Tesis de Maestría

CELERIDAD DE LA PANDEMIA: PRONÓSTICO DE COVID-19 EN ARGENTINA APLICANDO UN MODELO COMPARTIMENTAL DE APRENDIZAJE AUTOMÁTICO SIRD, ESTRUCTURADOS POR ALGORITMOS PREDICTIVOS, BAYESIANO Y AGRUPAMIENTO.

Carlos Alberto Hernández Sanabria

Carlos.sanabria@sas.com

Director de Tesis

Prof. Fernando Asteasuain

Buenos Aires, Argentina 2022

ARQUITECTURA DE REFERENCIA PARA CENTROS DE DATOS

ORIENTADOS A *CLOUD COMPUTING*

Carlos A. Hernández Sanabria

_____

Director de Tesis

_____

Presidente del Jurado

_____

Jurado

_____

Jurado Externo

Defensa de la Tesis

Ciudad Autónoma de Buenos Aires, a los ___ días del mes de_____

de _____.

RESUMEN

La mortalidad y las predicciones del crecimiento de los casos de covid-19 son importantes para las decisiones en el mundo de la salud, las empresas y las personas que vivieron con la pandemia. Esta labor predictiva es un desafío debido a la evolución del virus, los datos acotados y las respuestas basadas en evidencias de lo vividos en los últimos 3 años. Se define un modelo de serie temporal bayesiano basado en un algoritmo de bosque aleatorio dentro de un modelo compartimental epidemiológico SIRD, aplicado en una automatización de Machine Learning, que provee predicciones de COVID-19 a través de prácticas fundamentadas. El primer modelo mencionado, ajusta una curva específica de la distancia de la velocidad (primera derivada) del recuento de casos acumulados (operación de acotamiento), estos datos son transformados logarítmicamente (prolongación de datos en laso de tiempo), a través de 3 ubicaciones en el territorio Argentino e incorporando información previa, sondear y obtener una repartición posterior, con el fin de proyectar las trayectorias de los casos de infecciones, muertes y recuperados. El modelo compartimental usa esta distribución, predice muertes por medio del algoritmo Random Forest, en base a los datos obtenidos del dataset raíz, lo que genera estimaciones de intervalos y muertes en 3 zonas de Argentina con proyecciones diarias. Se evalúa el modelo entrenándolo en períodos progresivos durante de la pandemia y calculando su precisión predictiva en pronósticos de 15 días, para las etapas del modelo; 3 meses para el algoritmo Random Forest; y 4 meses para el algoritmo Bayesiano. La variación sustancial en las trayectorias pronosticadas y la incertidumbre asociada en el epicentro de la pandemia, estas trayectorias se ilustran en una representación de datos, en tres ubicaciones: Buenos Aires, CABA, Córdoba. La precisión y conteo de este modelo COVID-19 ofrece predicciones confiables y estimaciones de incertidumbre para la trayectoria actual de la pandemia.

Keywords: *Dataset, Machine Learning, Algoritmo de Bosque, Teorema Bayesiano, Modelo Compartimental SIRD.*

1. INTRODUCCIÓN

La rápida propagación del virus SARS-CoV-2 ha precipitado una epidemia sanitaria mundial, generando muerte e infectando a un alto porcentaje de humanos. Las entidades de Estado de todo el mundo han intervenido sin precedentes a reducir la propagación y la letalidad del virus. Estas mediaciones se han basado en gran medida en las predicciones de mortalidad y el aumento exponencial de los casos de COVID-19.

Los modelos de predicción de COVID-19 se pueden clasificar en varias ramas de la epidemiologia, nos copete seleccionar los modelos matemáticos que simulan enfermedades dentro de una población o modelos estadísticos que ajustan una función a los datasets observados y se extrapolan con una visión de resultados. Se plantean las características de ambos tipos de modelos. Por lo general de los modelos de COVID-19 son modelos compartiméntales *(Li R, Pei S, Chen B, Song Y, Zhang T, Yang W, 2020) (Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N, 2020) (Walker PG, Whittaker C, Watson O, Baguelin M, Ainslie K, Bhatia S, 2020)* **[1–3]**, un modelo matemático utilizado por epidemiólogos para simular epidemias de enfermedades infecciosas durante más de un siglo. Los modelos compartiméntales dividen una población en compartimentos mutuamente excluyentes que denotan el estado de la enfermedad a través de variables de entrada y salida, estas proporcionan un conjunto de ecuaciones diferenciales que definen el flujo de la población entre los compartimentos *(Lin Q, Zhao S, Gao D, Lou Y, Yang S, Musa SS, 2020)* **[4]**. Tradicionalmente reciben el nombre de sus compartimentos con los modelos SIRD (susceptible-infeccioso-recuperado, muerte) *(Mandal S, Bhatnagar T, Arinaminpathy N, Agarwal A, Chowdhury A, Murhekar M, 2020)* **[5]** y SEIR (susceptible-expuesto-infeccioso-recuperado) como ejemplos clásicos.

En un modelo compartimental de enfermedades infecciosas, S(t) es el número de individuos susceptibles en el tiempo t (velocidad de propagación), y las nuevas infecciones están

representadas por el flujo de individuos fuera del compartimento S. Esto se rige por la primera derivada de S(t) con respecto al tiempo, dS(t)/dt. Los modelos clásicos SIR y SIRD expresan esto como proporcional al producto de S(t), I(t), D (t dx) y una constante de velocidad β.

1.1. ALCANCE

La proyección principal del presente estudio es priorizar los casos de infecciones y muertes de COVID-19 implementando una composición de aprendizaje automático para asimilar funciones de transición, a un modelo compartimental y métodos analíticos de datos (Bayesianos- Forrest). Demostrando la velocidad de los casos acumulativos logarítmicos como un objetivo importante para predecir el crecimiento de los casos y proponer un modelo de serie programado de tiempo bayesiano que proporciona trayectorias específicas del punto que se extrapolan correctamente dentro de un modelo de probabilidad general, incluida la cuantificación de la incertidumbre y el ajuste del error. Usamos un algoritmo de bosque aleatorio para la función de transición de muerte, aprende la relación entre los casos de COVID-19 y las características de la población funcionales a través del podado del algoritmo Forest.

En este modelo, dividimos infecciones en dos clases: (i) personas que superaron los síntomas; (ii) individuos que presentan síntomas confirmados, es decir, infecciones respiratorias y contacto, estas son valoradas como casos positivos.

Estas dos clases de infección tienen tasas de transmisión separadas: ***b***, la transmisión tasa debida a individuos infectados y documentados; y ***mb***, la tasa de transmisión debido a personas que no superaron los síntomas, este último se reduce por un factor ***m*** (divisor de transmisión).

Sin embargo, este es un modelo conceptual preliminar, con la finalidad de fundamentar las bases de datos para futuros estudios de modelado, pudiendo ajustar fácilmente nuestro modelo para que los resultados estén en actuales con estudios previos.

1.2. CONTEXTO DE INVESTIGACION

Las observaciones de los individuos Covid-19 que convivieron con la cuarentena es un

proceso esencial para llevar a cabo esta investigación, por tal motivo permite mantener actualizada la problemática que vive el planeta con respecto a la salud de los pacientes y la evolución de la enfermedad. Todo el seguimiento debe estar canalizado y escalado de acuerdo con el avance de los contagios con respecto a las muertes, permitiendo actuar pronto de acuerdo con la valoración predictiva de cada individuo y generar una información solida sobre las predicciones y focos de contagio.

En la información descrita a lo largo del presente documento ofrece un marco de proyecto de Datos Informáticos, siendo apoyados por la data del Ministerio de Salud del Gobierno Nacional de Argentina. El proyecto se titula 'Celeridad de la pandemia: pronóstico de covid-19 en argentina aplicando un modelo compartimental de aprendizaje automático sird estructurados por algoritmos predictivos, bayesiano y agrupamiento', integrando un conjunto particionado de soluciones y evidencias que permitan controlar por medio de la programación una acción rápida de identificar la problemática y disponerla para próximas epidemias con base a algoritmos predictivos y epidemiológicos.

El objetivo final de la investigación es: 'Incrementar la efectividad del proceso a los individuos con Covid-19 con variantes evolucionadas con respecto a la transición de la virosis, ya sean muertes, recuperación, cambio de variante medica'. Estos casos recaen en escenarios probables o supuestos que dependerán de una serie de constantes que se ajustarán dependiendo del modelo/resultado de programación a implementar. De por medio se proporcionan plataformas analísticas y predictivas que integren diferentes fuentes de algoritmos con respecto al tratamiento de los datos, incluyendo la información comparativa por otros países y considerar siempre que los modelos referentes en la investigación pueden proveer una analítica de datos, relevante a las simulaciones automáticas, desplegadas a partir de los datos inyectados. Es posible encontrar modelos mas completos y mas optimizados para esta investigación, pero al considerar una problemática evolutiva y dependiente de una variante, se valida la implementación de los algoritmos que serán mencionados y detallados en los próximos capítulos de la

investigación.

1.3. PLANTEAMIENTO DEL PROBLEMA.

En diciembre de 2019, apareció en Wuhan, China, el síndrome respiratorio agudo severo coronavirus 2 (SARS-CoV-2), la nueva variante del coronavirus. Desde entonces, la ciudad de Wuhan ha tomado medidas sin precedentes en respuesta al brote, incluidos cierres prolongados de escuelas y lugares de trabajo. Los efectos de las medidas de distanciamiento físico en la progresión de la epidemia de COVID-19, con la esperanza de proporcionar algunos conocimientos para el resto del mundo **[6]**. Las consecuencias del crecimiento de casos de COVID-19 y la mortalidad son fundamentales para las decisiones de los líderes políticos, las empresas y las personas que luchan contra la pandemia. Esta tarea predictiva es un desafío debido a la novedad del virus, los datos limitados y las respuestas políticas y sociales dinámicas **[7]**. La compleja tarea predictiva es un desafío, debido a la novedad del virus, los datos limitados y las respuestas políticas y sociales dinámicas referente a la recepción de cargas de información basadas en la propagación y muertes causadas por el Covid-19. Son los modelos de series de tiempo y algoritmos de agrupamiento aleatorio los encargados de recepcionar y resolver dicha problemática, a partir de un modelo compartimental epidemiológico se logra favorecer las predicciones del virus ,siendo estas fundamentadas a partir de ajustes en una curva de ubicación específica a la velocidad (primera derivada) del recuento de casos acumulativo, transformado logarítmicamente, tomando como referencia la fuerza entre ubicaciones geográficas e incorporando información previa para obtener una distribución posterior para las trayectorias de los casos. *(Julien Arino, Stéphanie Porte, 2020)* **[8].**

En cuanto a la dificultad que presentan la implementación y el conocimiento de los modelos de predicción matemáticos/automatizados mencionados anteriormente versus los respectivos escenarios programáticos de compartimentos utilizando modelos epidemiológicos dentro del conjunto IA (Inteligencia Artificial), imposibilita totalmente un marco de modelización para generar una serie de trabajos futuros. *(Amy Hurford,*

Maria M. Martignoni, J.C, 2022) **[9]**. Es importante señalar que el modelo más frecuente para albergar escenarios de compartimientos es el modelo SIRD (susceptible - infeccioso - recuperado-muertes) , este cumple con 4 ítems que favorecen la identificación de variables de tiempo, aumento y otros modelos, el ítem de edad de la infección, lazos de fechas y etapas de la enfermedad proporcionan una aproximación razonable a los detalles de la progresión a través de la infección con un número mínimo de parámetros , pero resolver esta circunstancias no es tarea sencilla, puesto que la estructura del modelo es muy sensibles al valor del parámetro de entrada que se implemente y la cantidad de parámetros de salida deseados, generando un intensiva información y un desafío al programador, en manipular el mejor valor de los datos críticos para la solución de una comparativa en el tiempo. La evaluación de los modelos identificados y mencionados hasta el momento constan de periodos prologados en lapsos de tiempo periódicos y son estos ciclos los que definen la confiabilidad y las incertidumbres de la trayectoria de la velocidad en la que escalan los resultados, siendo éstos originados por la evolución del virus. A continuación, se ilustra un modelo de evaluación básico que ofrece SIRD a partir dos variables en el periodo de tiempo de incubación (infeccioso) del COVID, es decir contemplando una semana con su respectivo informe de contagiados (susceptibles). Luego, comparando la porción de individuos infectados en el tiempo 0 y aun incubando en el tiempo ***t*** (la constante a evaluar), obtenemos la siguiente ilustración.

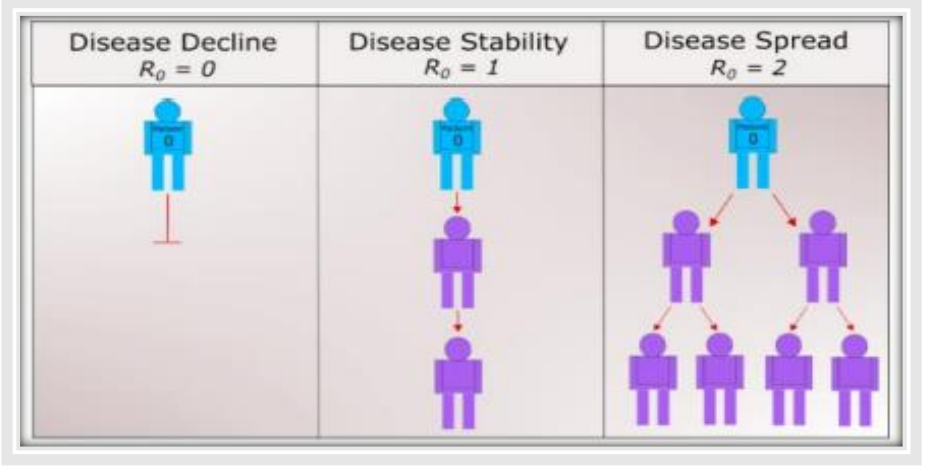


Ilustración 1- *Manifestación de la propagación de enfermedades sobre la base del valor R0*

1.4. ANTECENDENTES

Las investigaciones estudiadas y exploradas, la mayoría guardan relación a la problemática planteada, todas estas se indagaron y se hallan las siguientes líneas de investigación con su respectiva línea evaluativa.

1.4.1. Infección sustancial indocumentada facilita la rápida difusión de novedades coronavirus (SARS-CoV-2). [10]

Se usaron observaciones de infecciones reportadas dentro de China, junto con los datos de movilidad, un modelo meta-poblacional respecto a una dinámica en red, e Inferencia bayesiana, para inferir características epidemiológicas críticas asociadas con el SARS-CoV-2, incluida la fracción de infecciones indocumentadas y su contagio. Estimamos que el 86% de todas las infecciones eran indocumentados [intervalo de credibilidad (IC) del 95%: 82–90%] antes de las restricciones de enero de 2020. Los detalles de la iniciación del modelo, incluida la dispersión inicial de expuestos e indocumentados DataFrame, se proporcionan en diversos materiales de estudio. Para tener en cuenta los retrasos en la infección confirmación, también definimos un tiempo hasta el evento modelo de observación utilizando una distribución gamma (ver materiales complementarios). Específicamente, para cada nuevo caso en el grupo i (base grupo), un reportaje retraso ***td*** (en días) se generó a partir de una gamma distribución con un valor medio de Td . En ajuste tanto los brotes sintéticos como los observados, realizamos simulaciones con el modelo de referencia sistema que utiliza diferentes valores fijos de Td (6 días ≤ Td ≤ 10 días) y diferente máximo Dispersión (1500 ≤ Dispersión max≤ 2500) (ver materiales suplementarios) (fig. S2). El mejor modeloinferencia posterior se identificó por probabilidad logarítmica. *(Li R, Pei S, Chen B, Song Y, Zhang T, Yang W, Shaman J. , 2020).*

1.4.2. El efecto de las estrategias de control para reducir la mezcla social en resultados de la epidemia de COVID-19 en Wuhan, China: un estudio de modelado. [11]

Posibles informaciones que implican enfoques de variables incógnitas donde los cambios demográficos en la población (es decir, nacimientos, muertes y envejecimiento) que están siendo identificadas. Por esto, se deben considerar dichas variables para optimizar datos que permitan un estado de predicción, se divide la población según el estado de infección: en susceptibles (S), expuestos (E), infectados (I) y personas eliminadas (R), y de acuerdo con la edad, en 5 años listados hasta los 70 años y una categoría única a los 75 y mayores (resultando en 16 categorías de edad). Los individuos susceptibles pueden adquirir la infección a un ritmo determinado cuando entran en contacto con una persona infecciosa y entrar en el estado de enfermedad expuesta antes de que se conviertan en infecciosos y luego recuperarse o morir. Asumimos, la ciudad de Wuhan será un sistema cerrado con una población constante tamaño de 11 millones (es decir, S+ I + R + D = 11 millones) en todo el curso de esta epidemia. Usamos el modelo SEIR presentado en la figura 1. Los patrones de mezcla son específicos por edad de individuos, en el grupo de edad modificaron su probabilidad de ser expuesto al virus dado un cierto número de infecciones por personas en la población. Adicionalmente, se incorporan aportes de casos asintomáticos y subclínicos. Sin embargo, la pregunta de si tales individuos son capaces de transmitir la infección sigue sin resolverse en el momento de la escritura, aunque la evidencia sugiere que es probable también. *(Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N, 2020)*

1.4.3. Inteligencia artificial como apoyo a intervenciones no farmacológicas para combatir la COVID-19. [12]

El proceso de investigación que se han pronunciado desde la aparición del Covid-19, argumenta en tres enfoques, sucedidos en distintos momentos con la ayuda de la IA y Machine Learning. El primer enfoque abordado fue el uso de IA para la detección temprana de síntomas gripales o alérgicos mediante un análisis de términos claves en plataformas digitales. El segundo enfoque fue el uso de algoritmos de aprendizaje para

diagnosticar, clasificar y agrupar anomalías producidas por el covid-19. El tercer enfoque fue el pronóstico de pacientes con covid-19, con el fin de focalizar la asistencia y planificar la utilización de recursos médicos para pacientes críticos. Este proyecto de investigación favorece el entendimiento de como la inteligencia artificial de la mano de machine Learning han permitido explorar por medio de métricas y recursos toda la información referente a las anomalías y problemas de salud producido por el Covid-19. *(Mujica Rodríguez IE, Toribio Salazar LM, Cóndor Cámara DF., 2020)*

1.3.4. Inteligencia artificial y Big Data como soluciones frente a la COVID-19 [13]

Se exponen algunos proyectos de investigación relacionados con el aprendizaje profundo, el aprendizaje automático, el Big Data y la ciencia de datos, tendientes a dar soluciones plosibles bien en el monitoreo, detección, diagnóstico y tratamiento de las enfermedades asociadas con el virus. De hecho, se muestra la correspondencia entre las tecnologías disruptivas y la información crítica, creando sinergias que permiten elaborar sistemas más avanzados de estudio y análisis facilitando la obtención de datos relevantes para la toma de decisiones sanitarias. *(Jairo Eduardo Márquez Díaz, 2020).*

1.3.5. Enfermedad por coronavirus, COVID-19 [14]

El presente anexo es un documento que provee una información completa del comportamiento, estudio y conceptos ofrecidos por la ciencia epidemiológica, pero se tomó los conceptos referentes descritos en el *Capítulo 1.3*, titulado transmisión. Este apartado especifica las variantes, los estados y los medios de propagación que ha tenido este virus, dando a conocer un arduo y consolido estudio de cada etapa que han permitido la celeridad desde su aparición hasta el presente año. El análisis de la información anterior abarca todo el concepto de transmisión, por ende, ofrece un aporte muy concreto al considerar el intervalo serial de la velocidad como principal medio de observación epidemiológico en periodos de incubación. *(Centro de Coordinación de Alertas y Emergencia Sanitaria, 2021)*

1.3.6. Velocidad de la pandemia: pronóstico de COVID-19 en los EE. UU. [15]

Basado de un modelo compartimental de aprendizaje automático y series de tiempo bayesiano, pretende ofrecer una serie de procesos predictivos para estimar la velocidad y las novedades del virus, y los datos limitados que están siendo validados por grandes organizaciones médicas y científicas. Se incorporará un modelo de serie de tiempo bayesiano y un algoritmo de árbol aleatorio dentro de un modelo compartimental epidemiológico para las predicciones de COVID-19 a través de un dataset. El modelo de caso bayesiano ajusta una curva de ubicación específica a la velocidad (primera derivada) del recuento de casos acumulativo transformado logarítmicamente, tomando prestada la fuerza entre ubicaciones geográficas e incorporando información previa para obtener una distribución posterior para las trayectorias de los casos. El modelo compartimental usa esta distribución y predice muertes usando un algoritmo forestal aleatorio entrenado en datos de COVID-19 y características a nivel de población, produciendo proyecciones diarias y estimaciones de intervalo para casos y muertes en los estados de EE. UU. *(Watson GL, Xiong D, Zhang L, Zoller JA, Shamshoian J, Sundin P, Bufford T,* Rimoin AW, Suchard MA, Ramirez CM., 2021)

1.4. JUSTIFICACION

El presente modelo, va dirigido a todas aquellas personas encargadas del área de investigación epidemiológica o estudios relacionados al Sars-Cov-2 (Covid 19), el cual les permite conocer en tiempo real los datos de velocidad de contagio a partir de modelos de respuesta, con los cuales avanzan y complementa sus proyectos investigativos, a su vez mantiene informado a todas persona sobre estos valores exponenciales ; es decir es un modelo comprensible al alcance de cualquier persona interesada.

1.5. OBJETIVOS.

1.5.5. Objetivo General

Diseñar un modelo epidemiológicos SIRD, estructurado en un esquema estadístico de ajuste de curvas que integren una función matemática/algorítmica del DataFrame

resaltando a detalle cada información, basando la información en El Ministerio de Salud de la Nación Argentino, la data ubicada allí se extrapola, se integran y se modelan trayectorias de velocidad de transmisión del Covid-19, a partir de un pronóstico de crecimiento automatizado y técnicas de aprendizaje supervisado en un escenario de programación Python-Jupyter Network, identificando las variables de entrada /salida (atributos/columnas) pertinentes con respecto a los casos confirmados de Covid-19. A lo largo de la investigación se implementan técnicas de automatizadas en Inteligencia Artificial en conceptos definidos en el subconjunto de Machine Learning.

1.5.1. Objetivos específicos:

• Validar y Comparar el intervalo de los primeros registros de datos de casos positivos versus las muertes a partir de la estadística concebida por parte del MSNA (Ministerio de Salud de la Nación Argentino) para contemplar la implementación de sus datos con respecto a la celeridad de casos confirmados por COVID-19.

• Seleccionar un modelo matemático/automatizado SIRD (susceptible -infeccioso - recuperado) dentro de una solución de aprendizaje supervisado con respecto a los facture requeridos (*muestra acumulada de casos, total infectados, total fallecidos*) y las variables de entrada/salida para facilitar la solución del modelo.

• Identificar aquellos algoritmos de aprendizaje supervisados que se adapten a las funcionalidades de mis variables de entrada y salida, proporcionando una evaluación clara de la rapidez de transmisión de casos confirmados y que favorezcan al modelo matemático SIRD ajustando la velocidad del registro de casos acumulados por Covid-19.

• Ejecutar los algoritmos de predicción decisition trees / Bayesiano, seleccionados en un entorno de programación Python llamado Jupyter y el modelo de velocidad por el cual se proyecta el crecimiento de casos confirmados de muertes e infectados que se ajustan a los datos observados, para luego comparar su extrapolación obtenida en ambos escenarios estimulando un aceptado ajuste de curvas de ambos modelos.

• Obtener los resultados cuantitativos y cualitativos para ajustar los modelos, producir constantes pronosticadas y generar resultados estimados a partir de herramientas de cálculos lógicos intercalando atributos de velocidad.

• Adaptar los resultados obtenidos que permitan visualizar los modelos evaluados y hacer un registro de toda la información relevante en un documento de investigación.

2. FUNDAMENTOS TEÓRICOS

En este apartado se describen los conceptos, la elaboración de ideas, las características del modelado y la tecnología implementada que dan origen a la problemática a resolver. Se comienza describiendo la manera habitual del modelo y las practicas tecnológicas, las nuevas áreas del aprendizaje de datos predictivos y el impacto que tienen estos modelos en la inteligencia de datos. Seguidamente se tratarán los nuevos avances conceptuales y herramientas de validación, por el cual se obtiene una respuesta al problema planteado.

2.3. Modelo Epidemiológico. [16]

Representación de un proceso o sistema que está diseñado para aumentar la apreciación y comprensión de un posible autómata o serie de procesos automáticos. Los modelos se desarrollan para comprender el efecto de las influencias externas en los resultados, a través de la representación de las interacciones entre los componentes del sistema, y para comunicar ideas sobre el comportamiento de este.

Desde una perspectiva de salud, los modelos epidemiológicos tienen una variedad de aplicaciones/técnicas de evaluación. Po ende se implementan las siguientes:

– Generar hipótesis de datos (que luego pueden utilizarse para orientar estudios posteriores) sobre los factores que intervienen en la persistencia de enfermedades epidémicas en las poblaciones.

– Evaluar el impacto de mortalidad de las enfermedades

– Evaluar la eficacia y estrategias de los modelos determinísticos y de control de constantes. *(Carlos Castillo-Chávez, 2020)*

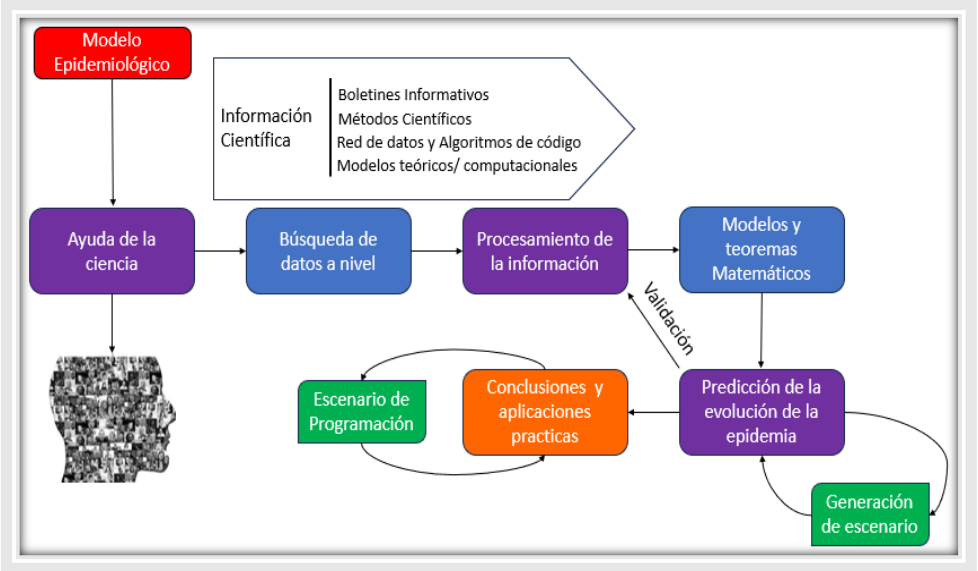


Ilustración 2*- Proceso Epidemiológico, autómatas de datos*

2.4. Modelo SIRD [16 17]

El modelo SIRD es un modelo compartimental/casillero, su función principal es modelar datos de una enfermedad que se propaga a través de una población y se particionan en divisiones de bloque de datos. Acrónimo de Susceptible, Infectado, Recuperado, Muerte. Suponiendo que la población total es constante y se divide en tres variables.

Se emplea el modelo SIRD (estándar) donde la población N se divide en pequeñas poblaciones de susceptible (S), infectado (I), recuperados (R) y muertes (D) para lazos de tiempo t. Por lo consiguiente, N= S+I+R+D.

Por medio de un conjunto de ecuaciones.

$$\frac{dS(t)}{dt} = -\beta SI/N \quad (1)$$

$$\frac{dI(t)}{dt} = \beta SI/N - \gamma I - \delta I \quad (2)$$

$$\frac{dR(t)}{dt} = \gamma I \quad (3)$$

$$\frac{dD(t)}{dt} = \delta I \quad (4)$$

Ilustración 3*- Modelo de ecuaciones de modelo SIRD*

Incógnitas o constantes, β, λ, δ intervienen como parámetros que especifican las características de infección, recuperación y muerte respectivamente (*Ilustración. 4*). Para coordinar consecutivamente este documento se considera que, ɪ representa la población de infección sintomática. La mejor practica para nuestro modelo *(Sunita Tiwari, 2020)* implica, en promedio, la infección tiene sus indicios de 2 a 3 días antes de que los síntomas se agudicen, es decir se contarían más de 15 días para tomar en cuenta dicho reporte. Es importante definir que la infecciosidad aumenta a un punto máximo antes de la aparición de los síntomas y perdura durante aproximadamente entre siete a diez días después del alza de la infección. Por consiguiente, una persona infectada permanece infecciosa durante aproximadamente 15 días en promedio y luego presentaría un cuadro de mejoría o nueva asignación de cuadro clínico. Este análisis preliminar, se establece la tasa de recuperación λ~1/2 (λ taza índice), que sin embargo no es el mejor ajuste para todos los casos que se pueda estudiar. En general, los valores numéricos de los parámetros del modelo se logran del mejor arreglo. Los valores iniciales (tiempo t = 0 días) del número de infectados, recuperados y fallecidos (I_0, R_0, D_0) se eligen a partir de datos reales suministrado.

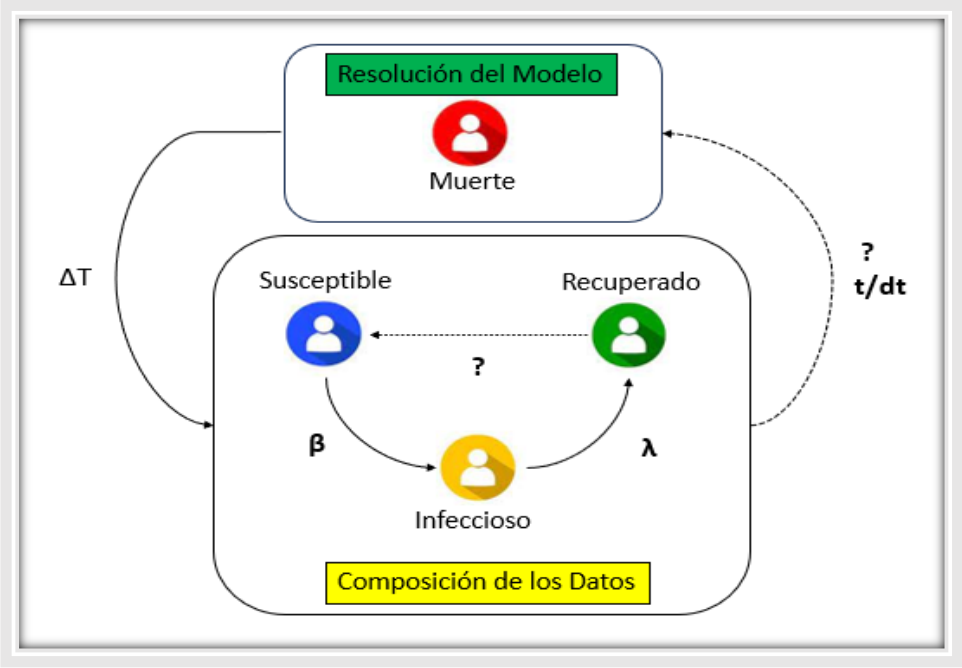


Ilustración 4*-Intervension de parámetros, condición modelo*

La selección del número inicial de susceptibles (S_0) es tema complejo. Empleamos dos enfoques diferentes para obtener una estimación de S_0. En primera instancia, estudiamos los datos de las cada una de las poblaciones de cada ciudad mencionada donde los casos positivos acumulados se han acercado a una altura aproximada a la estándar. Sin embargo, la población infectada, solo se puede determinar cuando la epidemia haya terminado. Al dividir este número por la población total del evaluados, se obtiene una

fracción que parece ser del orden de 10-1 para CABA, 10-2 Buenos Aires, 10-3 Córdoba. Por tanto, se puede obtener una estimación de los susceptibles multiplicando la población de un país por una fracción de índice seleccionada.

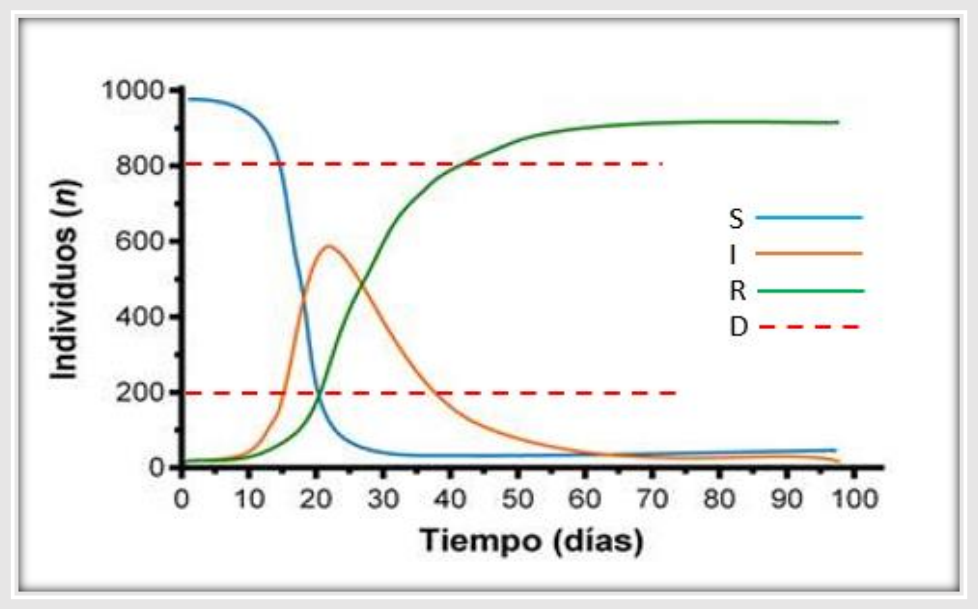


***Ilustración 5**- Selección inicial, modelo grafico SIRD*

Sin embargo, el número de susceptibles obtenido de esta manera indica un límite inferior, ya que muchas personas con síntomas leves o sin síntomas. La otra posibilidad dispuesta para estimar la fracción es probar el número de casos positivos por el número de pruebas realizadas. Este número sería un límite superior, ya que hay muchas regiones, solo se dispone de los territorios mencionados y las poblaciones en esos focos no serían susceptibles. Es importante tener una relación entre el número de casos positivos y el número total de pruebas, por ende, tomamos referencia de las tasas registradas para diferentes países.

Países	**Fracciones (Casos confirmados / Casos informados)**
Argentina	0,18
EE. UU.	0.159
Corea del Sur	0.0016
España	0,1072
Alemania	0.034

***Tabla 1**- Tasas registradas de Covid por % en el mundo.*

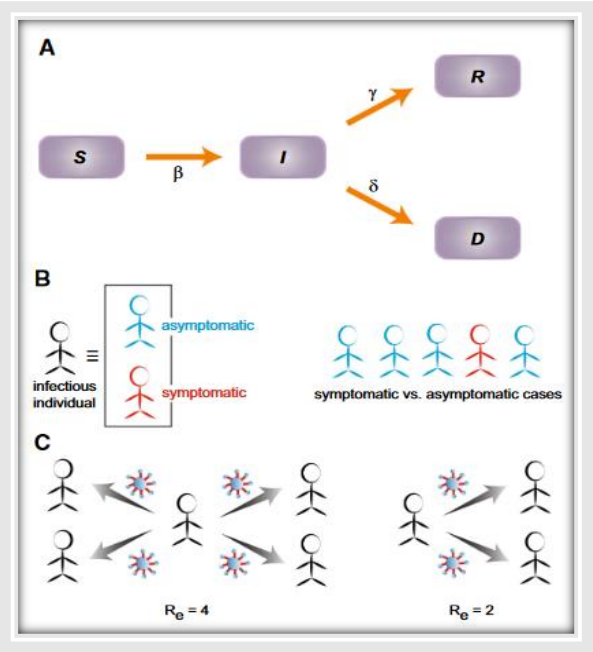


Ilustración 6*- Diagrama de Modelo SIR*

Con la conformación del modelo, viene la estimación cuantitativa de la velocidad por el cual, la enfermedad se propaga a través de una población. Es decir, a partir del modelo determinista SIRD, el objetivo es evaluar qué tan rápido un portador humano infectaría a personas pertenecientes a la población de susceptibles. La cantidad que determina la velocidad de transmisión de la pandemia es el número de reproducción efectivo o número de sucesiones *(Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study., 2020)*. Es de considerar que el número de reproducción básico R_0, definido como el número medio de infecciones secundarias que ocurren cuando una persona tiene un estado infeccioso (tramite o principio de infección) se coloca en una población susceptible, el valor R_0 se puede estimar desde la etapa más temprana de la infección, cuando la persona infectada se mezcla libremente con la población susceptible. Estimación de R_0 a menudo es un desafío debido a la falta de datos no sesgados, ya que no se pueden determinar con exactitud todas las infecciones secundarias; especialmente para COVID-19, donde los casos asintomáticos fueron muy comunes y sin síntomas para identificar (*Ilustración 3*).

El número de reproducción efectiva (R_e), implementados para este estudio, valora correctamente el número medio de nuevas infecciones (infectadas del grupo susceptible/clasificadas) transmitidas / inducidas directamente por una persona infectada típica (contacto/respiratorio) y pueden variar durante toda la duración de la

infección. En el modelo SIRD, R_e se puede representar como $\beta/(\lambda+\delta)$. Considerando el mejor ajuste de los datos, encontramos que $\lambda > \delta$, produciendo $R_e \sim \beta/\lambda$. Si Re > 1, la enfermedad comienza a propagarse en una población que se ve infecta masivamente, pero la propagación no ocurre si R_e se ubica por debajo de 1 Es fácil notar que más tiempo una persona permanece infecciosa (es decir, 1/ λ días), puede dar lugar a una R_e muy grande incluso si el número de interacciones infecciosas por día (i, e … β) es pequeño.

2.5. Inteligencia Artificial [18]

La ocupación de una gestión de Big Data (Grandes volúmenes de datos) y la construcción de sistemas inteligentes es el uso de conceptos y metodologías de Data Science, Inteligencia Artificial (tratar, analizar y presentar informes a partir de datos ruidosos, estructurados y no estructurados), Data Mining (proceso de análisis de grandes cantidades de información dispersa para darle sentido y convertirla en conocimiento) y Machine Learning (desarrollo de algoritmos y modelos estadísticos que utilizan los sistemas de computación con el fin de llevar a cabo tareas sin instrucciones explícitas).

Implementar máquinas inteligentes, conscientes de sí mismas no es algo que surgió repentinamente, el pasar del tiempo las computadoras son más rápidas, mejor procesamiento, mejor poder de cómputo y más almacenamiento, hemos estado viviendo en lo que me gusta llamar la "era de la información" o la "era de los datos".

Procesos que se llevan a cabo en el cerebro humano que pueden ser analizados, a un nivel de abstracción, como procesos computacionales de algún tipo datos. El propósito de la IA es efectuar computacionalmente el conocimiento humano por procedimientos simbólicos. Buscamos es un programa de ordenador sobre una máquina específica desarrollado a partir del conocimiento que supuestamente usaba el operador humano (experto) que realizaba esa tarea. *(University of Cincinnati, 2020).*

La investigación no presenta una elaboración y estructura correspondiente con este

concepto de manera directa, pero indirectamente ofreció un medio relacionado con el mundo la automatización y considerando el conjunto global que alberga todas las ciencias de automatización y entrenamiento de los datos. Por ende, se puede catalogar como un tema esencial para disponer de conceptos más específicos y elaborar a futuro autómata de máquina, programando para llevar la tarea de buscar escenarios con nuevas variantes de gripa y asignarlas por si solas a un marco donde se supervise un trabajo final.

2.6. Machine Learning [19]

Pertenece a una rama de la Inteligencia Artificial ,el principal desafío de ML es permitir a las maquinas aprender sin ser continuamente programadas, favorecer el reconocimiento de patrones/esquemas , y es capaz de convertir una muestra de datos en un programa informático, facilitando extracción de nuevos conjuntos de datos para los que no ha sido entrenado previamente por una serie de algoritmos definidos dentro de este subconjunto. De hecho, el concepto principal de Machine Learning es disponer de los ordenadores/unidad de cómputo, estos poseen la habilidad de aprender sin ser explícitamente ser proyectadas a partir de una novedosas técnicas de código en múltiples ámbitos de la informática, incluida la disponibilidad de recursos programables y buen almacenamiento, automatizar tareas, realizar análisis complejos, detectar anomalías y responder a las entradas (constantes definidas). *(Dymaxion Labs, 2021).*

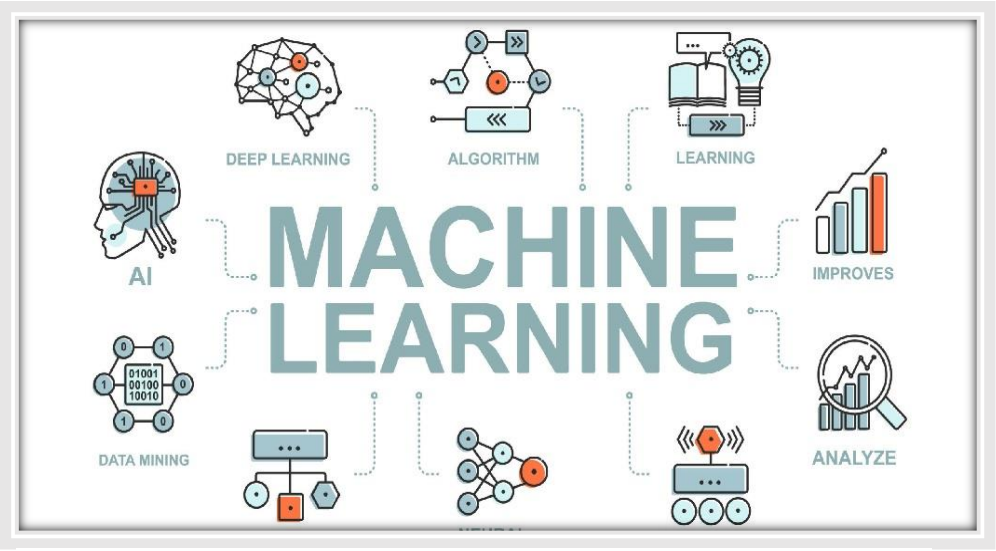


Ilustración 7*-Tecnología Machine Learning*

A lo largo de la investigación, la rama de Machine Learning es fundamental para el axioma de la investigación. Partiendo del conocimiento y el despliegue que brinda esta diciplina. El uso inicial va de la mano del conocimiento que abarca la Inteligencia artificial por el simple motivo de contener dicha tecnología como herramienta de algoritmos que se encuentran computados dentro de terminales o consolas en algún dispositivo inteligente. Partiendo del comentario anterior, supimos identificar los modelos y los tipos de algoritmos que daban un buen apoyo al modelo base, identificado previamente por sondeos y aciertos de la ciencia en su investigación en la pandemia. Una vez identificado el aspecto anterior, se debe interpretar los modelos internos que posee ML para la codificación de cada algoritmo, tenemos modelo supervisado y el no supervisado.

Disponer estos conocimientos más claros, nos permite avanzar con la investigación y considerar una data para ser modelada a partir de lo mencionado. Una vez tomado la fuente de los datos, optamos por transformarlos a la apertura correcta de simplificarlos en nuestro entorno Anaconda, Python. La codificación de código fue necesaria para consolidar la base del armazón del modelo SIRD.

2.7. Deep Learning [20]

También conocido como aprendizaje por refuerzo, esta rama utiliza un sistema de recompensa y castigo para entrenar el algoritmo. En esto, el algoritmo o un agente aprenden de su entorno y de sus variables (datos). El comportamiento aplicado se transforma en un agente/medidor, este obtiene recompensas por desempeño correcto y penalización por desempeño incorrecto. Por ejemplo, considere el caso de un automóvil autónomo, el agente obtiene una recompensa por conducir de manera segura hasta el destino y una penalización por salirse de la carretera. De manera similar, en el caso de un programa para jugar al ajedrez, el estado de recompensa puede ser ganar el juego y la penalización por recibir jaque mate. El agente trata de maximizar la recompensa y minimizar la penalización. En el aprendizaje por refuerzo, no se le dice

al algoritmo cómo realizar el aprendizaje; sin embargo, soluciona el problema por sí solo. *(Alba Centeno Franco, 2020).*

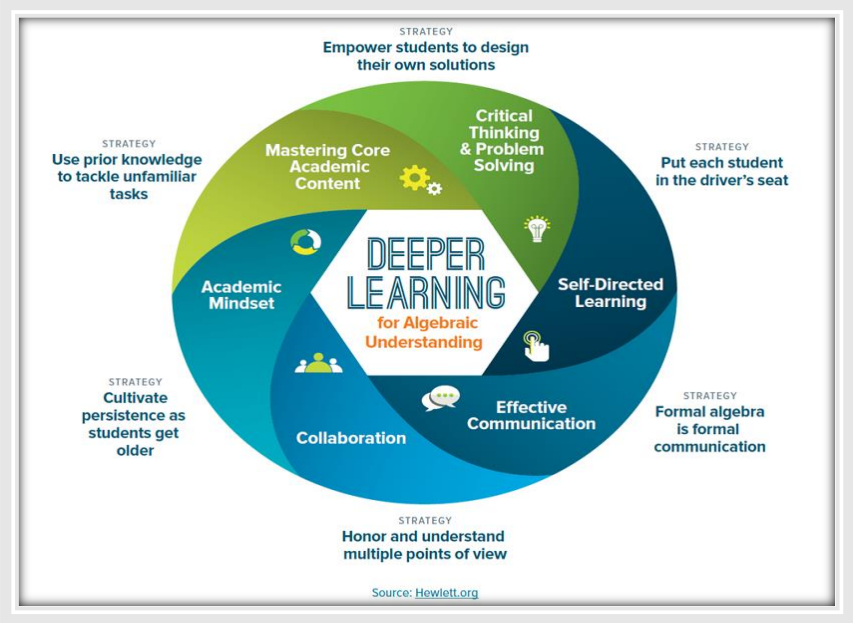


Ilustración 8*-Conector de funciones, Deep Learning*

2.8. Definición de la tarea de un agente T [21-27]

Por lo general es una muestra de datos o un punto entre las variables de entrada/salida, consiste en múltiples atributos de datos (también llamados características) al igual que los diversos parámetros del algoritmo/modelo. También podemos definir como una serie de datos típicos de características cualitativas/cuantitativas. *(Niklas Kühl, Marc Goutier, Robin Hir, Gerhard Satzger, 2020).*

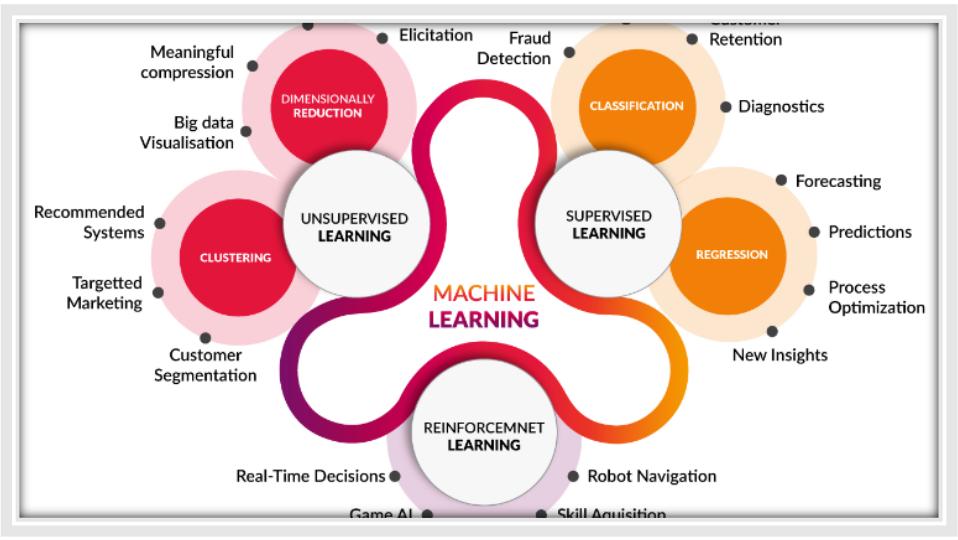


Ilustración 9*-Agente de datos , algoritmos soluciones de ML*

Estas tareas se pueden clasificar como tareas de Machine Learning, la siguiente lista describe algunas tareas populares:

- **Clasificación o Categorización:** es una lista de problemas o tareas en las que la máquina debe tomar puntos de datos o muestras y asignar una clase o categoría específica a cada una de ellas. *(Fernández Villafañez, 2022)*
- **Regresión:** tareas que generalmente implican realizar una predicción de modo que un valor numérico real sea la salida en lugar de una clase o categoría para un punto de datos de entrada. La mejor manera de entender una tarea de regresión sería tomar el caso de un problema del mundo real de predecir los precios de la vivienda considerando el área del espacio, el número de pisos, los baños, los dormitorios y la cocina como atributos de entrada para cada punto de datos. *(Fernández Villafañez, 2022).*

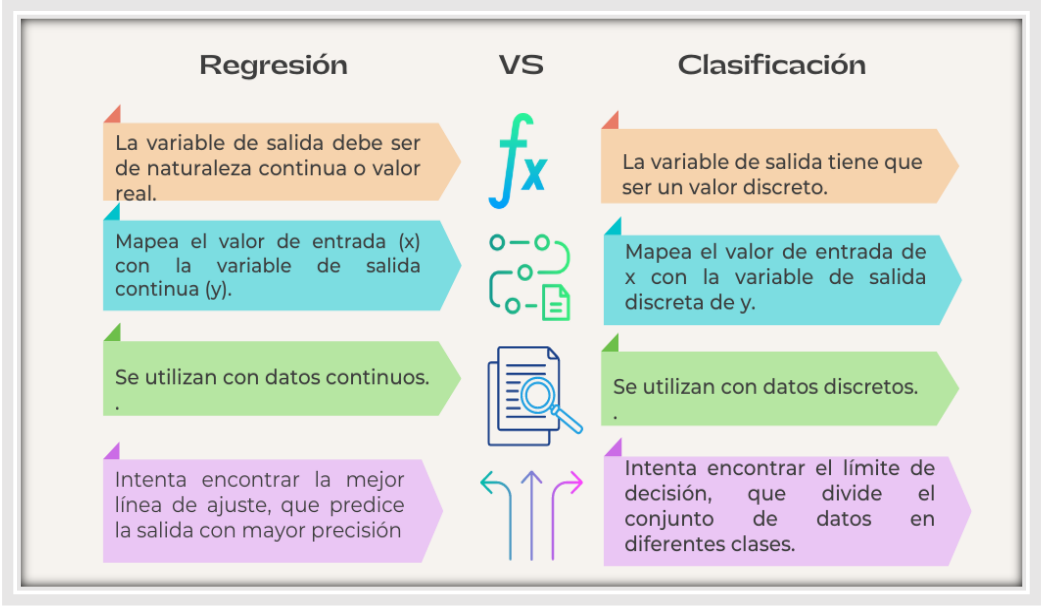


Ilustración 10*- Cuadro comparativo, 2 algoritmos fundamentales*

- **Detección de anomalías**: Son tareas que implican máquinas, revisan los registros de eventos, los registros de transacciones y otros puntos de datos para que pueda encontrar patrones o eventos anómalos o inusuales que sean diferentes del comportamiento normal. Ejemplos de esto incluyen tratar de encontrar ataques de denegación de servicio a partir de registros, indicaciones de fraude, etc. *(Alonso del*

Saso, Javier, 2020).

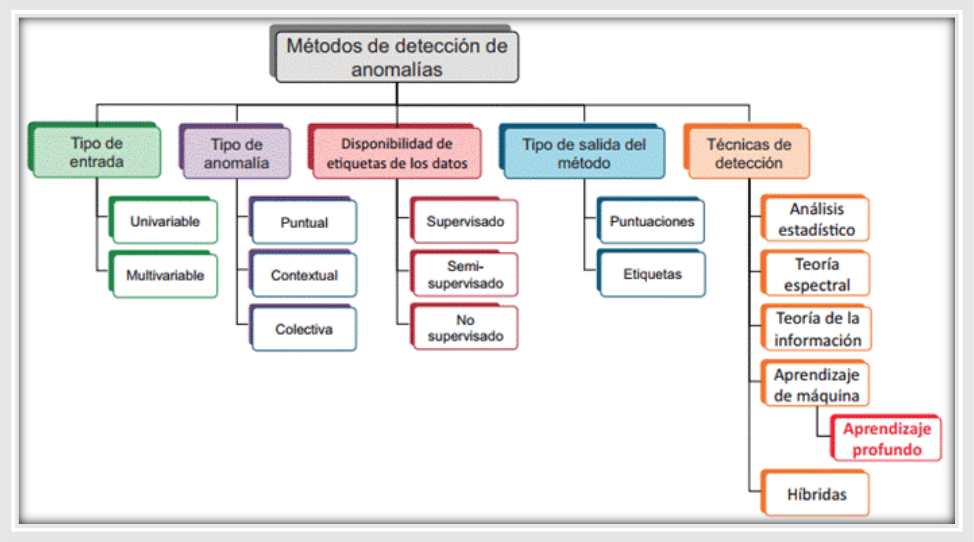


***Ilustración 11**-Estructura, detección de anomalías*

- **Anotación estructurada:** Realiza algún análisis en puntos de datos de entrada y agregar metadatos (descripción de todos los datos) estructurados como anotaciones a los datos originales que representan información adicional y relaciones entre los elementos de datos. Los ejemplos simples serían anotar variables de texto con sus a un disertación, entidades nombradas, gramática y sentimiento. También se pueden realizar anotaciones para imágenes, como asignar categorías específicas a píxeles de imagen, anotar áreas específicas de imágenes según su tipo, ubicación, etc. (Edian F. Franco, Rommel J. Ramos, 2019).

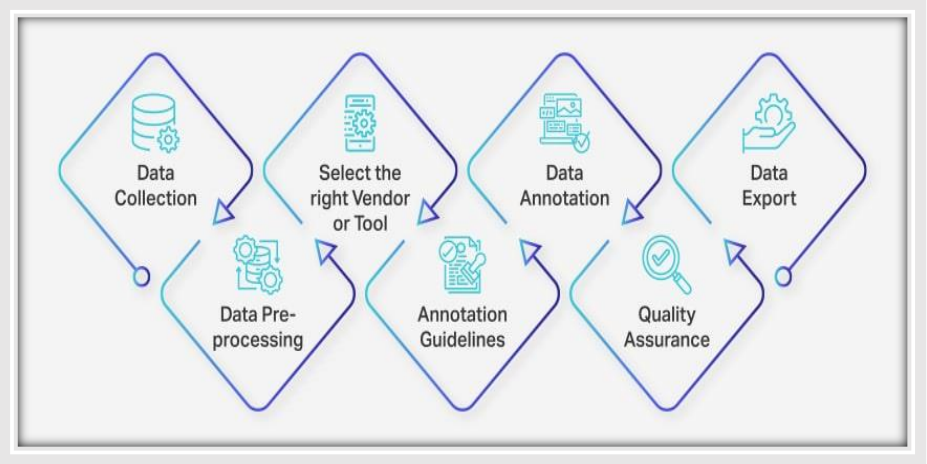


***Ilustración 12**-Ciclo de la anotación de datos*

• **Traslación de datos**: son tareas de traducción automática suelen ser de naturaleza tal que ,si tiene muestras de datos de entrada que pertenecen a un condición específica, las traducen a resultados poliformicos a la estructura deseada. La traducción basada en lenguaje natural es definitivamente un área enorme que maneja una gran cantidad de datos de texto. *(Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang L, 2019).*

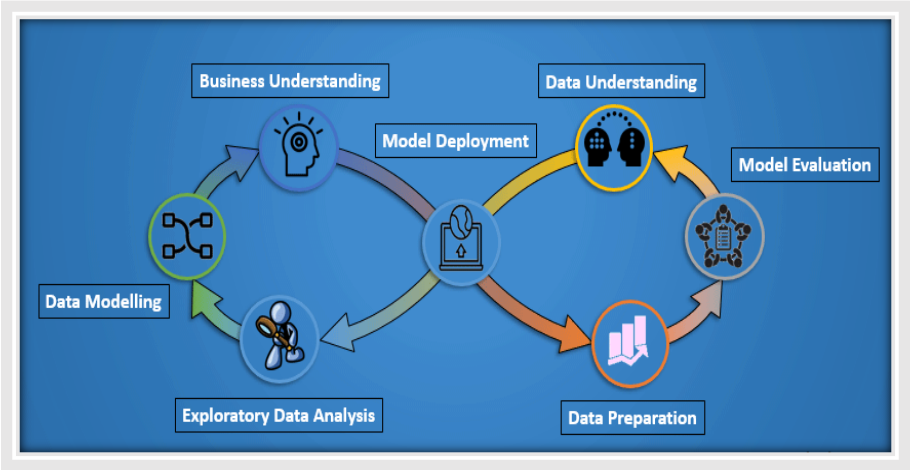


***Ilustración 13**-Codificacion de datos, estructura poliformica de datos.*

•**Agrupación de datos/ Clustering:** Los clústeres (almacenes de datos) o grupos de datos generalmente se forman a partir de muestras/porciones de datos de entrada al hacer que la máquina aprenda u observe patrones, relaciones y similitudes latentes inherentes entre los puntos seleccionados por los datos de entrada mismos. Por lo general, faltan datos pree-etiquetados o anotados previamente para estas tareas, por lo que forman parte del aprendizaje automático no supervisado (del que hablaremos más adelante). Los ejemplos serían agrupar productos, eventos y entidades similares. *(Aprendizaje no supervisado, 2020).*

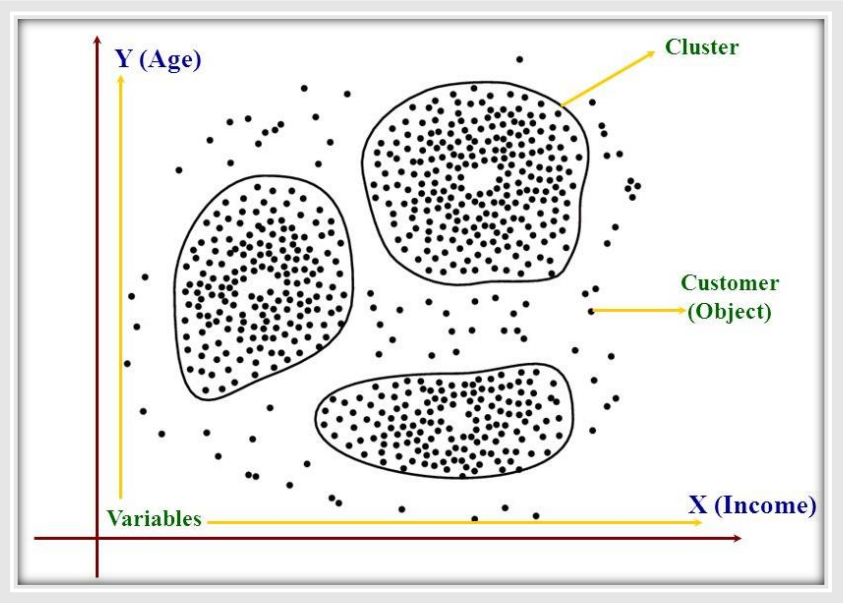


Ilustración 14*- Agrupamiento, muestra del modelo.*

•Árboles de Decisión/Decision Trees [28]

Los Árboles de Decisión o Decision Trees forman parte de los algoritmos de clasificación de aprendizaje supervisado en Machine Learning (ML) y su objetivo es trabajar sobre datos históricos y resultados conocidos para predecir el comportamiento de un dato nuevo de entrada e identificar patrones. Permiten que se construyan modelos con datos de entrenamiento para definir los parámetros del modelo y se validen los resultados con datos de prueba. Su funcionamiento se basa en reglas de decisión donde la cantidad total de datos está ubicada en el nodo raíz y éste se divide en ramas, llamadas nodos intermedios, representando una variable o atributo del conjunto de datos. (Jose A. Alonso Jim´enez, 2020).

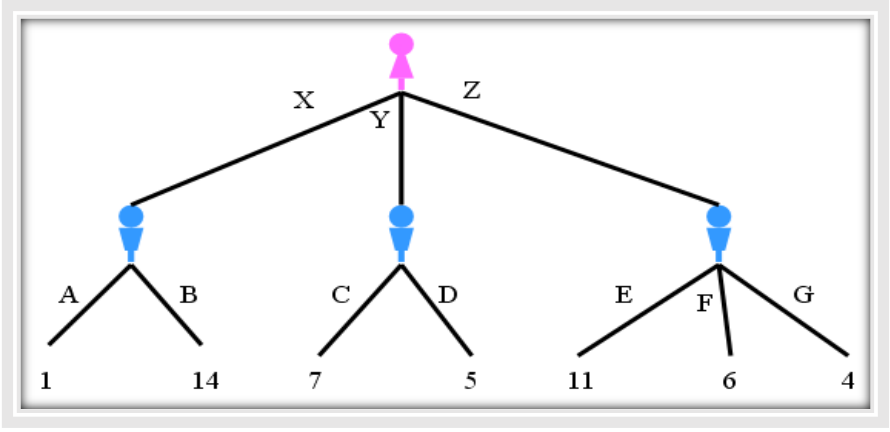


Ilustración 15*-Patrones de agrupamiento ,Árbol de decisión*

3. RESULTADOS

Se lleva a cabo el análisis del modelo SIRD sobre la progresión/agrupación Bayencina de covid-19 en el contexto de muertes y virus respiratorios pos-Covid en una porción de Argentina con variantes de simulación aproximada. Los resultados son previamente filtrados por la aplicación del algoritmo Random Forest, su función será detallar y comprimir las variables de entrada/salida, estos procedimientos se irán representando en las ilustraciones correspondientes a cada sub-apartado de este capítulo.

3.1. Incorporación del efecto de las medidas de aplicación.

Se implementa el efecto de bloqueo en el modelo modificando la tasa de infección (valor inicial/casos totales) y logrando el ajuste más optimo. Seleccionando de manera funcional y tipificando la tasa de infección dependiente del tiempo β(t) (constante en días) esta disminuye gradualmente después de que se aprovechan las medidas de contención *(Archana Singh Bhadauria, Hom Nath Dhungana, 2022)* **[29,30]**. Antes de proporcionar un cerco (límite de información) de datos, la tasa de infección es β=0, siendo nuestra segunda constante. Cuando persiste la contención de casos iniciales, se aplica el tiempo (días/día) t (punto inicial de t=0 o el día 0 aplicando muestras de 7 a 15 días), la tasa de infección siempre será dependiente del tiempo β(t) disminuye con cada día que progresa, lo que supone que varía exponencialmente en carácter siguiente:

$$\tilde{\beta}(t) = \beta(1-\zeta)e^{-[(t-\tau)/T]} + \zeta\beta \quad \text{for } t > \tau$$
$$\tilde{\beta}(t) = \beta \quad \text{for } t \leq \tau$$

***Ilustración 16**- Parámetro de infección*

En la ***Ilustración* 16**, ک Ɛ {0,1} es el parámetro de infección (o paramento de interacción) y *T* es la demora en el número de días antes de que el efecto del bloqueo sea visible en la propagación de la infección, hasta el momento se observa la formula general del modelo, con el fin de representar un teorema . Sin restricción ک = 1, se

refiere a una infección rápida, mientras que گ = 0 expresa una infección contenido (No hay interacción entre la población infectada y la susceptible, por lo que no hay transmisión). گ ε {0,1} refleja la mitigación asintótica de la tasa d infección β(t), cuando se imponen medidas de contención. Menor el valor de گ, más estrictas son las medidas de contención (o las manifestaciones reaccionadas) *(Archana Singh Bhadauria, Hom Nath Dhungana, 2022)* **[29,30]**.

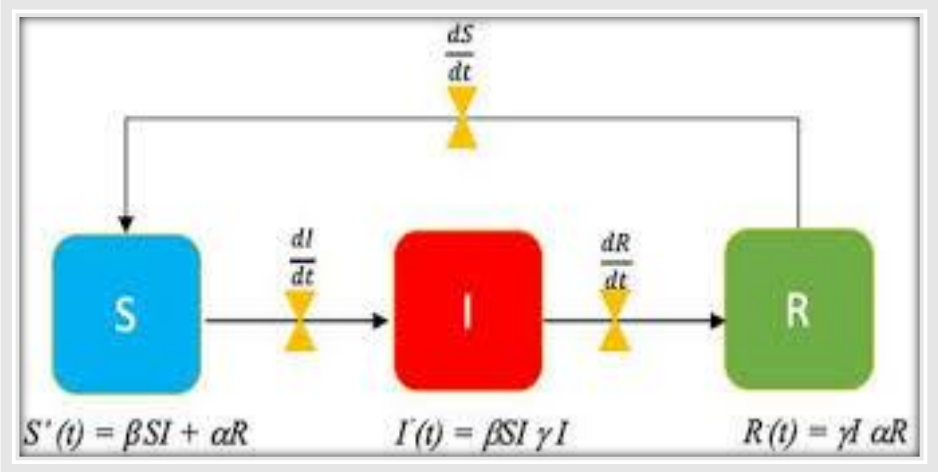


Ilustración 17- *Representación teorema, tasa 'd' infección β (t)*

Aquí, β(t) tiene un valor inicial β (Beta) siendo constante β(t) disminuye con el tiempo y alcanza un valor inicial un valor گ β a medida que continúan las medidas de contención. Especialmente, el valor inicial de β determina las propiedades, características de la enfermedad que dependen de la interacción efectiva de las personas en una región, comportamiento social densidad de población, etc. El valor terminal گ β muestra el efecto de la contención, y como el distanciamiento social se está manteniendo. En la configuración del modelo actual, β(t) está destinado a explicar los cambios en el comportamiento de la propagación de la infección debido al distanciamiento social y las medidas de contención. Este es un parámetro externo que se espera que disminuya con el tiempo a un valor menor گ β cuando se evita el contacto físico. Siguiendo el ejemplo de los estudios anteriores, la forma funcional de β(t) se construye para tener en cuenta los cambios aparentes en el crecimiento de la infección debido a las medidas de contención y el distanciamiento social. *T* (variable

tiempo) controla la velocidad efectiva a la que se produce un momento tardío de la transmisión de la enfermedad debido a las medidas de contención impuestas.

La simulación del modelo, el análisis de datos y el trazado se llevan a cabo en Python. El análisis de los datos de COVID-19, utilizando el modelo determinista compartimental SIRD, arroja buenos resultados sobre las características primarias de la evolución temporal de la pandemia.

3.2. Extraer, transformar y cargar recursos de dataset

Las Metodologías de Aprendizaje Automático *(Natasha K. Niemann and Raymond G. Blockmon, 2021)* **[31]** se utilizan a partir de contenidos de una única tabla o una única locker de datos de estadística , siendo estas recolectadas a partir de una problemática, representando a una constante, dado el caso de registrados del conjunto de datos valorados. Para comprender comportamiento de los datos del COVID-19 que afecta a las personas, su confirmación y predicciones de recuperación, se ocupa dicha valoración en la página de la gobernación *(Government of Buenos Aires City., 2021)* **[32]**, en el apartado de actualización de datos actualizados de Buenos Aires data.

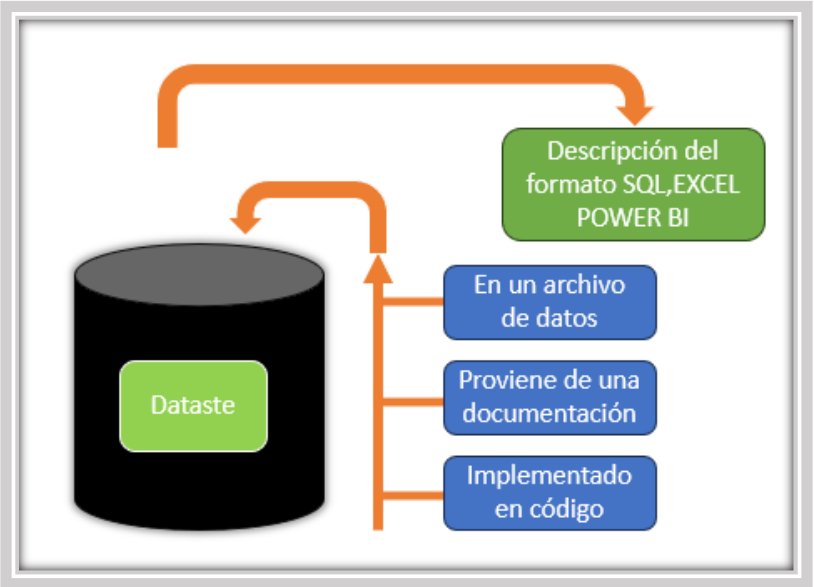


Ilustración 18*-Dataset, proceso de registros del conjunto de datos valorados.*

Habilitando los datos obtenidos en la página mencionada, el procedimiento a seguir es tomar el archivo .CSV (formato del archivo), transformarlo en un tablero

inteligente; para este escenario se implementa la herramienta Power BI para obtener los valores y determinar la variable de entrada (Edades, Infecciones, Sexo, Muertes). A partir de estas variables de entrada, facilitando el entrenamiento del modelo mencionado.

Cabe destacar, la transformación de los datos del archivo raíz fue particionada por bloques, ya que se pretende a ajustar todas las muestras dentro de los datos de entrenamiento, agrupamiento de esas características mencionadas, también estimar los valores perdidos, ya que mantiene la precisión en caso de faltar una parte de los datos en un periodo dado, en los años establecidos (3 años).

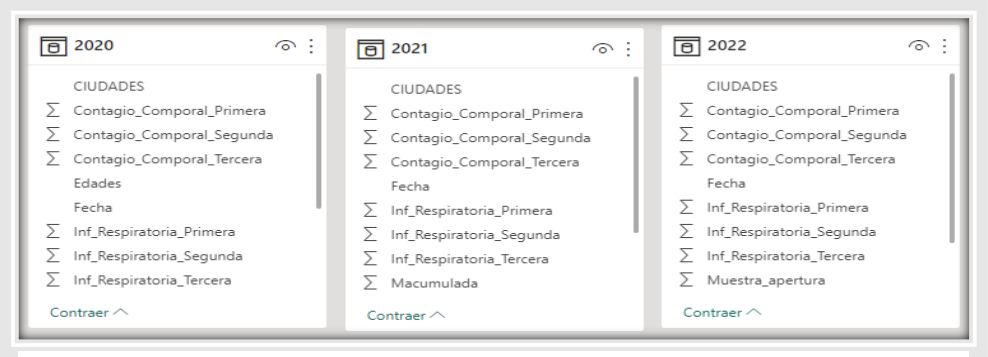


Ilustración 19- *Transformación de datos, panorama pandemia 2020 – 2021.*

Las Ilustraciones a continuación, muestran los diversos grupos de edades y el porcentaje de casos obtenido, en el conjunto de datos. Los grupos de edades de 25 a 82 años tienen una alta probabilidad de infectarse con COVID-19.

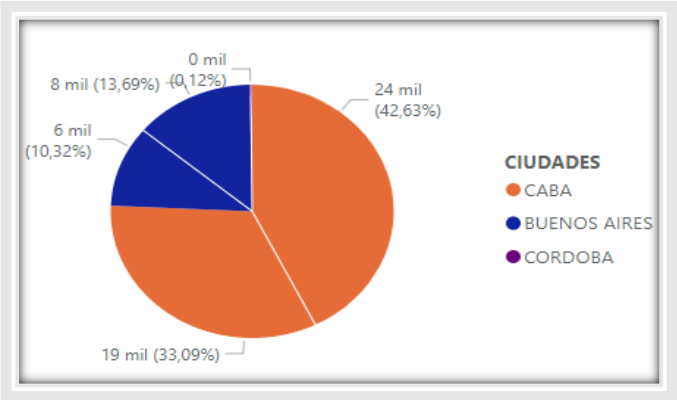


Ilustración 20-*Distribución de casos por edades año 2020*

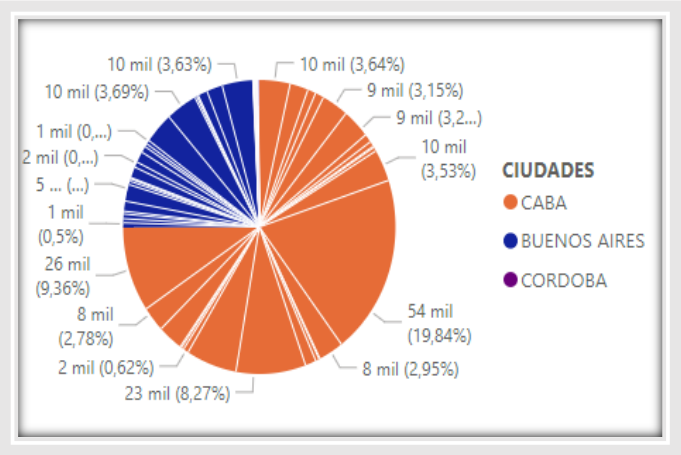


Ilustración 21*-Distribución de casos por edades año 2021*

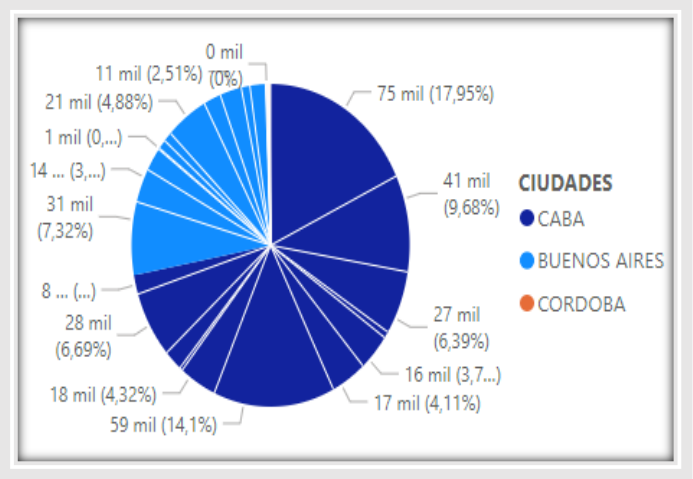
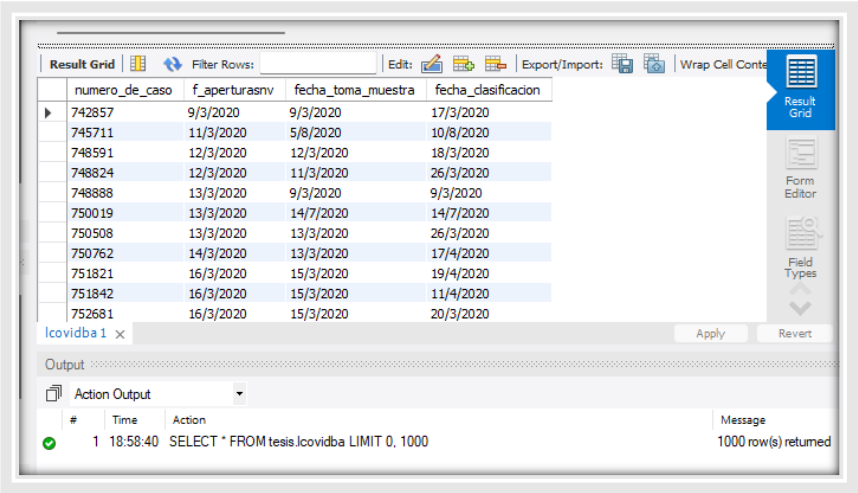


Ilustración 22*- Distribución de casos por edades 2021.*

En relación con la estructura de datos, se completa la manipulación, la extracción de la información (ETL) agrupados por la constante edad, para culminar este procedimiento se implementa la transformación de los datos facilitando el tratamiento de estos. Asimilar los datos transformados proporciona buen manejo de los algoritmos. Es por esto por lo que se efectúa la transformación en un motor de SQL anexado con complementos estadísticos y matriz de cálculo, para la investigación damos uso del motor WORBENCH con el complemento Infostat/Excel. Por ende, el archivo plano tomado del .CSV (archivo base), ya se complementa las variantes de datos, las extensiones permiten representar el modelado de los algoritmos de una manera más volátil.



numero_de_caso	f_aperturasnv	fecha_toma_muestra	fecha_clasificacion
742857	9/3/2020	9/3/2020	17/3/2020
745711	11/3/2020	5/8/2020	10/8/2020
748591	12/3/2020	12/3/2020	18/3/2020
748824	12/3/2020	11/3/2020	26/3/2020
748888	13/3/2020	9/3/2020	9/3/2020
750019	13/3/2020	14/7/2020	14/7/2020
750508	13/3/2020	13/3/2020	26/3/2020
750762	14/3/2020	13/3/2020	17/4/2020
751821	16/3/2020	15/3/2020	19/4/2020
751842	16/3/2020	15/3/2020	11/4/2020
752681	16/3/2020	15/3/2020	20/3/2020

Ilustración 23*-Transformación de datos .CSV a SQL/IDB/XLSX*

3.3. Soporte, despliegue y discusión de la herramienta Jupyter- Algoritmo de árboles de decisiones.

Un árbol de decisión se caracteriza por ser una ramificación de nodos, donde cada nodo muestra características (atributo), cada ramificación (rama) muestra una decisión (condición/regla) y cada hoja muestra una solución a la condición resuelta (categórico o valor continuo) *(Jadhav SD, Channe HP, 2016)* **[33].** Los AD (arboles de decisión) reproducen las ideas conceptuales de las personas, por el motivo de recolectar la información en porciones de datos y efectuar divisiones de mentales para luego ser interpretadas con condiciones. La finalidad de crear un AD en el presente capitulo es para separar las condiciones e identificar las variables/contantes que definan una estructura limpia para el uso de múltiples caminos de simulación y corrección de todos los datos.

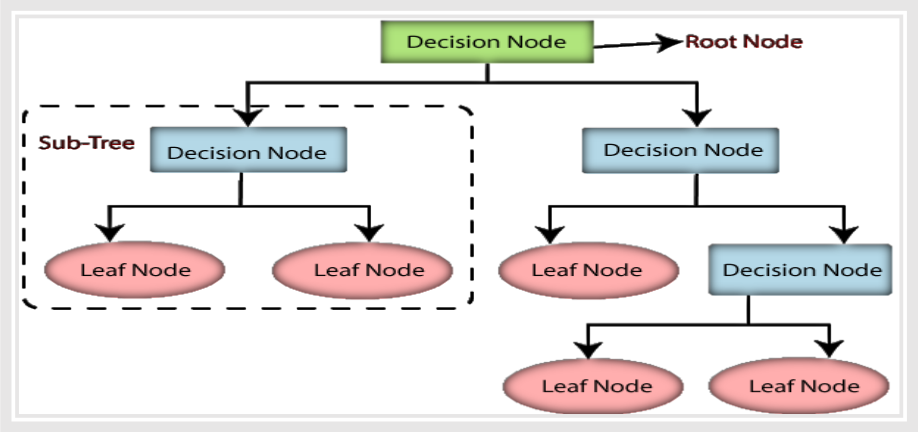


Ilustración 24*- Representación de nodos, motivo de recolectar los datos*

Las características tomadas para la construcción del árbol de decisión, dependerá de un escenario de programación que disponga de los tipos de algoritmos de árboles existente. Estos algoritmos ofrecerán beneficios al dividir el árbol y a la ejecución directa de las condiciones, una vez comprendido este escenario, se procede a seleccionar el algoritmo Random Tree para ser codificado en el ambiente de programación Python con cada uno de los conjuntos de datos que van a simbolizar un resultado categórico **[*Anexo 1*].**

	edad	Muestra	Condicion_decision
0	33	0	2
1	21	0	2
2	73	0	2
3	64	0	2
4	32	24	1
...	...	...	...
67	45	14	0
68	35	25	0
69	32	22	0
70	54	64	0
71	19	341	0

***Tabla 2**-Características de los datos iniciales*

La ***tabla 2***, facilita la visibilidad de los datos a implementar en primera instancia, para luego ofrecer la conclusión de una sola vista y comparar las alternativas que conformaran la data posterior a la ejecución de los siguientes algoritmos, SIRD (modelo base) y BAYES (modelo de estructura).

El AD hace explicitas todas las alternativas posibles y rastrea cada opción, estas alternativas deben y cumplir con las características de un árbol. Las características deben ser Objetiva y Explicativas. Por ende, seleccionamos las 3 columnas bases, luego estas se ramifican en el árbol transición, aplicando la secuencia de decisión final. Las características Objetivas despliegan una naturaleza de comprensibilidad numérica, en este escenario la columna 'Muestra' de casos de una posible virosis toma la identidad de una característica Objetiva. Las características Explicativas son interpretaciones categóricas, estas son representadas por la columna 'Condicion_desicion'.

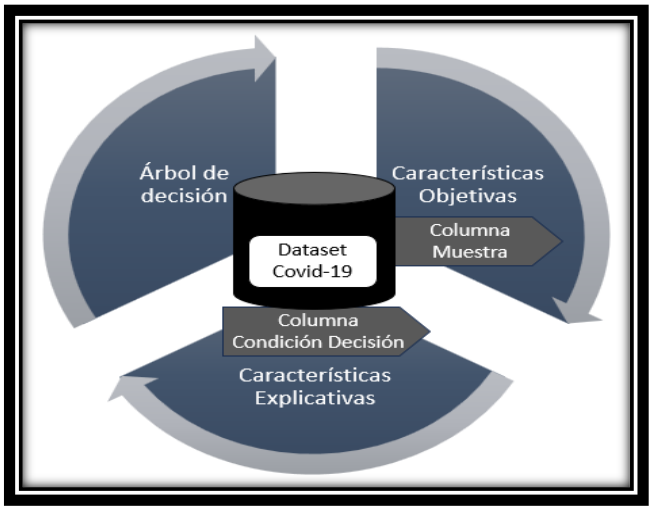


***Ilustración 25**-funcionamiento de las características del AD*

La selección de las variables y la sección de las características son lo suficientemente buenas para consolidar el primer árbol de decisión. En cuanto al su rendimiento, las posibles variables externas (nuevas) no afectaran ningún parámetro del árbol de decisión, siempre y cuando se respete las columnas para cada característica.

La división de los atributos se prueba en cualquiera de los nodos, para determinar si la es 'mejor' en clases individuales. Pero en esta ocasión los mejores nodos resultantes se particionarán nuevamente para logar una precisa evolución de los datos que van a ser implementados para los algoritmos solución.

Ampliar los factores calificativos de los datos en cada una de las clases intervenidas, para esta primera instancia se debe visualizar las clasificación generadas por los datos y el proceso de entropía (Medida de la incertidumbre) por el cual se ofrece un dictamen de salud y permite agrupar los datos funcionales para la segunda fase.

Las ***ilustraciones 10-12*** ofrecen una ventaja, con respecto a la capacidad de seleccionar las características más sesgadas y provee una compresibilidad latente en los 3 años donde transcurre las variantes epidemiológicas en la población.

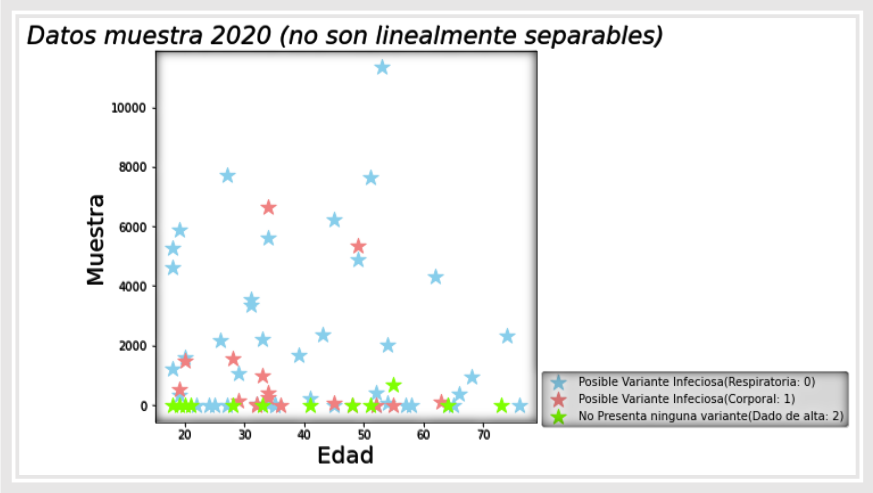


Ilustración 26- *Diagnostico 2020, primer agrupamiento*

La ***Ilustración 26***, dispone de los siguientes resultados: a medida que va aumentando la edad, va incrementando alguna posible variante epidemiológica (fase de inicio Covid19). Esta gráfica muestra que los datos no son linealmente separables, es decir que si tenemos una línea no la podríamos clasificar y esto nos muestra también parte de la etapa de los árboles de decisión.

Rudimento	Clasificación	Descripción
Algoritmo ID3 (edades entre 0-20)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	Aplicando división numérica, en los datos se percibe pocos datos confirmados, reflejando un rendimiento de ninguna variante y otras en estudio.

Algoritmo ID3 (edades entre 20-30)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	La distribución de los valores para este rango agrupa la mayoría de la población con sospechas de alguna infección debido a cantidad de individuos que se sumaban a los reportes, definiendo un valor simple de ser la población más fuerte.
Algoritmo ID3 (edades entre 30-40)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	La porción con más Covid identificada, representando el rango de edad con más infecciones confirmadas y agrupadas debido a que no cumplen con la condición.
Algoritmo ID3 (edades entre 40-50)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	La agrupación para este filtro pertenece a los resultados imparciales, conteniendo nuevos casos positivos y una porción recuperada de la infección.
Algoritmo ID3 (edades entre 50-60)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	Las infecciones (positivas), tuvieron una constante en este rango de edades puesto que siempre presentaron otro tipo de variante epidemiológica o de salud.
Algoritmo ID3 (edades entre 60-70)	Variante infecciosa (asintomática) Variante infecciosa (confirmada) Ninguna variante	La muestra de datos para cualquier variante es muy mínima

Tabla 3-*Distribución y componentes del árbol de decisión 2021*

Con este procedimiento, podemos apoyar una clasificación más limpia, cuenta con un bloque de resultados de datos con valores separables, al no comprensibles para los algoritmos planteados, disponemos de este procedimiento para lograr un procesamiento de valores en conjunto. La ***ilustración 11***, reúne los progresos del conjunto de datos del año 2021 que nos permitirá crear los árboles de decisión con mayor precisión de agrupamiento de datos, para este escenario es evidente el análisis obtenido, va aumentando la edad, ya los casos son identificados con mayor claridad y

sin supuestos clasificatorios, incrementa la variante epidemiológica (fase de intermedia del Covid19). **[*Anexo 1*].**

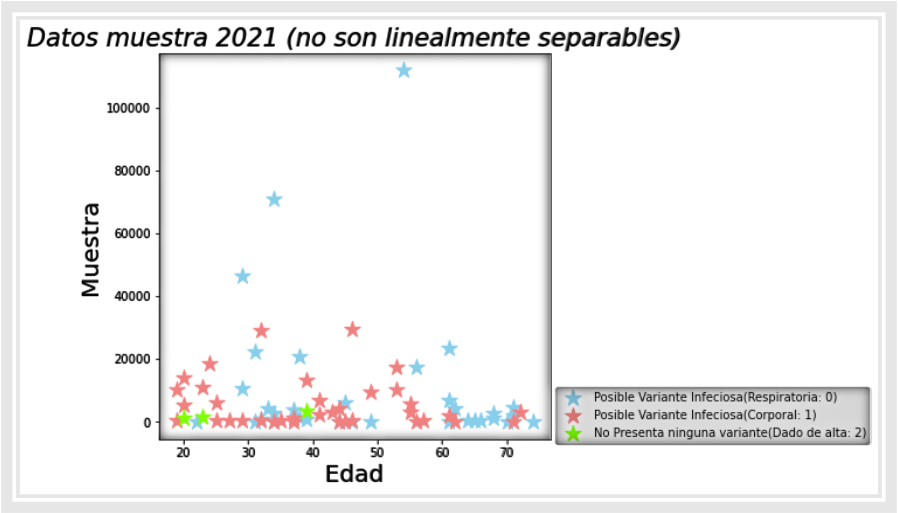


Ilustración 27*-Diagnostico 2021, primer agrupamiento*

Rudimento	Clasificación	Descripción
Algoritmo ID3 (edades entre 0-20)	Variante infecciosa Ninguna variante	La partición inicial, señala casos puntales de jóvenes de 18-19 años, con la variante marcada de covid-19, son muy escasos los datos, con ninguna variante que se replique para las siguientes edades en porciones de datos menores al anterior caso.
Algoritmo ID3 (edades entre 20-30)	Variante respiratoria Variante infecciosa Ninguna variante	Una única observación con los valores discretos. Por lo tanto, el conjunto de datos continuos dando ningún caso grave.
Algoritmo ID3 (edades entre 30-40)	Variante respiratoria Variante infecciosa Ninguna variante	Estimar la primera porción de valores negativos y los agrupo como principal atributo para repartir el caso a una nueva estructura de datos resultados.
Algoritmo ID3 (edades entre 40-50)	Variante respiratoria Variante infecciosa Ninguna variante	Se ocupa de un conjunto de datos más dispersos ofreciendo un incremento de muchas desconfianzas debido a pertenecer a la población más sensibles, pero con pocos casos encontrados como peligrosos.
Algoritmo ID3 (edades entre 50-60)	Variante respiratoria Variante infecciosa	Estima los valores negativos con una frecuencia más rápida y con una población

	Ninguna variante	más baja debido a su función de asimilar otros atributos o repartir el caso estadísticamente entre los resultados.
Algoritmo ID3 (edades entre 60-70)	Variante respiratoria Variante infecciosa Ninguna variante	Los supuestos disminuyen al existir poca población valorada y el algoritmo de árbol a esta altura cumple la condición inicial aleatoria por muestra de casos

***Tabla 4**-Distribución y componentes del árbol de decisión 2021*

La ventaja principal de este procedimiento fue seleccionar la característica más sesgada con una la naturaleza más comprensible. En la ***ilustración 12***, los resultados se encuentran totalmente limpios aplicando los conceptos que no son realmente separables, los conjuntos de datos continuos y reflejan esta última parte del año donde los casos positivos se manifestaron solamente en sondeos de poblaciones superiores 20000 personas y descartando la totalidad de infecciones o en su defecto personas recuperadas. **[Anexo 1].**

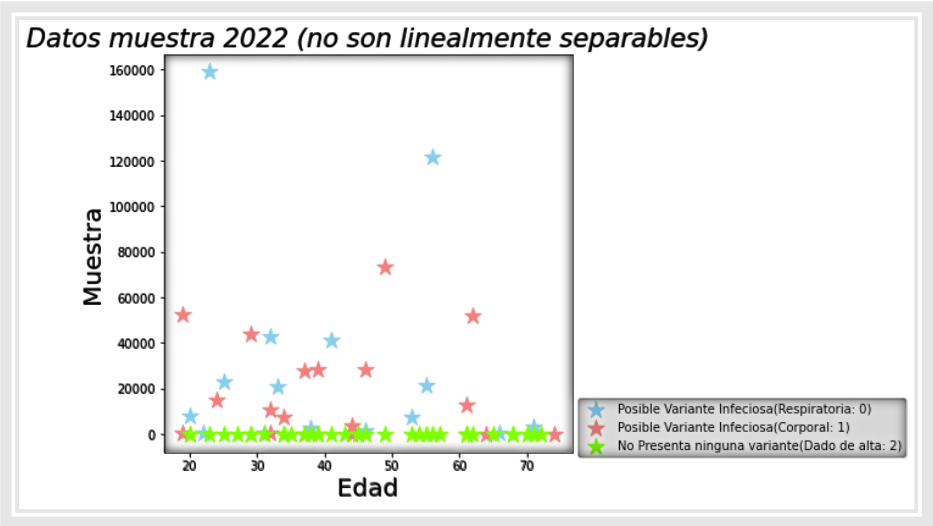


***Ilustración 28**-Diagnostico 2022, primer agrupamiento*

En el presente escenario no se ve la necesidad de plantear una tabla debido a la continuidad estadística de los casos agrupados con resultados altamente similares con respecto a la pandemia vivida.

3.4. Armado y toma de condiciones- Algoritmo de árboles de decisiones.

Según los valores del atributo de división del capítulo anterior, los datos de entrenamiento se dividen en varios subconjuntos correspondientes a los años dispuestos para efectuar la investigación, asumiendo las características de no ser linealmente separables. Todas las instancias se moldearon al subconjunto perteneciente a la misma clase, permitiendo que cualquier árbol de decisión los agrupe de forma más compacta y los algoritmos proceden a estar mejor alimentados con sus datos. A continuación, se

plantea un recorrido de cómo se arma y toman las condiciones para la realización del árbol de decisiones.

- **Cargar los nuevos datos:** La transformación y consolidación de los nuevos datos realizados en el subcapítulo anterior ,permitirá construir un árbol de decisión con variables más completas. Por ende, se carga el nuevo modelo de tabla para luego seleccionar las variables competentes.

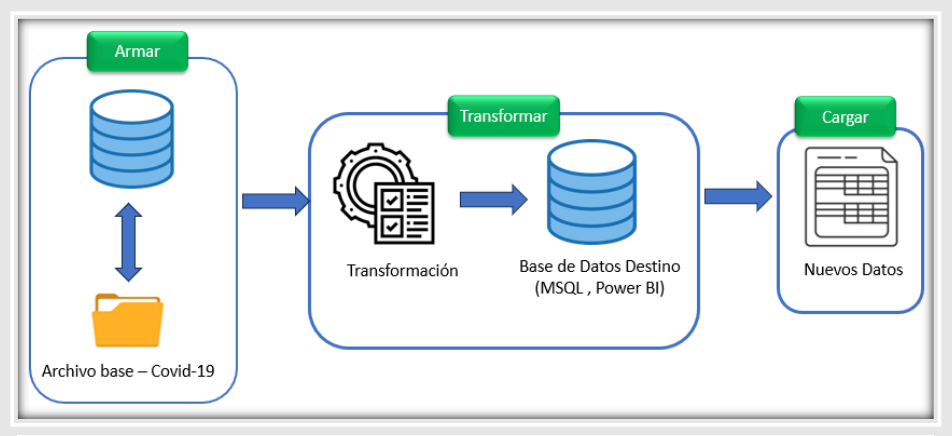
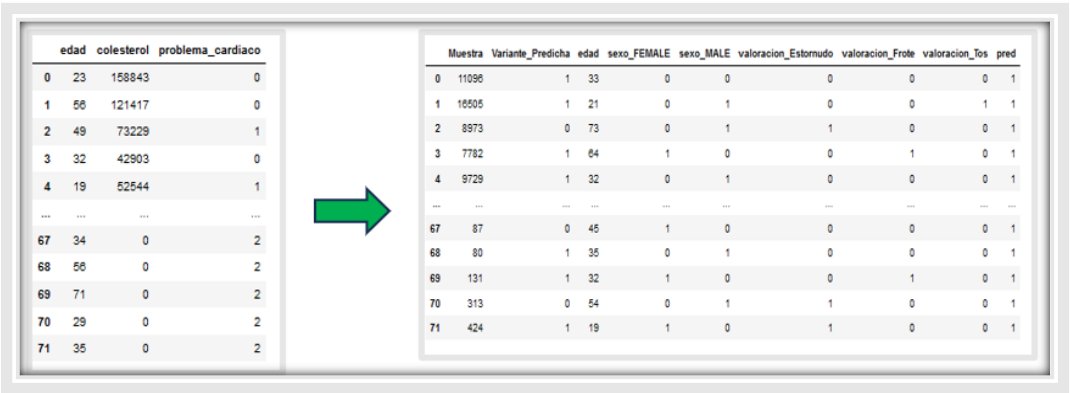


***Ilustración 29**-Procedimiento de cargar datos.*

- **Seleccionar variables:** El número de variables óptimas para realizar el bosquejo del árbol de decisión dependerá al cumplimiento de las condiciones y el criterio categorías planteadas, cuyo escenario se dispone de dos categorías. Se establecen dos variables: Objetiva y Explicativa. Dentro de la variable Objetiva se toma toda la columna del DataFrame 'Variante Predicha' y la objetiva se calcula a partir de las dos constantes numéricas 'muestra y edad'.

  En segundo escenario se toman 4 variables transformadas, estas se estructuraron a partir de los datos resultantes de la primera capa del árbol de decisión del procedimiento anterior por lo cual se construye una nuevo DataFrame en base a los datos resultantes. **[*Anexo 2*].**



	edad	colesterol	problema_cardiaco
0	23	158843	0
1	56	121417	0
2	49	73229	1
3	32	42903	0
4	19	52544	1
...	...	...	...
67	34	0	2
68	56	0	2
69	71	0	2
70	29	0	2
71	35	0	2

	Muestra	Variante_Predicha	edad	sexo_FEMALE	sexo_MALE	valoracion_Estornudo	valoracion_Frote	valoracion_Tos	pred
0	11096	1	33	0	0	0	0	0	1
1	16505	1	21	0	1	0	0	1	1
2	8973	0	73	0	1	1	0	0	1
3	7782	1	64	1	0	0	1	0	1
4	9729	1	32	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...
67	87	0	45	1	0	0	0	0	1
68	80	1	35	0	1	0	0	0	1
69	131	1	32	1	0	0	1	0	1
70	313	0	54	0	1	1	0	0	1
71	424	1	19	1	0	1	0	0	1

***Ilustración 30**- Transformación de datos, árbol de decisión a RJ48*

A partir del nuevo DataFrame se agrupan 4 variables nuevas (***Ilustración 13***), en dos sub-grupos, objetivas y explicativas. Para el subgrupo de variables Objetivas seria la columna Muestra y Sexo, para el subgrupo de variables Explicativas se toman las columnas de variaciones: Estornudo, Frote, Tos.
Con estas nuevas variables se estructura el otro modelo de ***árbol RJ48.***

- **Entrenar el modelo:** Para entrenar modelo se implementa la librería ***sklearn. model_selection import train_test_split (funciones/métodos codificados),***con ella resolvemos el tema de consolidar nuestro árbol en un objeto con las variables, consideradas en el punto anterior. Este algoritmo tomara las variables Objetivas y Predichas, la clase asociada a esos pacientes (Objetiva), no va a separar los datos de entrenamiento, simplemente creara el modelo y en datos de prueba (Explicativa) y así determinar qué tan bueno o malo es nuestro modelo de aprendizaje de máquina. La prueba se ejecutará para los años estudiados (2020-2023), se utiliza un tamaño de testeo=0.30 facilitando la investigación de los contactos y detectan infección aun 30%,con esto ya tengo mis datos preparados para el entrenamiento de árbol de decisión.

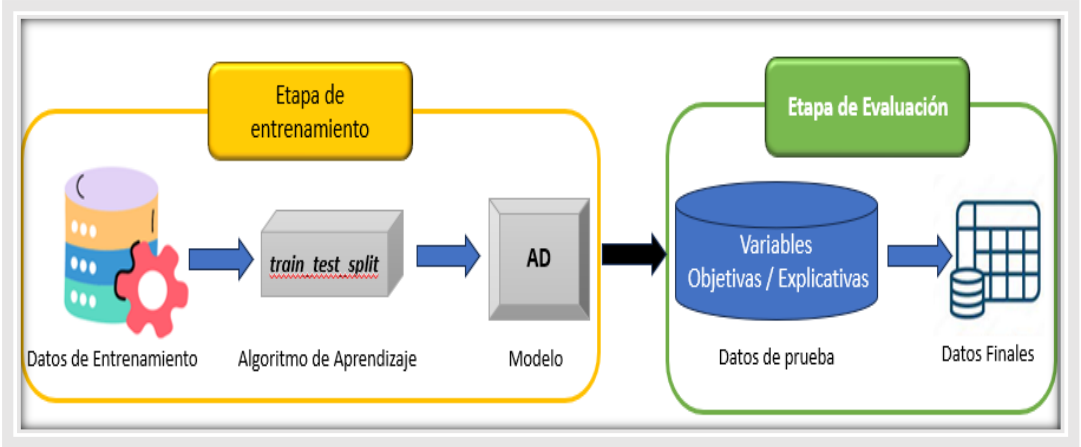


Ilustración 31*-Entrenar el Modelo, Etapa del AD*

Para el segundo modelo se implementan los mismos procedimientos de programación del escenario anterior, en el entorno Python. Este algoritmo presentara un cambio en cuanto a sus variables, debido a: se modelan 4 variables nuevas (fueron transformadas), sin embargo, las clases asociadas siguen siendo las mismas. La prueba se ejecuta para la misma fase (en años), se utiliza un tamaño de testeo= 0.20, disminuyendo un 10% con respecto a la anterior medición, todo lo demás permite preparar los datos correctamente para el entrenamiento del árbol de decisión RJ48. Este algoritmo ofrece un análisis practico y óptimo de los datos ingresados, sus características estuvieron modificadas para adaptarlo al nuevo árbol, este se caracteriza por ser menos denso y más comprensible en los datos, alcanzo un 78.5% de clasificación de las variables, dejando un 2% de sesgo de un posible error de programación. [***Anexo 3***]

- **Visualizar el modelo:** Con el método ***tree.DecisionTreeClassifier*** **[Anexo-código]** en Python creamos el árbol, con el criterio de entropía (probabilidad categórica), podemos calcular la profundidad del árbol, es decir cuantas ramas se calculan y van a ser consideradas, las ramas no tienen un numero especificado de profundidad para cumplir una condición, estas actuaran como evaluadores del modelo entre más ramas se profundice más acertado se volvería el modelo, estas predicciones se basan en la cantidad de datos. A continuación, se ilustra los resultados para los años de evaluación con las siguientes características. Dos variables, criterio de entropía, el objeto creado "arbol_decision" con su metodo *fit.* ***[Anexo 4***].

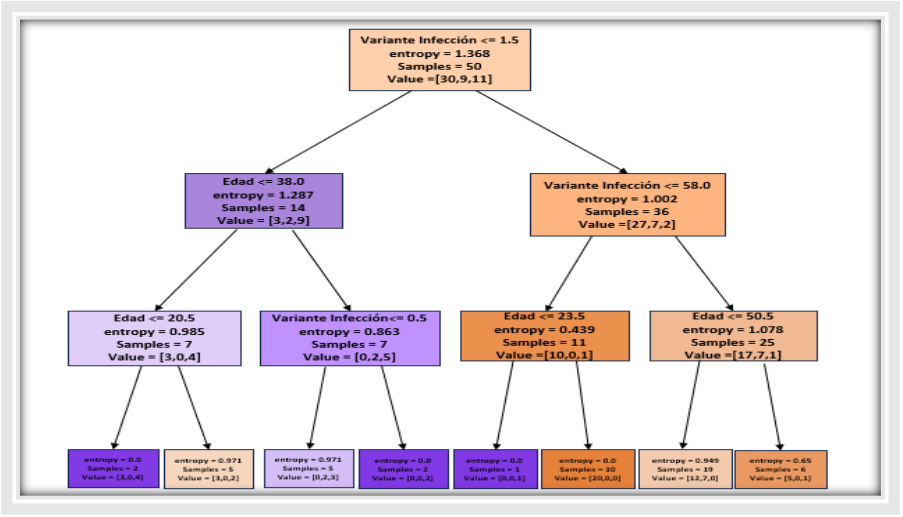


***Ilustración 32**- Árbol de decisión, datos de entrenamiento 2020*

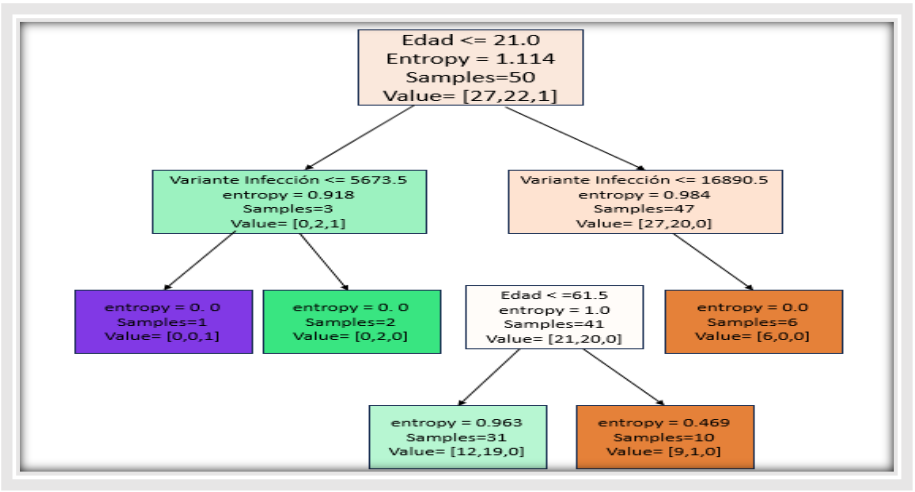


***Ilustración 33**- Árbol de decisión, datos de entrenamiento 2021*

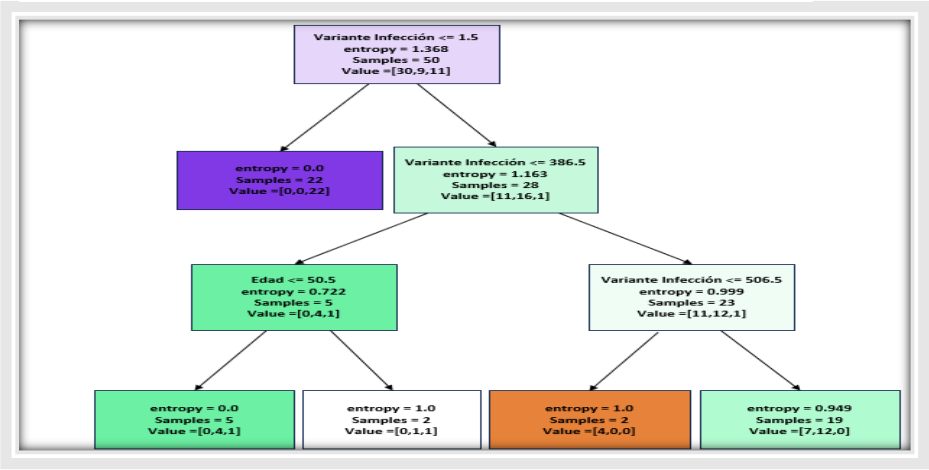


***Ilustración 34**- Árbol de decisión, datos de entrenamiento 2022*

Las Ilustraciones anteriores ofrecieron una ganancia de información, donde está fue sesgada (comprimir los datos a favor) hacia los atributos desempeñados a ser las variables de entrada/salida del algoritmo, el principal funcionamiento de la ganancia de información. La división desequilibrada de datos de la pandemia a través de los

años, determino en cada uno de los nodos secundarios, presentara más entradas de clases (columnas de datos) en comparación con los demás nodos. La relación de ganancia generalmente prefiere este escenario, depender de pocas ramas para poder visualizar la capacidad del modelo árbol de decisión. Los índices de Variante infecciosa proyectan resultados desfavorables a inicios del año 2020 y todo el año de 2021 porque tiene más de dos categorías en el conjunto de datos donde determino más de una variante infecciosa.

A continuación, se efectúa otro tipo de modelo de árbol de decisión para otras dos variables (Objetiva y Explicativas). Las nuevas variables están asignadas a las columnas 'sexo' y '*muestra*' respectivamente. Las ilustraciones a continuación ofrecen otro panorama, una mayor precisión con respecto al desempeño anterior.

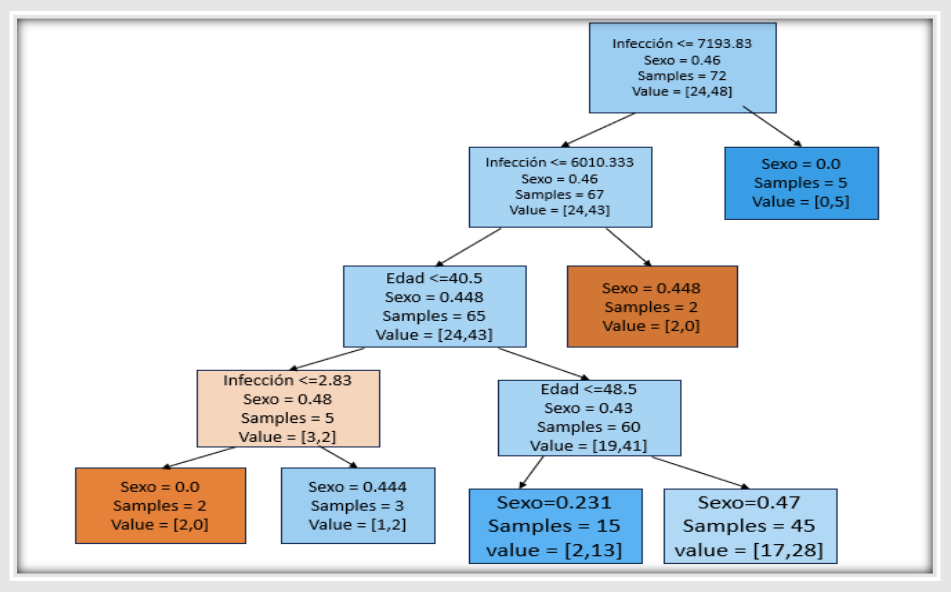


Ilustración 35- *Árbol de decisión J48, datos de entrenamiento (+4 variables) 2020*

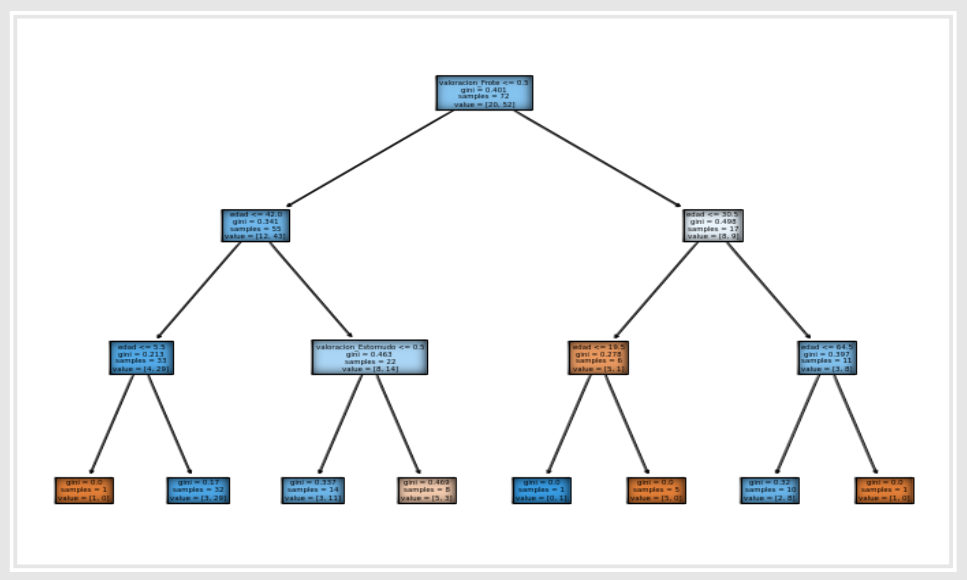


Ilustración 36- *Árbol de decisión J48, datos de entrenamiento (+4 variables) 2021*

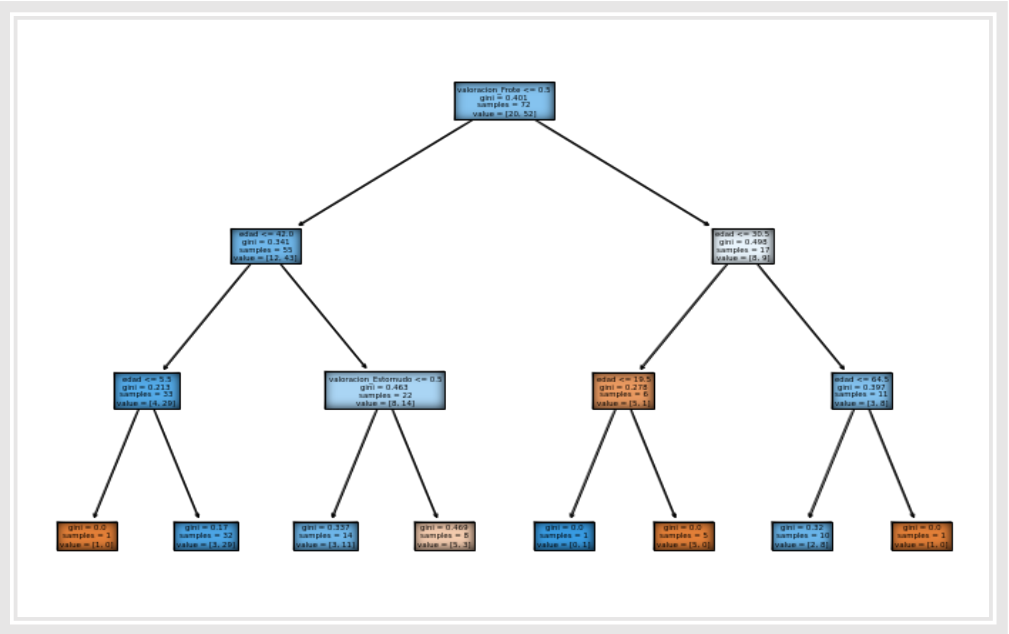


Ilustración 37- *Árbol de decisión J48, datos de entrenamiento (+4 variables) 2022*

Para esta presentación el ***algoritmo J48*** utilizado en paralelo, logra ser un análisis óptimo de los datos ingresados, las características se adaptaron mejor al modelo anterior, puesto que presenta un árbol como más subdivisiones y menos extenso en porción de cantidad con respecto a sus condiciones, alcanzando una clasificación más rápida.

La función principal de este caso fue agregar nodos o ramas con el fin de minimizar la diferencia de los datos. Por tal motivo si se desea explicar y predecir características u observaciones pertenecientes a los objetos de una clase cuyas bases pueden ser variables explicativas o cualitativas, utilizar modelos de árboles de clasificación y por otro lado para un modelo explicativo y predictivo para una variable cuantitativa dependiente cuyas bases sean variables cuantitativas de igual forma.

- **Interpretar modelo:** Concluimos interpretando los resultados obtenidos en ambos modelos de árboles de decisiones, estas interpretaciones van desde los valores internos de cada nodo hasta la comparativa de ambos modelos y su consolidada estructura para implementar y ejecutar en los próximos modelos de algoritmos, las ilustraciones a continuación (***Ilustración 20,21,22***) brindan una explicación más sensible a lo desarrollado.

Nombre	Cantidad de Niveles	Condición inicial	Numero de Capas	Mejor Condición	Condición no Favorable
Árbol RJ 48 - 2020	De nivel 0 a nivel 4	Toda infección debe ser menor a 7193, es decir las 3 sub-clases de infección debían sumar un resultado menor.	Corresponde a 4 capas, es decir el ultimo nivel posee 4 nodos decisión.	Nivel 1, capa 2 correspondiente al nodo con los siguientes valores: Sexo: 0.0 Samples: 5 Value: [0,5]	Nivel 1, capa 1 correspondiente al nodo con los siguientes valores: Infección <=6010 Sexo: 0.46 Sample: 67 Value: [24, 43]
		La valoración	Corresponde a	Nivel 3, capa 5	Nivel 1, capa 1

Árbol RJ 48 - 2021	De nivel 0 a nivel 3	del brote debe ser menor a 0.5, es decir debe existir un caso sospechoso por frote	la 8 capa, es decir el ultimo nivel posee 8 nodos decisión	correspondiente al nodo con los siguientes valores: gini: 0.0 Samples: 1 Value: [0,1]	correspondiente al nodo con los siguientes valores: Edad <=42 gini: 0.341 Sample: 55 Value: [12, 43]
Árbol RJ 48 - 2022	De nivel 0 a nivel 3	La valoración del brote debe ser menor a 0.5, es decir debe existir un caso sospechoso por frote	Corresponde a 8 capas, es decir el ultimo nivel posee 8 nodos decisión	Para este escenario existen dos escenarios favorables. Nivel 3, capa 5 correspondiente al nodo con los siguientes valores: gini: 0.0 Samples: 1 Value: [0,1] Nivel 3, capa 8 correspondiente al nodo con los siguientes valores: gini: 0.0 Samples: 1 Value: [1,0]	Nivel 3, capa 2 correspondiente al nodo con los siguientes valores: Valoración estornudo <=0.5 gini: 0.463 Sample: 22 Value: [8, 14]

***Tabla 5**-Valores y características, bosquejo de resultados de los datos.*

En este orden de ideas, se visualizó en primera instancia una serie de nodos con nomenclatura y condiciones similares que estuvieron presentes en ambos modelos de árbol de decisión y RJ-48, por ende, en la ***tabla 5*** resaltamos una serie de consideraciones básicas para comprender un poco los valores obtenidos . Cada nodo debía ser particionado de acuerdo con su condición, en ambos modelos de árboles las condiciones se basaban de acuerdo con sus características definidas dentro de la variable explicitas/objetivas, es decir las columnas involucradas en el apartado '*Seleccionar Variables*', estas condiciones regían una condición aleatoria para cada nodo raíz.

En el primer modelo de árbol de decisión, la condición rige sus filtros a partir de los atributos del dataFrame, siendo la columna Infección. Dicha columna tomara las condiciones numéricas para elegir las 3 ramas o subdivisiones creadas por el algoritmo de árbol de decisión. A continuación, se aborda una breve descripción a los recorridos de cada árbol, a partir de las ilustraciones 14-21. ***[Anexo 4]***

- ✓ ***Árbol de decisión, año 2020:*** La selección aleatoria realizada por el primer nodo refleja un resultado optimo, en tomar una condición referente a los casos (muestra) donde el filtro principal se basa en un caso y medio (0.5), siendo un alto porcentaje de infección referente al Covid-19. Este procedimiento es favorable para adaptarlo al modelo SIRD, ya que el modelo dispone de una condición necesaria, la cual es iniciar con un individuo infectado. Por ende, la simulación inicial se ajusta de la mejor manera al modelo de datos, por el cual facilita procedimientos a futuro.

En el nivel 0, los nodos se distribuyen para dos condiciones, la primera: seguir distribuyendo el número de casos de personas infectadas y la otra rama consolida al disponer por la columna '*Sexo'* de las personas que poseen un síntoma indiferente a la variante Covid-19. (Ver ***Ilustración 14***)

Cada nodo dispone de constantes internas, estas se calculan con el fin de consolidar una información más certera, por ende, cumplen una característica en la selección de los datos. Las características son:

- **Entropy =** Corresponde al criterio de entropía, es la cantidad de aleatoriedad en los datos. Este nodo depende de la cantidad de datos aleatorios encontrados, para cada nodo. El resultado de 1287, es el valor más pequeño seleccionado para el árbol, esté dispone de una entropía más pequeña en sus nodos y por ende devolvió el valor con más alto, en ganancia de información.
- **Samples =** es el valor de muestra que cumplen con condición correspondientes a la evidencia, es decir si la muestra de datos es > (mayor) de 1.5, personas con la infección desarrollada, esa seleccionara el lado derecho, en su defecto ira al lado izquierdo. En este nodo, Samples:14 es la muestra de valores que cumplen con la condición menor 1.5, es decir 14 personas no presentan ninguna condición infecciosa por Covid-19.
- **Value=** es una etiqueta vector que representa las posiciones de las filas versus las columnas. Continuando con la condición de la izquierda (x <=1.5), el valor resultante Value: [3,2,9] determina lo siguiente: dada la columna 3, fila 2 logro conseguir el valor óptimo de este nodo.
- **Variante de infección**= visualizando el árbol donde la condición es mayor a 1.5, el algoritmo sigue agregando automatizaciones referentes a las muestras aleatorias de los casos evaluados. El nodo con la siguiente condición <=58, traslada la inspección a una muestra mayor.
- **Sexo=** la característica surge cuando la condición se cumple puesto que, al conocer la población afectada, se debe definir el número de personas agrupadas por Masculino/Femenino.

Los demás nodos siguen la misma dinámica para evaluar cada rama/nivel del árbol, a medida que se cumpla la condición esta ira señalando las características con un nuevo filtro. Si la condición no cumple solo mostrara un nodo con valores y el vector posicional

- **Árbol de decisión año 2021:**

La selección aleatoria en este escenario vario significativamente y realizo un cambio posicional de una característica, es decir vio la necesidad de definir de inicio la edad como condición principal, esto se debe a al estado de datos, debido a que las muestras ya presentan casos confirmados por infección Covid-19. Sin embargo, el algoritmo de árbol de decisión cumple con su función de inspeccionar todos los valores y agruparlos con la condición más optima, esta condición por consiguiente posee inconsistencia referente a la población que no presenta una variante infecciosa, esto se debe a que existieron casos donde las personas tenían los síntomas y su calificación siempre fue Asintomática.

Para este escenario, el procedimiento vario levemente, el nivel 0, los nodos se distribuyen para dos condiciones, ambas distribuyen la condición 'Variante de Infección'. Las condiciones difieren, una confirma directamente las personas con Covid 19 y la otra simplemente reubica las personas con valores de incertidumbre bajos, con buenas probabilidades de mantener una tasa de personas sin una posible

variante de infección.

Como en el escenario anterior, cada nodo dispone de constantes. Las características son:

- ✓ **Edad =** Corresponde a la condición principal, con la que el algoritmo efectua las comparaciones y así estructura los demás niveles del árbol. El valor obtenido <=21.0, es acorde a la situación debido a que a partir de esa edad se pronuncia un el análisis de una posible variante y al estar en un año con mayor presencia de variante covid-19 no es ajeno a la precisión del algoritmo de tomar la edad promedio de los casos frecuentes.
- ✓ **Entropy =** El valor de 1.14 fue el valor más pequeño seleccionado para alimentar este árbol, como en el escenario anterior la entropía más pequeña es seleccionada y por ende devolvió el valor con más alta ganancia de información y a partir de allí efectúa medidas de asignación de valores menores.
- ✓ **Samples =** En esta ocasión se convierte en una característica del nodo principal. Su valor 50 se instala a ser la muestra aleatoria que dispuso el árbol de modelo para este conjunto de DataFrame y no fue la condición inicial como en el escenario anterior.
- ✓ **Value=** Cumple la misma condición del árbol anterior, el cambio que se genera, es un nuevo vector con otro tipo de valores [27,22,1] avisa lo siguiente: dada la columna 27, fila 22 logro conseguir el valor óptimo de este nodo que fue 1.
- ✓ **Variante de infección =** no prevalece como condición inicial debido que el algoritmo ya asume e identifica al instante el cambio de casos de variante infecciosa. Por ende, se convierte en una condición simultánea en las demás capas/ramas del árbol. Para ambos partes las condiciones están simplificadas por el factor ruido (izquierda de la gráfica) y su evidencia. El factor ruido refleja que el algoritmo aun percibe de manera perfecta los casos confirmados con variante Covid y sigue extrapolando valores semejantes a una persona sin infección. El valor para considerar es <= 56323, esta escala es muy alta y se debe a la duda que persiste en el algoritmo, puesto que para esta instancia ya había muchos casos confirmados y casos reajustados para otra valoración.

El factor evidencia define con un criterio más acertado las condiciones simultaneas que va considerando el algoritmo para ir identificando los rangos basados en edades de las personas con infecciones confirmadas.

Los nodos de los siguientes niveles cumplen la misma consideración del nivel/rama 0, si la condición no cumple tomara el camino del ruido/error y si este cumple su condición inicial tomar el camino del factor evidencia, por el cual mostrara un nodo con valores y los vectores posicionales.

- **Árbol de decisión año 2022:**

Las condiciones presentes para este árbol son de transición, con respecto al tiempo de la pandemia, su nodo inicial dispone de una condición primaria, se percibe un coste bajo de predicción, esto se debe a sus características con valores altos y un vector promedio que cumple un estado intermedio de los valores.

- ✓ **Variación Infección =** El valor de <= 1.5, estipula como un valor nominal con respecto a la identificación del próximo modelo, es decir a partir de este valor se construye el próximo DataFrame para evaluar las distribuciones posteriores a este escenario. Regresando a este valor refleja un mismo valor inicial del árbol construido para el 2020. Se refuerza el nodo inicial con la característica '*Value'*, es decir si el valor es <=1.5 esta decisión se ubicará en otro nodo con el

agrupamiento de vectores posicionales en los extremos de los datos.

- ✓ **Samples =** La muestra inicial es de 50 individuos seleccionados y agrupados para ser evaluados. Para esta instancia los valores de la muestra son muy escasos y su procedimiento de agrupación con respecto a la condición inicial permite que el árbol se resuelva con pocas ramas. Los niveles de la derecha conservan muestras superiores a 25 personas, esto se debe a que el algoritmo esta mencionando que la data ya debe ser modificada para otro modelo de datos y los niveles de la derecha ofrecen una data mínima para indicar que ya no podría evaluar más condiciones.

- ✓ **Value =** El vector raíz identificado [30,9,11], es un resultado aleatorio tomado por el algoritmo RJ-45, los demás vectores poseen una característica en común y el referente a sus posiciones 0 con valores 0, estos valores los identifica de inmediato debido a lo que se mencionó, al no haber valores identificados el algoritmo simplemente cierra el sondeo en posición columna o fila.

- **Árbol RJ-48 año 2020:**

Las condiciones percibidas en el nodo inicial marcan una predicción y agrupamiento más fuerte, debido al análisis de más columnas de evaluación. El valor obtenido genera un poco de ruido debido al aprendizaje inicial del algoritmo RJ-48, este asimila todas las transmisiones por variante Covid, como casos confirmados y aun no se logra considerar la diferencia de estos en casos asintomáticos o definir si la población cumple con un alto índice de infección. Es natural presentar este escenario debido a su primera evaluación, se refleja en el refuerzo que tuvo el primer nodo con características muy sólidas y específicas. Las características ofrecen un porcentaje de precisión en cuanto al valor más optimo que puedo encontrar, viendo el valor del '*Sexo'*, la población seleccionada es la femenina, al indicar los individuos menos afectados con el nuevo modelo de datos. A su vez se visualiza un cierre en su lado derecho, donde efectuar un agrupamiento para el sexo masculino y no logro conseguir un nodo fuerte para seguir expandiendo sus condiciones, por ende, la división de las condiciones es rápidas, el lado derecho prosigue agrupando condiciones y características de forma continua y su lado derecho no logro ampliar más condiciones. **[*Anexo 5*]**

Como en el escenario anterior, cada nodo dispone de constantes. Las características son:

- ✓ **Infección =** El valor inicial de <= 7193 seleccionado por este algoritmo es un importe alto, como se mencionó en el introductorio del árbol RJ-48 que es un número con un cierto grado de ruido, pero que a su vez fue reforzado por una serie de características que ofrecieron un mejor panorama en sus ramas/niveles. En los demás nodos refleja caminos acotados por la característica fuerte, Sexo. Dicha característica disminuye su porcentaje a medida que este nodo es dividido y finalizado.
- ✓ **Samples =** La muestra inicial es de 72 individuos seleccionados y agrupados para ser evaluados. Para esta instancia los valores de la muestra son relativos, su aprendizaje es lento y por ende selecciona un rango poblacional inferior a 100 individuos, corresponde a su procedimiento por defecto. Las muestras son cerradas cuando la característica '*Sexo*' se encuentra de por medio y a su vez la característica fuerte va disminuyendo con respecto a los valores correctamente visualizados.
- ✓ **Value =** El vector raíz, identificado: [24,48], como en todos los escenarios es un resultado aleatorio tomado por el algoritmo RJ-48, este presenta una variación con respecto a su posición y es al no ser señalado con su coordenada valor, esto

se debe a un aspecto básico pero conciso y es porque se basó la mayor parte de su selección en las columnas con valores iguales a femenino. Al poseer condiciones cerradas este no ve la necesidad de señalar específicamente sus valores. Los demás vectores presentan una condición muy marcada y se mencionó anteriormente, está presente en la mayoría de los nodos decisión que involucren la característica '*Sexo'*, puesto que sus posiciones suelen repetirse o no buscan posiciones tan alejadas de los valores iniciales***.*** ***[Anexo 6]***

- **Árbol RJ-48 año 2021:**

En este escenario el nodo creado por el algoritmo es muy diferente, aunque conserva la profundidad de agrupamiento de 3 niveles/ ramas, la diferencia recae en su nueva característica de nodo. La característica Gini determina el porcentaje de aceptación del nodo y su complejidad de agrupamiento, con esta nueva característica el nodo raíz debe conservar muestras homogéneas sin existir un criterio de observación cerrado, es decir en la ***Ilustración 18***, las muestras poblacionales seleccionas por este nodo se distribuyen con rangos de diferencias muy bajas. La característica '*infección'* determina la condición inicial para este cuadro de valores, el algoritmo interpreto de mejor manera, el escenario anterior puesto que comprendió que a partir de la característica '*Sexo'* podría agruparlos por igual y no depender de esta misma variable solo para un escenario, es decir impone los dos valores contenidos en esta característica (Masculino/Femenino) y a partir de ellos evalúa los 3 niveles asignados. [**Anexo 5]**

A continuación, se indaga los valores/características obtenidas:

- ✓ **Valoración Brote =** Siendo la condición inicial, con un valor de <= 0.5, indica una estimación primaria de 50%, que selecciono el algoritmo como una porción optima para dar comienzo al armado de este árbol, a su vez esta característica está muy marcada con las demás y no depende ubicarse en los demás nodos para interpretar los datos agrupados.
- ✓ **Gini =** el valor inicial de 0.41, determina el porcentaje de aceptación, es decir si este valor esta mas cerca de 0, se considera optimo por la ausencia de ruido o errores, por ende, su nodo 0 distribuye de buena manera este porcentaje de error y lo va reduciendo de acuerdo con sus componentes. Si se observa el ultimo nivel/rama ya viene reducido ese porcentaje a 0, esto quiere decir que la densidad de los datos es buena para modelar a otros algoritmos.
- ✓ **Edad =** siendo la condición secundaria con respecto a los nodos y a su rama/nivel de evaluación. Gracias a la interpretación y mejora del algoritmo para este DataFrame. Los valores <= 42 y <=30, vienen precedidos de dos características que a su vez determinan una nueva condición, esta es: '*Sexo*' puesto que este nodo divide los dos atributos (Masculino/Femenino) de la columna para efectuar su inspección a los agrupamientos posteriores. Los resultados mencionados anteriormente poseen una variante y es que, al distribuir de sus nodos a una condición no cumplida, esta se apegara al otro valor de la condición inicial (Valoración Brote, Valoración tos).
- ✓ **Samples =** Ya ha sido considerado esta característica en los anteriores modelos de árboles, para este escenario conlleva el mismo significado. La muestra inicial fue de 75 personas evaluadas.
- ✓ **Value =** Provee los vectores posicionales, los valores no fueron seleccionados por el algoritmo, la novedad se debe a la fuerte cohesión de los datos y no considera seleccionar valores mínimos debido al importe división de la columna '*Infección tos'*, '*Infección estornudo'* e '*Infección frote*'. En el ultimo nivel las posiciones seleccionadas mantienen una similitud al conservar posiciones entre los 5 primeros valores, esto se debe al peso que tiene la condición inicial (Nodo

0), esta posición arrastra las características ya consolidadas en el atributo infección.

- **Árbol RJ-48 año 2022:**

El ultimo nodo establecido es muy similar al anterior, la diferencia más fuerte radica en las características '*Gini*' debido a los valores óptimos de la 2 rama para abajo, se consolidan varios nodos con un valor a 0, es decir 0% de ruido por parte de los datos. Los datos finiquitaron la deuda de estimación de los demás nodos, este árbol se considera el más claro y definido hasta el momento, este acercamiento es válido comentarlo por los siguientes motivos: la muestra de datos del DataFrame están reducidas en sus columnas '*Infección frote'*, '*Infección tos*' y '*Infección estornudo*'; también se puede mencionar que el algoritmo comprendió mejor como construir el árbol y por último se puede deducir que el año de construcción del árbol favorece mucho en las condiciones de los datos, puesto que ya no debe existir ruido en los datos debido a que estos ya están definidos e identificados. **[Anexo 5]**

Para esta ocasión no será necesario especificar los resultados obtenidos debido a las reiteradas ocasiones que se mencionaron y a su vez los valores predichos hasta esta instancia reflejan un final de la pandemia y por ende existen fuertes agrupamientos que no necesitaron ser alimentados.

Lo más sobresaliente de diseñar los Árboles de Decisiones y RJ-48, es consolidar nuevas variables de entradas y salidas que se implementaran en los próximos algoritmos definidos en la investigación. En definitiva, el procedimiento concreto en este capítulo cumple una función fundamental, las estructuras quedan compactas y precisas para aplicar correctamente el modelo SIRD, dicho modelo requiere de un filtrado de los datos, esta información requiere y necesita variables modeladas con características, muestras en conjunto, por ende, los Árboles desarrollados facilitaron esta labor.

Luego de culminar el despliegue, diseño y distribución de los datos a través de árboles de agrupamiento, se procede a desarrollar una comparativa entre los algoritmos descritos a lo largo de este capítulo, el árbol de decisión se implemento en base a una DataFrame más desordenado que a medida que se alimentaba el algoritmo este permitió organizar los valores y agruparlos aleatoriamente con condiciones acordes a la necesidad; mientras el árbol RJ-48 implemento la mejora de datos resultantes del Árbol de Decisión, estos valores se replantearon con la idea de agregar más columnas de evaluación, al observar los resultados del anterior DataFrame, los resultados favorecieron el planteamiento del problema del modelo SIRD.

En consecuencia, a lo visto hasta el momento, se efectúa una leve comparativa de los resultados obtenidos entre los dos Algoritmos de árbol. A partir de las siguientes ilustraciones podemos observar la eficacia obtenida por ambos factores de decisión. ***[Anexo 6]***

- ***La ilustración 20 ofrece la siguiente interpretación:*** los dos modelos de algoritmo de árbol están representados en función a la eficacia con respecto a la población anual, por ende, se interpretan los escenarios de cada uno de los árboles descritos anteriormente. Los dos grupos, Árbol de decisión (0, Gris) y Árbol RJ-48 (1, Naranja) reflejan la efectividad, desempeño y visibilidad al favorecimiento con respecto a la muestra. El factor X (Eje X) de la tabla comparativa se basa siempre en la condición inicial de los modelos definidos anteriormente, es decir '*Variante Infección'* para el Árbol de Decisión (color gris) e '*Infección*' para el árbol RJ-48

(color naranja). Conociendo la interpretación de la gráfica solo resta mencionar que para esta comparativa el modelo de árbol de RJ-48 posee mejor ajuste de condiciones para el respectivo DataFrame del año 2020, sin embargo, no queda descartado el rendimiento que tuvo el algoritmo árbol de decisión, considerando que este tomo valores desordenados.

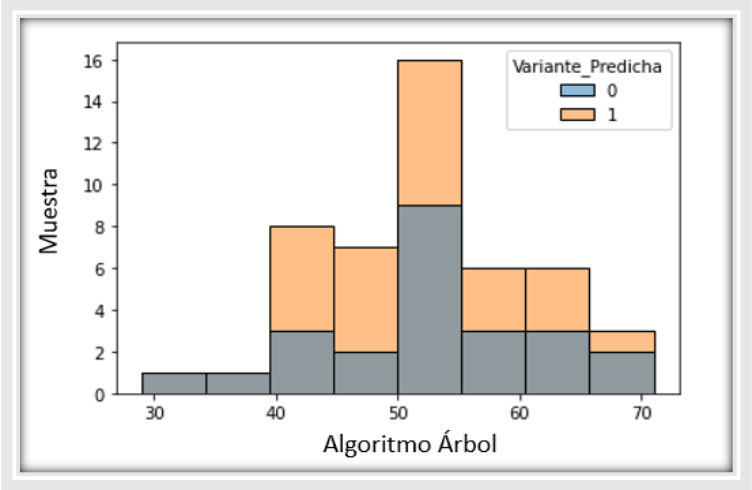


Ilustración 38*-Interpretación Árbol de Decisión vs RJ48 2020*

- ***La ilustración 21 ofrece la siguiente interpretación:*** Los dos grupos, Árbol de decisión (0, Gris) y Árbol RJ-48 (1, Naranja) reflejan la efectividad, desempeño y visibilidad de favorecimiento con respecto a la muestra. El factor X (Eje X) de la tabla comparativa se basa siempre en la condición inicial de los modelos definidos anteriormente, es decir '*Edad'* para el Árbol de Decisión (color gris) y variante '*Infección Brote*' para el Árbol RJ-48 (color naranja). Conociendo la interpretación de la gráfica anterior y visualizando esta nueva gráfica se puede indicar que a partir de 65 o más condiciones ambos algoritmos mantiene estrechamente una evaluación similar y efectiva, para esta interpretación se repite la efectividad del algoritmo RJ-48 y a menor condiciones ,mayor es la efectividad para evaluar los datos y el algoritmo de decisiones al tener más condiciones mejor efectividad.

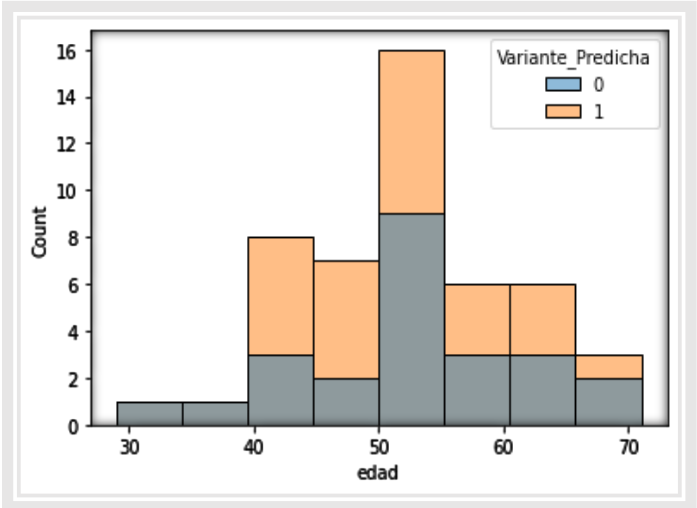


Ilustración 39*- Interpretación Árbol de Decisión vs RJ48 2021*

- **La ilustración 22 ofrece la siguiente interpretación:** En las interpretaciones anteriores, dimos a conocer el contexto y la evaluación reflejada en las gráficas. El factor X de la tabla comparativa se basa siempre en la condición inicial de los modelos definidos anteriormente, es decir '*Variante infección*' para el árbol de

decisión (color gris) y '*Variante Infección Brote*' para el árbol RJ-48 (color naranja). Ahora para esta interpretación final, se minimiza el rendimiento del algoritmo de decisión con respecto al algoritmo RJ-48, sin embargo, este último sigue conservando mejor rendimiento en todas las interpretaciones estudiadas hasta el momento. Para este escenario queda demostrado nuevamente que a menor condiciones para evaluar el algoritmo RJ-48 es más eficaz y a mayores condiciones el algoritmo Árbol de Decisión mantiene una eficacia más alta.

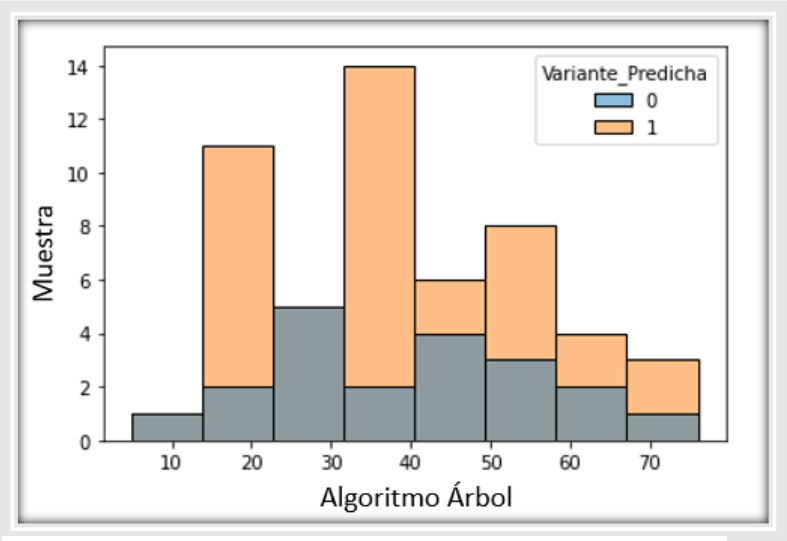


Ilustración 40- *Interpretación Árbol de Decisión vs RJ48 2022*

3.5. Soporte y despliegue de la herramienta Jupyter - Algoritmo de investigación SIRD.

Machine Learning como modelo de datos provee una serie compleja de algoritmos como Random Forest Classifier, SVM, KNN+NCA, Gaussian Naive Bayes, Decision Tree Classifier, Multilinear Regression, Logistic Regression, etc. Al crearse un modelo descrito anteriormente, este utiliza las transformaciones de los datos,en el dataset raíz de datos *(University of Liège, 2015)* **[10]**. El cálculo y la precisión de los algoritmos tomados para la investigación dependerán de una serie la regresiones o procesamiento posteriores a los datos con el fin de interpretar los coeficientes determinísticos con respecto a las variables que mantienen el modelo.

Después de conocer el procedimiento y la funcionalidad del modelo SIRD en este apartado. La tasa de transmisión β(t), es el punto inicial para estimar los modelos mencionados, a partir de una serie temporal de valores de infecciones activa, distribuidos en las 3 zonas estudiadas (BA, CABA, Mendoza).

La primera propuesta se deriva a partir de un ajuste exponencial de los datos transformados por el algoritmo de Árbol de Decisión y RJ-48, estos valores se complementan con informes diarios de la transmisión de la variante Covid-19 en cada una de las áreas estudiadas y mencionadas por el Gobierno, estos informes brindan evolución; usando las variables de datos de los infectados. La segunda propuesta se desprende a partir de la identificación los períodos anuales con diferentes fases asociadas a la evolución con respecto a las medidas gubernamentales (dadas mensualmente). La tercera propuesta se deduce a partir de las muertes agrupadas por 'edad/sexo' de la cantidad de infectados activos.

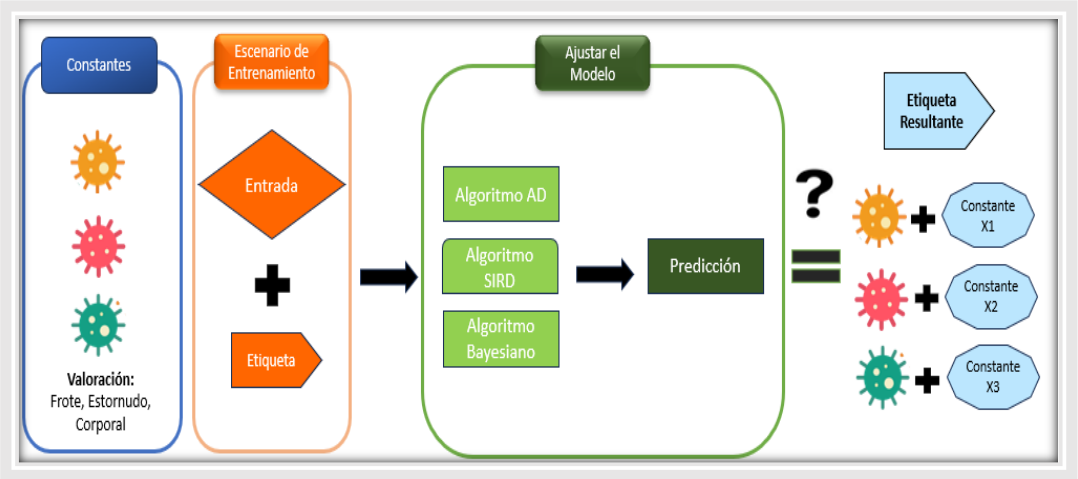


Ilustración 41- *Propuesta, procedimiento del modelo de algoritmos con su etiqueta inicial y final*

La primera propuesta mencionada anteriormente, permite visualizar una serie temporal de datos basados en algoritmos bases de *Python*, dando una proyección escalada de los métodos a implementar. La finalidad de este procedimiento es dar inicio al entorno de predicción de datos en Jupyter Network, para ello es importante tener instalado en su totalidad los Packets del lenguaje, versión de programación Python y sus respectivos complementos. Para ello fue vital haber procesado y transformado los datos referentes a los estudios de la variante covid-19 en formatos más dúctiles de manipular (.bd, .pbix), tener instalado la herramienta Anaconda con sus respectivas librerías

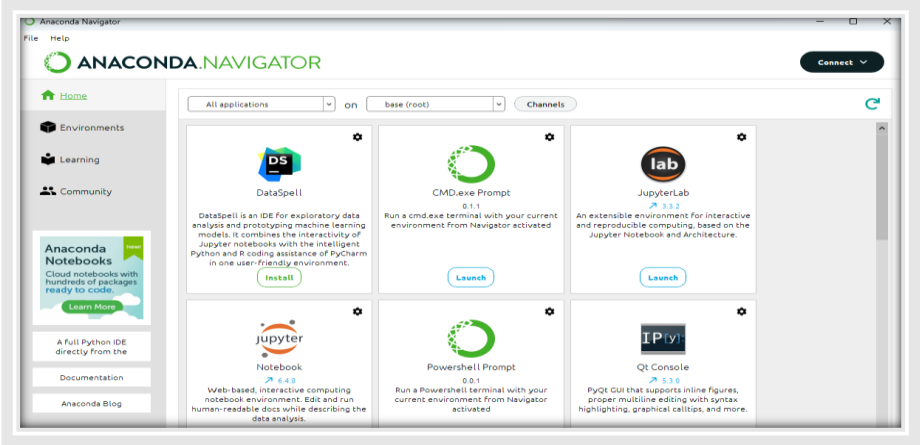


Ilustración 42- *Plataforma Anaconda, procesamiento de volúmenes de datos y aprendizaje de datos.*

La segunda propuesta mencionada, es identificar los periodos anuales, con esto se amplía la agrupación e implementación de código predictivo, imprimiendo una brecha clara de cómo disminuyen o aumentan la correlación de muertes, contagios hasta predecir el final de estos incrementos en base a las tazas de error calculados que promedian resultados mínimos para luego así compararlos con los reportes del gobierno y también determinar la fase final de esta pandemia, resaltándolo en el modelo SIRD. ***[Anexo 7]***

```
[10]: muestras2020=pd.read_excel(ruta+"yyy.xlsx", parse_dates=True,sheet_name='2020')
      muestras2020.head(5)
```

[10]:

	CIUDADES	Fecha	Muestra_apertura	Muestrapert_acum	Segunda_muestra	TerceraM_clasificacion	Inf_Respiratoria_Primera	Contagio_Comporal_Primera	Inf_Respiratoria_Segun
0	CABA	01/01/2020 - 15/01/2020	0	0	5	4	0	0	
1	CABA	15/01/2020 - 30/01/2020	0	0	1	1	0	0	
2	CABA	01/02/2020 - 15/02/2020	0	0	2	1	0	0	
3	CABA	15/02/2020 - 30/02/2020	0	0	2	1	0	0	
4	CABA	01/03/2020 - 15/03/2020	24	24	28	15	5	19	

Ilustración 43*- Periodo Anual 2020, vista de datos en Jupyther Network*

```
[12]: muestras2020=pd.read_excel(ruta+"yyy.xlsx", parse_dates=True,sheet_name='2021')
      muestras2020.head(5)
```

[12]:

	CIUDADES	Fecha	Macumulada	Muestra_apertura	Segunda_muestra	TerceraM_clasificacion	Inf_Respiratoria_Primera	Contagio_Comporal_Primera	Inf_Respiratoria_Segunda
0	CABA	01/01/2021 - 15/01/2021	10975	10975	11401	10911	1001	9974	5431
1	CABA	15/01/2021 - 30/01/2021	28395	17420	15913	16183	9321	8099	6210
2	CABA	01/02/2021 - 15/02/2021	45815	9378	8659	8882	3410	5968	3412
3	CABA	15/02/2021 - 30/02/2021	55193	737	6834	15776	413	324	2341
4	CABA	01/03/2021 - 15/03/2021	55930	10083	9520	9582	7431	2652	4591

Ilustración 44*-Periodo Anual 2021, vista de datos en Jupyther Network*

```
[13]: muestras2022=pd.read_excel(ruta+"yyy.xlsx", parse_dates=True,sheet_name='2022')
      muestras2022.head(5)
```

[13]:

	CIUDADES	Fecha	Muestra_apertura	Segunda_muestra	TerceraM_clasificacion	Inf_Respiratoria_Primera	Contagio_Comporal_Primera	Inf_Respiratoria_Segunda	Contagio_Comp
0	CABA	01/01/2022 - 15/01/2022	158843	213054	210199	83425	75418	111291	
1	CABA	15/01/2022 - 30/01/2022	121417	413403	128667	62189	59228	231091	
2	CABA	01/02/2022 - 15/02/2022	73229	60072	61809	32541	40688	30036	
3	CABA	15/02/2022 - 30/02/2022	42903	31881	33584	24761	18142	15670	
4	CABA	01/03/2022 - 15/03/2022	52544	43501	44055	25681	26863	23774	

Ilustración 45*-Periodo Anual 2022, vista de datos en Jupyther Network*

La última propuesta se basa en identificar las columnas soluciones del teorema SIRD y a la agrupación de salidas de esa clase o en su defecto la salida de un valor usando la data de entrenamiento. Estas salidas son consideradas por las variables discretas o continuas de acuerdo con el algoritmo requerido y la resolución que se planteó en los fundamentos teóricos. El procedimiento se basa más en identificar las variables politómicas, para tener toda la información sobre una condición

concreta de la variable X (Muerte, Recuperados, Síntoma final) cualitativa/cuantitativa, es decir se codifica cada categoría y plasma la información a una muestra final compuesta. ***[Anexo 8]***

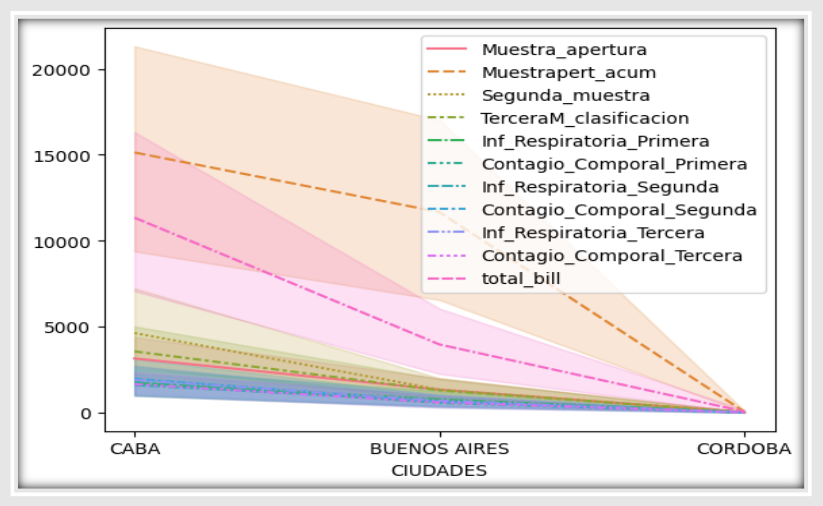


Ilustración 46*-Identificación de variables politómicas, condición Sexo/Muertes*

3.6. Modelado, Evaluación y Regresión de los datos.

Una vez definidas las propuestas, se procede a efectuar técnicas de exploración de datos que se utilizaran para la mayoría de los casos de predicciones y conlleven a una solución del modelo SIRD, linealizando cada una de las regresiones de los datos. Al implementar una solución predictora, con dicha técnica consideramos ajustar la mejor línea de valores en consideración a las constantes requeridas por el teorema. Dicha línea comprende relacionar el predictor con los valores de predicción obtenidos por el patrón de cada fase del modelo SIRD. ***[Anexo 9]***

3.6.1. Primeras Muestra, Agrupación del modelo de datos – Primer Criterio.

Los detalles se proporcionan a partir de los modelos de datos complementarios. Ajustamos el modelo a partir de agrupaciones referentes a las variables categóricas/ numéricas basados en aprendizajes supervisados y no supervisados, siendo las muertes modeladas e informadas por la distribución de infecciones considerando en poblaciones referentes a la edad, sexo y muertes. La distribución y proporción de datos deben cumplir el patrón ***N*** del modelo SIRD, el cual fue estimado con 15 días del mes; ajustada en la estructura mencionada a la población referentes. ***[Anexo 10, 11,12]***

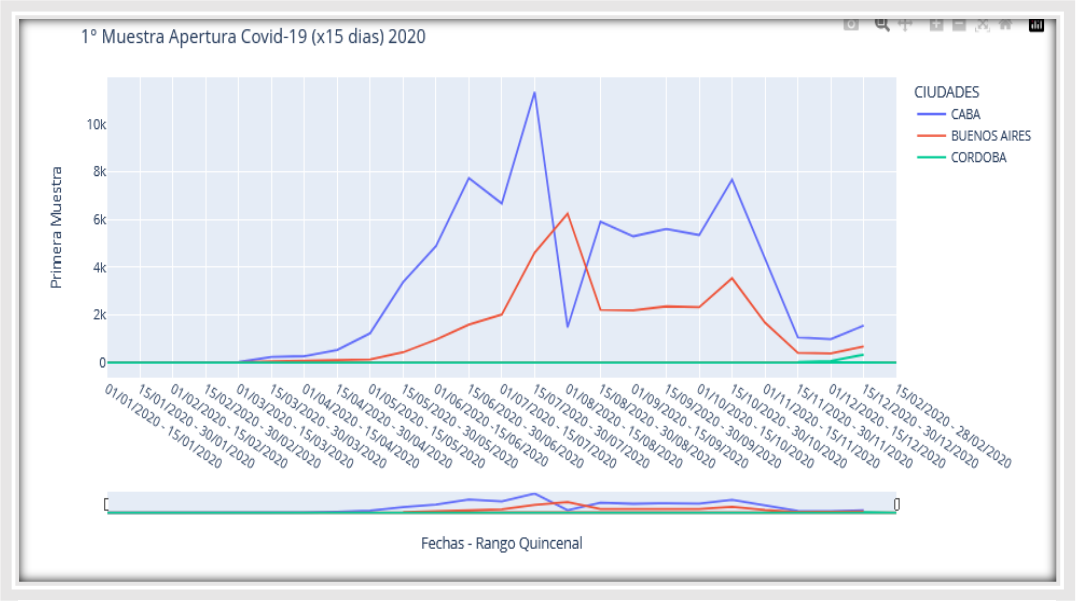


Ilustración 28*- Segunda Muestra de Apertura Covid 19 año 2020.*

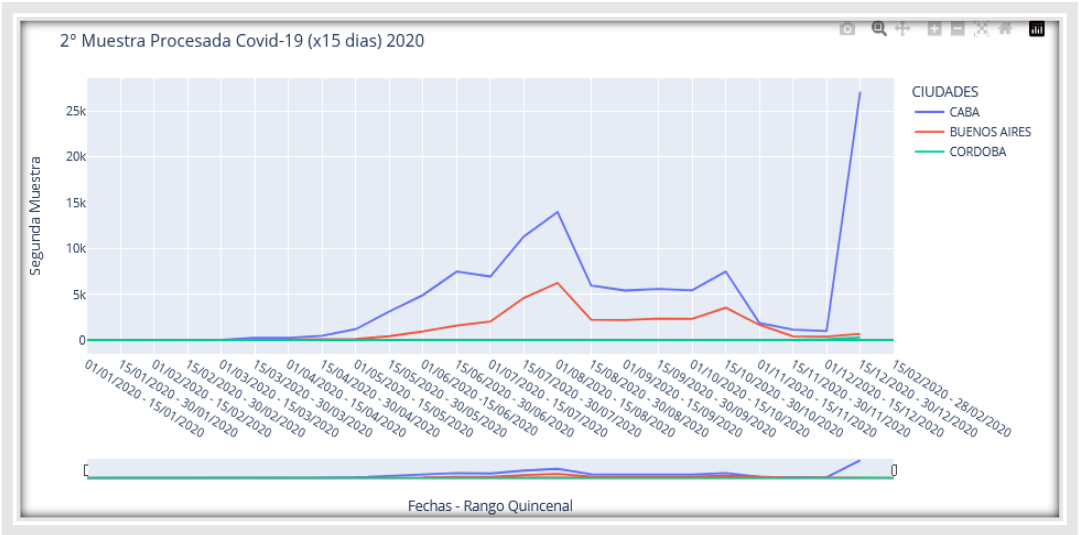


Ilustración 48*-Segunda Muestra de Apertura Covid 19 año 2020.*

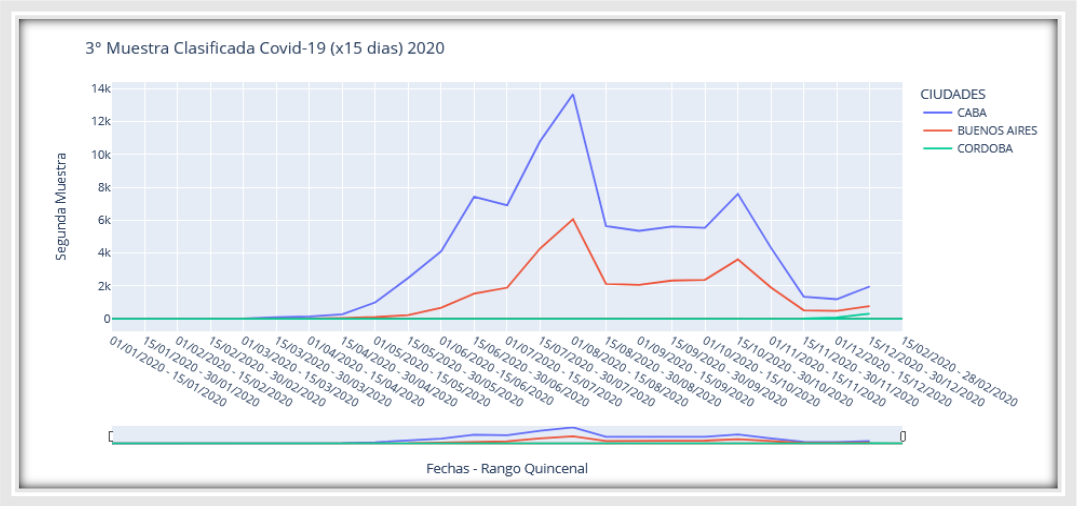


Ilustración 47*-Tercera Muestra de Apertura Covid 19 año 2020.*

Las Ilustraciones 47,48,49 representan en gran medida la agrupación referente a la muestra determinada para cumplir el primer criterio de aceptación del modelo sugerido para la investigación. Asociar los datos de las ciudades estudiadas y señaladas en 3 muestras particionadas en rangos quincenales durante cada año, donde la variable Dummy (continua, cuantitativa/cualitativa) asignada fortaleza a la línea predictora.

La primera muestra consolida las muestras iniciales, ofrecida por el gobierno de salubridad de Argentina brindando un escalar continuo hasta cumplir un periodo semestral donde se remarca un incremento exponencial en la ciudad de CABA , el incremento en ciudad de Buenos Aires se particiona en un balance promedio con la muestra anterior y para la ciudad de Córdoba el valor estadístico es totalmente inferior por la dispersión del virus en sus inicios.

La segunda muestra cataloga una proporción semejante en ambas muestras de las estudiadas, los boletines confirmados y el incremento mundial del virus, en el mundo provee un alza del 6,9 % en nuestros resultados con respecto al primer año y la variante del virus.

La tercera muestra describe le evidencia final de la pandemia puesto que los casos disminuyeron por factores de discrepancia poblacional (desigualdad de datos no coincidentes) final debido a la constante muerte, constante recuperación, constantes de variantes de infecciones que en el transcurso del proyecto fueron apareciendo.

En definitiva, en el transcurso de este segmento de resultados cumplen con el camino del primer criterio para construir el modelo SIRD, despejando todas las variables dicotómicas (toma dos valores Infectado/No infectado) que faciliten el flujo del modelo y por ende se agrupan primero las porciones de datos anuales, cumpliendo con la variable tiempo en porciones de n= 15 días.

3.6.2. Afecciones Persistentes, Agrupación de los datos – segundo criterio.

Las infecciones por virus COVID-19, mantienen efectos a largo plazo, conociendo este escenario y cumpliendo con la condición del modelo. Esta investigación determina dos tipos de afecciones persistentes en la pandemia, estas se identifican, se agrupan como en el *apartado 3.4.0.* Las afecciones persistentes señaladas se definen en términos generales como infección respiratoria y contagio corporal.

Esta definición de afecciones persistentes por COVID-19, fue recreada para ajustar el segundo criterio y adicionar todas las posibles variables Dummy (variable categórica) esta concuerda con las variantes de las salidas del modelo junto con valores determinísticos. ***[Anexo 13]***

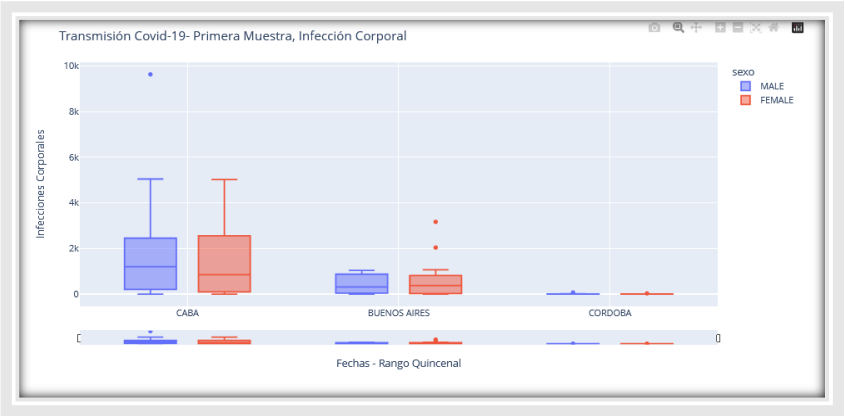


Ilustración 48*-° Muestra, Afección Corporal agrupado por sexo.*

Definir las afecciones producidas por el Covid-19 favorece una apertura de esquema que alimentara el modelo, el segundo criterio congrega las ilustraciones con cada una de las compilaciones programadas por muestras quincenales (patrón N del modelo). En las ilustraciones que se verán a continuación, se busca observar los límites de cada frecuencia a través del modelo de Python llamado upper fence.

Con la ayuda del algoritmo upper fence, proyectamos una barrera limite, inferiores y superiores de los datos y cualquier valor que se concentre fuera de estos límites definidos, puede considerarse un valor atípico. En la ***Ilustración 50***, calculamos los cercos (aproximaciones) porcentuales encontrando los limites superiores e inferiores para la infección corporal en la primera muestra, los resultados fueron los siguientes:

Ciudad	Límite Superior		Límite Inferior		Mediana	
Caba	4.66 %	4.64%	0.1 %	0.1%	1.108%	0.7%
Buenos Aires	0.96%	2.92%	0.03%	0.01%	0.2%	0.34%
Córdoba	0.064%	0.25%	0%	0%	0.001%	0.00092%

Tabla 6*-Algoritmo Upper fence, afección corporal 1° Muestra, 2020.*

En la ***Ilustración 51,*** calculamos los cercos porcentuales encontrando los limites superiores e inferiores para la infección respiratoria en la primera muestra, los resultados fueron los siguientes ***[Anexo 14]***:

Ciudad	Límite Superior		Límite Inferior		Mediana	
Caba	5.82%	2.97%	0.176%	0.053%	0.49 %	1.10%
Buenos Aires	2.31%	1.77%	0.028%	0.015%	0.11%	0.40%
Córdoba	0.25%	0.004%	0%	0%	0.0018%	0.0004%

Tabla 7*-Algoritmo Upper fence, afección respiratoria 2° Muestra, 2020.*



Ilustración 49*-1° Muestra, Afección Respiratoria agrupado por sexo.*

En la ***Ilustración 51***, calculamos los cercos porcentuales encontrando los limites inferiores e inferiores para la infección corporal en la segunda muestra, los resultados fueron los siguientes: ***[Anexo 15]***:

Ciudad	Límite Superior		Límite Inferior		Mediana	
Caba	4.73%	3.24%	0.001%	0.01%	0.88%	16.10%
Buenos Aires	1.68%	2.57%	0.01%	0.0064	0.26%	0.43%
Córdoba	0.162%	0.041%	0%	0%	0.0013%	0.0092%

Tabla 8*- Metodo Upper fence, afección corporal 3° Muestra, 2020.*

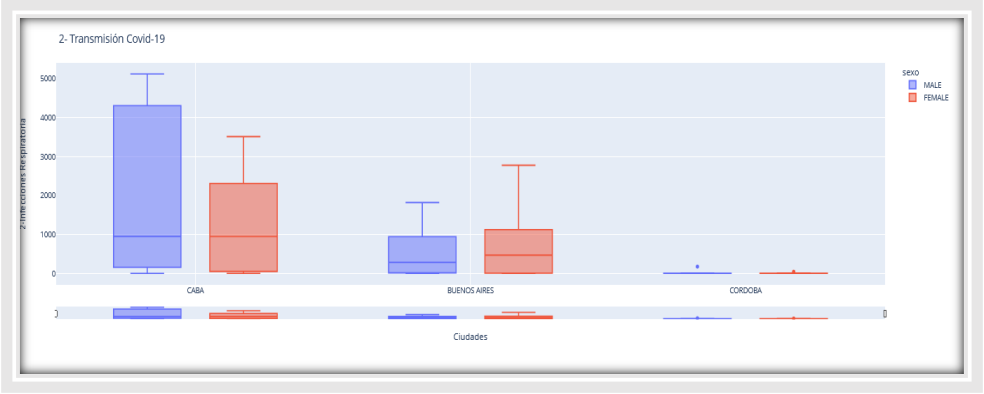


Ilustración 50*-2° Muestra, Afección Respiratoria agrupado por sexo.*

En la ***Ilustración 51***, calculamos los cercos porcentuales encontrando los limites inferiores e inferiores para la infección corporal en la segunda muestra, los resultados fueron los siguientes ***[Anexo 16]***:

Ciudad	Límite Superior		Límite Inferior		Mediana	
Caba	5.248%	3.49%	0.09%	0.07%	1.14%	1.27%
Buenos Aires	1.65%	3.64%	0.023%	0.004%	0.20%	0.23%
Córdoba	0.126%	0.041%	0%	0.0055%	0.0013%	0.0003%

Tabla 9*- Algoritmo Upper fence, afección corporal 1° Muestra,2021.*

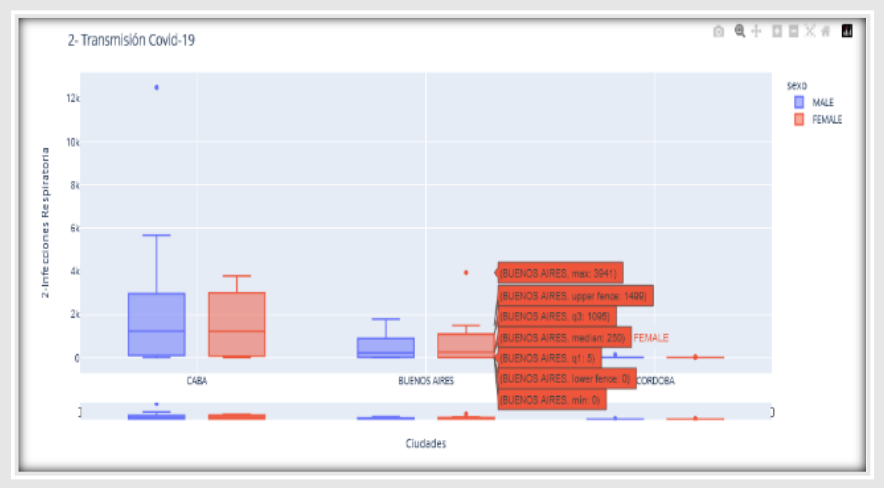


Ilustración 49*-2° Muestra, Afección corporal agrupado sexo, 2021.*

En la ***Ilustración 49***, calculamos los cercos porcentuales encontrando los limites inferiores e inferiores para la infección corporal en la segunda muestra, los resultados fueron los siguientes:

Ciudad	Límite Superior		Límite Inferior		Mediana	
Caba	3.5%	2.93%	0.06%	0.030%	1.01%	1.14%
Buenos Aires	1.125%	1.156%	0.01%	0.06%	0.21%	0.34%
Córdoba	0.092%	0.04%	0%	0.005%	0.0013%	0.004%

Tabla 10*- Algoritmo Upper fence, afección corporal 1° Muestra, 2022*.

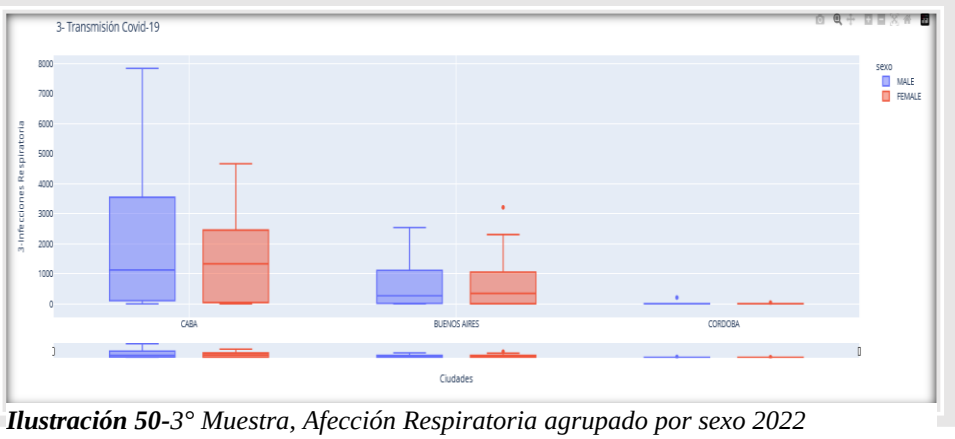


Ilustración 50*-3° Muestra, Afección Respiratoria agrupado por sexo 2022*

Las personas contraen las afecciones persistentes al COVID-19 por varios nombres, tales como afecciones posteriores al COVID-19, COVID-19 de larga duración, COVID-19 pos-agudo, efectos a largo plazo del COVID-19 y COVID-19 crónico, todas estas afecciones representan las columnas y atributos de la tabla base . También se utiliza el

término secuelas pos-agudas de la infección por el SARS-CoV-2 (PASC, por sus siglas en inglés) para referirse a un subgrupo de afecciones persistentes al COVID-19.

3.7. Discusión, enfoque y selección del Modelo aplicativo SIRD.

El procedimiento SIRD fue precedido por muchas investigaciones como elección funcional de la mayoría de los procedimientos ante su capacidad alta de asimilar datos correspondientes a múltiples variables de salida y en escenarios de pandemias, con respecto a poblaciones referentes a datos simples y escasos para desplegar. Sin embargo, los periodos largos determinan un buen balance dependiendo de la integración de los datos, por tal motivo se desarrolla cada una de las fases de la investigación para ir asimilando una correlación de datos más continua lo cual permita una apropiada valides de predicciones con respecto a sus variables de salida [muertes, recuperaciones]. La propuesta de tomar el modelo SIRD es el aspecto de incorporar variables fragmentadas para representar mayor grado de detalle cada ciclo del modelo (Susceptibles-Infectados-Recuperados), por lo tanto, se requirió múltiples datos. ***[Anexo 17]***

Por ejemplo, el algoritmo adaptado al modelo SIRD, ira modificando las muestras estipuladas por el capítulo anterior con respecto a su ciclo, partiendo de cada etapa con su respectivo algoritmo final de Machine Learning. Todo lo anterior rige un paso a paso por el cual captura mejor el resultado de dicha pandemia para nuestro escenario.

En este contexto, la propuesta del algoritmo SIRD en Python describe paralelos de soluciones que conservan el mismo número de variables que se ajustan libremente al algoritmo, desde una variable -hasta *n* variables determinando los grados de libertad para estipular una predicción requerida por los datos.

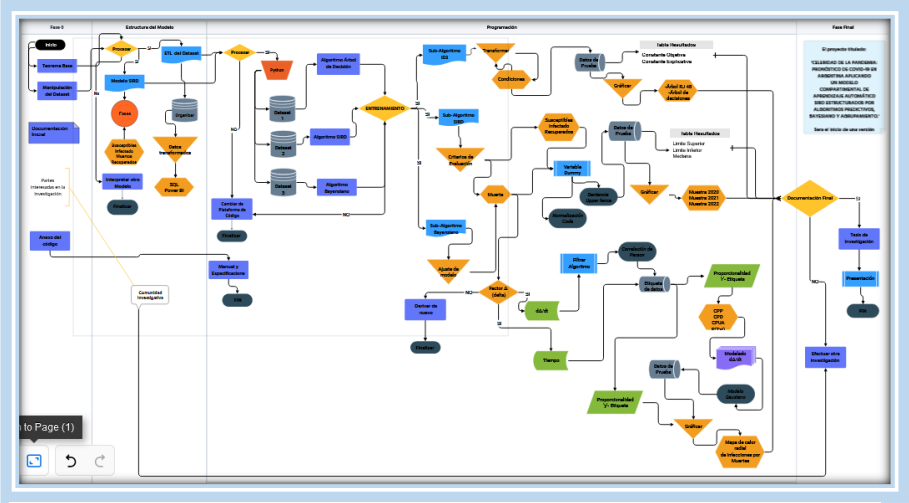


Ilustración 51*-Diagrama de Proceso de la investigación.*

3.7.1. Sustracción, transmisión como parámetros del modelo SIRD.

El propósito principal es definir una tasa de transmisión inicial β, se visualiza con distintas aproximaciones en tasas de tiempo variables, permitiendo mejores predicciones con respecto a cada dato filtrado en un lapso de la muestra en días [15,30 días] , referentes a la mutación de la variante con respecto a los lapsos de tiempo de pandemia. Cada resultado debe ser coherente con las variantes del algoritmo, cuando son encontradas en base a las fases y el tiempo de cada tasa de contagio, introduciendo la primera equivalencia, $\beta(t) > 0$, disminuye o incrementa con respecto al tiempo. Las

funciones proveen mitigar tipos de valores logísticos, β_0 representa la tasa de transmisión antes de aplicar cada regla del algoritmo, esta es la medida es el aislamiento y la recuperación a partir de variantes de salud que no incriminan la defunción de las personas y permite ver con claridad la proyección de cada flujo predictivo con respecto al parámetro comparativo de la variable velocidad que define el descenso de los pronósticos.

Lo mismo se resuelve con cada tasa de reducción γ, dejando que sean las variables las capaces de captar mejor los cambios con la fase incorpórea de recuperación. Sin embargo, la fase inicial de la pandemia predomina resultados constantes, porque no se justificó la opción de un algoritmo de predicción de epidemia.

4. Aspectos Tecnológicos y Metodológicos del Caso de Estudio Desarrollado

En el transcurso de la investigación se proyectaron variables precisas correspondientes a las infecciones y muertes hasta el fin, reportando una mortalidad en 2022 para toda Argentina.

Las 5 muestras anteriores son previstas para los casos acumulados que se implementan para este apartado, debido al algoritmo SIRD se separa por una mediana, los subconjuntos de infecciones activas, el recuento de muertes en las ciudades mencionada. Estas 3 ciudades fueron seleccionados como ejemplo referente porque son zonas diversas en tamaño de población, espacio y dispersión de progresión de área capital/estado diversas del Covid-19. Las cifras no mencionadas en esta investigación se respaldan en otra versión de tesis con la epidemia correspondiente de una gripe viral postpandemia.

Buenos Aires, especialmente con la mayoría de la población, fue el epicentro de la pandemia en Argentina, con más de 10(millones) de casos confirmados a final de toda la pandemia. El crecimiento exponencial fue reflejado en cada fase planteada, generando aproximaciones constantes de la velocidad y picos de casos finales desde el 2020. **[Anexo 21]**

Córdoba, por el contrario, ha tenido menos casos que la totalidad de la muestra tomada de 13071 personas involucradas, en el epicentro a finales del 2021. En lugar de mostrar un alza pronunciada, en simultaneo presenta bajas de crecimiento de casos, los casos de Córdoba impactan, con alzas constantes por oleadas de crecimiento que es propagado desde capital. Sus estimaciones de intervalos se proyectaban por muestras acumuladas con evaluaciones de error para ser adaptadas de forma periódica ofreciendo una fase neutral con respecto a las demás concentraciones elegidas, debido al algoritmo epidemiológico que especifica una trayectoria mejor estimada para cualquier concentración de predicción y agrupación estimada. Córdoba también muestra una variación de mayor recuento de personas recuperadas, seguido de Buenos Aires, los recuentros de Caba a través del algoritmo son exponenciales, debido a que allí el aprendizaje supervisado fue estipulado como la segunda mayor mortalidad de Sudamérica.

El territorio estudiado de Buenos Aires se diferencia del anterior por un aspecto referente a más de 10.000 casos predictivos con combinación de variantes de covid-19 y gripa A hasta final de la pandemia, pero con un lento crecimiento de recuperados sin alguna variante de enfermedad durante la transición de ambos sucesos de cuarentena. Con el aprendizaje controlado por el algoritmo los crecimientos fueron exponenciales.

La visualización de las cifras se incluye en su comienzo por intervalos de credibilidad de 90% por variable de entrada "*Sexo*" alrededor de criterios de muestras simuladas dentro

de cada región establecida como estudio. Estos intervalos determinan un comienzo creíble para alimentar los algoritmos bayesianos y SIRD, ya con estas partituras de cada modelo podemos ir segregando cada aprendizaje, ofrecer una probabilidad inteligente por parte del lenguaje de programación definido. Sin embargo, representa toda la explicación razonable de la investigación basada en la incertidumbre del modelo desenlace, ya que incorporo todas las variables de entrada acopladas a cada modelo aplicado sobre la duración de la enfermedad y estimaciones agrupadas en intervalos de velocidad con la determinación de ajustar los porcentajes predichos y aprendidos por la muestra de valores creadas hasta el momento.

Dicho procedimiento fortalece cada variable necesaria para construir la estructura completa del algoritmo que se realizó previamente para la predicción evolutiva de los casos de COVID que provocaron un porcentaje de tasa de mortalidad y de recuperaciones en los 3 escenarios estudiados (*Todos los códigos de programación generados se agregaran al anexo*). A continuación, se efectúa la normalización de la muestra para los casos de muertes producidas por rangos de edades en el transcurso de cada año. ***[Anexo 22,23,24]***

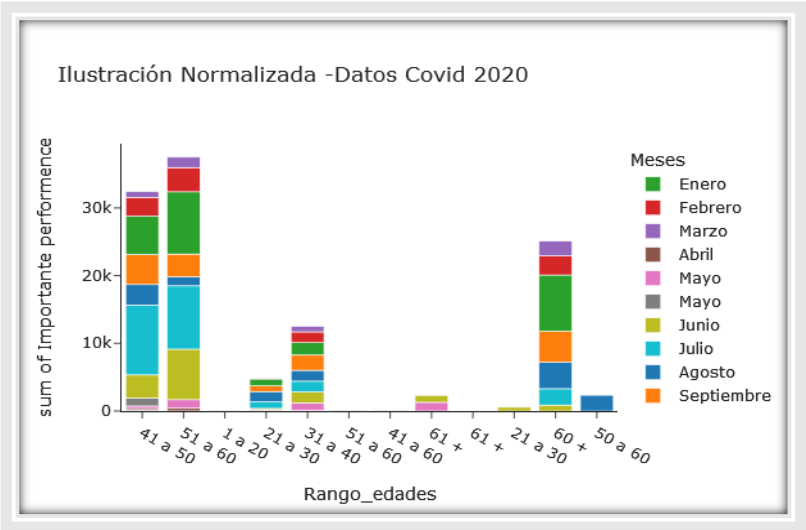


Ilustración 52- *Muertes por rango normalizadas 2020, ultima fase SIRD*

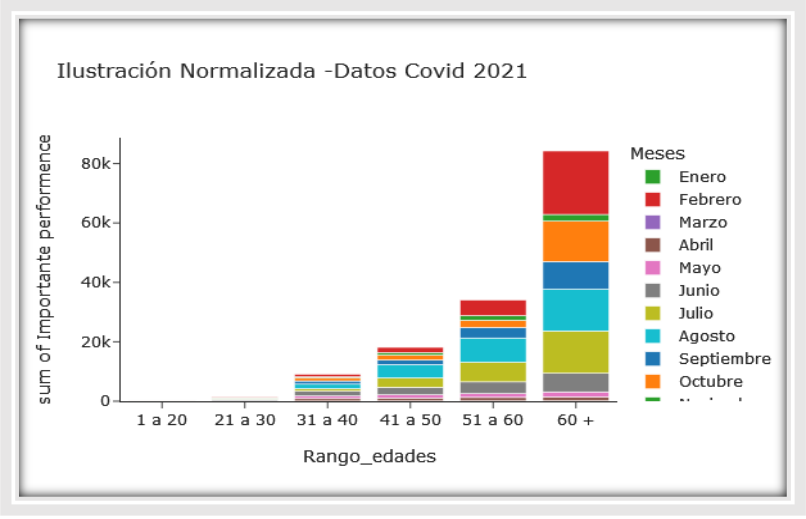


Ilustración 53- *Muertes por rango normalizadas 2021, ultima fase SIRD*

Siempre se buscó el beneficio de estimar cada fase del modelo SIRD a través de cada población semejante a cada variable esperando las variables de salida de recuperación/muerte con su evidencia, el algoritmo ha adquirido con porcentajes de aceptación mínimos en predicciones temporales con el fin de ofrecer la próxima inmunidad y un soporte estadístico comparativo ante la gripe A que fue precedida finalizando la pandemia.

El modelo presentando hasta aquí, se centra en modelar casos y muertes confirmadas, ofreciendo un aproximado número de personas recuperadas en una escala real y con variantes progresivas con respecto a la velocidad tomada como punto de propagación.

Es importante saber que, en Argentina, los casos confirmados fueron tomados a partir de conteos sustancialmente aproximados al número real de infecciones (muertes/recuperaciones) por la variante Covid-19. Estos resultados, estiman el número de personas que pudieron evitar muertes o casos de propagación del virus. En otros países se citó cuantificar la recuperación de las personas solo con las dos últimas fases del modelo SIRD.

A continuación, se muestran los resultados de la fase completa de muertes, implementando variables de entrada Sexo y Muertes, estas fueron agregadas a la fase final del modelo SIRD, las pruebas dieron los siguientes resultados: ***[Anexo 23,24,25]***

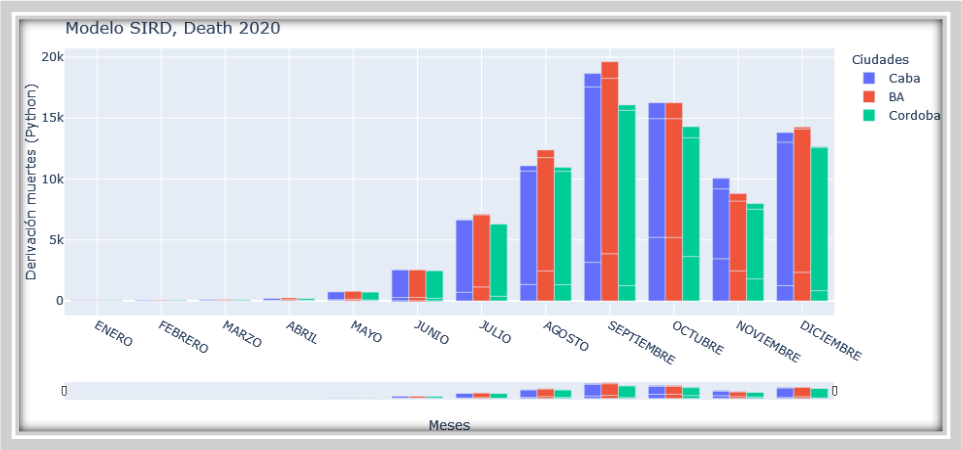


Ilustración 54*-Derivación de muertes 2020, fase final.*

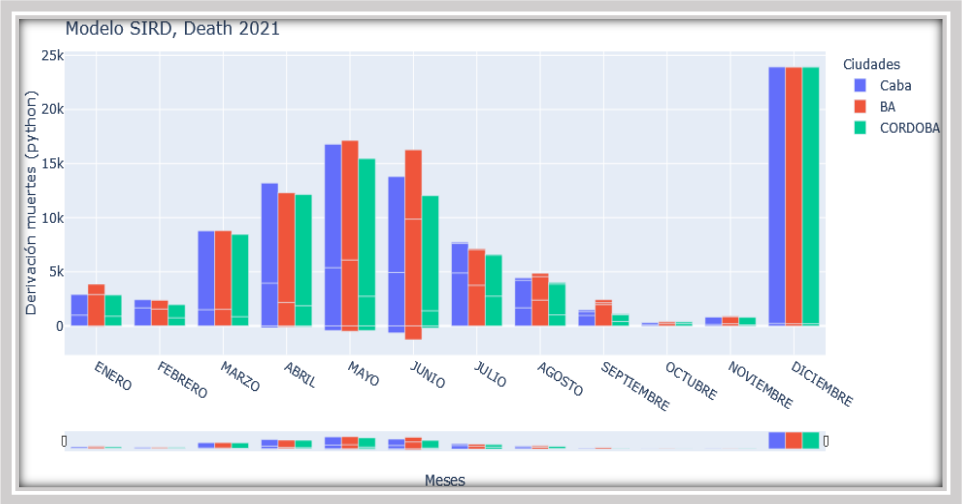


Ilustración 55*-Derivación de muertes 2020, fase final.*

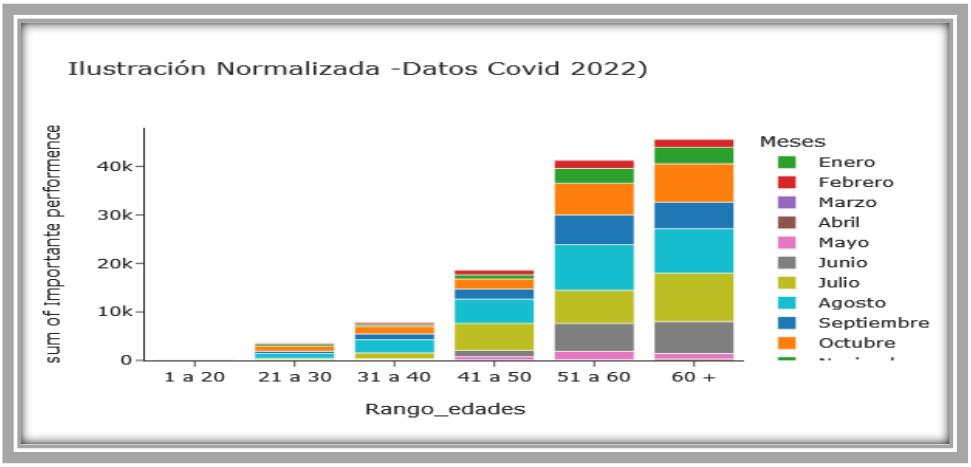


Ilustración 56*-Muertes por rango normalizadas 2023, última fase SIRD.*

La conclusión de este capítulo, sin la adicción de covariable el modelo de velocidad efectuado no podría predecir picos altos o bajos de los casos determinados durante los 3 años de muestra, por ende, el resultado siempre fue un retorno al comportamiento secuencial de cada una de las fases, marcando así sucesos que ocurrieron durante el distanciamiento social o alguna intervención del gobierno de salubridad. Por consiguiente, los datos (.csv) se adaptan correctamente para cada tipo de evento por las agrupaciones de cada variable implementada. La velocidad creciente, se asoció con el aumento de casos de gravedad con respecto a cada una de las variables de entrada que se inyectaron al algoritmo. ***[Anexo 26***]. La mayoría de los incidentes iniciales no reflejaban resultados importantes puesto que la cantidad de datos eran escasos, la data fue alimentada para concurrir con agrupaciones de variables de pacientes con valores no favorables a su recuperación y la propagación que fue predicha a en los lazos de tiempos fijos (15 días) para evaluar puntos mínimos y máximos, con este método los valores incrementaban para cada una de las variables. Se apunta siempre a esta velocidad (15 días, 1 muerto, 0 recuperados), el modelo pronostica trayectorias muy importantes a lo largo de plazos mensuales, distribuidas por edades sin necesidad de repetir el modelo SIRD para cada feature de las tablas, determinando por criterio modelado los resultados de cada fase, puesto que implementar de a grupos de columnas resultaba complejo de predecir y con errores de programación altos.

4.1. Modelo Bayesiano, filtrado al algoritmo SIRD.

El Análisis de Puntos de Cambio (CPA) para las 3 áreas de estudio es el primer procedimiento aplicado en este capítulo, con la finalidad de percibir la propagación de la variante, con la finalidad de referenciar la distribución entre lo que se calculó. Antes de analizar cada fase del algoritmo. Se analizan los datos con este nuevo modelo, ofreciendo un promedio de datos actualizados y referentes a las muertes, basadas en el brote y los diferentes estados mencionados anteriormente.

Los datos correspondientes al número de casos de estados seleccionados recopilan el CPA utilizado para el rastreo de la variante COVID-19 (rastreador COVID-19). La CPA se realiza sobre los datos que se muestran en la *Tabla 3*, para determinar el punto de inflexión de los casos de COVID-19 en cada estado respectivo. ***[Anexo 28]***

	ID	Meses	Ciudades	Muertes	Promed_mensual	Error_estandar	Variacion_muertes	acumulados	Conclusiones	Skill
0	1	ENERO	Caba	3	45	21.0	-0.150	131.0	NaN	7.0
1	2	FEBRERO	Caba	4	67	31.5	-0.098	172.0	NaN	18.0
2	3	MARZO	Caba	8	88	40.0	-0.249	292.0	Del total de casos (966), el 42,5% son mujeres y el 57,5% son hombres.	27.0
3	4	ABRIL	Caba	55	143	44.0	-4.527	1168.0	(20,7%) son importados, (43%) son contactos estrechos \nde casos confirmados, (25,9%) son casos de circulación comunitaria y el resto se encuentra en investigación epidemiológica.	218.0
4	5	MAYO	Caba	99	637	269.0	-1.606	8206.0	970 (5,8%) son importados, 7.308 (43,4%) son contactos estrechos de casos confirmados, 5.975 (35,5%) son casos de circulación comunitaria y el resto se encuentra en investigación epidemiológica.	539.0
...	...	...	...	...	...	...	...	...	...	...
31	32	AGOSTO	Cordoba	1325	9309	3992.0	317.000	8522.0	NaN	NaN
32	33	SEPTIEMBRE	Cordoba	1245	14392	6573.5	436.000	34203.0	NaN	NaN
33	34	OCTUBRE	Cordoba	3643	9745	3051.0	907.000	85415.0	NaN	NaN
34	35	NOVIEMBRE	Cordoba	1801	5726	1962.5	451.000	114207.0	NaN	NaN
35	36	DICIEMBRE	Cordoba	834	11765	5465.5	56.350	126917.0	NaN	NaN

Tabla 11 *- Fuente de valores a partir de el análisis de puntos de cambio (CPA)*

El CPA se realiza para cada una de las áreas planteada, analizado según la metodología explicada anteriormente. ***La Ilustración 57,*** muestra el punto de cambio de cada estado. Vale la pena señalar que también se ve una franja tenue en las imágenes de los estados. La franja indica que hubo un cambio gradual durante ese período; sin embargo, el punto de cambio, este está precedido de cada vacío que se ve en cada grafica. ***[Anexo 29]***

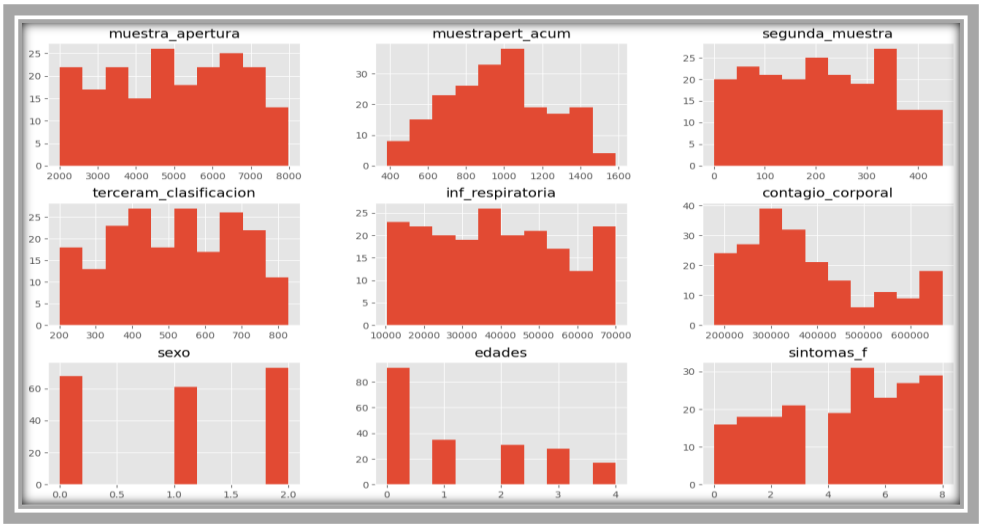


Ilustración 57– *Resultados CPA por área estudiada, punto de cambio agrupados.*

De los gráficos anteriores, se identifica el CPA y la correlación en delta de x (dΔ) , se calcula utilizando los datos: rango de notificación del primer caso y la dΔ (muestra), la discrepancia en las semanas de los casos, antes del cierre y la semana del punto de cambio de los estados de la variante COVID-19. ***La Tabla 11*** muestra los valores dΔ, que es una medida indirecta de la rapidez de respuesta de cada impacto con respecto al cambio en el ritmo de crecimiento de las muertes por casos de COVID-19 de los respectivos estados. ***[Anexo 30]***

	muestra_apertura	muestrapert_acum	segunda_muestra	terceram_clasificacion	inf_respiratoria	contagio_corporal	sexo	edades	sintomas_f	muerte
0	6000	1000	0	600	50000	400000	0	2	2	1
1	6745	944	123	429	43240	636897	1	3	6	0
2	6455	1033	98	795	57463	321779	2	1	8	1
3	7098	1278	15	254	54506	660933	0	0	3	0
4	6167	863	223	520	41512	348932	0	0	3	1
...	...	...	...	...	...	...	...	...	...	...
95	4517	1310	89	529	25960	227634	2	2	6	1
96	4505	1171	343	627	37943	266595	0	0	4	0
97	3080	585	285	521	16260	377254	0	0	4	0
98	3543	992	91	372	30918	327691	1	4	6	0
99	2039	387	258	362	25537	210701	0	0	4	0

Tabla 8- *Muestra de datos asignadas para cada dΔ por cada crecimiento.*

Según el recuadro 2 (‘*Muestra de apertura acumulada’*), los valores de Δ son inferiores a las demás muestras, debido al año con más afectados y muertes de la pandemia. Por otro lado, las infecciones corporales tienen los casos más inferiores, desde el segundo trimestre (*empezando 1 de junio*) del 2021 hasta el final de la cuarentena, indicando un valor comparativo y agudo por parte de la infección respiratoria Δ, siendo la predicción correcta, ya que en resultados específicos, las muertes frecuentes fueron por la transmisión respiratoria. Los resultados de la CPA representan los enfoques adoptados por cada recuadro de la ***Ilustración 56***, el propósito de mitigar el COVID-19. La población de los territorios también tiene un impacto significativo en el número de casos de COVID-19. El análisis se lleva a cabo mediante la recopilación de datos relacionados con la población y el área estudiada. Tomando cada proporcionalidad de los datos se precede a inyectarlos a una medida de dependencia lineal entre nuestras variables infección corporal e infección respiratoria. ***[Anexo 31]***

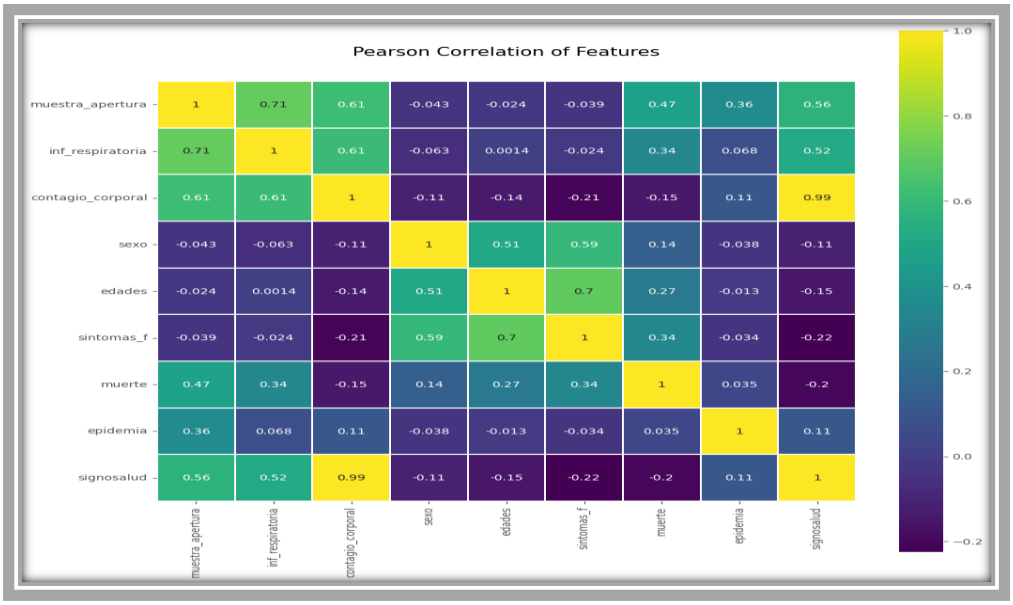


Ilustración 58-– *Correlación de Pearson, dependencia lineal (Inf Respiratoria/corporal).*

La Ilustración 57, muestra los datos completos necesarios para el análisis. La correlación de Pearson dispone de los estados y factores anteriores para comprender las relaciones lineales de los variables anteriores. El análisis lineal se ajusta a su diagonal como un modelo completo y acoplado, por tener un modelado por capas que fueron uniendo cada fase y por tener un área comparativa pequeña (BA, CABA, CORDOBA), otorgando un sesgo preciso al conjunto de datos. El resto de los estados se analizan en busca de correlación lineal (cumpliendo la condición). ***[Anexo 32]***

```
Index(['muestra_apertura', 'inf_respiratoria', 'edades', 'sintomas_f',
       'signosalud'],
      dtype='object')

Veamos la correlación entre las 5 mejores Features
```

Ilustración 59*- La selección de las feacture correctas por el algoritmo.*

La Ilustración 59, proporciona los valores ‘*r*' para cada uno de los factores. En la ***Tabla 13,*** se observa una clara correlación inversa de delta con CPP (Casos por unidad

de Población), CPUA (Casos por unidad de Área) y CPD (casos por densidad de población). Aunque el valor de ‘r varía entre − 0,019 y − 0,024, ofreciendo correlación lineal frágil, todos los parámetros están inversamente relacionados. Esto indica que cuanto mayor sea el valor de Δ, menor será el CPP, CPD o CPUA. Por lo tanto, cuanto antes hubiese sido la acción, menores habrían sido los casos y la tasa de progresión. ***[Anexo 33]***

	Delta (Δ)	**CPP**	**CPD**	**CPUA**	**P (T ≤ t)**
Delta (Δ)	1	0.71	0.024	0.039	0.36
CPP	0.71	1	0.0014	0.024	0.52
CPD	-0.024	0.0014	1	0.7	0.15
CPUA	-0.019	-0.024	0.7	1	0.22
P(T ≤ t)	0.36	0.52	0.15	0.22	1

Tabla 13- *Correlación lineal entre parámetros clave.*

La importancia de un valor de la correlación de Pearson, indica no solo una interacción lineal no fuerte sino unidireccional (efectiva a una predicción) , se realizan pruebas para delta y CPP, CPD y CPUA. Los valores de infección y los signos de salud están dentro del rango de ± 1 para todos los parámetros, cuya condición de ajuste este ofreciendo un determinístico más viable para una predicción final. Las estadísticas descriptivas se dan en el código fuente. ***[Anexo 34]***

Los resultados del modelado td/dv se muestran en la siguiente gráfica:

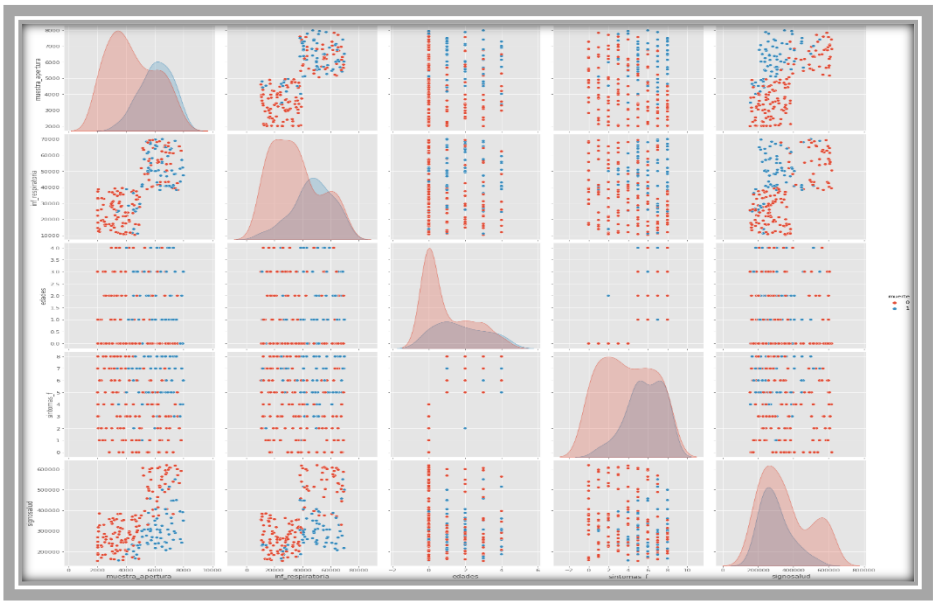


Ilustración 60-*Modelado dΔ/dt, predicción con valores determinísticos.*

En la ***Tabla 14***, se visualiza una significativa correlación entre Δ con CPP y Δ con CPUA. Los valores de p (población) son menores que 0,05 e Inicio (t) > t crítico tanto para una pila como para dos pilas. La relación entre delta y CPD muestra una correlación débil con solo un valor de p cercano a 0,05 para una cola. Por lo tanto, se puede concluir con seguridad que los casos por unidad de población o área se habrían controlado si la delta hubiera sido mayor.

	delta (Δ) con CPP	delta (Δ) con CPD	Delta (Δ) con CPUA
Inicio (t)	11.306	-1.869	2.980
P(T ≤ t) ,1°PILA	0.000	0.052	0.008
t Critico, 1° PILA	1.895	1.895	1.833
P(T ≤ t) ,2°PILA	0.000	0.104	0.015
t Critico, 2° PILA	2.365	2.365	2.262

Tabla 14*- Estadísticas de casos de COVID-19, dt/P (estructura de datos)*

4.2. Creación del modelo Gaussian Naive Bayes

Dados los resultados anteriores, se podrían analizar los estados en búsqueda de posibles puntos críticos y zonas seguras utilizando IP (infecciones/poblaciones) como se explica a continuación. El área es un factor inherente al IP que se decide en función de los valores de puntos. Por lo tanto, IP está experimentado una variante de la población y los casos de COVID-19 de cada estado para comprender la importancia de la correlación. Los datos de población y COVID-19 de cada ciudad considerada, se proporcionan como datos complementarios. La discusión se agrega por estados como se ve a continuación. ***[Anexo 35]***

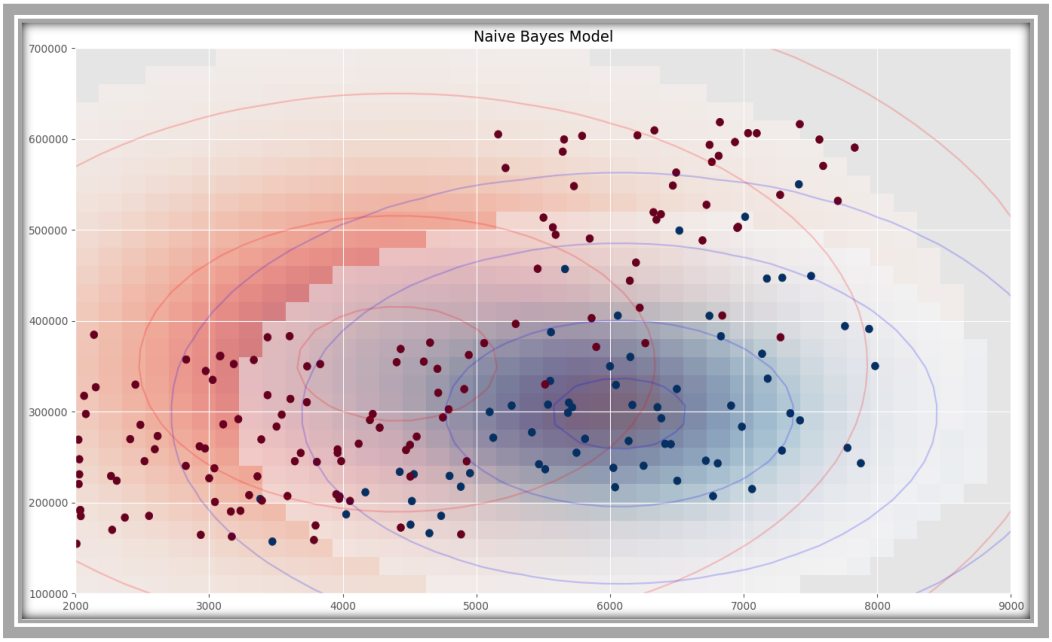


Ilustración 61*-Mapa de calor radial de infecciones, concluyentes a una predicción solución.*

Vemos que en ***la Ilustración 61,*** cuantas más infecciones se tienen en la población, menor va hacer es la cantidad de muertes, esto se debe a evolución que posee el organismo y la intervención de salubridad de este tipo de modelos que mantendrá alerta cada cambio de estado del individuo afectado y proporcionando a tiempo una posible resolución del problema, por ende se mejora la opción de recuperación (en azul). De otra manera, conviene que las infecciones sean latentes debido a que estas pueden ser tratadas con otra regulación médica, ejemplo: La nueva gripe A, aleja cualquier opción de muerte.

4.2.1. Discusión por estado

La discusión se agrega por estado a continuación:

- En la ***Ilustración 60*** se puede ver que, en el foco de pigmentos, una gran población y un gran número de casos de infecciones tratadas que se presentan en una escala nominal. La región central tiene una población alta en infecciones y casos evidentemente bajos de muerte por COVID-19, mientras que la región alta tiene un rango inicial de muertes elevadas.
- Es interesante ver que el COVID-19 ya se dibuja está alejando hacia el lado oeste, ofreciendo un alza de casos que no terminan en muertes.
- El IP del estado del área estudiada, la parte oriental de la gráfica está gravemente afectada por la COVID-19, debido al inicio de la pandemia por la cantidad de población no restringida al contacto. Los resultados son sólidos, con una población media, con casos altos debido a la transición del tiempo y al ajuste de pandemias registradas en el país.
- CABA, con lleva una característica dinámica debido a su ubicación territorial compacta y rodeada del mayor epicentro de salubridad. En la gráfica 29, el Modelado td/dv , es una muestra con picos altos (tonalidad roja) a una tendencia de transiciones de muertes, infecciones superiores a las demás áreas estudiadas con acotes exponenciales del gran golpe que tuvo en las cuarentenas ,similar a otras capitales del mundo, siendo el área central para internar la mayor población.
- BUENOS AIRES, para este territorio podría asimilar las mismas razones expuestas anteriormente relacionadas con a la frecuencia y la densidad de población. Además, al ser Buenos Aires un lugar limítrofe a otros países con un problema mayor como lo fue Brasil, ya que este produjo muchos vuelos de población nativa y extranjera, lo que podría haber dado lugar a datos crecidamente dispersos.
- CORDOBA, el mapa IP ***Ilustración 29*** del estado de este territorio refleja un bajón de la población correspondiente a muertes, debido a que allí las infecciones fueron más recurrentes. La parte central de cada grafica esta dispersa debido a estos focos dibujados allí, reflejando su ubicación a las características de infecciones al no ser tan poblado y, por lo tanto, es testigo de la mayoría de los casos de diferentes anomalías que a futuro fueron convertidos en otras infecciones no pertenecientes al covid-19. Los límites de TP también siguen una tendencia similar.

4. ASPECTOS TECNOLÓGICOS Y METODOLÓGICOS DEL CASO DE ESTUDIO DESARROLLADO

El siguiente apartado describirá las componentes, recursos y herramientas implementadas a lo largo de toda la investigación. Proporcionando un índice correspondiente a lo realizado en Python y en otras aplicaciones de manipulación de datos. Por lo tanto, se procede a recorrer nuevamente todos los capítulos definidos en este documento, con la iniciativa de argumentar y explicar los pasos que intervienen en la investigación, en el área de programación o en su defecto debatir posibles inquietudes que no fueron resueltas.

▪ Requerimientos Básicos

Los Requerimientos dispuestos en la presente investigación recaen en la arquitectura de la computadora, debido a que la mayoría son aplicaciones de programación que requieren recursos del equipo, con el fin de compilar/procesar todo tipo de información. Por ende, se calcula las siguientes componentes:

- ✓ Servidor físico o máquina virtual.
- ✓ CPU: 2 CPU de 64 bits, 2,8 GHz, 8,00 GT/s o superior. Verificar la arquitectura

de la máquina.

- ✓ Memoria: tamaño mínimo de RAM de 8 GB, o 16 GB de RAM con DDR3 de 1600 MHz instalado, para una instalación típica con 1/10 usuarios habituales. Verifique los requisitos de memoria.

- ✓ Almacenamiento: Mínimo recomendado de 650 GB para un espejo de repo.anaconda.com o al menos 1,5 TB para un entorno con espacio libre. Se recomienda espacio adicional si se utiliza el Repositorio para almacenar paquetes creados por su organización. Verificar los requisitos de almacenamiento.

- ✓ Acceso a Internet para descargar los archivos, componentes y versiones de cada aplicación, o una unidad USB que contenga todos los archivos que necesita con instrucciones alternativas para instalaciones.

Ahora bien, los requerimientos generales dependen de la agente del Sistema Operativo, en nuestra investigación se utilizó la versión de Windows 10. Damos mención a esto, por temas de terminales/consolas que no acoplan algunas componentes, sin embargo, en las páginas principales de cada una de las herramientas implementadas ofrecen manuales de usuarios para cada Sistema Operativo.

Es importante destacar el nombre de las aplicaciones, con el fin de facilitar los requerimientos de instalación de cada una de ellas, recomendamos ingresar en cada una de sus páginas, los enlaces se describirán en el punto '***Instalación de aplicaciones***'.

- ✓ Worbench SQL.
- ✓ Power BI.
- ✓ Anaconda Navigator: Jupyter Network, Spyder.
- ✓ Weka.

- **Búsqueda de información**

La mayor parte de la investigación se llevó a cabo en bibliotecas virtuales, revistas de investigación, libros de Instituciones educativas y los portales web de salubridad generados por el gobierno Argentino. La otra parte de la investigación fue proporcionada por el tutor de la tesis, quien por medio de bibliografías aplicadas y referentes a las variantes de covid-19 determino un peso agregado a lo desarrollado en esta investigación.
El mayor aporte obtenido en la investigación, la otorgo la revista Plos Computacional, en dicho portal web, proveía constantemente información actualizada con respectivos dataset de datos y múltiples libros con el tema de SARS COVID-19.

Para concluir con este apartado, solo resta comentar que cada documento consultado estaría en disposición de todo aquel lector que desee darle continuidad a la investigación. Se pretende crear un drive en la web con todos los archivos .pdf , Dataset, bases de datos (transformadas).

- **Instalación de aplicaciones:**

Anteriormente mencionamos las aplicaciones implementadas en la investigación, ahora el deber es especificar las páginas webs principales de cada una de ellas con su respectivo apartado de descarga.

- ✓ **Workbench SQL:** https://dev.mysql.com/downloads/workbench/
- ✓ **Power BI:** https://powerbi.microsoft.com/es-es/desktop/
- ✓ **Anaconda Navigator:** Jupyter Notebook, Spyder. https://www.anaconda.com/download
- ✓ **Weka:** https://weka.sourceforge.io/packageMetaData/

▪ **Procedimiento (Paso a Paso)**

La investigación siguió una serie de pasos o procedimientos claros para lograr una buena sinergia y cumplir con cada una de las condiciones, que pedía el algoritmo de salubridad SIRD. Previo a esto se debía realizar una integración de datos referentes a tres pasos: extraer, transformar, cargar los datos previos de un dataset.

A continuación, se profundiza el procedimiento completo de cómo se trabajó en la investigación al pasar cada uno de los capítulos:

1. Disponer de un conjunto de datos referentes a la problemática mencionada en la investigación y al comportamiento del Covid-19. Esta información se obtuvo descargando el dataset anual de la virosis, en la página: *https://data.buenosaires.gob.ar/dataset/* .
   Una vez visualizado el contendido a priori de los datos que allí vienen, se procede a tratar todos los valores que deseamos implementar.

2. Efectuar una integración de los datos obtenidos anteriormente, para este procedimiento disponemos de la herramienta y aplicación Power BI, SQL Workbench. La principal función de este paso sería conocer los datos y moldearlos a la necesidad del cada uno de los modelos, es decir tomar los datos con mayor presión para nuestros algoritmos.

3. Luego de efectuar y convertir los datos de nuestro archivo raíz, este dataset debe tener una estructura más moldeable a la manipulación de los atributos, por ende, se facilitó el escenario: importando este nuevo ETL en una simetría de datos de SQL.

4. El archivo raíz paso de tener una estructura de datos .CSV a una estructura de datos MWB/.PBIX (SQL). La transformación de formatos de los datos favorece en gran medida, al tratamiento y sencilla manipulación de los datos y explotar algunas operaciones constantes, efectuando rápidas consultas necesitaras para agilizar la toma de variables, por ejemplo: *la media de los datos, máximos y mínimos de los datos, varianza de los datos, etc*.

5. Crear y estructurar del algoritmo de árboles. El procedimiento realizado para este escenario consta de 3 fases:
   La primera fase: Organizar los atributos necesarios para trabajar cómodamente, la elaboración se realizó en el entorno mencionado, MSQL Workbench. Por medio de consultas SQL, se construiría y se distribuyen de mejor manera los atributos y valores de una nueva tabla. Esta nueva tabla se interpreta como un DataFrame (formato de datos volátiles para un escenario en Python).
   La segunda fase: Programar por líneas de código los dos patrones de árboles que fueron comparados para determinar su mayor grado de solides con la información. Estos dos modelos de árboles cumplirán la función de condicionar y aproximar los valores constantes, para estructurar el armazón que se efectuará después de integrar toda la información al modelo SIRD.
   La tercera fase: Comparar los resultados y definir el nuevo DataFrame de valores que van a ser inyectados a los demás algoritmos definidos a lo largo de la investigación.

6. Al conservar múltiples DataFrame, uno para cada etapa del algoritmo SIRD, se dispone en crear y desplegar, este gran conjunto de incertidumbres y el principal escenario de la investigación. El modelo SIRD depende directamente de la agrupación de las constantes que se fueron identificando en la creación de los árboles de decisiones, una vez consolidado el DataFrame a implementar, se despliegan 3 propuestas o fases para resolver y acondicionar el modelo SIRD con lo desarrollado hasta el momento.

   La primera propuesta, cumple con visualizar una serie temporal de datos basados en algoritmos bases de Python, con esto se enfoca una proyección escalada de los métodos a implementar. La conclusión de este primer momento es dar inicio al entorno de predicción de datos en Jupyter Network, previo a este procedimiento se debe tener instalado los paquetes de las siguientes bibliotecas: *matplotlib, seaborn, collections, mitosheet, plotly*.
   Nota: las bibliotecas se deben instalar en la terminal de Anaconda, a partir de comandos, la página principal de ***ANACONDA DEVELOPERS*** te explican cómo funcionan y como se deben instalar.

   La segunda propuesta, cumple con identificar los periodos anuales, favoreciendo una amplia agrupación e implementación de código predictivo, logramos imprimir una brecha clara de cómo disminuyen o aumentan la correlación de muertes, contagios hasta predecir el final de estos incrementos en base a las tazas de error calculadas y también determinamos la fase final de esta pandemia, restándolo en el modelo SIRD.

   La última propuesta, se basó en identificar las columnas solucionadas por el teorema SIRD y a la agrupación de salidas de esa clase, es decir la salida de un valor se consideró como data de entrenamiento. Estas salidas son efectuadas por variables discretas/continuas, todo depende del momento requerido por el algoritmo. La conclusión de esta fase fue en identificar las variables creadas para cada año y con esto resolvemos la información sobre cada una de las condiciones concretas de la variable X (Muerte, Recuperados, Síntomas finales).

7. Proporcionar detalles de la primera muestra para cada año del Modelo SIRD, ajustando el modelo a partir de agrupaciones referentes a las variables categóricas/numéricas basados en el aprendizaje supervisado/no supervisados, donde las muertes modeladas e informadas por la distribución de las infecciones se consideran referentes a la *edad, sexo y muertes*. Se debía cumplir el estimado *N* de 15 días del mes por muestra. Posterior a esto se efectuaban las gráficas del año 2020,2021,2022 con las ciudades mencionadas.

8. Desarrollar el segundo criterio correspondiente a las Afecciones Persistentes con respecto a la agrupación de los datos. Con esta definición de afecciones persistentes por Covid-19 se crea un ajuste y se adiciona todas las variables Dummy (categóricas), y estas variables concordaron con las salidas del modelo junto con sus valores determinísticos.

9. Apoyar y agregar el seudo-algorimto Upper Fence dentro del segundo criterio, con el fin de proyectar una barrera con los limites inferiores/superiores de los datos y cualquier dato que se concentre fuera de estos límites definidos, puede considerarse valores atípicos. Por ende, disponer de 3 tablas y su respectiva la gráfica con los resultados por cada año estudiado es un esquema idóneo para el código en Python.

10. Finalizar el modelo SIRD. Dar fina a este modelo, dependia de la *Sustracción, Transmisión* como parámetros principales de sus fases. El propósito fue definir una tasa de transmisión inicial ***Beta***, con ella se visualizan las distintas aproximaciones en tasas de tiempo variables, permitiendo mejor las predicciones con respecto a cada dato filtrado

con los días ya definidos. Estos resultados se muestran coherentes con las variantes del algoritmo de programación, cuando son encontradas en base a las fases y el tiempo de cada tasa de contagio con una equivalencia en tiempo de Beta > 0 y disminuye o incrementa con respecto al tiempo.

11. Se concluye con una breve discusión de las muestras y resultados predictivos del modelo SIRD. Se toma cada área estudiada, con el fin de profundizar algunas pautas que no fueron resueltas, una de esas pautas fue las bajas de crecimiento de los casos y la aparición de una nueva variante gripal, pero sin demasiados epicentros o información referente a esta. Se concluye con la última fase del modelo SIRD, muertes. Estas muertes son normalizadas y graficadas, dando fin a todas las etapas de este modelo con resultados favorables.

12. Damos inicio al filtrado del algoritmo SIRD a través del Modelo Bayesiano. Este modelo se trabajó por varias capas, estas capas determinaban el estado del CPA (Análisis de puntos de cambio) para las 3 áreas de estudio. El procedimiento inicial es percibir la propagación de la variante, con la finalidad de referenciar la distribución entre lo que se calculó antes de analizar cada fase del algoritmo con la fase obtenida después de finalizar el algoritmo SIRD. Previamente analizar los datos con este nuevo modelo, reflaja una medida de actualizar los datos de las muertes, con datos de nueva variante o la disminución del brote del covid-19.

    Se aplica el CPA en cada área planteada en el DataFrame final, el punto de cambio de cada estado indico que hubo muchos cambios graduales durante cada periodo evaluado.El punto de cambio esta precedido de cada valor nulo que se ven en la muestra. Esta muestra es graficada por agrupamiento de los puntos de cambios.
    Se efectúa el cálculo de la correlación en delta de x (dΔ), utilizando los datos: rango de notificación del primer caso y la dΔ muestra, la discrepancia en las semanas de los casos, antes del cierre y la semana del punto de cambio de los estados de la variante COVID-19. Por otro lado, los resultados CPA representan los enfoques adaptados por cada recuadro de la gráfica estipulada.
    Luego se efectúa la correlación de Person en nuestro modelo CPA, esta dispone de los datos y factores calculados hasta el momento. El análisis lineal se ajustó a la diagonal obtenida en la correlación de Person, con esta dinámica otorgamos un sesgo preciso al conjunto de datos.
    Después buscamos los valores *'r'* (coeficiente de correlación) para cada uno de los factores. Los factores determinan la correlación inversa para cada factor, calculamos los factores de delta como: CPP (casos por unidad de Población), CPUA (Casos por unidad de área) y CPD (casos por densidad de población). Indica que cuanto mayor sea el valor de Δ, menor será el CPP, CPD o CPUA. Por lo tanto, cuanto antes hubiera sido la acción, menores habrían sido los casos y la tasa de progresión.
    Al finalizar se procesa el modelado de dt/dv con las predicciones con valores determinísticos, los valores de *P* (población) son menores que 0,05 e Inicio (t) > t crítico, tanto para una pila como para dos pilas (o más). La relación entre delta y CPD muestra una correlación débil con solo un valor de *P* cercano a 0,05 para una cola. Por lo tanto, se puede concluir con seguridad que los casos por unidad de población o área se habrían controlado si la delta hubiera sido mayor.

13. Graficar el modelo Gaussian Naive Bayes, dados los resultados obtenidos por el modelo SIRD, los resultados de la última fase (Muertes), se analizaron cada uno de los estados en búsqueda de posibles puntos críticos y zonas seguras utilizando IP (infecciones/poblaciones) como se explica a continuación. El área es un factor inherente al IP que se decide en función de los valores de puntos resueltos. Por lo tanto, IP está experimentado una variante de la población y los casos de COVID-19 de cada estado para comprender la importancia de la correlación,

14. Para finalizar el recorrido de la investigación total se procede a concluir con una discusión de cada estado obtenido, reflejando los resultados a favor, las predicciones resueltas y una posible versión a trabajar.

5. CONCLUSIÓN

La infección transmitida por contextos de salubridad por Covid-19, permitieron establecer medidas de control destinadas a la población, estas medidas pertinentes estuvieron presentes en diferentes áreas de investigación con el fin de reducir o determinar un modelado predictivo de amenazas potenciales de altas y bajas probabilidades resultantes de la epidemia.

El trabajo de investigación analizo en detalle las razones en el momento de los confinamientos y su eficiencia para controlar la propagación del virus, y, por consiguiente, de la virosis misma. El Modelo Bayesiano se solucionó utilizando un entorno de programación Python, para comprender el punto de inflexión de 3 áreas de Argentina, es decir, para comprender el punto de partida de cual la tasa de propagación de la enfermedad cambio significativamente.

El punto de cambio se comparó con las semanas que transcurrieron antes de que llegara el primer caso a Capital Federal y el cierre nacional de Argentina. Se demostró que el factor dx es correlacionado con CPP y CPUA, como se demostró en el último apartado de resultados como variable de entrada t(velocidad), es decir, inicio en tv > tv critico y $p < 0.005$. Sin embargo, la relación es inversamente proporcional como lo mestra la ‘r’ de Person. Indicando que la predicción tuvo que haber empezado desde el primer cuatrimestre del año.

La estructura del código e investigación definida a lo largo de este documento orientó algoritmos predictivos basados en modelos matemáticos, fundamentales para particionar las etapas del esparcimiento de la enfermedad con sus respectivos modelados necesarios para permitir manejar los capítulos de la investigación. El procedimiento evaluativo de la pandemia generada por el virus COVID-19 (SARS COV-2) se basó en diferentes modelos de estudio, cada uno con aproximaciones de evaluación diferentes que con llevan a la programación epidemiológica básica del modelo SIRD. Esta resolución practica/investigativa se planteó aprovechando cada fase del modelo mencionado con la finalidad de medir, predecir y modelar la pandemia en Argentina, se reducen las matrices de datos requeridos para las condiciones del algoritmo, brindaron un valor de libertad mas amplio a partir de cada ajuste que se realizo en los capítulos de resolución con el fin de reemplazar cada etapa con dicho procedimiento, cada uno de ellos dependieron de la variable de entrada velocidad/tiempo.
Por consiguiente, los 3 métodos efectuados para cada fase (Susceptibles, Infectados, Muertes, Recuperados) proporcionaban variables de salida con parámetros ajustados a cada regla del algoritmo, delta de x (Δx) siendo la variable de salida mas justificada para los datos reales que se iban prediciendo durante los 3 años, a partir de los valores actualizados del Ministerio de Salud Argentino. El modelado de cada una de las fases obtenidas con su respectivo ajuste (indicados por los modelos) contempla un modelo moldeable a ETLs (cajones de datos) preparados para acoplarse al modelo epidemiológico SIRD, el análisis y la valoración correcta y cambiante en cada evaluación en las 3 áreas estudiadas en plazos acotados a muestras quincenales, para el algoritmo;

muestras trimestrales para el rango de errores y muestras mensuales para las muertes/recuperaciones, cumpliendo plazos acordes a la necesidad.
Sin embargo, las mejoras planeadas para trabajos futuros que se deseen trabajar en base a una comparativa de otra problemática con la idea desarrollada en este documento puede basarse en lapsos de tiempos diferentes, una transición de clases de entrada y salida diferentes como Gamma γ, representando la transición de una nueva variante de salud y como esta constante pueda predecir sus efectos a la población con respecto a las provocadas por el Covid-19, por ejemplo en un apartado se menciona una breve comparativa referente a Influenza (H1N1) post Covid-19, esta nueva variante de salud se acoplaría de manera continua ya que con una nueva entrada (tiempo) y parámetro de salida γ , se evaluaría este modelo SIRD con ajustes mas reales debido al conocimiento de las nuevas muestras y la habilidad de desempeñar el algoritmo para esta nueva problemática; incorporando posibles métodos sistema dinámico y cambiante en el transcurso del tiempo, conformado por variables aleatorias y es analizado en términos de probabilidad.

La simulación de infecciones e intervenciones de agrupamiento, cumplen una finalidad de reducir probabilidades en varios escenarios de componentes basado en matrices de volumen de datos agudos, por ende la programación cumple un papel importante en la presente investigación, por el motivo de asimilar algoritmos de modelos matemáticos, estos pueden ayudar a comprender como sería el comportamiento del Covid-19 para aislar o indicar la patología que presente la población e informar las medidas de contingencia que permitan mitigar la transmisión futura. Por ende, se la simulación efectuada siempre busco la trayectoria de la pandemia utilizando el modelo SIRD con algoritmos presentes en PYTHON.

6. ANEXO [Código Fuente]

En el presente apartado, anexo se detallan los aspectos de implementación y código más relevantes de la solución desarrollada

- **Líneas de código (Consola)**

Valorando cada una de las instancias desarrolladas hasta le momento, solo resta en mencionar los bloques de código diseñados para compilar cada escenario de los modelos de programación planteados con su respectiva citación en el texto respectivo.

Anexo 1, Capitulo 3.3.

```
Input [1]: pandas as pd

Input [2]: matplotlib. pyplot as plt

Input [3]: numpy as np

Input [4]: import pandas as pd

Input [5]: import matplotlib.pyplot as plt

Input [6]:

 ruta1="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"

Input [7]:

pacientes=pd.read_excel
```

```
(ruta1+"pacientes_v2.xlsx”,parse_dates=True,sheet_name='pacientes_
cuatro’)

Input [8]: pacientes

Input [9]:

inf_contagio = pacientes[pacientes["problema_cardiaco"]==0]

Input [10]:

inf_respiratoria = pacientes[pacientes["problema_cardiaco"]==1]

Input [11]:

no_aplica = pacientes[pacientes["problema_cardiaco"]==2]

Input [12]: inf_contagio

Input [13]: inf_respiratoria

Input [14]: no_aplica-

Input [15]:

Inf_contagio = pacientes[pacientes["problema_covid"]==0]

Input [16]:

Inf_respiratoria = pacientes[pacientes["problema_covid"]==1]

Input [17]:

noresultados = pacientes[pacientes["problema_covid"]==2]

Input [18]: plt.figure(figsize=(6, 6))

Input [19]:

plt.title('Datos muestra 2022 (no son linealmente separables)',
fontsize = 21, style = "italic")

Input [20]: plt.xlabel('Edad', fontsize = 20.0)

Input [21]: plt.ylabel ('Muestra', fontsize = 20.0)

Input [21]:

plt.scatter    (Inf_respiratoria    ["edad"],    Inf_respiratoria
["problema_covid"],

Input [22]:

label="Posible  Inf_respiratoria  (Respiratoria:  0)",  marker="*",
c="skyblue", s=200)

Input [23]:

plt.scatter(cardiacos["edad"], cardiacos["problema_covid"],

Input [24]:

label="Posible  Variante  Infeciosa(Corporal:  1)",  marker="*",
c="lightcoral", s=200)

Input [25]:
```

```
plt. scatter(noresultados["edad"], noresultados["problema_covid"],
```

```
Input [26]:
```

```
label="No Presenta ninguna variante (Dado de alta: 2)", marker="*", c="chartreuse", s=200)
```

```
Input [27]: plt.legend(bbox_to_anchor=(1, 0.15))
```

```
Input [28]: plt.show()
```

El bloque de código descripto anteriormente es muy similar para los años 2021,2022. Solo se debe cambiar el archivo .CSV por el correspondiente al año, visto en la línea de código `Input [7]:`

Anexo 2, Capitulo 3.4.

```
Input [1]:  import pandas as pd
```

```
Input [2]: import matplotlib.pyplot as plt
```

```
Input [3]: import numpy as np
```

```
Input [4]: import pandas as pd
```

```
Input [5]: import matplotlib.pyplot as plt
```

```
Input [6]: df=pd.get_dummies(data=df,drop_first=True)
```

```
Input [7]: df.sample(10)
```

Anexo 2, Capitulo 3.4.

```
Input [1]: from sklearn.tree import DecisionTreeClassifier
```

```
Input [2]: model= DecisionTreeClassifier(max_depth=4)
```

```
Input [3]: model.fit(X=explicativas, y=objetivo)
```

Anexo 3, Capitulo 3.4.

```
Input [1]: from sklearn.tree import plot_tree
```

```
Input [2]: import matplotlib.pyplot as plt
```

```
Input [3]: plt.figure(figsize=(12,8))
```

```
Input [4]:
```

```
plot_tree(decision_tree=model,feature_names=explicativas.columns,filled=True).
```

Anexo 4, Capitulo 3.4.

```
Input [1]: import seaborn as sns
```

```
Input [2]: sns.histplot(x=df.edad ,hue=df.Variante_Predicha)
```

Anexo 5, Capitulo 3.4.

```
Input [1]: explicativas=df.drop(columns='Variante_Predicha')
```

```
Input [2]: objetivo=df.Variante_Predicha.
```

Anexo 6, Capitulo 3.4

```
Input [1]: from sklearn.tree import DecisionTreeClassifier

Input [2]: model= DecisionTreeClassifier(max_depth=4)

Input [3]: model.fit(X=explicativas, y=objetivo)
```

Anexo 7, Capitulo 3.5

```
Input [1]:

ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"

Input [2]: cov19=pd.read_excel(ruta+"yyy.xlsx", parse_dates=True)

Input [3]: cov19
```

Anexo 8, Capitulo 3.5

```
Input [1]: sns.lineplot(df)
```

Anexo 9, Capitulo 3.6

```
Input [1]:

sheet_df_dictonary

=pd.read_excel(r'C:\Users\carlo\Documents\TesisMaestria\2023\yyy.xlsx', engine='openpyxl', sheet_name=['2020', '2021', '2022'], skiprows=0)

Input [2]: df_2020 = sheet_df_dictonary['2020']

Input [3]: df_2021 = sheet_df_dictonary['2021']

Input [4]: df_2022 = sheet_df_dictonary['2022']
```

Anexo 10, Capitulo 3.6.1

```
Input [1]: import plotly.express as px

 # Construct the graph and style it. Further customize your graph
 by editing this code.

 #       See       Plotly       Documentation       for       help:
 https://plotly.com/python/plotly-express/

Input     [2]:     fig     =     px.line(df_2020_1,     x='Fecha',
y='Muestra_apertura', color='CIUDADES', line_shape='linear')

Input [3]: fig.update_layout(

    title='1º Muestra Apertura Covid-19 (x15 días) 2020',

     xaxis = dict(

        title='Fechas - Rango Quincenal',

        showgrid=True,

        rangeslider = dict(

            visible=True,
```

```
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='Primera Muestra',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
)
Input [4]: fig.show(renderer="iframe")
```

Anexo 11, Capitulo 3.6.1

```
Input [1]: import plotly.express as px
 # Construct the graph and style it. Further customize your graph
 by editing this code.
 #       See       Plotly       Documentation       for       help:
 https://plotly.com/python/plotly-express/
Input [2]:
fig    =    px.line(df_2020,    x='Fecha',    y='Segunda_muestra',
color='CIUDADES', line_shape='linear')
Input [3]: fig.update_layout(
    title='2° Muestra Procesada Covid-19 (x15 Dias) 2020',
    xaxis = dict(
        title='Fechas - Rango Quincenal',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='Segunda Muestra',
        showgrid=True
```

```
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
)
Input [4]: fig.show(renderer="iframe")
```

Anexo 12 ,Capitulo 3.6.1

```
Input [1]: import plotly.express as px
# Construct the graph and style it. Further customize your graph
by editing this code.
#       See       Plotly       Documentation       for       help:
https://plotly.com/python/plotly-express/
Input [2]:
fig  =  px.line(df_2020,  x='Fecha',  y='TerceraM_clasificacion',
color='CIUDADES', line_shape='linear')
Input [3]: fig.update_layout
  ( title='3° Muestra Clasificada Covid-19 (x15 días) 2020',
    xaxis = dict(
        title='Fechas - Rango Quincenal',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='Segunda Muestra',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
 )
```

```
  Input [4]: fig.show(renderer="iframe")
```

Anexo 13, Capitulo 3.6.2

```
Input [1]: import plotly.express as px

# Construct the graph and style it. Further customize your graph by editing this code.

#       See       Plotly       Documentation       for       help: https://plotly.com/python/plotly-express/

Input [2]:

fig = px.box(df_2020, x='CIUDADES', y='Inf_Respiratoria_Primera', color='sexo', points='outliers')

fig.update_layout(

    title='Transmisión Covid-19- Segunda Muestra',

    xaxis = dict(

        title='Ciudades',

        showgrid=True,

        rangeslider = dict(

            visible=True,

            thickness=0.05

        )

    ),

    yaxis = dict(

        title='Infecciones Respiratoria',

        showgrid=True

    ),

    legend = dict(

        orientation='v'

    ),

    paper_bgcolor='#FFFFFF'

 )

Input [4]: fig.show(renderer="iframe")
```

Anexo 14, Capitulo 3.6.2

```
Input [1]: import plotly.express as px

# Construct the graph and style it. Further customize your graph by editing this code.

#       See       Plotly       Documentation       for       help: https://plotly.com/python/plotly-express/
```

```
Input [2]:
fig = px.box(
 df_2020, x='CIUDADES',
 y='Contagio_Comporal_Segunda',
 color='sexo', points='outliers'
)
Input [3]: fig.update_layout(
    title='2- Transmisión Covid-19',
    xaxis = dict(
        title='Ciudades',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='2-Infecciones Respiratoria',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
)
Input [4]: fig.show(renderer="iframe")
```

Anexo 15, Capitulo 3.6.2

```
Input [1]: import plotly.express as px
# Construct the graph and style it. Further customize your graph by
editing this code.
#       See       Plotly       Documentation       for       help:
https://plotly.com/python/plotly-express/
Input [2]:
fig = px.box(df_2020, x='CIUDADES', y='Inf_Respiratoria_Tercera',
color='sexo', points='outliers')
```

```
Input [3]: fig.update_layout(
    title='3- Transmision Covid-19',
    xaxis = dict(
        title='Ciudades',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='3-Infecciones Respiratoria',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
 )
Input [4]:fig.show(renderer="iframe")
```

Anexo 16, Capitulo 3.6.2

```
Input [1]: import plotly.express as px
# Construct the graph and style it. Further customize your graph by
editing this code.
#       See       Plotly       Documentation       for       help:
https://plotly.com/python/plotly-express/
Input [2]:
fig = px.box(
 df_2020, x='CIUDADES',
 y='Contagio_Comporal_Tercera',  color='sexo',
points='outliers')
Input [3]: fig.update_layout(
    title='3- Transmisión Covid-19',
    xaxis = dict(
        title='Ciudades',
```

```
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='3-Infecciones Corporal',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
    ),
    paper_bgcolor='#FFFFFF'
 )
Input [4]:fig.show(renderer="iframe")
```

Anexo 17, Capitulo 3.7

```
Input [1]: from sympy import Derivative, diff, simplify,Symbol
Input [2]: import numpy as np
Input [3]: import matplotlib.pyplot as plt
Input [4]: from math import log
Input [5]: from sympy import *
Input [6]:
ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"
#index_col=0


Input [7]:
df=pd.read_excel(ruta+"yyy.xlsx", parse_dates=True)
Input [8]: df
```

A continuación, se complementan aalgunos bloques de código indispensables para la conclusión de la tesis, pero por temas de argumentación no hubo la necesidad de citarlo en el documento. Sin embargo, para futuras investigaciones seria ideal conservar el orden y poder citarlo con su respectiva disposición.

Anexo 18, Primera Muestra 2020.

```
Input [1]: import numpy as np
```

```
Input [2]: from scipy.integrate import odeint

Input [3]: import matplotlib.pyplot as plt

Input [4]: muestraCaba= 26385

Input [5]: muestraBA=32084

Input [6]: muestraCordoba=532

Input [7]: ncasos=muestraCaba+muestraBA+muestraCordoba

Input [8]: ncasos

Input [9]: Npersonas= 263352

Input [10]: TI=ncasos/Npersonas

Input [12]: TI*7

 # Población total, N

Input [13]: N =   263352

 # Valores iniciales de personas Infectadas (I0) y personas que se
 recuperaron (R0)

Input [14]: I0, R0 = 1, 0

 # El resto de la población, S0 son las personas que están sujetos
 a infección inicialmente.

Input [15]: S0 = N - I0 - R0

 # Tasa de transmisión, tasa de recuperación en 1/días

Input [16]: beta, gamma = 1.56, 1/15

 # Las ecuaciones diferenciales del modelo SIR

Input [17]: def deriv(y, t, N, beta, gamma):

Input [18]: S, I, R = y

Input [19]: dSdt = -beta * S * I / N

Input [20]: dIdt = beta * S * I / N - gamma * I

Input [21]: dRdt = gamma * I

Input [22]: return dSdt, dIdt, dRdt

   # Vector de las condiciones iniciales

  Input [23]: y0 = S0, I0, R0

   #  Resolver  el  sistema  de  ecuaciones  diferenciales,  en  la
 secuencia de días que ya definimos

Input [24]: ret = odeint(deriv, y0, t, args=(N, beta, gamma))

Input [25]: S, I, R = ret.T

 # Plot the data on three separate curves for S(t), I(t) and R(t)

Input [26]: import plotly.graph_objects as go
```

```
Input [27]: fig = go.Figure()

Input [28]: fig.add_trace(go.Scatter(x=t,
y=I,mode='lines',name='Infected'))

Input [29]: fig.show()

 ##Para la parte de las gráficas:

Input [30]: import cufflinks as cf

Input [31]: import plotly.offline as py

Input [32]: py.iplot([{

   'x': t,

   'y': modelo(*valor),

   'name': str(valor),

 }  for valor in valores])
```

Anexo 19, Segunda Muestra 2020

```
Input [1]: muestraCaba2= 111076

Input [2]: muestraBA2=32066

Input [3]: muestraCordoba2=526

Input [4]: ncasos2=muestraCaba2+muestraBA2+muestraCordoba2

Input [5]: ncasos2

Input [6]: Npersonas= 263352

# Población total, N

Input [7]: N1 =   263352

# Valores iniciales de personas Infectadas (I0) y personas que se
recuperaron (R0)

Input [8]: I1, R1 = 1323, 990

# El resto de la población, S0 son las personas que están sujetos a
infección inicialmente.

Input [9]: S1 = N1 - I1 - R1

 # Tasa de transmisión, tasa de recuperación en 1/días

Input [10]: beta1, gamma1 = 3.5, 1/15

 # Puntos en la gráfica (En días)

Input [11]: t1 = np.linspace(0, 122,122)

Input [12]: TI2=ncasos/Npersonas

Input [13]: TI2*7

 # Población total, N

Input [14]: N1 =   263352
```

```
 # Valores iniciales de personas Infectadas (I0) y personas que se
 recuperaron (R0)

Input [15]: I1, R1 = 1323, 990

 # El resto de la población, S0 son las personas que están sujetos
 a infección inicialmente.

Input [16]: S1 = N1 - I1 - R1

 # Tasa de transmisión, tasa de recuperación en 1/días

Input [17]: beta1, gamma1 = 3.5, 1/15

 # Puntos en la gráfica (En días)

Input [18]: t1 = np.linspace(0, 122,122)

# Las ecuaciones diferenciales del modelo SIR

Input [19]: def deriv(y1, t1, N1, beta1, gamma1):

Input [20]: S1, I1, R1 = y1

Input [21]: dSdt1 = -beta1 * S1 * I1 / N1

Input [22]: dIdt1 = beta1 * S1 * I1 / N1 - gamma1 * I1

Input [23]: dRdt1 = gamma1 * I1

Input [24]: return dSdt1, dIdt1, dRdt1

# Vector de las condiciones iniciales

Input [25]: y1 = S1, I1, R1

# Resolver el sistema de ecuaciones diferenciales, en la secuencia
de días que ya definimos

Input [26]: ret1 = odeint(deriv, y1, t1, args=(N1, beta1, gamma1))

Input [27]: S1, I1, R1 = ret1.T

# Plot the data on three separate curves for S(t), I(t) and R(t)

Input [28]: import plotly.graph_objects as go

Input [29]: fig = go.Figure()

Input [30]:

fig.add_trace(go.Scatter(x=t1, y=I1,mode='lines',name='Infected'))

Input [31]: fig.show()

 ##Para la parte de las gráficas:

Input [32]: import cufflinks as cf

Input [33]:import plotly.offline as py

Input [34]: py.iplot([{

   'x': t1,

   'y': modelo(*valor),
```

```
   'name': str(valor),
  }  for valor in valores])
```

Anexo 20, Tercera Muestra 2020

```
Input [1]: muestraCaba31= 85490

Input [2]: muestraBA31=31079

Input [3]: muestraCordoba31=528

Input [4]: ncasos31=muestraCaba31+muestraBA31+muestraCordoba31

Input [5]: ncasos31

Input [6]: Npersonas31= 263352

Input [7]: TI31=ncasos31/Npersonas31

Input [8]: TI31*6

# Población total, N

Input [9]: N31 =   263352

# Valores iniciales de personas Infectadas (I0) y personas que se
recuperaron (R0)

Input [10]: I31,R31= 69591, 9759

# El resto de la población, S0 son las personas que están sujetos a
infección inicialmente.

Input [11]: S31= N31- I31- R31

# Tasa de transmisión, tasa de recuperación en 1/días

Input [12]: beta31, gamma31 = 3, 1/15

# Puntos en la gráfica (En días)

Input [13]: t31 = np.linspace(0, 30,30).

# Las ecuaciones diferenciales del modelo SIR

Input [14]: def deriv31(y31, t31, N31, beta31, gamma31):

Input [15]: S31, I31, R31 = y31

Input [16]: dSdt31 = -beta31 * S31 * I31 / N31

Input [17]: dIdt31 = beta31 * S31 * I31 / N31 - gamma31 * I31

Input [18]: dRdt31 = gamma31 * I31

Input [19]: return dSdt31, dIdt31, dRdt31

# Vector de las condiciones iniciales

Input [20]: y31 = S31, I31, R31

# Resolver el sistema de ecuaciones diferenciales, en la secuencia
de días que ya definimos

Input [21]: ret = odeint(deriv31, y31, t31, args=(N31, beta31,
gamma31))
```

```
Input [22]: S31, I31, R31 = ret.T

# Plot the data on three separate curves for S(t), I(t) and R(t)

Input [23]: import plotly.graph_objects as go

Input [24]: fig = go.Figure()

Input              [25]:              fig.add_trace(go.Scatter(x=t31,
y=I31,mode='lines',name='Infected'))

Input [26]: fig.show()

##Para la parte de las gráficas:

Input [27]: import cufflinks as cf

Input [28]: import plotly.offline as py

Input [29]: py.iplot([{

   'x': t31,

   'y': modelo(*valor),

   'name': str(valor),

}  for valor in valores])
```

Nota: El mismo procedimiento es cconsecuente para las siguientes muestras 2021, 2022. Solo es cambiar la ruta y el archivo correspondiente al año.

Anexo 21, Capitulo 4

```
Input [1]:

ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"

 #index_col=0

Input [2]:

dt=pd.read_excel(ruta+"datos_c.xlsx", parse_dates=True)

Input [3]: dt
```

Anexo 22, Capitulo 4

```
Input [1]: import plotly.express as px

Input [2]: fig = px.histogram(dt, x="Rango_edades", y="total_bill",
color="Meses", title="Ilustración Normalizada -Datos Covid 2020",

width=600, height=400, labels=

 { # replaces default labels by column name

         "sex": "Meses 2020",

        "day": "Normalización por edades","total_bill":

        "Importante performence"

            },

          category_orders={ # replaces default order by column name
```

```
        "day": ["Thur", "Fri", "Febrero", "Enero"],
        "sex": ["Male", "Female"]
            },
            color_discrete_map=
        { # replaces default color mapping by value
                "Male": "RebeccaPurple", "Female": "MediumPurple"
            },
            template="simple_white"
            )
Input [3]: fig.show()
```

Anexo 22, Capitulo 4

```
Input [1]:
 ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"
 #index_col=0
Input [2]:
du=pd.read_excel(ruta+"death_final.xlsx",
parse_dates=True,sheet_name='2020_death')
Input [3]: du
Input [4]: import plotly.express as px
Input [5]:
fig  =  px.histogram(dcs,  x="Promed_mensual_CA",  y="total_bill",
color="Meses",
            title="Ilustración Normalizada -Datos Covid 2020",
            width=600, height=400,
            labels={ # replaces default labels by column name
               "sex": "Meses 2020",
              "day": "Normalización por edades",
              "total_bill": "Importante performence"
            },
            category_orders={
              # replaces default order by column name
                "day": ["Thur", "Fri", "Febrero", "Enero"],
                "sex": ["Male", "Female"]
            },
            color_discrete_map=
```

```
            { # replaces default color mapping by value
                "Male": "RebeccaPurple", "Female": "MediumPurple"
            },
            template="simple_white"
            )
Input [5]: fig.show()
```

Anexo 23, Capitulo 4

```
Input [1]:
u=pd.read_excel(ruta+"datos_c.xlsx",
parse_dates=True,sheet_name='20212')
Input [2]: u
Input [3]: import plotly.express as px
Input [4]:
fig = px. histogram(
   u, x="Rango_edades", y="total_bill", color="Meses",
       title="Ilustración Normalizada -Datos Covid 2022)",
       width=600, height=400,
      labels={ # replaces default labels by column name
           "sex": "Meses 2021",
           "day": "Normalización por edades",
           "total_bill": "Importante performence"
            },
            category_orders={
             # replaces default order by column name
                "day": ["Thur", "Fri", "Febrero", "Enero"],
                "sex": ["Male", "Female"]
            },
            color_discrete_map=
            { # replaces default color mapping by value
                "Male": "RebeccaPurple", "Female": "MediumPurple"
             },
            template="simple_white"
            )
Input [5]: fig.show()
```

Anexo 23, Capitulo 4

```
Input                                                                [1]:
ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"

  #index_col=0

Input [2]:

muertes2020=pd.read_excel(ruta+"final_death.xlsx",
parse_dates=True,sheet_name='death_2020')

Input [3]: muertes2020

Input [4]:

muertes2021=pd.read_excel(ruta+"final_death.xlsx",
parse_dates=True,sheet_name='death_2021')

Input [5]: muertes2021

Input [6]:

muertes2022=pd.read_excel(ruta+"final_death.xlsx",
parse_dates=True,sheet_name='death_2022')

Input [7]: muertes2022

Input [8]: import plotly.express as px

 # Construct the graph and style it. Further customize your graph
 by editing this code.

 #See         Plotly         Documentation         for         help:
 https://plotly.com/python/plotly-express/

Input [9]:

fig   =   px.bar(muertes2020,   x='Ciudades',   y='Cantidad_muerts',
color='Meses')

Input [10]:fig.update_layout(

    title='Transmisión Covid-19',

    xaxis = dict(

        title='Ciudades',

        showgrid=True,

        rangeslider = dict(

            visible=True,

            thickness=0.05

        )

     ),

      yaxis = dict(

        title='Infecciones Respiratoria',

        showgrid=True
```

```
      ),
      legend = dict(
        orientation='v'
       ),
       paper_bgcolor='#FFFFFF'
        )
Input [11]: fig.show(renderer="iframe")
Input [12]:
 muertes2023=pd.read_excel
   (
        ruta+"final_death.xlsx",
       parse_dates=True,sheet_name='death_2023'
    )
Input [13]: muertes2023
```

Anexo 24, Capitulo 4

```
Input [1]: import plotly.express as px
 # Construct the graph and style it. Further customize your graph
 by editing this code.
 #       See       Plotly       Documentation       for       help:
 https://plotly.com/python/plotly-express/
Input [2]:
 fig = px.bar(
  muertes2021, x='Ciudades', y='Cantidad_muerts', color='Meses'
 )
Input [3]: fig.update_layout(
    title='Transmisión Covid-19',
      xaxis = dict(
        title='Ciudades',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
     ),
      yaxis = dict(
```

```
        title='Infecciones Respiratoria',
        showgrid=True
     ),
      legend = dict(
        orientation='v'
      ),
    paper_bgcolor='#FFFFFF'
 )
Input [4]:fig.show(renderer="iframe")
```

Anexo 25, Capitulo 4

```
Input [1]: import plotly.express as px
  # Construct the graph and style it. Further customize your graph
  by editing this code.
  #       See       Plotly       Documentation       for       help:
  https://plotly.com/python/plotly-express/
Input [2]:
fig = px.bar(
  muertes2022, x='Ciudades', y='Cantidad_muerts', color='Meses')
Input [3]: fig.update_layout(
    title='Transmisión Covid-19',
    xaxis = dict(
        title='Ciudades',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
       ),
       yaxis = dict(
        title='Infecciones Respiratoria',
        showgrid=True
      ),
       legend = dict(
        orientation='v'
     ),
```

```
    paper_bgcolor='#FFFFFF'
   )
Input [4]: fig.show(renderer="iframe")
```

Anexo 26, Capitulo 4

```
Input [1]: import matplotlib.pyplot as plt
Input [2]: import seaborn as sns
Input [3]: import plotly
Input [4]: import plotly.express as px
# Construct the graph and style it. Further customize your graph by editing this code.
#       See       Plotly       Documentation       for       help: https://plotly.com/python/plotly-express/
Input [5]:
 fig = px.box
 (
  death_2020, x='Ciudades',
 y=['Variacion_muertes', 'Error_estandar',
 'Promed_mensual'], color='Meses', points='outliers'
 )
Input [6]: fig.update_layout(
    title='Variante de muertes mensuales /Error Estandar 2020',
    xaxis = dict(
        title='Valor x Muertes',
        showgrid=True,
        rangeslider = dict(
            visible=True,
            thickness=0.05
        )
    ),
    yaxis = dict(
        title='Ciudades',
        showgrid=True
    ),
    legend = dict(
        orientation='v'
```

```
    ),
    paper_bgcolor='#FFFFFF'
)
Input [7]: fig.show(renderer="iframe")
```

Anexo 27, Capitulo 4.1

```
Input [1]: import pandas as pd
Input [2]:
 ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"
  #index_col=0
Input[3]:
df2020=pd.read_excel(ruta+"naive_bayes.xlsx",
parse_dates=True,sheet_name='2020')
Input [4]: df2020.
Input [5]:
df2021=pd.read_excel(ruta+"naive_bayes.xlsx",
parse_dates=True,sheet_name='2021')
Input [6]: df2021
Input [7]:
df2022=pd.read_excel(rut", parse_dates=True,sheet_name='2022')
Input [8]: df2022.
```

Anexo 28, Capitulo 4.1

```
Input [1]: import pandas as pd
Input [2]: import numpy as np
Input [3]: import matplotlib.pyplot as plt
Input [4]: from matplotlib import colors
Input [5]: import seaborn as sb
 %matplotlib inline
Input [6]: plt.rcParams['figure.figsize'] = (16, 9)
Input [7]: plt.style.use('ggplot')
Input [8]: from sklearn.model_selection import train_test_split
Input [9]: from sklearn.metrics import classification_report
Input [10]: from sklearn.metrics import confusion_matrix
Input [11]: from sklearn.naive_bayes import GaussianNB
Input [12]: from sklearn.feature_selection import SelectKBest
Input [13]:
```

```
 ruta="C:\\Users\\carlo\\Documents\\TesisMaestria\\2023\\"

  #index_col=0

Input [14]:

dataframe =pd.read_excel

 (

 ruta+"comprar_alquilar.xlsx",
 parse_dates=True,sheet_name='comprar_alquilar')

Input [15]: dataframe.head(200)
```

Anexo 29, Capitulo 4.1

```
Input [1]: print(dataframe.groupby('muerte').size())

Input [2]: dataframe.drop(['muerte'], axis=1).hist()

Input [3]: plt.show()
```

Anexo 30, Capitulo 4.1

```
Input [1]:

dataframe['epidemia']=(dataframe['muestrapert_acum']+dataframe['ter
ceram_clasificacion']+dataframe['segunda_muestra'])

Input [2]:

 dataframe['signosalud']=dataframe['contagio_corporal']-
dataframe['inf_respiratoria']

Input [3]:

dataframe.drop(['muestrapert_acum','terceram_clasificacion','segund
a_muestra'], axis=1).head(10)
```

Anexo 31, Capitulo 4.1

```
Input [1]: colormap = plt.cm.viridis

Input [2]: plt.figure(figsize=(12,12))

Input [3]:

plt.title('Pearson Correlation of Features', y=1.05, size=15)

Input [4]:

sb.heatmap

(reduced.astype(float).corr(),linewidths=0.1,vmax=1.0,  square=True,
cmap=colormap, linecolor='white', annot=True).
```

Anexo 32, Capitulo 4.1

```
Input [1]: X=dataframe.drop(['muerte'], axis=1)

Input [2]: y=dataframe['muerte']

Input [3]: best=SelectKBest(k=5)

Input [4]: X_new = best.fit_transform(X, y)
```

```
Input [5]: X_new.shape

Input [6]: selected = best.get_support(indices=True)

Input [7]: print(X.columns[selected])
```

Anexo 33, Capitulo 4.1

```
Input [1]: tures =X.columns[selected]

Input [2]: colormap = plt.cm.viridis

Input [3]: plt.figure(figsize=(12,12))

Input [4]: plt.title('Pearson Correlation of Features', y=1.05, size=15)

Input [5]:

sb.heatmap(

dataframe[used_features].astype(float).corr(),linewidths=0.1, vmax=1.0, square=True, cmap=colormap, linecolor='white', annot=True

)
```

Anexo 34. Capitulo 4.1

```
Input [1]:

sb.pairplot(dataframe,
hue='muerte',size=4,vars=used_features,kind='scatter').
```

Anexo 35, Capitulo 4.1

```
Input [1]:used_features2 =[

    "muestra_apertura",

    "signosalud"

  ]

Input [1]: X=dataframe[used_features2].values

Input [2]: y=dataframe["muerte"]

Input [3]: fig, ax = plt.subplots()

Input [4]: ax.scatter(X[:,0], X[:,1], c=y, s=50, cmap='RdBu')

Input [5]: ax.set_title('Naive Bayes Model', size=14)

Input [6]: xlim = (2000, 9000)

Input [7]: ylim = (100000, 700000)

Input [8]: xg = np.linspace(xlim[0], xlim[1], 40)

Input [9]: yg = np.linspace(ylim[0], ylim[1], 30)

Input [10]: xx, yy = np.meshgrid(xg, yg)

Input [11]: Xgrid = np.vstack([xx.ravel(), yy.ravel()]).T
```

```
Input [12]: for label, color in enumerate
 (['red', 'blue']):
    mask = (y == label)
    mu, std = X[mask].mean(0), X[mask].std(0)
    P = np.exp(-0.5 * (Xgrid - mu) ** 2 / std ** 2).prod(1)
    Pm = np.ma.masked_array(P, P < 0.03)
    ax.pcolorfast(xg, yg, Pm.reshape(xx.shape), alpha=0.5,
                  cmap=color.title() + 's')
    ax.contour(xx, yy, P.reshape(xx.shape),
               levels=[0.01, 0.1, 0.5, 0.9],
               colors=color, alpha=0.2)
    input [13]: ax.set(xlim=xlim, ylim=ylim)
```

7. BIBLIOGRAFÍA

Alba Centeno Franco. (2020). *DEEP LEARNING.* Sevilla: idUS.

Alonso del Saso, Javier. (2020). *Métodos de detección de anomalías y clustering en series temporales.* Cantabria : UniCan.

Amy Hurford, Maria M. Martignoni, J.C. (2022). Pandemic modelling for regions implementing an elimination strategy. *Journal of Theoretical Biology*.

Aprendizaje no supervisado. (2020). LIMA: CTI UNIC.

Archana Singh Bhadauria, Hom Nath Dhungana. (2022). Epidemic theory: Studying the effective and basic reproduction numbers, epidemic thresholds and techniques for the analysis of infectious .

Carlos Castillo-Chávez. (2020). *Epidemic Models.* Arizona: ResearchGate.

Centro de Coordinación de Alertas y Emergencia Sanitaria. (2021). EVALUACIÓN RÁPIDA DE RIESGO. *Ministerio de Sanidad*.

Dymaxion Labs. (2021). *Aprendizaje-Automatico.* New York: United Nation.

Edian F. Franco, Rommel J. Ramos. (2019). *Aprendizaje de máquina y aprendizaje profundo en biotecnología: aplicaciones, impactos y desafíos.* Brasil: Universidad Federal de Par.

Fernández Villafañez. (2022). *MÉTODOS DE REGRESIÓN Y CLASIFICACIÓN.* VALLADOLID: Universidad de Valladolid.

Government of Buenos Aires City. (2021). *Gobierno de la Nación* . Obtenido de BA Data: https://data.buenosaires.gob.ar/dataset/?groups=covid-19

Gregory L. Watson, Di Xiong, Lu Zhang, Joseph A. Zoller, John Shamshoian, Phillip Sundin, Teresa Bufford, Anne W. Rimoin. (2020). Fusing a Bayesian case velocity model with random forest for predicting COVID-19. *U.S*.

Jadhav SD, Channe HP. (2016). Efficient recommendation system using. *Int. Res. J. Eng.*, 3:2113.

Jairo Eduardo Márquez Díaz. (2020). Artificial intelligence and Big Data as solutions to COVID-19. *Revista de Bioética y Derecho*.

Jose A. Alonso Jim´enez. (2020). *Arboles de decisión.* Sevilla: Universidad de Sevilla .

Julien Arino, Stéphanie Porte. (2020). A simple model for COVID-19. *Journal Pre-proof*.

Kiesha Prem,Yang Liu, Timothy W Russell, Adam J Kucharski, Rosalind M Eggo, Nicholas Davie. (2020). The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan. *China:a modelling study*.

Li R, Pei S, Chen B, Song Y, Zhang T, Yang W. (2020). Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV2). *Science*.

Li R, Pei S, Chen B, Song Y, Zhang T, Yang W, Shaman J. . (2020). Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). *Science*.

Lin Q, Zhao S, Gao D, Lou Y, Yang S, Musa SS. (2020). A conceptual model for the coronavirus disease 2019 (COVID-19) outbreak in Wuhan, China with individual reaction and governmental action. *Int J Infect Dis*.

Mandal S, Bhatnagar T, Arinaminpathy N, Agarwal A, Chowdhury A, Murhekar M. (2020). Prudent public health intervention strategies to control the coronavirus disease 2019 transmission in India: A mathematical model-based approach. *Indian J Med Res*.

Mujica Rodríguez IE, Toribio Salazar LM, Cóndor Cámara DF. (2020). Artificial intelligence as a support for non-pharmacological interventions against COVID-19. *pubmed*.

Natasha K. Niemann and Raymond G. Blockmon. (2021). USING MACHINE LEARNING TO PREDICT THE INSIDER THREAT IN A NETWORK ENVIRONMENT. 107.

Niklas Kühl, Marc Goutier, Robin Hir, Gerhard Satzger. (2020). *Machine Learning in Artificial Intelligence.* New York: arXiv .

Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study. (2020). *Gunther Eysenbach*, 10.

Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N. (2020). The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. *Lancet Public Health.*

Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N. (2020). The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study . *Lancet Public Health.*

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang L. (2019). *Learning Deep Transformer Models for Machine Translation.* Shenyang: University of Macau.

Sunita Tiwari, S. K. (Octubre de 2020). Outbreak Trends of Coronavirus Disease-2019 in India: A Prediction. 38.

University of Cincinnati. (2020). *AN INTRODUCTION TO ARTIFICIAL INTELLIGENCE.* Cincinnati: University of Cincinnati .

University of Liège. (Junio de 2015). *U N D E R S TA N D I N G R A N D O M F O R E S T S*. Obtenido de https://arxiv.org/pdf/1407.7502.pdf

Walker PG, Whittaker C, Watson O, Baguelin M, Ainslie K, Bhatia S. (2020). The global impact of COVID-19 and strategies for mitigation and suppression. *Imperial College*.

Watson GL, Xiong D, Zhang L, Zoller JA, Shamshoian J, Sundin P, Bufford T, Rimoin AW, Suchard MA, Ramirez CM. (2021). Pandemic velocity: Forecasting COVID-19 in the US with a machine learning & Bayesian time series compartmental model. *PLoS Comput Biol*.