Argentina programa 4.0

Ministerio de Desarrollo Productivo Argentina

Secretaría de Economía del Conocimiento

PROGRAMADOR JUNIOR EN MACHINE LEARNING I

UNIVERSIDAD NACIONAL DE MISIONES

FACULTAD DE CIENCIAS EXACTAS QUÍMICAS Y NATURALES

CURSO 1: Fundamentos de Machine Learning (ML) Python - Nivel 1 -

FUNDAMENTACIÓN

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes.

Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Python puede utilizarse en proyectos de inteligencia artificial, para crear sitios web escalables, realizar cálculos estructurales complejos con elementos finitos, y diseñar videojuegos, entre otras muchas aplicaciones

Objetivos

- Familiarizar al estudiante con el entorno y las herramientas de programación de Python.
- Desarrollar las capacidades de plasmar una propuesta de resolución a un problema sencillo planteado utilizando las herramientas planteadas.
- Identificar las ventajas del trabajo colaborativo y las herramientas disponibles.
- Promover la indagación sobre los temas planteados.

Introducción a los algoritmos de Machine Learning

¿Qué es el Machine Learning?

El machine learning es una disciplina de la inteligencia artificial que permite a las aplicaciones aprender de los datos y mejorar su precisión o toma de decisiones sin ser programadas para ello. Se basa en algoritmos y modelos estadísticos que procesan grandes cantidades de datos, identifican patrones e inferencias, y hacen clasificaciones o predicciones . El machine learning es un componente importante de la ciencia de datos y tiene diversas aplicaciones en diferentes campos .

Existen dos tipos principales de machine learning: supervisado y no supervisado. El machine learning supervisado consiste en hacer predicciones a futuro basadas en comportamientos o características que se han visto en los datos históricos. Para ello, se utiliza un campo especial llamado campo objetivo, que indica la categoría o el valor que se quiere predecir. Por ejemplo, se puede predecir si un correo electrónico es spam o legítimo, o el precio de una vivienda según sus características. El machine learning no supervisado usa datos históricos que no están etiquetados para explorarlos y encontrar alguna estructura o forma de organizarlos. Por ejemplo, se puede agrupar clientes con características o comportamientos similares para hacer campañas de marketing personalizadas .

Cómo funcionan los algoritmos

Los algoritmos de machine learning son métodos computacionales que permiten aprender de los datos y hacer predicciones o decisiones basadas en ellos. Los algoritmos supervisados requieren que se les proporcione un conjunto de datos etiquetados, es decir, que contienen la respuesta correcta o deseada para cada ejemplo. Los algoritmos no supervisados, en cambio, sólo reciben un conjunto de datos sin etiquetas y tienen que encontrar patrones o estructuras ocultas en ellos. Algunos ejemplos de algoritmos supervisados son la regresión lineal, la clasificación logística, las redes neuronales y los árboles de decisión. Algunos ejemplos de algoritmos no supervisados son el análisis de componentes principales, el agrupamiento k-means y el aprendizaje por refuerzo. En este curso se introducirán los conceptos básicos de los algoritmos de machine learning, así como las técnicas y herramientas para aplicarlos a problemas reales.

Clasificación de los algoritmos de machine learning

Aprendizaje supervisado

Los algoritmos de Machine Learning supervisados son aquellos que aprenden a partir de datos etiquetados, es decir, que tienen una variable objetivo o de salida que se quiere predecir o clasificar. Estos algoritmos se pueden dividir en dos tipos: clasificación y regresión. La clasificación consiste en asignar una categoría a una muestra, entre un conjunto finito de posibilidades. Por ejemplo, determinar si un correo electrónico es spam o no, o si una imagen contiene un rostro humano o no. La regresión consiste en estimar un valor numérico a partir de una o más variables de entrada. Por ejemplo, predecir el precio de una casa, o el consumo de energía de un edificio.

Existen muchos algoritmos de Machine Learning supervisados que se pueden aplicar a diferentes problemas y dominios. Algunos ejemplos son:

- Regresión lineal: es un algoritmo que busca ajustar una línea recta a los datos, minimizando la distancia entre los puntos y la línea. Se utiliza para modelar relaciones lineales entre variables, como el peso y la altura de una persona, o el número de horas de estudio y la calificación de un examen.
- Árboles de decisión: son algoritmos que construyen una estructura jerárquica de reglas para clasificar o predecir los datos. Se utilizan para modelar problemas con muchas variables categóricas, como el diagnóstico médico, o la detección de fraudes.
- K-vecinos más cercanos: es un algoritmo que clasifica o predice los datos basándose en la similitud con los k ejemplos más cercanos del conjunto de entrenamiento. Se utiliza para problemas donde la distancia entre los datos es relevante, como el reconocimiento facial, o la recomendación de productos.
- Redes neuronales: son algoritmos que imitan el funcionamiento del cerebro humano, mediante capas de unidades de procesamiento llamadas neuronas. Se utilizan para problemas complejos y no lineales, como el procesamiento de imágenes, el reconocimiento de voz, o la generación de texto.

Aprendizaje no supervisado

Los algoritmos de machine learning no supervisados son aquellos que pueden analizar y agrupar conjuntos de datos sin etiquetas, es decir, sin un conocimiento previo de las categorías o clases a las que pertenecen los datos. Estos algoritmos son capaces de descubrir patrones ocultos o estructuras en los datos que pueden ser útiles para explorar, comprender o transformar la información. Algunas de las aplicaciones más comunes de los algoritmos de machine learning no supervisados son:

- La agrupación o clustering, que consiste en dividir los datos en grupos homogéneos según su similitud o distancia. Por ejemplo, se puede utilizar el algoritmo de K-means para segmentar a los clientes de una empresa según sus características o comportamientos.

- La asociación o association rule learning, que consiste en encontrar reglas que relacionan elementos frecuentes o co-ocurrentes en los datos. Por ejemplo, se puede utilizar el algoritmo de A priori para identificar productos que se compran juntos en un supermercado.
- La reducción de dimensionalidad o dimensionality reduction, que consiste en simplificar los datos eliminando variables irrelevantes o redundantes, o proyectando los datos en un espacio de menor dimensión. Por ejemplo, se puede utilizar el algoritmo de PCA para reducir el número de características de un conjunto de datos sin perder mucha información.

Los algoritmos de machine learning no supervisados pueden ser muy útiles para descubrir conocimiento nuevo o extraer valor de los datos sin tener que depender de una supervisión humana. Sin embargo, también presentan algunos desafíos, como la dificultad para evaluar su rendimiento, la necesidad de ajustar parámetros o la sensibilidad al ruido o a los valores atípicos.

Aprendizaje por refuerzo

Los algoritmos de Machine Learning por refuerzo son una rama de la inteligencia artificial que se basa en el aprendizaje a partir de la experiencia y la recompensa. Estos algoritmos son capaces de aprender de forma autónoma y mejorar su comportamiento mediante la interacción con el entorno y el feedback que reciben. Algunos ejemplos de la vida real donde se aplican estos algoritmos son:

- Los coches autónomos, que aprenden a conducir de forma segura y eficiente mediante la observación de las condiciones del tráfico, las señales, los obstáculos y las recompensas que obtienen por llegar a su destino.
- Los videojuegos, donde los agentes inteligentes aprenden a jugar y a superar los desafíos mediante la exploración de las posibilidades, las acciones y las recompensas que consiguen por ganar o perder.
- Los sistemas de recomendación, que aprenden a sugerir productos o servicios personalizados a los usuarios mediante el análisis de sus preferencias, su historial y las recompensas que reciben por generar satisfacción o fidelidad.
- Los robots, que aprenden a realizar tareas complejas o a adaptarse a entornos cambiantes mediante la experimentación, el ensayo y error y las recompensas que obtienen por cumplir sus objetivos o evitar daños.

Criterios de elección y medición de algoritmos

¿Cómo elegir el algoritmo?

El primer criterio es el tipo de tarea que se quiere realizar con el aprendizaje automático. Existen tres tipos principales de tareas: supervisadas, no supervisadas y por refuerzo. Las tareas supervisadas son aquellas en las que se dispone de datos etiquetados, es decir, datos que tienen una salida o respuesta conocida. El objetivo es entrenar un modelo que pueda predecir la salida a partir de los datos de entrada. Dentro de las tareas supervisadas, se pueden distinguir dos subtipos: regresión y clasificación. La regresión consiste en predecir un valor numérico continuo, como el precio de una casa o la temperatura de una ciudad. La clasificación consiste en predecir una categoría o clase, como el tipo de flor o el género musical.

Las tareas no supervisadas son aquellas en las que no se dispone de datos etiquetados, es decir, datos que no tienen una salida o respuesta conocida. El objetivo es encontrar patrones o estructuras ocultas en los datos, sin tener una guía previa. Dentro de las tareas no supervisadas, se pueden distinguir dos subtipos: agrupamiento y reducción de dimensionalidad. El agrupamiento consiste en dividir los datos en grupos o clústeres según su similitud, como los clientes según sus preferencias o los documentos según su temática. La reducción de dimensionalidad consiste en reducir el número de variables o características de los datos, manteniendo la mayor cantidad de información posible, como las imágenes según sus componentes principales o los textos según sus palabras clave.

Las tareas por refuerzo son aquellas en las que el sistema aprende mediante la interacción con un entorno y la recepción de recompensas o castigos por sus acciones. El objetivo es encontrar la mejor estrategia o política para maximizar la recompensa acumulada a largo plazo. Dentro de las tareas por refuerzo, se pueden distinguir dos subtipos: basados en modelos y sin modelos. Los basados en modelos son aquellos en los que el sistema tiene un conocimiento previo o estimado del entorno y sus transiciones, como un juego de mesa o un simulador. Los sin modelos son aquellos en los que el sistema no tiene ningún conocimiento previo del entorno y debe explorarlo por sí mismo, como un robot o un agente virtual.

El segundo criterio es el tipo y la cantidad de datos disponibles para el aprendizaje automático. Los datos son la materia prima del aprendizaje automático y su calidad y cantidad influyen directamente en el rendimiento del algoritmo. Algunos aspectos a tener en cuenta son:

- La naturaleza de los datos: pueden ser numéricos, categóricos, textuales, visuales, auditivos, etc. Cada tipo de dato requiere un tratamiento específico y puede ser más adecuado para ciertos algoritmos que para otros.
- La dimensionalidad de los datos: se refiere al número de variables o características que describen cada dato. Una alta dimensionalidad puede dificultar el aprendizaje y requerir técnicas de reducción o selección de características.
- La distribución de los datos: se refiere a cómo se reparten los datos según sus valores o clases. Una distribución desequilibrada puede sesgar el aprendizaje y requerir técnicas de balanceo o ponderación.

- La limpieza de los datos: se refiere a la ausencia o presencia de ruido, errores, valores faltantes o atípicos en los datos. Una baja limpieza puede afectar negativamente al aprendizaje y requerir técnicas de preprocesamiento o imputación.

El tercer criterio es el objetivo deseado con el aprendizaje automático. El objetivo puede ser de diferente naturaleza y complejidad, y puede implicar diferentes requisitos o restricciones. Algunos aspectos a tener en cuenta son:

- La precisión del algoritmo: se refiere a la capacidad del algoritmo de predecir correctamente la salida o encontrar los patrones adecuados en los datos. La precisión se puede medir con diferentes métricas, como el error cuadrático medio, la exactitud, la precisión, el recall, el F1-score, etc. La precisión suele ser el aspecto más importante a la hora de elegir un algoritmo, pero no siempre es el único.
- El tiempo de entrenamiento del algoritmo: se refiere al tiempo que tarda el algoritmo en aprender de los datos y ajustar sus parámetros. El tiempo de entrenamiento depende de la complejidad del algoritmo, del tamaño y la dimensionalidad de los datos, y de los recursos computacionales disponibles. Un tiempo de entrenamiento largo puede ser un inconveniente si se requiere una respuesta rápida o si se dispone de pocos recursos.
- La interpretabilidad del algoritmo: se refiere a la capacidad del algoritmo de explicar cómo y por qué ha llegado a una determinada predicción o patrón. La interpretabilidad es un aspecto cada vez más relevante en el aprendizaje automático, especialmente en ámbitos sensibles como la medicina, el derecho o la ética. Algunos algoritmos son más interpretables que otros, como los árboles de decisión, las reglas de asociación o los modelos lineales.
- La generalización del algoritmo: se refiere a la capacidad del algoritmo de adaptarse a nuevos datos o situaciones que no ha visto durante el entrenamiento. La generalización es un aspecto clave para evitar el sobreajuste o subajuste del algoritmo, que ocurre cuando el algoritmo memoriza los datos de entrenamiento o no aprende lo suficiente de ellos. Algunas técnicas para mejorar la generalización son la validación cruzada, el conjunto de datos de prueba, la regularización o el ensamble.

A continuación, presentamos algunos ejemplos de la vida real en los que se aplican estos criterios para elegir el algoritmo más adecuado según el problema y los datos disponibles.

- Ejemplo 1: Se quiere predecir el precio de venta de una vivienda en función de sus características, como el número de habitaciones, el tamaño, la ubicación, etc. Se dispone de un conjunto de datos con miles de viviendas vendidas y sus precios reales. Este es un problema de regresión supervisada, ya que se tiene una salida numérica conocida. Un posible algoritmo para resolverlo es la regresión lineal múltiple, que asume una relación lineal entre las variables y es fácil de interpretar. Sin embargo, si se sospecha que hay relaciones no lineales entre las variables, se podría optar por un algoritmo más flexible y potente, como una red neuronal artificial o un bosque aleatorio.
- Ejemplo 2: Se quiere segmentar a los clientes de una empresa según sus hábitos de consumo, preferencias y satisfacción. No se dispone de ninguna etiqueta o categoría previa para los clientes. Este es un problema de agrupamiento no supervisado, ya que se busca encontrar grupos similares sin tener una guía previa. Un posible algoritmo para resolverlo es el K-Means, que asigna cada cliente a uno de los K clústeres según su distancia a los centroides. Sin embargo, si se desconoce el número óptimo de clústeres

o si los datos tienen formas irregulares, se podría optar por un algoritmo más robusto y flexible, como el DBSCAN o el agrupamiento jerárquico.

- Ejemplo 3: Se quiere entrenar a un agente virtual para que juegue al ajedrez contra un humano o contra otro agente. El agente recibe una recompensa por cada movimiento que realiza y debe aprender a maximizar su puntuación final. Este es un problema de aprendizaje por refuerzo sin modelo, ya que el agente debe explorar el entorno y encontrar la mejor estrategia sin tener ningún conocimiento previo del juego. Un posible algoritmo para resolverlo es el Q-learning, que estima el valor esperado de cada acción en cada estado y actualiza su tabla Q según la recompensa recibida.

Técnicas de optimización de los algoritmos

Los algoritmos de Machine Learning son métodos computacionales que permiten aprender de los datos y hacer predicciones o clasificaciones. Sin embargo, para obtener buenos resultados, es necesario aplicar algunas técnicas de optimización que mejoran el rendimiento y la generalización de los modelos. Estas técnicas son:

- Validación cruzada: consiste en dividir el conjunto de datos en varias partes, y usar una de ellas como prueba y el resto como entrenamiento. Se repite el proceso varias veces, cambiando la parte de prueba, y se calcula el error medio. Esto permite estimar la capacidad de generalización del modelo y evitar el sobreajuste.
- Selección de características: consiste en elegir las variables más relevantes para el problema, y descartar las que no aportan información o son redundantes. Esto permite reducir la dimensión de los datos, simplificar el modelo y mejorar su interpretabilidad.
- Ajuste de hiperparámetros: consiste en encontrar los valores óptimos de los parámetros que controlan el comportamiento del algoritmo, como el número de iteraciones, la tasa de aprendizaje, el grado del polinomio, etc. Esto se puede hacer mediante métodos de búsqueda exhaustiva, aleatoria o basada en gradiente.

Estas técnicas se pueden ilustrar con algunos ejemplos:

- Validación cruzada: supongamos que queremos entrenar un modelo de regresión lineal para predecir el precio de una casa a partir de sus características. Podemos usar validación cruzada de 10 iteraciones para evaluar el error cuadrático medio del modelo.
- Selección de características: supongamos que queremos entrenar un modelo de clasificación binaria para detectar si un correo electrónico es spam o no. Podemos usar un método de filtrado basado en la información mutua para seleccionar las palabras más relevantes que aparecen en el texto del correo.
- Ajuste de hiper parámetros: supongamos que queremos entrenar un modelo de clasificación multiclase para reconocer dígitos escritos a mano. Podemos usar un método de búsqueda aleatoria para encontrar el número óptimo de capas y neuronas de una red neuronal artificial.

Evaluación del rendimiento y la calidad de los algoritmos

La evaluación del rendimiento y la calidad de los algoritmos es un aspecto fundamental en el aprendizaje automático. Existen diferentes métricas que nos permiten medir la eficacia de un algoritmo para clasificar o predecir datos. Algunas de estas métricas son:

- Matrices de confusión: Son tablas que muestran el número de aciertos y errores de un algoritmo al comparar las etiquetas reales con las predichas. Las matrices de confusión nos permiten calcular otras métricas como la precisión, la exhaustividad, el valor F1, etc.
- Curvas ROC: Son gráficos que representan la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) de un algoritmo para diferentes umbrales de clasificación. Las curvas ROC nos permiten comparar el rendimiento de diferentes algoritmos y elegir el umbral óptimo para maximizar la TPR y minimizar la FPR.
- Área bajo la curva ROC (AUC): Es una medida numérica que resume la calidad de una curva ROC. El AUC es el porcentaje de área que queda bajo la curva ROC. Cuanto más cercano sea el AUC a 1, mejor será el rendimiento del algoritmo. Un AUC de 0.5 indica que el algoritmo no tiene capacidad discriminativa.

Veamos algunos ejemplos de estas métricas aplicadas a problemas reales de clasificación.

Ejemplo 1: Supongamos que tenemos un algoritmo que predice si un paciente tiene o no una enfermedad cardíaca basándose en ciertas características clínicas. El algoritmo produce una salida binaria: 1 si tiene la enfermedad y 0 si no la tiene. Para evaluar el rendimiento del algoritmo, podemos usar una matriz de confusión que compare las etiquetas reales con las predichas por el algoritmo. La matriz de confusión tendría esta forma:

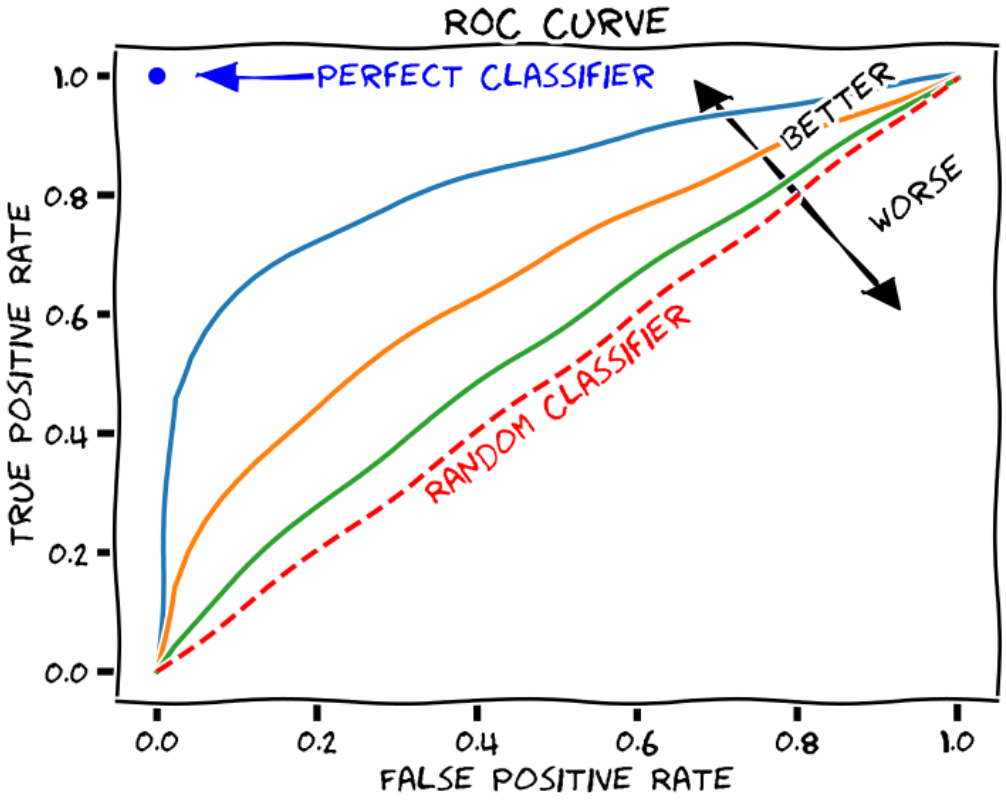
	Predicción 1	Predicción 0
Real 1	TP	FN
Real 0	FP	TN

Donde TP son los verdaderos positivos (pacientes con enfermedad correctamente clasificados), FN son los falsos negativos (pacientes con enfermedad incorrectamente clasificados como sanos), FP son los falsos positivos (pacientes sanos incorrectamente clasificados como enfermos) y TN son los verdaderos negativos (pacientes sanos correctamente clasificados). A partir de esta matriz, podemos calcular otras métricas como:

- Precisión: Es la proporción de predicciones positivas que son correctas. Se calcula como TP / (TP + FP).
- Exhaustividad: Es la proporción de casos positivos reales que son correctamente clasificados. Se calcula como TP / (TP + FN).
- Valor F1: Es una medida que combina la precisión y la exhaustividad en un solo valor. Se calcula como 2 * (Precisión * Exhaustividad) / (Precisión + Exhaustividad).

Estas métricas nos dan una idea de cómo de bien el algoritmo clasifica a los pacientes con y sin enfermedad cardíaca.

Ejemplo 2: Supongamos que tenemos otro algoritmo que predice la probabilidad de que un cliente compre o no un producto basándose en su historial de compras. El algoritmo produce una salida continua entre 0 y 1, donde 0 significa que el cliente no comprará el producto y 1 significa que sí lo hará. Para evaluar el rendimiento del algoritmo, podemos usar una curva ROC que muestre la relación entre la TPR y la FPR para diferentes umbrales de clasificación. Por ejemplo, si usamos un umbral de 0.5, el algoritmo clasificará como positivos (compradores) a los clientes con una probabilidad mayor o igual a 0.5, y como negativos (no compradores) a los clientes con una probabilidad menor a 0.5. La curva ROC tendría esta forma:



Donde el eje x representa la FPR y el eje y representa la TPR. La línea punteada representa el comportamiento de un algoritmo aleatorio, que no tiene capacidad discriminativa. La curva azul representa el comportamiento de nuestro algoritmo, que tiene una capacidad discriminativa moderada. Podemos ver que a medida que aumentamos el umbral de clasificación, la TPR y la FPR disminuyen, y viceversa. El punto óptimo sería aquel que maximiza la TPR y minimiza la FPR, es decir, el punto más cercano a la esquina superior izquierda del gráfico. Para cuantificar la calidad de la curva ROC, podemos usar el AUC, que en este caso sería el área sombreada en gris bajo la curva azul. Un AUC cercano a 1 indicaría que el algoritmo tiene una alta capacidad discriminativa, mientras que un AUC cercano a 0.5 indicaría que el algoritmo no tiene capacidad discriminativa.

Estas métricas nos dan una idea de cómo de bien el algoritmo predice la probabilidad de compra de los clientes y cómo elegir el umbral de clasificación más adecuado para nuestro objetivo.

Cuestionario guia de lectura del material

Este es un cuestionario guía de cinco preguntas sobre algoritmos de Machine Learning. El objetivo es evaluar su nivel de conocimiento y comprensión de los conceptos básicos de esta disciplina. Las preguntas son las siguientes:

1. ¿Qué es un algoritmo de Machine Learning y para qué sirve?
2. ¿Qué diferencia hay entre aprendizaje supervisado y no supervisado?
3. ¿Qué son las características y las etiquetas en un conjunto de datos?
4. ¿Qué es el sobreajuste y cómo se puede evitar?
5. ¿Qué es la validación cruzada y cuál es su utilidad?

Bibliografía:

1. Bagnato, J. i., (2020). Aprende Machine Learning en Español: Teoría + Práctica Python. Editorial Leanpub.
2. Britos, P. V., & García Martínez, R. (2009). Propuesta de Procesos de Explotación de Información. In XV Congreso Argentino de Ciencias de la Computación.
3. Chazallet, S. (2016). Python 3: los fundamentos del lenguaje. Ediciones ENI.
4. Geron, A., (2020). Aprende Machine Learning con Scikit-Learn, Keras y Tensor Flow: Conceptos, herramientas y técnicas para construir sistemas inteligentes. Editorial O'Reilly y Anaya
5. Hilera, J. R. y Martinez, V. J. (2000) Redes Neuronales Artificiales. Fundamentos. modelos y aplicaciones. Alfaomega Ed
6. JIMÉNEZ, R. O., (2021). Python a fondo. Editorial Marcombo
7. Kuna, H. D., Caballero, S., Rambo, A., Meinl, E., Steinhilber, A., Pautsch, J., ... & Villatoro, F. (2010). Avance en procedimientos de la explotación de información para la identificación de datos faltantes, con ruido e inconsistentes. In XII Workshop de Investigadores en Ciencias de la Computación.
8. Kuna, H., Pautsch, G., Rey, M., Cuba, C., Rambo, A., ... & Villatoro, F. (2012). Obtenido de COMPARACIÓN DE LA EFECTIVIDAD DE PROCEDIMIENTOS DE LA EXPLOTACIÓN DE INFORMACIÓN PARA LA IDENTIFICACIÓN DE OUTLIERS EN BASES DE DATOS:
9. Matthes, E. (2021) Curso intensivo de Python, 2ª edición: Introducción práctica a la programación basada en proyectos. Editorial Anaya Multimedia
10. Ochoa, M. A. (2004). Herramientas inteligentes para explotación de información. Trabajo Final Especialidad en Ingeniería de Sistemas Expertos url: https://ri.itba.edu.ar/server/api/core/bitstreams/a848d640-0277-459d-9104-b37017309d31/content