ANÁLISIS PREDICTIVO DE CASOS DE COVID-19 BASADO EN CONDICIONES CLIMÁTICAS

DANILO PLAZAS IRREÑO

UNIVERSIDAD INTERNACIONAL DE VALENCIA
FACULTAD DE MAESTRÍAS
MÁSTER EN BIG DATA Y DATA SCIENCE
BOGOTÁ D.C.
2022

ANÁLISIS PREDICTIVO DE CASOS DE COVID-19 BASADO EN CONDICIONES CLIMÁTICAS

DANILO PLAZAS IRREÑO
viudanilo0221p@gmail.com

Trabajo de grado para optar al titulo de:
Magister en Big Data y Data Science

DIRECTOR:
MSc. BENJAMÍN ARROQUIA CUADROS
Docente Universidad Internacional de Valencia

UNIVERSIDAD INTERNACIONAL DE VALENCIA
FACULTAD DE MAESTRÍAS
MÁSTER EN BIG DATA Y DATA SCIENCE
BOGOTÁ D.C.
2022

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

1. INTRODUCCIÓN

El COVID-19 es una enfermedad respiratoria causada por el virus SARS-CoV-2. Desde su aparición en Wuhan, China a finales de 2019, ha afectado a millones de personas en todo el mundo. Los gobiernos de todo el mundo han implementado diversas medidas para prevenir la propagación del virus y proteger la salud pública.

Algunas de las medidas más comunes fueron: cierre de fronteras, distanciamiento social (como el cierre de escuelas, lugares de trabajo y eventos públicos), uso de mascarillas, pruebas y rastreo de contactos (para identificar a las personas infectadas y rastrear a aquellos con los que han tenido contacto cercano), cierre de empresas y restricciones de actividades no esenciales (para reducir la cantidad de personas que se congregan en lugares públicos), campañas de concientización y educación pública. Todas estás medidas ayudan a mitigar la propagación del virus y aunque la transmisión del virus se produce principalmente por contacto cercano con personas infectadas, se ha investigado sobre la posible influencia de las condiciones climáticas en la propagación del virus.

En general, se cree que el clima cálido y húmedo puede reducir la propagación del virus, ya que el calor y la humedad pueden debilitar la capacidad del virus para sobrevivir en el aire y en las superficies. Sin embargo, los expertos señalan que no hay suficiente evidencia científica para afirmar que las altas temperaturas y la humedad reducen significativamente la transmisión del virus. Por otro lado, el invierno y el clima frío pueden aumentar la transmisión del virus, ya que las personas tienden a pasar más tiempo en espacios cerrados y con poca ventilación, lo que facilita la propagación del virus de persona a persona. [1][2]

En este proyecto se desarrollará un estudio y análisis sobre el impacto de las condiciones climáticas en la propagación del virus covid-19 en España y determinar si existe algún factor relacionado con la transmisión.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

- Identificar las características principales que afectan e influyen el aumento de personas contagiadas del virus de COVID-19 en España.

2.2. OBJETIVOS ESPECÍFICOS

- Extraer, transformar y obtener conocimiento de las diferentes fuentes de información o bases de datos de COVID-19 en España, centrándonos en características climáticas.
- Crear, comparar y contrastar los diferentes modelos de predicción y/o clustering sobre el COVID-19.
- Seleccionar el modelo que predice o explica lo más exacto posible la influencia de las características climáticas en los casos de virus de COVID-19.
- Precisar los efectos de otro tipo de características o variables en los modelos utilizados y evaluar sus desempeños.

3. MACHINE LEARNING Y ESTADO DEL ARTE

El Machine Learning es una técnica de Inteligencia Artificial que permite a los sistemas informáticos aprender de manera automática a partir de datos y experiencias previas, sin ser programados explícitamente para cada tarea. En lugar de seguir un conjunto fijo de instrucciones, los sistemas de Machine Learning pueden aprender a partir de datos, identificando patrones y tendencias, y utilizando esta información para realizar predicciones o tomar decisiones.

El aprendizaje automático se basa en la idea de que los sistemas informáticos pueden aprender de manera similar a como lo hacen los seres humanos, mediante la identificación de patrones y la adaptación a nuevas situaciones. En lugar de requerir que se programen todas las posibles situaciones y resultados, también nos permite que los sistemas aprendan a partir de datos históricos y experiencias previas, y así puedan tomar decisiones informadas y precisas en tiempo real. Programar computadoras para aprender de la experiencia eventualmente debería eliminar la necesidad de gran parte de este esfuerzo de programación detallado. Conforme a la definición de ML de Tom M. Mitchell: "Se dice que un programa de computadora aprende de la experiencia E con respeto a alguna clase de tareas T y medida de rendimiento P, si su rendimiento en tareas en T, medido por P, mejora con la experiencia E" [3]. Existen varios tipos de técnicas de Machine Learning, incluyendo el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. En el aprendizaje supervisado, el sistema aprende a partir de datos etiquetados previamente, mientras que, en el aprendizaje no supervisado, el sistema busca patrones y similitudes en los datos sin etiquetar. En el aprendizaje por refuerzo, el sistema aprende a partir de la retroalimentación del entorno.

Se aplica en una variedad de campos, como la detección de fraudes, la clasificación de imágenes, el análisis de sentimientos y la predicción de ventas, entre otros. A medida que los datos se vuelven cada vez más importantes y abundantes, el Machine Learning se está convirtiendo en una herramienta esencial para empresas e investigadores que buscan automatizar tareas y mejorar la toma de decisiones.

3.1. APRENDIZAJE SUPERVISADO

El aprendizaje supervisado es una técnica de ML que se basa en el uso de datos etiquetados previamente para entrenar un modelo de predicción o clasificación. En el aprendizaje supervisado, el modelo se entrena utilizando un conjunto de datos de entrenamiento que contiene ejemplos de entrada y salida esperada. El objetivo del modelo es aprender una función que pueda predecir la salida correcta para nuevas entradas nunca antes vistas.

Por ejemplo, en la clasificación de correos electrónicos como spam o no spam, el modelo se entrena con una gran cantidad de correos electrónicos etiquetados previamente como spam o no spam. Utilizando esta información, el modelo aprende a identificar patrones en los correos electrónicos que le permiten clasificarlos correctamente. Una de las principales ventajas del aprendizaje supervisado es que puede proporcionar predicciones precisas y confiables en una variedad de tareas. Sin embargo, el aprendizaje supervisado también tiene algunas limitaciones. En particular, requiere grandes cantidades de datos etiquetados, lo que puede ser costoso y laborioso en algunos casos. Además, el modelo

puede ser susceptible al sobreajuste si se entrena con demasiados datos, lo que significa que se adapta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos nunca antes vistos. Este a su vez se divide en problemas de regresión y clasificación.

3.1.1. REGRESIÓN

Los problemas de regresión en el aprendizaje supervisado son aquellos en los que se busca establecer una relación funcional entre una variable de entrada (también llamada variable independiente o predictor) y una variable de salida (también llamada variable dependiente o respuesta) que toma valores continuos en lugar de discretos. Es decir, se busca predecir un valor numérico continuo en lugar de una etiqueta o clase discreta. El objetivo de la regresión es encontrar una función que mejor se ajuste a los datos observados, de manera que pueda ser utilizada para predecir el valor de la variable de salida para nuevas observaciones de la variable de entrada.

Sin embargo, los problemas de regresión pueden presentar algunos desafíos, como:

- La presencia de valores atípicos o datos faltantes, que pueden afectar negativamente el ajuste del modelo y la precisión de las predicciones.
- La elección de la función de regresión adecuada y de los parámetros del modelo, que pueden depender de la distribución de los datos y del objetivo de la predicción.
- La evaluación de la calidad del modelo, que puede requerir el uso de medidas de error y de rendimiento específicas para problemas de regresión.

Para superar estos desafíos, se pueden utilizar técnicas de preprocesamiento de datos, selección de características, validación cruzada y ajuste de hiperparámetros, entre otras. Estos modelos pueden ser lineales o no lineales. Los modelos lineales, como la regresión lineal simple o múltiple, buscan establecer una relación lineal entre las variables de entrada y la variable de salida [4]. Por otro lado, los modelos no lineales, como la regresión polinómica, la regresión logística, regresión de árbol de decisión, random forest o redes neuronales, permiten establecer relaciones no lineales entre las variables.

3.1.2. CLASIFICACIÓN

Los modelos de clasificación binaria son utilizados cuando se desea predecir una variable de salida que puede tomar únicamente dos valores posibles, como “sí” o “no”, “verdadero” o “falso”, etc. Ejemplos de modelos de clasificación binaria incluyen la regresión logística y la máquina de vectores de soporte. Por otro lado, los modelos de clasificación no binaria son utilizados cuando se desea predecir una variable de salida que puede tomar más de dos valores posibles, como la clasificación de imágenes en diferentes categorías o la predicción de resultados deportivos. Ejemplos de modelos de clasificación no binaria incluyen los árboles de decisión, los bosques aleatorios y las redes neuronales.

Aunque los modelos de clasificación binaria y no binaria utilizan diferentes técnicas y algoritmos, el proceso general de construcción de modelos es el mismo. Se trata de identificar las variables de entrada más importantes, elegir un modelo adecuado, ajustar sus parámetros y evaluar su rendimiento utilizando medidas de evaluación adecuadas. binaria como no binaria. La elección del modelo adecuado dependerá del tipo de datos y del objetivo de la predicción. [5]

3.2. APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado es un tipo de aprendizaje automático en el que el algoritmo se entrena con datos no etiquetados, es decir, sin información previa sobre las categorías a las que pertenecen los datos. A diferencia del aprendizaje supervisado, donde los algoritmos aprenden a partir de datos etiquetados, en el aprendizaje no supervisado, los algoritmos buscan patrones y estructuras en los datos sin ninguna orientación sobre lo que se debe buscar. Se utiliza para descubrir patrones ocultos y estructuras en los datos, como grupos de datos similares, tendencias en los datos y relaciones entre variables. Algunos de los algoritmos de aprendizaje no supervisado más comunes incluyen la agrupación (clustering), la reducción de dimensionalidad y la asociación.

3.2.1. CLUSTERING

El clustering, también conocido como agrupamiento, es una técnica de aprendizaje no supervisado en la que se agrupan datos similares en grupos o clústeres. El objetivo del clustering es dividir un conjunto de datos en grupos, donde los objetos en cada clúster son similares entre sí y diferentes de los objetos en otros clústeres. Esto se logra mediante el uso de medidas de similitud o distancia para medir la distancia entre objetos. Existen varios algoritmos de clustering, cada uno con sus propias fortalezas y debilidades. El algoritmo K-means es uno de los algoritmos más populares y ampliamente utilizados en el clustering. Funciona dividiendo el conjunto de datos en K clústeres y asignando cada objeto al clúster más cercano. Luego, se recalcula el centroide de cada clúster y se repite el proceso hasta que se alcanza una solución óptima.

Otro algoritmo popular de clustering es el clustering jerárquico. Este algoritmo construye una jerarquía de clústeres mediante la combinación iterativa de clústeres en subgrupos más grandes. El resultado es un árbol jerárquico que representa la estructura de agrupamiento de los datos. El clustering se utiliza en muchas aplicaciones, como la segmentación de clientes, la clasificación de imágenes y la agrupación de documentos. Por ejemplo, en la segmentación de clientes, el clustering se puede utilizar para agrupar a los clientes en función de sus patrones de compra o preferencias. En la clasificación de imágenes, el clustering se puede utilizar para agrupar imágenes similares para su posterior análisis o clasificación.

En la agrupación de documentos, el clustering se puede utilizar para agrupar documentos similares en temas específicos.Sin embargo, es importante tener en cuenta que el clustering no siempre es la mejor opción para todos los conjuntos de datos y situaciones. Algunas limitaciones del clustering incluyen la necesidad de definir el número de clústeres de antemano, la sensibilidad a los valores atípicos y la dificultad de evaluar los resultados. Es importante seleccionar el algoritmo y los parámetros adecuados para obtener resultados precisos y significativos. [6]

3.2.2. REDUCCIÓN DE LA DIMENSIONALIDAD

La reducción de la dimensionalidad es una técnica que se utiliza para reducir el número de variables o características en un conjunto de datos, mientras se mantiene la mayor cantidad posible de información útil. La reducción de la dimensionalidad es importante ya que a menudo los conjuntos de datos pueden contener muchas variables o características, lo que puede hacer que los modelos sean complejos y difíciles de interpretar. Además, muchos

algoritmos de aprendizaje automático pueden tener dificultades para manejar conjuntos de datos con un gran número de variables. Existen varios algoritmos de reducción de dimensionalidad, pero uno de los más populares es el análisis de componentes principales (PCA). PCA es un método lineal que utiliza una transformación matemática para encontrar una nueva representación de los datos en un espacio de menor dimensión, mientras se mantiene la mayor cantidad posible de información. La idea detrás de PCA es encontrar una nueva combinación de las variables originales que explique la mayor cantidad posible de la varianza en los datos.

Otro algoritmo popular de reducción de la dimensionalidad es el t-distributed stochastic neighbor embedding (t-SNE). t-SNE es una técnica no lineal que se utiliza para visualizar datos en un espacio de menor dimensión. t-SNE es especialmente útil para visualizar datos de alta dimensión en dos o tres dimensiones. La técnica funciona encontrando una representación de los datos en un espacio de menor dimensión que mantiene la estructura de similitud de los datos originales. La reducción de la dimensionalidad se utiliza en muchas aplicaciones, como la visualización de datos, la detección de anomalías y la clasificación de datos. Por ejemplo, en la visualización de datos, la reducción de la dimensionalidad se puede utilizar para visualizar datos de alta dimensión en dos o tres dimensiones. En la detección de anomalías, la reducción de la dimensionalidad se puede utilizar para identificar patrones o grupos de datos anómalos. En la clasificación de datos, la reducción de la dimensionalidad se puede utilizar para mejorar la precisión de los modelos al reducir la complejidad del conjunto de datos.

Sin embargo, es importante tener en cuenta que la reducción de la dimensionalidad también puede tener algunas limitaciones. Por ejemplo, puede perder información importante durante el proceso de reducción de la dimensionalidad. Además, algunos algoritmos pueden ser sensibles a los valores atípicos o pueden ser difíciles de interpretar. [7]

3.2.3. ASOCIACIÓN

La asociación se utiliza para encontrar patrones interesantes en conjuntos de datos grandes y complejos, para descubrir relaciones entre variables en los datos y, a menudo, se utiliza en el análisis de transacciones, como el análisis de cestas de la compra, el análisis de clics de página web y el análisis de registros de transacciones financieras. La asociación se basa en el concepto de que los elementos en un conjunto de datos pueden estar relacionados de alguna manera. Por ejemplo, en el análisis de cestas de la compra, es posible que los clientes que compran leche también compren pan y huevos. La asociación se utiliza para encontrar patrones como este en los datos.

Un algoritmo popular de asociación es el algoritmo Apriori. El algoritmo Apriori se utiliza para encontrar conjuntos de elementos frecuentes en un conjunto de datos. Un conjunto de elementos es frecuente si aparece con frecuencia en el conjunto de datos. El algoritmo Apriori utiliza la propiedad de que cualquier subconjunto de un conjunto frecuente también es frecuente. El algoritmo comienza encontrando los conjuntos de elementos de un solo elemento más frecuentes, luego encuentra los conjuntos de elementos de dos elementos más frecuentes, y así sucesivamente. Otro algoritmo popular de asociación es el algoritmo FP-Growth. El algoritmo FP-Growth utiliza una estructura de árbol llamada árbol FP para encontrar conjuntos de elementos frecuentes. El árbol FP almacena los conjuntos de elementos frecuentes y sus frecuencias de manera eficiente, lo que permite que el

algoritmo sea mucho más rápido que el algoritmo Apriori para conjuntos de datos grandes.

La asociación se utiliza en muchas aplicaciones, como la recomendación de productos, la segmentación de clientes y la detección de fraudes. Por ejemplo, en la recomendación de productos, la asociación se puede utilizar para recomendar productos relacionados con los que el cliente ya ha comprado. En la segmentación de clientes, la asociación se puede utilizar para identificar grupos de clientes que tienen patrones de compra similares. En la detección de fraudes, la asociación se puede utilizar para identificar patrones de transacciones sospechosos, como un cliente que compra varios artículos caros en un corto período de tiempo.

3.3. APRENDIZAJE POR REFUERZO

El aprendizaje por refuerzo es una técnica de aprendizaje automático que se basa en el concepto de que un agente debe aprender a tomar decisiones óptimas en un entorno determinado para maximizar una recompensa acumulativa. En el aprendizaje por refuerzo, el agente recibe información del entorno en forma de recompensas y castigos y debe aprender a seleccionar acciones que maximicen la recompensa a largo plazo. El agente toma una acción en un estado particular y el entorno responde con una recompensa. El objetivo del agente es aprender una política que le permita maximizar la recompensa acumulativa en el largo plazo. La política es una función que mapea estados a acciones y puede ser aprendida utilizando técnicas de aprendizaje por refuerzo.

Un algoritmo popular de aprendizaje por refuerzo es el algoritmo Q-Learning. En Q-Learning, el agente aprende una función Q que le permite evaluar la calidad de una acción en un estado particular. La función Q se puede utilizar para seleccionar la mejor acción en un estado determinado y el algoritmo Q-Learning utiliza una técnica llamada exploración, explotación para equilibrar el aprendizaje de nuevas acciones y la selección de acciones que se sabe que son buenas. El aprendizaje por refuerzo se utiliza en muchas aplicaciones, como los juegos, la robótica y la optimización de procesos. Por ejemplo, en los juegos, el aprendizaje por refuerzo se puede utilizar para crear agentes que aprendan a jugar juegos complejos, como el ajedrez y el Go, en la robótica, el aprendizaje por refuerzo se puede utilizar para crear robots que aprendan a realizar tareas complejas, como caminar y manipular objetos y en la optimización de procesos, el aprendizaje por refuerzo se puede utilizar para crear sistemas que aprendan a optimizar procesos, como la producción de energía y la gestión de inventarios.

3.4. REDES NEURONALES

Las redes neuronales son un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano, estas redes están compuestas por unidades de procesamiento llamadas neuronas artificiales, que se organizan en capas y se interconectan mediante conexiones ponderadas. La información se propaga a través de la red de neuronas a través de una función de activación no lineal, lo que permite que la red realice una amplia variedad de tareas de aprendizaje.
El proceso de entrenamiento de una red neuronal implica ajustar los valores de los pesos de las conexiones para que la salida de la red se acerque a la salida deseada. Esto se logra

mediante el uso de algoritmos de aprendizaje supervisado o no supervisado. En el aprendizaje supervisado, se proporciona a la red un conjunto de datos etiquetados de entrada y salida esperada, mientras que, en el aprendizaje no supervisado, la red debe descubrir patrones en los datos de entrada por sí misma. Las redes neuronales han demostrado ser muy efectivas en una amplia variedad de tareas de aprendizaje automático, como la clasificación de imágenes, el reconocimiento de voz, la traducción automática y la generación de texto y música, entre otras. Además, las redes neuronales profundas, que tienen muchas capas ocultas, han llevado a grandes avances en áreas como el procesamiento del lenguaje natural y la visión por computadora.

Sin embargo, el entrenamiento de redes neuronales puede ser un proceso muy intensivo en términos de recursos computacionales y de tiempo, y la interpretación de los resultados obtenidos puede ser difícil debido a la naturaleza no lineal y altamente distribuida de las redes neuronales. Además, las redes neuronales pueden ser propensas a sobreajustarse a los datos de entrenamiento, lo que puede llevar a un rendimiento deficiente en datos nuevos.

3.4.1. CONVOLUCIONALES

Las redes neuronales convolucionales son un tipo de red neuronal que ha sido especialmente diseñada para procesar datos de tipo imagen y otros datos de alto dimensionalidad. Las CNN (Convolutional Neural Network) utilizan una técnica de procesamiento conocida como convolución, que implica desplazar un filtro sobre una imagen y realizar una operación de multiplicación punto a punto entre el filtro y la sección de la imagen correspondiente.

Las CNN son especialmente útiles para tareas como la clasificación de imágenes y la detección de objetos, ya que son capaces de extraer características de las imágenes y otros datos de alto dimensionalidad con gran precisión, estas características se utilizan luego para realizar la clasificación o detección de objetos, según sea el caso. Las CNN son particularmente efectivas en la detección de patrones en datos de tipo imagen, ya que los filtros convolucionales permiten capturar características locales de la imagen, como bordes, texturas y patrones repetitivos, de manera eficiente. Además, las CNN pueden aprender automáticamente las características relevantes de los datos de entrenamiento, lo que las hace muy efectivas para tareas de clasificación y detección de objetos en las que las características relevantes no son conocidas de antemano. Aunque las CNN son particularmente útiles para tareas de procesamiento de imagen, también se han utilizado con éxito en tareas como el procesamiento del lenguaje natural, la clasificación de secuencias de tiempo y la detección de anomalías. Además, las redes neuronales convolucionales profundas, que contienen varias capas de convolución, han llevado a grandes avances en áreas como el procesamiento de imágenes médicas y la visión por computadora. [8]

3.4.2. RECURRENTES

Las redes neuronales recurrentes (RNN, recurrent neural network) son un tipo de red neuronal que se utiliza para procesar datos secuenciales, como el lenguaje natural y las series de tiempo. A diferencia de las redes neuronales convolucionales, que procesan los datos de manera independiente en cada posición, las RNN tienen memoria y son capaces de procesar secuencias de datos de longitud variable. En una red neuronal recurrente, cada neurona tiene una conexión consigo misma, lo que permite que la información fluya a través de la red en bucles. Esta arquitectura de bucle permite que la red neuronal recuerde información

de los pasos anteriores y la utilice para procesar la entrada actual. La principal ventaja de las RNN es su capacidad para modelar la dependencia temporal de los datos. Esto significa que la red puede aprender patrones en secuencias de datos, como la estructura sintáctica del lenguaje natural o los patrones de fluctuación en una serie de tiempo. Además, las RNN son capaces de procesar secuencias de datos de longitud variable, lo que las hace muy útiles para tareas como la traducción automática, donde la longitud de la entrada y la salida puede variar.

Una variante de las RNN son las redes neuronales LSTM (Long Short-Term Memory), que fueron diseñadas para evitar el problema del desvanecimiento del gradiente, que puede ocurrir cuando se entrena una red neuronal recurrente profunda. Las redes LSTM utilizan un mecanismo de compuertas para controlar el flujo de información a través de la red y evitar que la señal degradada afecte al entrenamiento de la red. Las redes neuronales recurrentes y las redes LSTM han tenido un gran éxito en una amplia variedad de aplicaciones, incluyendo la generación de texto, la traducción automática, el reconocimiento de voz y la predicción de series de tiempo. Sin embargo, las redes neuronales recurrentes también tienen algunas limitaciones, como la dificultad para manejar dependencias a largo plazo y el costo computacional elevado de entrenar redes profundas. [9]

3.4.3. DE ATENCIÓN

Las redes neuronales de atención (attention-based neural networks) son un tipo de red neuronal que se utiliza para procesar datos secuenciales y modelar la dependencia temporal de los datos, al igual que las redes neuronales recurrentes. Sin embargo, a diferencia de las redes neuronales recurrentes, las redes neuronales de atención no procesan todos los datos secuenciales de manera uniforme, sino que prestan atención a partes específicas de la secuencia en cada paso de procesamiento. En una red neuronal de atención, cada paso de procesamiento se divide en dos partes: la codificación y la decodificación. Durante la codificación, la red procesa la entrada secuencial y la transforma en una serie de vectores de características. Durante la decodificación, la red genera la salida secuencial a partir de los vectores de características generados durante la codificación.
La atención se utiliza para determinar qué vectores de características se deben utilizar en cada paso de decodificación. En lugar de procesar toda la secuencia de entrada en cada paso de decodificación, la red presta atención a las partes más relevantes de la entrada en cada paso, utilizando una función de atención para asignar pesos a cada vector de características de la secuencia de entrada.

La principal ventaja de las redes neuronales de atención es su capacidad para enfocarse en las partes más importantes de la entrada secuencial en cada paso de procesamiento, lo que las hace muy útiles para tareas como la traducción automática, donde la atención se puede utilizar para identificar las partes más relevantes de la oración en la que se está trabajando en cada paso del proceso de traducción. Una variante de las redes neuronales de atención son las redes neuronales de atención múltiple (multi-head attention networks), que utilizan múltiples funciones de atención para procesar diferentes aspectos de la entrada secuencial de manera simultánea.

Las redes neuronales de atención han tenido un gran éxito en una amplia variedad de aplicaciones, incluyendo la traducción automática, el procesamiento del lenguaje natural y la generación de texto. Sin embargo, al igual que todas las redes neuronales, también

tienen algunas limitaciones que deben ser consideradas, como la necesidad de grandes cantidades de datos de entrenamiento y el costo computacional elevado de entrenar redes profundas.

3.5. ESTADO DEL ARTE

El aprendizaje automático ha experimentado un gran avance en las últimas décadas gracias al aumento de la cantidad y calidad de datos disponibles y a la mejora de los algoritmos uno de los más importante ha sido el desarrollo de algoritmos de aprendizaje semi-supervisado y de transferencia de aprendizaje, que permiten entrenar modelos con conjuntos de datos más pequeños y reducir el tiempo y los costos de entrenamiento. Estos enfoques son especialmente útiles en áreas donde la recopilación de datos es costosa o difícil, como en la medicina o la astronomía.

A su vez el aprendizaje federado es una técnica de aprendizaje supervisado que ha ganado mucha atención en los últimos años debido a su capacidad para entrenar modelos de manera distribuida y colaborativa sin la necesidad de compartir los datos subyacentes. Esto lo hace especialmente útil en situaciones donde la privacidad y la seguridad son una preocupación importante, como en el análisis de datos médicos o financieros. Un ejemplo es el proyecto NVIDIA FLARE, un kit de desarrollo de software que ayuda a las partes distribuidas a colaborar para desarrollar modelos de IA más generalizables *"El código abierto de NVIDIA FLARE para acelerar la investigación del aprendizaje federado es especialmente importante en el área de la salud, donde el acceso a conjuntos de datos multiinstitucionales es crucial, pero las preocupaciones sobre la privacidad del paciente pueden limitar la capacidad de compartir datos"*, Dr. Jayashree Kalapathy. [10]
Para nuestro proyecto el aprendizaje federado no es el adecuado ya que no contamos con información sensible y la seguridad de esta información no es relevante, ya que son datos públicos suministrados por el gobierno y páginas oficiales.

Sin duda el área en donde más se ha experimentado grandes avances ha sido las redes neuronales especialmente en áreas como las redes neuronales profundas, las arquitecturas de redes neuronales innovadoras (diversas arquitecturas que abordan problemas específicos), el aprendizaje por transferencia el cual implica reutilizar las capas ocultas de una red pre-entrenada en una tarea y adaptarla para una tarea nueva. Así como nuevas técnicas para mejorar los modelos como lo es la técnica de regularización que se utilizan para evitar el sobreajuste o el sobreentrenamiento de las redes neuronales, esto implica agregar términos de penalización a la función de costo de la red para evitar que los pesos de la red adquieran valores extremos. Algunos proyectos recientes en estas áreas son:

- **AlphaGo:** Es un programa de ordenador desarrollado por DeepMind que utiliza una combinación de redes neuronales y algoritmos de búsqueda para jugar al juego de mesa chino Go. En 2016, AlphaGo se convirtió en el primer programa de ordenador en derrotar a un campeón humano de Go, lo que fue considerado un hito importante en la inteligencia artificial. [11]
- **DALL-E:** Es un modelo de red neuronal desarrollado por OpenAI que genera imágenes a partir de descripciones de texto. El modelo utiliza una combinación de redes neuronales convolucionales y de atención para generar imágenes realistas y detalladas a partir de descripciones de texto. [12]

Figura 1: Astronauta sosteniendo una flor - DALL-E.
Fuente: [Dall-E 2: Why the AI image generator is a revolutionary invention]

- **StyleGAN2:** Es un modelo de red neuronal que se utiliza para generar imágenes fotorrealistas de alta calidad. El modelo utiliza una técnica llamada red neuronal generativa adversarial"para generar imágenes que son indistinguibles de las fotos reales. [13]

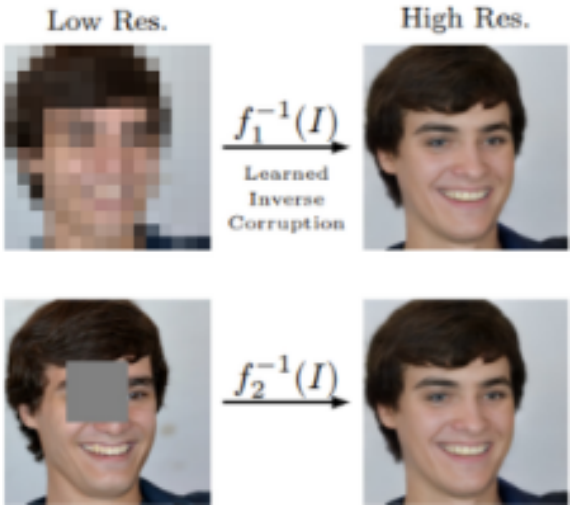

Figura 2: Reconstrucción de imagen - StyleGAN2.
Fuente: [MIT CSAIL Uses Deep Generative Model StyleGAN2 to Deliver SOTA Image Reconstruction Results]

- **BERT (Bidirectional Encoder Representations from Transformers):** Es un modelo de lenguaje natural basado en redes neuronales de atención que ha demostrado una gran precisión en una variedad de tareas de procesamiento de lenguaje natural, como la clasificación de texto y la respuesta a preguntas.

Estos avances muestran cómo las redes neuronales están transformando la forma en que las computadoras pueden aprender y procesar información, abriendo nuevas posibilidades en áreas como la generación de lenguaje natural, la visión por computadora y la toma de

decisiones inteligente. Para nuestro caso puntual no se utilizarán estas técnicas recientes debido al alcance del proyecto, en donde se evaluará modelos tradicionales y la comparación entre estos.

4. DESARROLLO DEL PROYECTO Y RESULTADOS

Para el desarrollo del proyecto se utilizó la metodología KDD (Knowledge Discovery in Databases) en los distintos orígenes de datos, el objetivo de es crear un set de datos refinado a partir de todos los orígenes y a partir de este extraer conocimiento a través de los modelos creados para darle una solución al planteamiento del problema. Se realizará un análisis sobre los resultados obtenidos y las posibles mejoras en trabajos futuros.

Todo el desarrollo del proyecto se realizará en lenguaje Python, a partir de notebooks de Jupyter, en estos se encontrará el desarrollo de todas las etapas del proceso de KDD y el porque se toman ciertas decisiones. Basado en los resultados de los notebooks se crean archivos .py con algunas etapas para realizar una CI/CD. Las fuentes de datos, los notebooks con los análisis de cada etapa y los archivos .py para la automatización de todo el flujo se encuentra en el repositorio personal para este TFM, a su vez se encontrará el código fuente de LaTeX el cual se desarrolló el presente documento: REPOSITORIO TFM

4.1. METODOLOGÍA

KDD (Knowledge Discovery in Databases), o Descubrimiento de Conocimiento en Bases de Datos, es un proceso integral que implica la identificación, extracción y transformación de patrones y conocimientos valiosos a partir de grandes conjuntos de datos. Es una disciplina que combina el uso de técnicas de minería de datos, estadísticas, aprendizaje automático y bases de datos para descubrir información útil y conocimientos ocultos en datos no estructurados o estructurados.

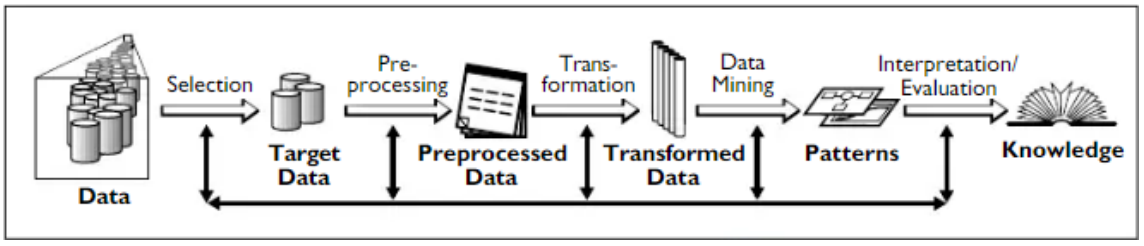

Figura 3: Descripción de los pasos que constituyen el proceso KDD
Fuente: [The KDD Process for Extracting Useful Knowledge from Volumes of Data]

El proceso de KDD consta de varias etapas, que incluyen:

- **Selección de datos:** Consiste en la identificación y recopilación de los datos relevantes para el análisis. Esto puede involucrar la obtención de datos de diversas fuentes, la limpieza y preprocesamiento de los datos para asegurar su calidad y consistencia.
- **Preprocesamiento de datos:** Implica la transformación y limpieza de los datos para prepararlos para el análisis. Esto puede incluir la eliminación de datos duplicados o inconsistentes, la normalización de los datos, la imputación de valores faltantes y la selección de características relevantes.
- **Transformación de datos:** Involucra la conversión de los datos preprocesados en formatos adecuados para el análisis. Esto puede incluir la transformación de datos

categóricos en datos numéricos, la discretización de datos continuos, la reducción de dimensionalidad, entre otros.

- **Minería de datos:** Es la etapa central de KDD, donde se aplican técnicas y algoritmos de minería de datos para descubrir patrones y conocimientos en los datos. Esto puede incluir técnicas de clasificación, regresión, agrupamiento, asociación, entre otras.
- **Interpretación y Evaluación de resultados:** Implica la interpretación y comunicación de los resultados obtenidos a través de técnicas de visualización y presentación de datos. Así como la utilización de métricas de evaluación y validación para medir la precisión, el rendimiento y la utilidad de los resultados obtenidos. Esto puede ayudar a comprender y utilizar los patrones y conocimientos descubiertos para tomar decisiones informadas y mejorar la toma de decisiones en diversas áreas de aplicación.

4.2. PLANTEAMIENTO DEL PROBLEMA

El COVID-19 es una enfermedad infecciosa altamente contagiosa que ha afectado a millones de personas en todo el mundo y ha causado la muerte de cientos de miles de personas. Si bien se sabe que la propagación del virus se produce principalmente por contacto cercano con personas infectadas, también hay evidencia emergente que sugiere que las condiciones climáticas como la temperatura, la humedad y la luz solar, pueden influir en la propagación del virus. Un modelo de machine learning puede ser una herramienta útil para evaluar la relación entre las condiciones climáticas y la propagación del COVID-19. El objetivo de este planteamiento del problema es desarrollar un modelo de machine learning que pueda predecir la propagación del virus en función de factores climáticos como la temperatura, la humedad y la luz solar.
El modelo podría utilizar datos históricos sobre la propagación del virus y las condiciones climáticas para predecir la propagación futura del virus en diferentes condiciones climáticas o más específicamente dependiendo del modelo utilizado a groso modo se podrían utilizar de la siguiente forma:

- **Modelo de regresión:** Utilizando datos históricos de propagación del virus y condiciones climáticas para predecir la propagación futura del virus. El modelo puede incluir variables relacionadas con la propagación del virus, como el número de casos confirmados y la tasa de reproducción.
- **Redes neuronales:** Se podría utilizar para analizar grandes conjuntos de datos de propagación del virus y condiciones climáticas. El modelo puede aprender patrones y relaciones entre las variables para predecir la propagación futura del virus en diferentes condiciones climáticas.
- **Análisis de series de tiempo:** Si contamos con datos históricos para identificar patrones y tendencias en la propagación del virus y las condiciones climáticas. El modelo puede predecir la propagación futura del virus en función de los patrones identificados.
- **Modelos de aprendizaje profundo:** Si tenemos datos de satélite y mapas climáticos para predecir la propagación del virus. Estos modelos pueden integrar datos

climáticos con información sobre la densidad de población y la movilidad humana para predecir cómo se propagará el virus en diferentes áreas geográficas.

Lo anterior es solo un bosquejo de un posible uso de cada tipo de modelo, conforme vayamos avanzando en nuestra investigación determinaremos cuál es el modelo que más se ajusta a nuestro requerimiento o que obtenga mejores resultados basado en sus métricas, el resultado de esto podría ayudar a informar la toma de decisiones sobre políticas de salud pública y permitir a las autoridades sanitarias tomar medidas preventivas antes de que se produzca un aumento en los casos de COVID-19. Al responder a esta pregunta, se pueden desarrollar mejores estrategias de prevención y mitigación para el COVID-19, y se pueden aplicar los hallazgos a futuras pandemias y enfermedades infecciosas.

4.3. DESARROLLO DEL PROYECTO

El proyecto se desarrollará siguiendo cada una de las etapas del KDD, ya que proporciona una estructura sistemática para la extracción de conocimiento a partir de los datos. Al seguir cada una de las etapas del KDD, se puede asegurar que el proceso es riguroso y que se obtienen resultados precisos y relevantes para el proyecto.

4.3.1. SELECCIÓN DE DATOS

Para la ejecución del proyecto se utilizaron distintas fuentes de datos las cuales se detallan a continuación:

- **Datos casos COVID-19 por provincias**

  Los resultados que se presentan se obtienen a partir de la declaración de los casos de COVID-19 a la Red Nacional de Vigilancia Epidemiológica (RENAVE) a través de la plataforma informática vía Web SiViES (Sistema de Vigilancia de España) que gestiona el Centro Nacional de Epidemiología (CNE). Esta información procede de la encuesta epidemiológica de caso que cada Comunidad Autónoma cumplimenta ante la identificación de un caso de COVID-19. Datos oficiales disponibles en el sitio web: https://cnecovid.isciii.es/covid19/#documentaci%C3%B3n-y-datos

  Se utilizan los siguientes sets de datos:

  casos_hosp_uci_def_sexo_edad_provres.csv: Datos desde el inicio de la pandemia, para todas las edades, hasta el 28 de marzo de 2022.
  casos_hosp_uci_def_sexo_edad_provres_60_mas.csv: Datos desde el inicio de la pandemia, para la población de 60 o más años.

  Para ambos casos cuentan con las mismas columnas, así como su descripción. Esta descripción se detalla en la tabla **4** del anexo I.

- **Datos códigos provincias**

  Archivo de elaboración propia que contiene el nombre de la provincia y el correspondiente código ISO de la provincia. Esta fuente de datos es necesaria para el cruce de información o unión de los sets de datos, ya que algunas fuentes tienen el nombre de la provincia y otros tiene su correspondiente código ISO de cada provincia.

Tabla 1: Código ISO y nombre de provincia.

Variable	Descripción
PROVINCIA_ISO	Código ISO correspondiente a la provincia
PROVINCIA	Nombre de la provincia

Fuente: Elaboración Propia

- **Datos climatológicos por provincia**

  Los datos climatológicos son diarios por provincia, estos datos oficiales fueron extraídos desde el portal datos abiertos de AEMET, que permite la difusión y la reutilización de la información meteorológica y climatológica de la Agencia, en el sentido indicado en la Ley 18/2015, de 9 de julio, por la que se modifica la Ley 37/2007, de 16 de noviembre, sobre reutilización de la información del sector público. Para poder acceder a AEMET OpenData, es necesario solicitar una API Key (https://opendata.aemet.es/centrodedescargas/altaUsuario?). Una API Key es un identificador, mediante el cual se contabilizan e imputan los accesos que un usuario realiza al API. Mediante el API KEY solicitado se obtiene la data en el siguiente sitio web: https://opendata.aemet.es/centrodedescargas/productosAEMET.

Tabla 2: Variables climatológicas de provincia.

Variable	Descripción
fecha	fecha del dia (AAAA-MM-DD)
indicativo	indicativo climatológico
nombre	nombre (ubicación) de la estación
provincia	provincia de la estación
altitud	altitud de la estación en m sobre el nivel del mar
tmed	Temperatura media diaria
prec	Precipitación diaria de 07 a 07
tmin	Temperatura Mínima del día
horatmin	Hora y minuto de la temperatura mínima
tmax	Temperatura Máxima del día
horatmax	Hora y minuto de la temperatura máxima
dir	Dirección de la racha máxima
velmedia	Velocidad media del viento
racha	Racha máxima del viento
horaracha	Hora y minuto de la racha máxima
sol	Insolación
presmax	Presión máxima al nivel de referencia de la estación
horapresmax	Hora de la presión máxima (redondeada a la hora entera más próxima)
presmin	Presión mínima al nivel de referencia de la estación
horapresmin	Hora de la presión mínima (redondeada a la hora entera más próxima)

Fuente: Elaboración Propia

- **Datos población por provincia**

  Los datos anuales demográficos por provincia fueron extraídos desde el instituto nacional de estadística de España, mediante el sitio web: https://www.ine.es/jaxi/Datos.htm?path=/t20/e245/p08/l0/&file=03003.px#!tabs-tabla, esta fuente de datos es necesaria para realizar una normalización de los casos de covid-19 por provincia o lo que llamamos la tasa de incidencia.

Tabla 3: Población anual desde 1998 por provincia.

Variable	Descripción
Provincias	Nombre de la provincia
Sexo	H (hombre), M (mujer), Ambos sexos
Edad (año a año)	Rango de edad - total edades (contempla todas)
Españoles/Extranjeros	Origen de la persona - total (comtempla ambos)
Año	Año de recopilación de la información
Total	Cantidad de personas

Fuente: Elaboración Propia

4.3.2. PREPROCESAMIENTO DE DATOS

Para cada una de las fuentes de datos se realiza el procesamiento de su información, así como el dataset total que esta compuesto de la unión de estás fuentes mencionados en el apartado anterior.

- **Datos casos COVID-19 por provincias**

  Ambos sets de datos de covid-19 por provincia cuenta con 8 columnas con los mismos nombres que fueron descritas en el apartado anterior. Se realiza la unión de ambos set de datos (*casos_hosp_uci_def_sexo_edad_provres.csv* y *casos_hosp_uci_def_sexo_edad_provres_60_mas.csv*), ya que la única diferencia entre estos dos es el rango de edad que presenta en una de sus variables. Se realiza un perfilamiento de los datos mediante la librería de pandas_profiling el cual a partir de un dataframe de pandas genera el perfilamiento de la data en el siguiente archivo df_covid_prof.html.
  Realizamos un análisis de la cantidad de nulos en todas sus variables mediante un heatmap.

Figura 4: Headmap variables dataset covid-19.

Fuente: [Elaboración propia]

Del gráfico anterior observamos que no se evidencian nulos, pero es una medida cualitativa ya que por la cantidad de datos pueda que haya muy pocos y no se evidencien, por lo que se procede a hacer una lista de porcentaje de valores nulos:

```
provincia_iso - 0.0%
sexo - 0.0%
grupo_edad - 0.0%
fecha - 0.0%
num_casos - 0.0%
num_hosp - 0.0%
num_uci - 0.0%
num_def - 0.0%
```

Figura 5: Porcentaje de nulos dataset covid-19.

Fuente: [Elaboración propia]

Los estadísticos obtenidos para el dataset se representan en la siguiente imagen:

	num_casos	num_hosp	num_uci	num_def
count	1461210.00	1461210.00	1461210.00	1461210.00
mean	8.74	0.43	0.04	0.08
std	48.38	2.49	0.30	0.77
min	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00
50%	0.00	0.00	0.00	0.00
75%	4.00	0.00	0.00	0.00
max	3749.00	271.00	35.00	100.00

Figura 6: Estadísticos dataset covid-19.

Fuente: [Elaboración propia]

La media de *num_casos* es de 8.74 y la mediana es 0 ya que en muchas fechas diarias no se reportaron casos, la media de *num_hosp* es de 0.43 y la mediana 0, la media de *num_uci* es 0.04 y la mediana 0, por último, la media de *num_def* es de 0.08 y la mediana 0. Las variables parecen ser asimétricas a la derecha dado que su media es mayor a su mediana.

Las desviaciones típicas son bajas, la mayoría de las observaciones se encuentran dispersas a no más de una desviación estándar a cada lado. La imagen anterior también muestra los valores máximos y mínimos que toman cada variable objeto de estudio además de los cuartiles calculados. Los datos menores al cuartil 1 (Q1) representan el 25 % de los datos, los que están por debajo del cuartil 2 (Q2) representan el 50 % de los datos y los que están por debajo del cuartil 3 (Q3) representan el 75 % de los datos.

Tras realizar un análisis de correlación entre sus variables de estudio del conjunto de datos se evidencia en la siguiente gráfica que las variables *num_hosp*, *num_uci* y *num_def* tiene una correlación positiva y fuerte entre ellas. Es de esperarse este

resultado ya por lo general son consecuentes una con la otra, es decir, por ejemplo, si hubo una difusión por covid-19 es muy probable que haya estado en uci y hospitalización previamente.

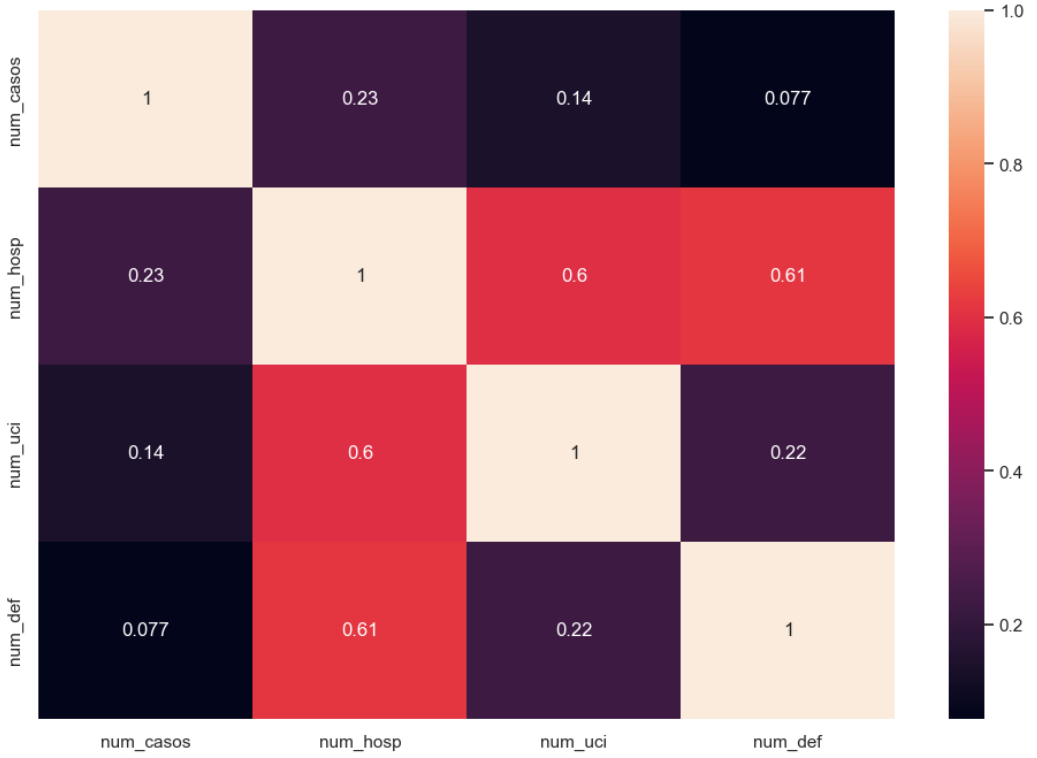

Figura 7: Correlación variables dataset covid-19.

Fuente: [Elaboración propia]

Se realizo un análisis de outliers para cada una de las variables, así como la distribución de los valores de las variables categóricas para identificar valores atípicos, para ningún caso se haya evidencia de alguno. Para evitar que existan palabras distintas y que simbolicen el mismo significado solo por el hecho de estar en minúscula o mayúsculas, para todas las variables tipo string las pasaremos a mayúscula, ya que por defecto todas viene así, también eliminaremos los espacios al principio y al final.

- **Datos códigos provincias**

  Esta fuente de datos fue de elaboración propia por lo cual se aseguro que no hubiera valores duplicado, valores nulos, valores atípicos, el nombre de las columnas está en mayúscula, los valores están en mayúscula sin ningún tipo de espacio por ende no necesita ningún tipo de transformación. Esta fuente cuenta con dos columnas y con 52 registros, el cual servirá para unir los datasets.

- **Datos climatológicos por provincia**

  Se realiza la concatenación de cada uno de los archivos con la información climatológica por provincia para poder obtener un dataset completo y su respectivo análisis, comenzando por la cantidad de nulos en todas sus variables mediante un heatmap.

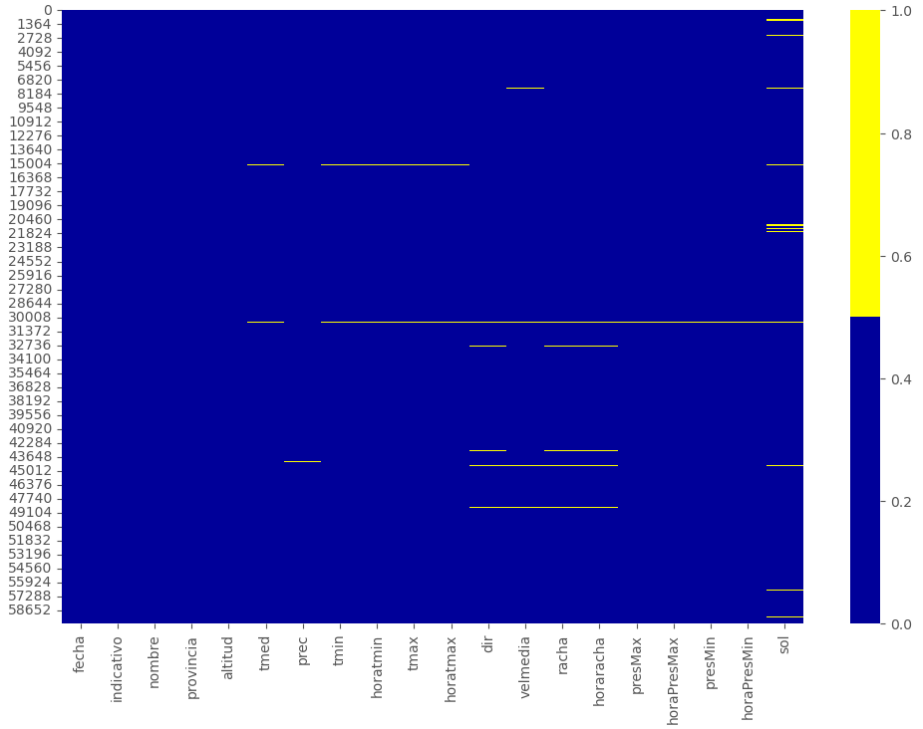

Figura 8: Headmap variables dataset clima.

Fuente: [Elaboración propia]

El gráfico anterior muestra los patrones de datos que faltan de todas las columnas, el eje horizontal muestra el nombre del atributo de entrada; el eje vertical muestra el número de observaciones/filas; el color amarillo representa los datos que faltan, mientras que el color azul, en caso contrario. Detallamos que todas las características tienen muy pocos valores perdidos o inclusive no tienen, para tener un valor exacto hacemos una lista de porcentaje de valores nulos:

```
fecha - 0.0%
indicativo - 0.0%
nombre - 0.0%
provincia - 0.0%
altitud - 0.0%
tmed - 0.3233711%
prec - 0.3133699%
tmin - 0.3200373%
horatmin - 0.3467071%
tmax - 0.3033687%
horatmax - 0.3250379%
dir - 1.0467888%
velmedia - 0.7134166%
racha - 1.0467888%
horaracha - 1.0501225%
presMax - 0.4100478%
horaPresMax - 0.4233827%
presMin - 0.4100478%
horaPresMin - 0.4300502%
sol - 2.3619422%
```

Figura 9: Porcentaje de nulos dataset clima.

Fuente: [Elaboración propia]

La imputación de los valores faltantes de las variables se estableció mediante fillna aplicado al dataframe por el método *ffill* como primera opción, como segunda opción el método *bfill*. El primer método usa la anterior observación válida para llenar el valor faltante y el segundo método usa la siguiente observación válida para llenar el valor faltante. La elección de estos métodos es debido a que son variables climatológicas, en donde el clima entre estaciones climáticas y periodos de tiempo corto son similares, la granularidad de este set de datos es diaria por lo que la imputación de valores faltantes se hará sobre el valor del día anterior o el más próximo. Se elimina las variables indicativo y nombre, ya que estás hacen referencia netamente a la información de la estación meteorológica en donde se obtuvieron los datos, esta información no aporta a nuestro estudio.

Tras realizar un análisis de correlación entre sus variables de estudio del conjunto de datos se evidencia en la siguiente gráfica que las variables *PREX_MAX* y *PREX_MIN* tiene una correlación positiva y fuerte entre ellas; así como las variables *TEMP_MIN*, *TEMP_MAX* y *TEMP_MED* tienen una correlación positiva alta.

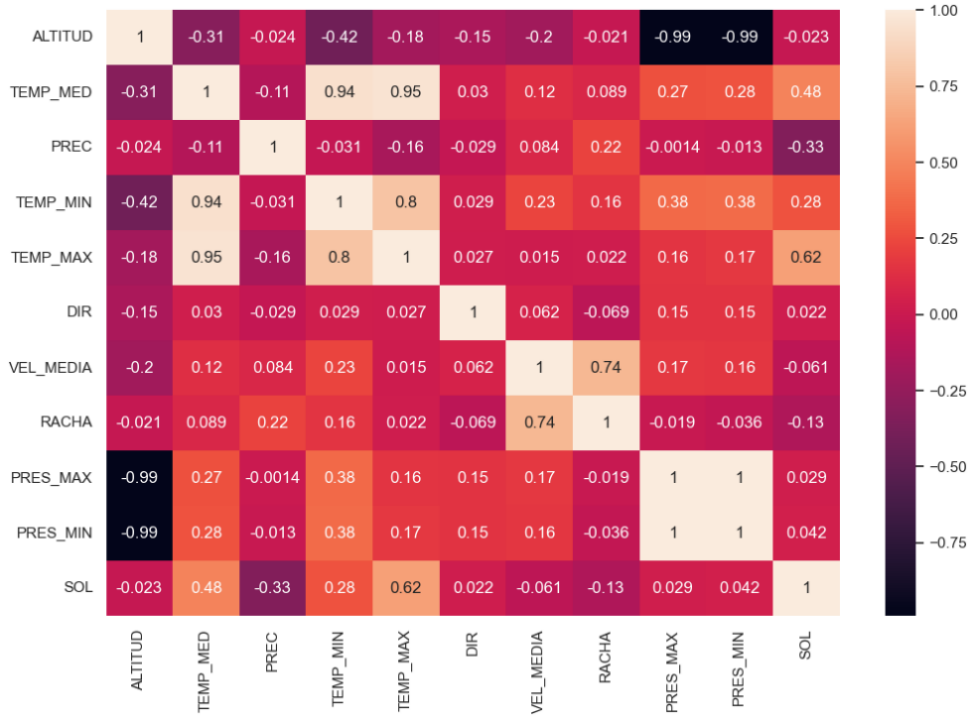

Figura 10: Correlación variables dataset clima.

Fuente: [Elaboración propia]

Se realizo un análisis de outliers por medio de boxplots para cada una de las variables, así como la distribución de los valores de las variables categóricas para identificar valores atípicos, se encontraron valores alejados de la media y poco comunes, pero son valores climatológicos posibles por esta razón para ningún caso se haya evidencia de outliers. Para evitar que existan palabras distintas y que simbolicen el mismo significado solo por el hecho de estar en minúscula o mayúsculas, para todas las variables tipo string las pasaremos a mayúscula, ya que por defecto todas viene así, también eliminaremos los espacios al principio y al final.

- **Datos población por provincia**

  Para este set de datos se eliminó las variables Sexo, Edad (año a año) y Españoles/Extranjeros, ya que solo nos interesa la población anual por provincia para poder generar la tasa de incidencia de covid-19 mensual. Como primer paso se crea una columna equivalente a la tasa de crecimiento mensual ya que tenemos la población anual, se utiliza una de las formulas más utilizadas para cálculos poblacionales como lo es el modelo geométrico [14]. A continuación, se describe la fórmula utilizada:

$$r = \left(\frac{P_f}{P_i}\right)^{\frac{1}{t}} - 1 \tag{1}$$

  donde:

  r Tasa de crecimiento mensual
  P_f Población final
  P_i Población inicial
  t Distancia en tiempo entre las dos poblaciones de referencia

  Tomaremos la población inicial del año 2019 y la población actual del año 2022, esto debido a que en los años anteriores en casi todos los casos aumento la población, pero en este periodo de tiempo el comportamiento fue diferente a razón del covid-19, el cálculo de la tasa de crecimiento mensual se utilizará para calcular el aproximado de la población mensual por provincia hasta el primer trimestre del 2023. Se realiza el calculo de la población mensual con proyección en un periodo t, mediante la siguiente ecuación:

$$P_f = P_i(1+r)^t \tag{2}$$

  donde:

  P_f Población final en ese caso mes a mes
  P_i Población Inicial en este caso del comienzo de cada año
  r Tasa de crecimiento mensual calculada anteriormente
  t La proyección en tiempo, en este caso el mes a calcular

  El valor de esta población mensual tomara el nombre de *POB_MEN*

- **Dataset total**

  Este dataset está conformado por el dataset climatológico unido a la fuente de datos cod_iso_provincias por medio del campo *PROVINCIA* esto con el fin de añadir la columna *PROVINCIA_ISO*, a su vez este dataset se unirá a la fuente de datos de covid por medio de la columna *PROVINCIA_ISO* y la *FECHA*, para generar el dataset total, este proceso se describe mediante los siguientes comandos en Python:

```
df_clima_iso = df_clima.merge(df_iso, how="inner", on="PROVINCIA")
df_total = df_covid.merge(df_clima_iso, how="inner",
                          on=["FECHA", "PROVINCIA_ISO"])
```

  A este dataset total se eliminaron las variables de Horas (*HORA_TEMP_MIN, HORA_TEMP_MAX, HORA_RACHA, HORA_PRES_MAX, HORA_PRES_MIN*) ya que

no deseamos un nivel de granularidad tan bajo, por el contrario, se tomara en cuenta la demás variables climatológicas diarias; se elimina la variable *PROVINCIA_ISO* ya que tenemos la variable *PROVINCIA* la cual hace referencia al mismo significado; se elimina las variables *NUM_HOSP, NUM_UCI, NUM_DEFU* esto debido a la explicación de la figura **7** y el objetivo principal del estudio es la propagación del virus covid-19 es decir el número de casos (*NUM_CASOS*) y no las defunciones y/o hospitalizaciones; se elimina las variables *TEMP_MIN, TEMP_MAX* esto debido a la explicación de la figura **10** y se conserva la variable *TEMP_MED*; por el momento se eliminan las variables *GRUPO_EDAD, SEXO* para centrar el estudio en la propagación del virus entorno a las variables climatológicas. Tras esta serie de pasos se realiza una gráfica de tendencia de los casos de covid de todas las provincias para tener un panorama más amplio del caso de estudio.

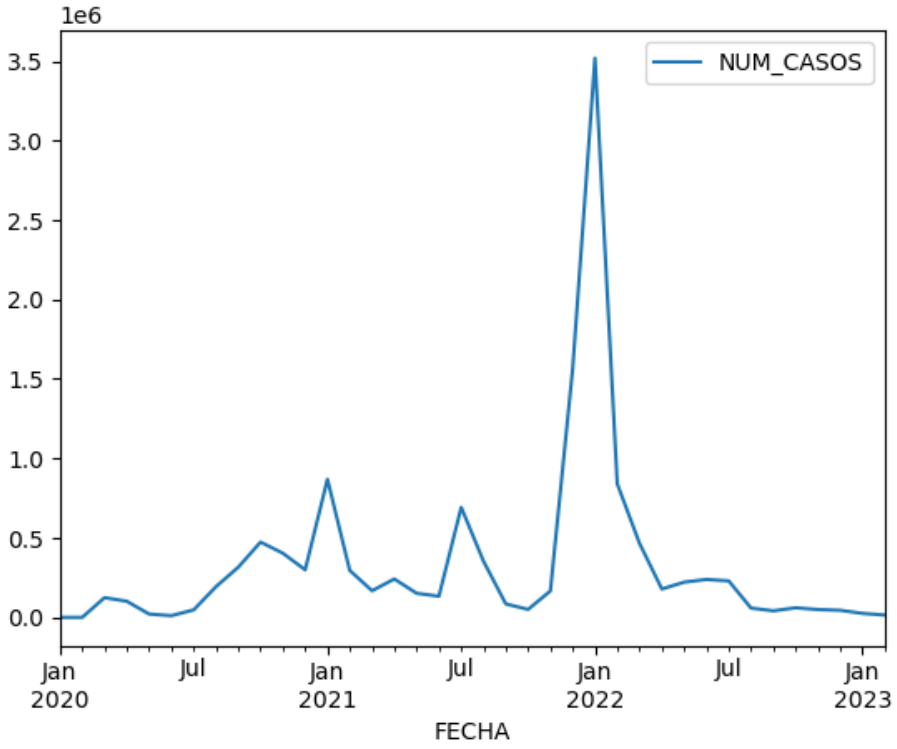

Figura 11: Tendencia de casos covid.

Fuente: [Elaboración propia]

Se crea la variable *TASA_INCIDENCIA* a partir de la normalización del número de casos, dada por la siguiente formula:

$$TASA\_INCIDENCIA = \left(\frac{NUM\_CASOS}{POB\_MEN}\right) \times 100000 \tag{3}$$

En donde *POB_MEN* (población mensual) se calculó en el ítem anterior para cada año desde el 2020 hasta el primer trimestre del 2023 por provincia. De igual forma se realiza una gráfica de tendencia de la tasa de incidencia de todas las provincias, como se puede observar en la siguiente figura la tasa de incidencia es una muy buena normalización con respecto a los casos covid de la figura **11**.

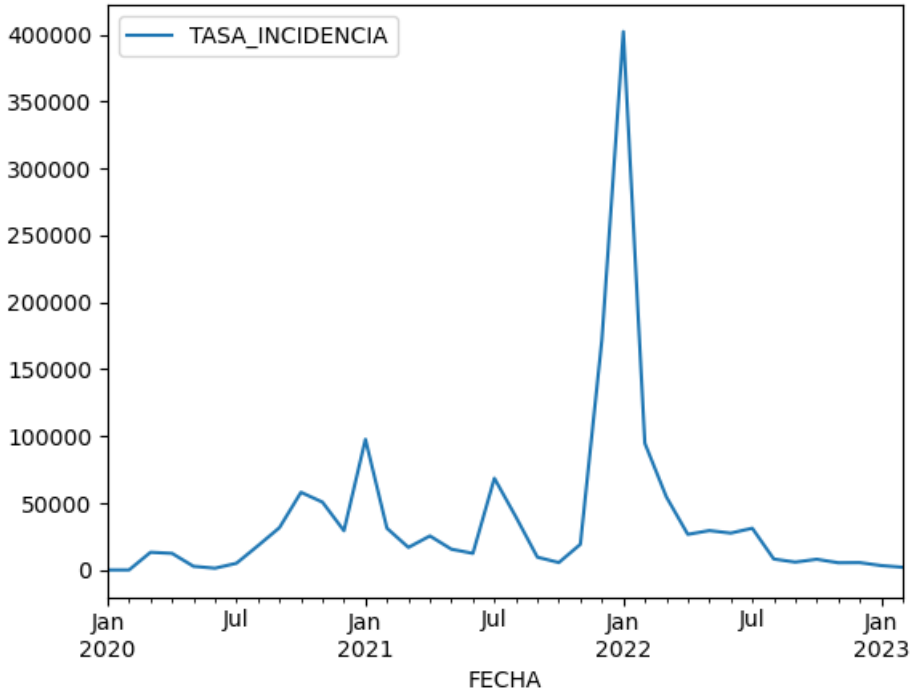

Figura 12: Tendencia tasa de incidencia.

Fuente: [Elaboración propia]

4.3.3. TRANSFORMACIÓN DE DATOS

Para cada una de las fuentes de datos descrita anteriormente se realiza la transformación de su información, así como el dataset total que esta compuesto de la unión de estás fuentes.

- **Datos casos COVID-19 por provincias**

  Se realizo la transformación del nombre de todas las columnas a mayúscula, de esta forma se trabajará en todos los dataset para manejar un estándar de nombramiento, se realiza conversión de la variable de tiempo *fecha* a tipo "datetime", las variables que estén tipo float y no tengan ningún valor decimal se convertirán en enteros. Normalización y homologación de los valores en campos categóricos con el fin de agrupar y estandarizar, obteniendo así los siguientes tipos de datos:

```
PROVINCIA_ISO            object
SEXO                     object
GRUPO_EDAD               object
FECHA             datetime64[ns]
NUM_CASOS                 int64
NUM_HOSP                  int64
NUM_UCI                   int64
NUM_DEFU                  int64
dtype: object
```

Figura 13: Tipos de datos dataset covid-19.

Fuente: [Elaboración propia]

- **Datos climatológicos por provincia**

  Se realizo la transformación del nombre de todas las columnas a mayúscula, la conversión de la variable de tiempo *fecha* a tipo "datetime", las variables que estén tipo float y no tengan ningún valor decimal se convertirán en enteros, así como las variables que son de tipo string y que en realidad todos son datos son numéricos decimales se realizará su correspondiente transformación a tipo float. A la variable prec (Precipitación diaria de 07 a 07) se transformó el valor 'Ip' (significa precipitación inapreciable, es decir, cantidad inferior a 0.1 mm) por 0,0.

  Para todas las variables de horas se transformo el valor "24" por "00" ya que hacen referencia a la misma hora, más adelante se explicará porque estas variables de horas no serán tomadas en cuenta para la etapa de minería de datos (creación de modelos). Inicialmente solo dos campos eran numéricos, tras realizar el proceso de transformación obtenemos los siguientes tipos de datos:

```
FECHA            datetime64[ns]
PROVINCIA                object
ALTITUD                   int64
TEMP_MED                float64
PREC                    float64
TEMP_MIN                float64
HORA_TEMP_MIN            object
TEMP_MAX                float64
HORA_TEMP_MAX            object
DIR                     float64
VEL_MEDIA               float64
RACHA                   float64
HORA_RACHA               object
PRES_MAX                float64
HORA_PRES_MAX            object
PRES_MIN                float64
HORA_PRES_MIN            object
SOL                     float64
dtype: object
```

Figura 14: Tipos de datos dataset clima.

Fuente: [Elaboración propia]

- **Datos población por provincia**

  Se transforma la variable *Provincias* y se hace la homologación con los mismos nombres de las provincias como en los demás conjuntos de datos, es decir, se reemplazan algunos nombres; la variable *Total* se transforma a entero ya que todos sus datos tienen decimales con ceros, además de que este debe ser un valor entero; una vez calculada la población mensual en el preprocesamiento se eliminan las variables intermedias y que no son de utilidad como *YEAR, YEAR_ACUM, TOTAL_POB, TASA_MENSUAL*

- **Dataset total**

  Se realiza la transformación de la variable *FECHA* cambiando su granularidad de diaria a mensual, de esta forma se procede a hacer la agrupación de los datos por

FECHA y *PROVINCIA* y todas las demás medidas se le hace un promedio excepto la variable *NUM_CASOS* que será la suma de todos los días para el mes correspondiente. Se eliminan las variables *NUM_CASOS, POB_MEN* tras el calculo de la tasa de incidencia de covid-19 en el ítem anterior, obteniendo así el dataset final con los siguientes tipos de datos:

```
FECHA             period[M]
PROVINCIA            object
ALTITUD             float64
TEMP_MED            float64
PREC                float64
DIR                 float64
VEL_MEDIA           float64
RACHA               float64
PRES_MIN            float64
SOL                 float64
TASA_INCIDENCIA     float64
dtype: object
```

Figura 15: Tipos de datos dataset total.

Fuente: [Elaboración propia]

Este dataset final será exportado como archivo data_refined.csv y será tomado como partida de referencia para la construcción de los modelos en nuestra etapa de minería de datos.

5. CONCLUSIÓN Y TRABAJOS FUTUROS

XXXXXX

6. REFERENCIAS

[1] Araujo, M. B., & Naimi, B. (2020). Spread of SARS-CoV-2 Coronavirus likely to be constrained by climate. MedRxiv, 2020.03.12.20034728. https://doi.org/10.1101/2020.03.12.20034728

[2] Wang, J., Tang, K., Feng, K., & Lv, W. (2020). High Temperature and High Humidity Reduce the Transmission of COVID-19. Available at SSRN 3551767. https://ssrn.com/abstract=3551767

[3] Mariette Award & Rahul Khanna. (2015). Efficient Learning Machines, Theories, Concepts, and Applications for Engineers and System Hesigners. https://link.springer.com/book/10.1007/978-1-4302-5990-9

[4] Ingry Nathaly Salamanca Rativa & Edgar Junior Castro Escorcia. (2019). Técnicas de aprendizaje automático aplicadas en los sistemas de predicción. https://revistas.udistrital.edu.co/index.php/tia/article/download/17325/17214/104552

[5] Valenzuela González, Gema. (2022). Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones. https://riuma.uma.es/xmlui/handle/10630/25147

[6] Jesús Bobadilla Sancho. (2020). Machine Learning y Deep Learning Usando Python, Scikit y Keras. https://www.perlego.com/book/2165268/machine-learning-y-deep-learning-pdf

[7] Raúl Benítez, Andrés Cencerrado Barraqué, Gerard Escudero & Samir Kanaan. (2020). https://openaccess.uoc.edu/bitstream/10609/140427/8/Inteligencia

[8] Aston Zhang, Zachary Lipton, Mu Li & Alexander J. Smola. (2021). Dive into Deep Learning, Convolutional Neural Networks. https://www.d2l.ai/chapter_convolutional-neural-networks/index.html

[9] Simeon Kostadinov. (2020). Recurrent Neural Networks With Python Quick Start Guide: Sequential Learning and Lenguage modeling

[10] Edgar Zepeda Urzua. (2021). Aprendizaje Federado con FLARE: NVIDIA Lleva la Inteligencia Artificial Colaborativa a el Área de la Salud y más allá. https://hardwareviews.com/aprendizaje-federado-con-flare-nvidia-lleva-la-inteligencia-artificial-colaborativa-a-el-area-de-la-salud-y-mas-alla/

[11] BBC Mundo. (2016). AlphaGo vs. Lee: la máquina venció al humano

[12] Alex Hughes. (2022). Dall-E 2: Why the AI image generator is a revolutionary invention

[13] SYNCED. (2020). MIT CSAIL Uses Deep Generative Model StyleGAN2 to Deliver SOTA Image Reconstruction Results

[14] Arnaldo Torres-Degró. (2011). Tasas de crecimiento poblacional (r): Una mirada desde el modelo matemático lineal, geométrico y exponencial. https://revistas.upr.edu/index.php/cidedigital/article/download/11774/9736/11342

APÉNDICE I

ANEXOS I

Tabla 4: Variables y descripción del set de datos de COVID-19 por provincias.

Variable	Descripción
provincia_iso	Código ISO de la provincia de residencia. NC (no consta)
sexo	Sexo de los casos: H (hombre), M (mujer), NC (no consta)
grupo_edad	Grupo de edad al que pertenece el caso: 0-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, $\geq$ 80 años. NC: no consta. Después del 28 de Marzo solo grupos de más de 60 años.
fecha	**Casos:** En los casos anteriores al 11 de mayo, se utiliza la fecha de diagnóstico, en su ausencia la fecha de declaración a la comunidad y, en su ausencia, la fecha clave (fecha usada para estadística por las CCAA). En los casos posteriores al 10 de mayo, en ausencia de fecha de diagnóstico se utiliza la fecha clave1. **Hospitalizaciones, ingresos en UCI, defunciones:** los casos hospitalizados están representados por fecha de hospitalización (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave, los casos UCI por fecha de admisión en UCI (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave) y las defunciones por fecha de defunción (en su defecto, la fecha de diagnóstico, y en su defecto la fecha clave).
num_casos	Número de casos notificados confirmados con una prueba diagnóstica positiva de infección activa (PDIA) tal como se establece en la Estrategia de detección precoz, vigilancia y control de COVID-19 y además los casos notificados antes del 11 de mayo que requirieron hospitalización, ingreso en UCI o fallecieron con diagnóstico clínico de COVID19, de acuerdo a las definiciones de caso vigentes en cada momento.
num_hosp	Número de casos hospitalizados
num_uci	Número de casos ingresados en UCI
num_def	Número de defunciones

Fuente: RNVD