

## Daniel Oliva González

## Máster de Programación Avanzada en Python para Hacking, BigData y Machine Learning

## Interfaz en Streamlit para realizar predicciones

Resumen

Durante estos últimos años la Inteligencia Artificial se encuentra en auge, puesto que prácticamente toda empresa o sector se puede ver beneficiado por esta rama de la informática al poder mejorar sus ventas e ingresos, adaptarse a los nuevos clientes…

Aunque existen gran cantidad de métodos para realizar predicciones, los algoritmos de predicción de aprendizaje supervisado (necesitan del conocimiento en los datos de la variable de salida) han adquirido una gran importancia, ya que, gracias a ello, las empresas pueden predecir todo tipo de aspectos, como sería el precio de venta al público. Lo único que necesitan es transmitirle el conocimiento al modelo de predicción y ese conocimiento reside en datos previos que serán usado para el entrenamiento del modelo.

Con el objetivo de amenizar la experiencia de realizar predicciones, surge la idea de desarrollar una aplicación de tal forma que el usuario pueda cargar cualquier tipo de datos y se vean los datos faltantes, cómo se encuentran distribuidos los datos, entrenar un modelo algorítmico, escogiendo el porcentaje de datos destinados a training y a test, obtener un resultado fiable mediante diferentes métricas y realizar predicciones tanto de regresión como de clasificación, simplemente realizando unos cuantos clics e introduciendo los datos de la instancia a predecir.

**Keyword: aprendizaje supervisado, predicciones, datos, aplicación**

Abstract

During these past years, Artificial Intelligence has been booming, since practically every company or sector can benefit from this branch of information technology by being able to improve its sales and income, adapt to new customers...

Although there are a large number of methods for making predictions, supervised learning prediction algorithms (they require knowledge of the output variable data) have gained great importance, since, thanks to this, companies can predict all kinds of events. aspects, such as the retail price. The only thing they need is to transmit the knowledge to the prediction model and that knowledge resides in previous data that will be used for training the model.

In order to liven up the experience of making predictions, the idea of ​​developing an application arises in such a way that the user can load any type of data and see the missing data, how the data is distributed, train an algorithmic model, choosing the percentage of data destined for training and testing, obtain a reliable result through different metrics and make both regression and classification predictions, simply by making a few clicks and entering the data of the instance to be predicted.

**Keywords: predictions, supervised learning, data, application**

Índice

[1. Introducción 9](#_Toc127985021)

[Motivación 9](#_Toc127985022)

[Objetivos 9](#_Toc127985023)

[Aplicaciones del aprendizaje automático 10](#_Toc127985024)

[Estructura de la memoria 10](#_Toc127985025)

[2. Estado del arte 13](#_Toc127985026)

[3. Contexto tecnológico 15](#_Toc127985027)

[Inteligencia Artificial 15](#_Toc127985028)

[Aprendizaje automático 15](#_Toc127985029)

[Streamlit 19](#_Toc127985030)

[4. Metodología de trabajo y herramientas usadas 21](#_Toc127985031)

[Metodología 21](#_Toc127985032)

[Herramientas de trabajo empleadas 22](#_Toc127985033)

[Python 23](#_Toc127985034)

[Github 23](#_Toc127985035)

[PyCharm 23](#_Toc127985036)

[5. Desarrollo del trabajo 25](#_Toc127985037)

[6. Problemas y soluciones encontradas 41](#_Toc127985039)

[6.1. Tamaño de las imágenes. 41](#_Toc127985040)

[6.2. Tabs y obtención de los nombres. 43](#_Toc127985041)

[6.3. Columnas 44](#_Toc127985042)

[6.4. Reinicio de la aplicación 44](#_Toc127985043)

[7. Resultados 47](#_Toc127985044)

[8. Conclusiones 53](#_Toc127985045)

[9. Futuras líneas y mejoras 55](#_Toc127985046)

[9.1. Selector para variables 55](#_Toc127985047)

[9.2. Implementación de más algoritmos de predicción 55](#_Toc127985048)

[9.3. Estudio de los outliers 57](#_Toc127985049)

[10. Referencias bibliográficas 59](#_Toc127985050)

[11. Anexos 61](#_Toc127985051)

[11.1. Datos Iris 61](#_Toc127985052)

[11.2. Datos Titanic 62](#_Toc127985053)

[11.3. Datos Boston 63](#_Toc127985054)

[11.4. Manual de usuario 64](#_Toc127985055)

Tabla de Ilustraciones

[Ilustración 1. Tipos de kernel para SVM de clasificación (Scikit-Learn, s.f.) 18](#_Toc127985056)

[Ilustración 2. Ejemplo gráfico de árbol de decisión (Heras, 2020) 19](#_Toc127985057)

[Ilustración 3. Fases de la metodología CRISP-DM (Wikipedia, s.f.) 22](#_Toc127985058)

[Ilustración 4. Zona de introducción del conjunto de datos 26](#_Toc127985059)

[Ilustración 5. Desplegable para seleccionar datos 27](#_Toc127985060)

[Ilustración 6. Fragmento de código para carga de datos 27](#_Toc127985061)

[Ilustración 7. Radio selector de tipo de predicción 28](#_Toc127985062)

[Ilustración 8. Fragmento de código para la construcción del radio selector 28](#_Toc127985063)

[Ilustración 9. Alerta de selección de predicción 28](#_Toc127985064)

[Ilustración 10. Gráfico de valores faltantes 29](#_Toc127985065)

[Ilustración 11. Mensajes de valores faltantes 29](#_Toc127985066)

[Ilustración 12. Estadística descriptiva del conjunto de datos 30](#_Toc127985067)

[Ilustración 13. Fragmento de código de valores faltantes y estadística descriptiva 30](#_Toc127985068)

[Ilustración 14. Botón de imputar valores 31](#_Toc127985069)

[Ilustración 15. Selector de variable a predecir 31](#_Toc127985070)

[Ilustración 16. Muestra de datos de dataset Titanic 31](#_Toc127985071)

[Ilustración 17. Cuadro de introducción de porcentaje de test 32](#_Toc127985072)

[Ilustración 18. Conjuntos de training y de test 33](#_Toc127985073)

[Ilustración 19. Tabs de Clasificación 34](#_Toc127985074)

[Ilustración 20. Tabs de Regresión 34](#_Toc127985075)

[Ilustración 21. Fragmento de código de tabs de Clasificación 34](#_Toc127985076)

[Ilustración 22. Matriz de correlación de dataset Iris 35](#_Toc127985077)

[Ilustración 23. Error producido al realizar la matriz de correlación 35](#_Toc127985078)

[Ilustración 24. Código generador del multiselector 35](#_Toc127985079)

[Ilustración 25. Boxplot para la variable age de dataset Titanic 36](#_Toc127985080)

[Ilustración 26. Displot para la variable age de dataset Titanic 36](#_Toc127985081)

[Ilustración 27. Facetgrid para las dos variables del dataset Iris 37](#_Toc127985082)

[Ilustración 28. Violinplot para la variable sepalLengthCm de dataset Iris 37](#_Toc127985083)

[Ilustración 29. Radio selector de regresión 38](#_Toc127985084)

[Ilustración 30. Radio selector de clasificación 38](#_Toc127985085)

[Ilustración 31. Pasos de entrenamiento y resultado de la predicción 38](#_Toc127985086)

[Ilustración 32. Mensaje de accuracy 39](#_Toc127985087)

[Ilustración 33. Mensajes de métricas de error 39](#_Toc127985088)

[Ilustración 34. Inputs creados para las variables del dataset Titanic 39](#_Toc127985089)

[Ilustración 35. Fragmento de código generador de los inputs 40](#_Toc127985090)

[Ilustración 36. Pantalla final de la aplicación. 40](#_Toc127985091)

[Ilustración 37. Imagen demasiado grande en Streamlit 42](#_Toc127985092)

[Ilustración 38. Imagen demasiado pequeña en Streamlit 43](#_Toc127985093)

[Ilustración 39. Resultado final de la gráfica en aplicación 44](#_Toc127985094)

[Ilustración 40. Fragmento de código para solucionar tabs 45](#_Toc127985095)

[Ilustración 41. Fragmento de código de botones y estados 46](#_Toc127985096)

[Ilustración 42. Clasificación según sepal-length y sepal-width 48](#_Toc127985097)

[Ilustración 43. Clasificación segun sepal-length y petal-width 49](#_Toc127985098)

[Ilustración 44. Clasificación según sepal-length y petal-length 49](#_Toc127985099)

[Ilustración 45. Clasificación según sepal-width y petal-width 50](#_Toc127985100)

[Ilustración 46. Clasificación según petal-width y petal-length 50](#_Toc127985101)

[Ilustración 47. Violinplot para la variable sepalWidthCm de dataset Iris 51](#_Toc127985102)

[Ilustración 48.Violinplot para la variable petallLengthCm de dataset Iris 51](#_Toc127985103)

[Ilustración 49. Violinplot para la variable petalWidthCm de dataset Iris 52](#_Toc127985104)

[Ilustración 50. Imagen Perceptrón Simple (Gutiérrez) 57](#_Toc127985105)

[Ilustración 51. Red Neuronal Artificial (Wikipedia, s.f.) 57](#_Toc127985106)

[Ilustración 52. Muestra dataset Iris (UCI Machine Learning, s.f.) 62](#_Toc127985107)

[Ilustración 53. Muestra conjunto de datos Titanic (Kaggle, s.f.) 63](file:///C:\Users\DEEPGAMING\Desktop\TFM-Interfaz_de_Usuario_Streamlit\OLIVA_GONZALEZ_DANIEL_Memoria_TFM.docx#_Toc127985108)

[Ilustración 54. Muestra datos de Boston (PERERA, s.f.) 64](#_Toc127985109)

[Ilustración 55. Datos sin separar por columnas 65](#_Toc127985110)

[Ilustración 56. Datos separados por columnas 65](#_Toc127985111)

# Introducción

## Motivación

La motivación para la realización de este trabajo surge al crear una aplicación de análisis de datos que puede ser usada para cualquier persona (utilizando el manual de usuario) sin que tenga unos conocimientos previos de programación y así realice predicciones para un conjunto de datos existente.

## Objetivos

El objetivo principal se trata de realizar una interfaz para que un usuario con unos conocimientos básicos sea capaz de realizar predicciones introduciendo un conjunto de datos cualquiera.

El usuario cargará su conjunto de datos en formato “.csv” para así escoger entre realizar una clasificación de una instancia o realizar una regresión. Además, el usuario que está en proceso de realizar la predicción podrá visualizar cualquier dato faltante existente en su conjunto de datos.

Así mismo, el usuario escogerá la variable a predecir y visualizará una matriz de correlación entre las variables y diferentes gráficas según las variables seleccionadas.

De esta forma, el usuario seleccionará un modelo algorítmico en función del tipo de predicción que desee realizar, para el que se realizará el entrenamiento y se mostrará la precisión o el error dependiendo de la predicción para que, por último, el introduzca los valores de las variables de la instancia que desee predecir.

## Aplicaciones del aprendizaje automático

**Finanzas**

El objetivo de la inteligencia artificial en las finanzas es la predicción de precios de las propiedades al considerar la ubicación y otros precios históricos,

**Gobierno**

Las agencias gubernamentales se sirven de la IA para así modernizar y digitalizar sus procesos y así reducir los posibles riesgos de ataques cibernéticos sobre las bases de datos nacionales, controlar gastos, aumentar la eficiencia de los empleados…

**Fabricación**

Los empresarios se ayudan de los pronósticos generados por la IA para disminuir la inactividad de la producción, incrementar la eficiencia o para mejorar el trato y la satisfacción del cliente. Así mismo, estas predicciones se suelen utilizar para el diseño de procesos, mantenimiento, etc.

**Cuidado de la salud**

La salud se sirve de la IA para realizar predicciones acerca de posibles afectados por diversas enfermedades como podría ser el cáncer.

**Seguro**

Las compañías de seguro utilizan la IA para predecir la satisfacción del cliente, además de establecer tarifas en función de las necesidades de sus clientes.

## Estructura de la memoria

**2. Estado del arte**

Se verán distintas tecnologías para realizar predicciones, así como una introducción de los diferentes usos del aprendizaje supervisado.

**3. Contexto tecnológico**

En este apartado se abordarán los temas referentes a la Inteligencia Artificial, aprendizaje supervisado…

Metodología de trabajo y herramientas usadas

**4. Metodología de trabajo y herramientas usadas**

Se explicará el tipo de metodología usada durante el desarrollo de la aplicación, así como las herramientas usadas para el mismo.

**5. Desarrollo del trabajo**

Durante este apartado se explicará todo lo realizado para construir la aplicación, así como diferentes funcionalidades incluidas para intentar obtener la mejor predicción posible.

**6. Problemas y soluciones encontradas**

Aunque Streamlit se trata de un framework muy potente y sencillo que permite crear aplicaciones, en la realización del proyecto se han encontrado ciertos problemas. En este apartado se expondrán los más relevantes y las soluciones pensadas.

**7. Resultados**

Se mostrarán los diferentes resultados que se pueden obtener para los conjuntos de datos que se han usado para la construcción de la aplicación y los diferentes algoritmos existentes.

**8. Conclusiones**

Se explicarán aquellos aspectos más destacados encontrados en la realización del proyecto.

**9. Futuras líneas y mejoras**

En este apartado, se aportarán ciertas líneas de posibles continuaciones a este trabajo con el fin de complementarlo y mejorarlo.

**10. Referencias bibliográficas**

Se mostrarán todas las referencias consultadas para la realización del trabajo.

**11. Anexos**

Los anexos se encontrarán compuestos por un manual de usuario con el objetivo que el usuario no se encuentre desubicado si decide usar la aplicación y la explicación de los diferentes conjuntos de datos usados (Iris, Titanic y Boston) para que así entienda cómo se trabaja con ellos.

# Estado del arte

En este apartado se va a realizar el estudio de otras posibles herramientas en la creación de modelos algorítmicos para la realización de predicciones:

* **H20 AI Cloud**: se trata de la mejor opción para empresas que quieren generar modelos y aplicaciones de IA, desarrollando los modelos en la nube. Presentan incorporado la función de autoML garantizando una predicción de datos precisa. H20 AI Cloud oferta una combinación única de algoritmos patentados y de código abierto, con un rendimiento óptimo de CPU y GPU
* **Neptune**: es una herramienta conocida por permitir realizar un seguimiento de los resultados del aprendizaje automático.
* **DataRobot**: tiene como objetivo acerca la IA a diferentes industrias utilizando modelos de aprendizaje automático que necesita código bajo para realizar predicciones en tiempo real.
* **Obviously AI**: realiza predicciones de ingresos y resultados comerciales sin código, haciendo uso de la IA. En función de estos datos, las empresas modifican su cadena de suministro además de establecer nuevas estrategias de marketing. Esta herramienta es fácil de integrar con fuentes de datos como Google Drive o Dropbox.
* **Futrli**: expone pronósticos comerciales rápidos sobre tendencias comerciales, ingresos, ventas, impuestos... Esta herramienta atiende sobre todo a la contabilidad de las empresas, analizando toda transacción comercial de la empresa para hacer un pronóstico informado.
* **Pecan**: genera datos predictivos para ventas, de tal forma que las empresas encuentran soluciones a sus problemas de negocio.
* **Qlik Sense**: aporta datos para que personas con diferente nivel de habilidad tomen decisiones informadas.
* **Dataiku**: esta herramienta incorpora 109 tipos de transformaciones de datos para realizar una limpieza de datos manualmente. Además, presenta una GUI interactiva ola cual permite acceder a los datos en un número mínimo de clics.

# Contexto tecnológico

En este apartado, el objetivo es situar el proyecto en la tecnología aplicada.

## Inteligencia Artificial

La **Inteligencia Artificial** o IA, se trata de un campo de la ciencia informática que presenta como objetivo la resolución de problemas asociados a los humanos, como podrán ser el aprendizaje o el reconocimiento de patrones.

Dentro de la Inteligencia Artificial hay dos campos existentes: el **aprendizaje automático** y el **aprendizaje profundo**.

A su vez, el aprendizaje automático se puede dividir en aprendizaje supervisado (será sobre el que se centre este proyecto), que necesitan datos de entrenamiento que incluyan la salida deseada, y aprendizaje no supervisado, que, por el contrario, no lo necesitan.

La Inteligencia Artificial aumenta su conocimiento y su rapidez al aprender cuantos más datos posee. A más datos, mayor conocimiento y mejores predicciones.

## Aprendizaje automático

El **aprendizaje automático** o **Machine Learning** (en la memoria se referirá a este concepto de las dos formas), se trata de una de las ramas de la Inteligencia Artificial. Su objetivo principal es el desarrollo de técnicas que permitan a las computadoras aprender, es decir, de crear modelos abstractos capaces de generalizar comportamientos y poder reconocer patrones a partir de una información suministrada para entrenarlo.

El aprendizaje supervisado se encarga de resolver problemas donde existe una variable de especial interés en el conjunto de datos, conocida como variable de salida. Fundamentalmente existen dos tipos de problemas de predicción que son resueltos utilizando aprendizaje supervisado y van a depender del tipo de dato que almacene la variable de salida:

* **Regresión**. El objetivo es realizar una predicción numérica (número real)
* **Clasificación**. La variable de salida es categórica

Este proyecto se centrará en esta parte de la Inteligencia artificial, el aprendizaje supervisado.

Además, para cada problema de predicción existen diversos modelos. En el proyecto para la regresión se usarán KNR y LinearRegression; mientras que, para la clasificación, SVC, KNN y DecisionTreeClassifier.

Los algoritmos de **regresión lineal** tienen como objetivo encontrar relaciones lineales entre los atributos y la clase. Se distinguen dos tipos de regresiones: simple y múltiple. La regresión simple se basa en obtener el valor de salida a partir de una única variable de entrada; mientras que, la regresión múltiple consiste en encontrar una combinación de valores para las variables de entrada y así predecir el resultado de la variable de salida.

Las ventajes e inconvenientes para destacar de la regresión lineal son:

* Simplicidad del modelo. Este algoritmo se trata de una ecuación matemática constituida por una combinación lineal de los atributos de entrada

En la ecuación y será el valor predicho por el algoritmo, los valores de w serán constantes y x serán los valores que adoptan las variables para predecir las diferentes instancias.

* Sencillo de interpretar. Es fácil de comprender la importancia de cada atributo gracias a los coeficientes asociados a los valores de entrada. A un valor mayor (en valor absoluto), mayor importancia presentará la variable de entrada.
* Tiempo de entrenamiento razonable. El entrenamiento del modelo consiste en ajustar los valores de los coeficientes w con el objetivo de conseguir el valor más acertado posible. Dependiendo de la tasa de aprendizaje, la convergencia al óptimo llevará más o menos tiempo.
* Tiempo de predicción prácticamente instantáneo. Al ser una ecuación matemática, conseguir las predicciones es casi inmediato.

En la regresión lineal se asumen ciertas hipótesis:

* Independencia entre atributos
* Relación lineal entre atributos y clase

Los algoritmos de **vecinos cercanos** (KNN para clasificación y KNR para regresión) encuentran los ejemplos similares en sus atributos con un comportamiento parecido en el valor de sus clases. En regresión, este algoritmo realiza un promedio de los “n” vecinos más cercanos obteniendo así un valor a predecir. Para clasificación, simplemente se encargará de definir la etiqueta de clase a predecir viendo cuál es la que predomina entre sus “n” vecinos más cercanos.

Los algoritmos de **support vector machines** son usados tanto para clasificación y regresión como para detección de outliers. En este caso, se usará únicamente el de clasificación.

Gráfico, Mapa, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Tipos de kernel para SVM de clasificación (Scikit-Learn, s.f.)

Como se observa en la “Ilustración 1. Tipos de kernel para SVM de clasificación”, este algoritmo dependiendo del tipo de kernel realiza diferentes agrupaciones, siendo efectivos en conjuntos de datos donde las dimensiones son mayores que el número de ejemplos.

El algoritmo de **árbol de decisión** será utilizado para clasificación, aunque también es usado para la regresión. Presenta una estructura de árbol jerárquica con nodo raíz, ramas, nodos internos y hojas.

Diagrama

Descripción generada automáticamente

Ilustración . Ejemplo gráfico de árbol de decisión (Heras, 2020)

Como se observa en la “Ilustración 2. Ejemplo gráfico de árbol de decisión”, este sería un ejemplo claro de un árbol de decisión. En este caso, según el valor de la variable “petal length” si es más pequeño que 2.45 será de tipo setosa; en el caso contrario, se estudia el valor de la variable “petal width”, si es menor que 1.75 será versicolor, si no, será virginica.

Este ejemplo es referente al conjunto de datos Iris (“Datos Iris “)

## Streamlit

Streamlit se trata de un framework de Python de código abierto que permite el desarrollo de una manera sencilla de aplicaciones mediante la interacción con otras librerías usado para Machine Learning y ciencia de datos. Para este proyecto, se ha usado la versión 1.15.2.

Streamlit posee una gran cantidad de elementos para dar forma a la aplicación, entre los que se han usado:

* Contenedores. Actúa a modo de recipiente donde se “guardan” todos los demás elementos.
* Columnas. Usando este comando se permite generar una división de la página, incluso aportando un tamaño a las columnas
* Selectbox. Este comando permite al usuario escoger una opción de un conjunto de posibilidades
* Botones. Al ser pulsados por el usuario, permiten la ejecución de otras partes del código.
* Tabs. Generan pestañas con nombres de tal forma que al escoger una se ejecuta cierto fragmento de código.
* Imagen
* Dataframe. Streamlit permite mostrar un conjunto de datos con una cierta
* Multiselect. Es similar al selectbox, salvo que el usuario en este caso puede seleccionar más de una opción
* Radio. En este caso, el usuario debe rellenar el círculo de la opción seleccionada.
* Entrada de texto o de número.
* Resaltadores de texto. Streamlit posee tres formas de mostrar un texto con importancia y son:
  + Success: muestra el texto en un fondo verde, simulando éxito.
  + Warning: el texto se muestra con un fondo amarillo, advirtiendo al usuario
  + Error: en este caso, el fondo será rojo e indicará un error.

# Metodología de trabajo y herramientas usadas

## Metodología

La metodología empleada durante la realización del trabajo será **CRISP-DM**, que se trata de la abreviación de ““Cross-industry standard process for data mining”

Esta metodología circular surge en la década de los años 90 y gracias a ella se explica el clico de vida de un proyecto de análisis de datos y de cómo se realiza un desarrollo de software de una aplicación. Se encuentra compuesta por unas fases no rígidas, es decir, que se pueden realizar unas antes que las otras, aunque habitualmente siguen la disposición mostrada a continuación:

Diagrama

Descripción generada automáticamente

Ilustración . Fases de la metodología CRISP-DM (Wikipedia, s.f.)

Como se observa en la “Ilustración 3. Fases de la metodología CRISP-DM (Wikipedia, s.f.)”, se explica el proceso de la metodología en cuestión siendo una metodología circular que no presenta un orden establecido, ya que se puede volver a las diferentes fases según las necesidades del proyecto. El siguiente paso, será explicar las fases que la componen:

• **Fase 1: Business Understanding**. Se podría definir como la primera fase, ya que se realiza un estudio de las necesidades iniciales del cliente, además de establecer y estudiar los objetivos a alcanzar.

En este proyecto serían los objetivos o el estudio de las otras posibles herramientas existentes. Esta fase no ha sido realizada una única vez, ya que al ir surgiendo diferentes ideas durante el mismo desarrollo se ha vuelto a esa para ampliar los objetivos.

• **Fase 2: Data Understanding.** En ella se estudia la organización de los datos realizando diferentes actividades para obtener el conocimiento necesario de los mismos.

En este proyecto, pertenecerían a esta fase el estudio de los diferentes tipos de variables que se pueden encontrar en un conjunto de datos.

• **Fase 3: Data Preparation.** Para la tercera fase, se establece un análisis de los datos, seleccionando aquellas características necesarias para los datos finales.

Para este proyecto esta fase no ha sido realmente cómo preparar los datos, sino cómo se puede generalizar los diferentes tipos de variables para que sean tratadas y usadas en la predicción de la instancia, aportando el valor necesario.

**• Fase 4: Modelling.** En ella se usan las técnicas de modelado para diseñar una solución al problema en cuestión. Como ha ocurrido en este proyecto, muchas veces se debe volver a los datos debido a los requerimientos específicos.

Esta fase es dentro del proyecto en la que más horas se han empleado, puesto que se deben conectar los diferentes elementos usados para conseguir el desarrollo correcto de la aplicación. Se trata realmente del desarrollo de la aplicación en sí.

• **Fase 5: Evaluation.** Como su propio nombre indica, se evalúa la aplicación para conseguir resultados. Además, se evalúan los diversos pasos seguidos para estudiar si existe alguna cuestión de negocio que no ha sido considerada suficiente.

• **Fase 6: Deployment**. En esta fase se realiza la puesta en producción.

## 

## Herramientas de trabajo empleadas

### Python

Python se trata de un lenguaje de programación que presenta un amplio abanico de posibilidades de uso, desde aplicaciones web hasta la ciencia de datos y Machine Learning, pasando por el desarrollo de software. Es utilizado por los desarrolladores puesto que eficiente y sencillo de aprender, así como ejecutable en diferentes plataformas. Algunas de las aplicaciones de este lenguaje son:

- Desarrollo web del lado del servidor

- Automatización con scripts de Python

- Realizar tareas de ciencia de datos y Machine Learning

- Desarrollo de software

- Automatización de pruebas de software

En este proyecto se han utilizado librerías para usar algoritmos de Machine Learning (scikit-learn), para construir gráficas para la visualización de datos (seaborn), para abrir y manipular datos (numpy y pandas) y para la creación de la interfaz de usuario.

La versión de Python utilizada durante la realización de este proyecto ha sido la 3.9.

### Github

Con el fin de realizar un control exhaustivo de las versiones, en el proyecto se ha utilizado la plataforma de desarrollo colaborativo como Github, creando diversos repositorios y utilizando diferentes ramas e incluso pudiendo recuperar los archivos subidos al repositorio. En este trabajo se ha accedido usando la página web y sus comandos, aunque también se puede emplear la aplicación de escritorio Github Desktop

### PyCharm

Se trata de un entorno de desarrollo integrado (IDE) multiplataforma utilizado concretamente para el lenguaje de programación Python. Ha sido desarrollado por JetBrains.

Se ha seleccionado este IDE por la experiencia previa usándolo, además de sus amplias funcionalidades incorporadas.

Se ha utilizado la versión Community 2022.2.3.

# Desarrollo del trabajo

En el desarrollo de la memoria se alternarán imágenes de las partes de código más difíciles o interesantes, así como de capturas del resultado del desarrollo de la aplicación. Durante el desarrollo del trabajo no se ha seguido el orden exacto que se seguirá en la memoria, pero resultará más sencillo de comprender si se explica en el orden en el que se encuentra en la aplicación antes que el orden real, aunque el inicio sí que coinciden, puesto que lo primero que se introdujo fue la utilidad de introducir los datos:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración . Zona de introducción del conjunto de datos

Como se observa en la “Ilustración 4. Zona de introducción del conjunto de datos”, se establece una zona en la que el usuario puede arrastrar su conjunto de datos o si lo prefiere pulsar el botón de “Browse files” para cargar manualmente el conjunto de datos navegando por las carpetas, como se ve en la “Ilustración 5. Desplegable para seleccionar datos”:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Desplegable para seleccionar datos

El código utilizado para generar esta funcionalidad junto al mensaje de información se muestra en la “Ilustración 6. Fragmento de código para carga de datos”

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código para carga de datos

Además, una vez se han cargado los datos, se procede a leerlos usando la librería Pandas.

En esta situación, una vez se disponen de los datos, se plantea un radio para seleccionar si se va a realizar una clasificación o una regresión:

Texto

Descripción generada automáticamente con confianza media

Ilustración . Radio selector de tipo de predicción

# Texto Descripción generada automáticamente

Ilustración . Fragmento de código para la construcción del radio selector

En la “Ilustración 8. Fragmento de código para la construcción del radio selector”, además se observa como en el caso de no haber seleccionado el tipo de predicción, se mostrará una alerta (“Ilustración 9. Alerta de selección de predicción”).



Ilustración . Alerta de selección de predicción

En el caso de ser seleccionado, se comenzará con el estudio del conjunto de datos mostrando si existen valores faltantes como se muestra en la “Ilustración 10. Gráfico de valores faltantes”, donde se observa un rectángulo por cada valor faltante en el conjunto de datos. Además, en este caso, al haber seleccionado una pequeña parte del dataset Titanic y unas variables en concreto se entiende bien el gráfico.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración . Gráfico de valores faltantes

En el caso que no se pudiera entender bien, se mostrará una alerta por cada valor faltante y la variable en la que falte. (“Ilustración 11. Mensajes de valores faltantes”)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Mensajes de valores faltantes

También, se mostrará estadística descriptiva de las variables numéricas del conjunto de datos:

Tabla

Descripción generada automáticamente

Ilustración . Estadística descriptiva del conjunto de datos

Estas últimas funcionalidades se muestran una al lado de la otra, para lo que se usará el elemento de **columns** de Streamlit.

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código de valores faltantes y estadística descriptiva

En el caso de haber algún valor faltante, se habilitará un botón (“Ilustración 14. Botón de imputar valores”) que permitirá imputar los valores de tal forma que ninguna de las variables presente “nulls”. Para la imputación de valores se ha usado la moda.



Ilustración . Botón de imputar valores

Cuando se tiene el conjunto de datos imputado, se debe seleccionar la variable sobre la que realizar la predicción.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Selector de variable a predecir

La variable utilizada para realizar la predicción, claramente, dependerá del conjunto de datos que se use. De esta forma, ya se podrá mostrar los datos como se muestra en la “Ilustración 16. Muestra de datos de dataset Titanic”

Tabla

Descripción generada automáticamente

Ilustración . Muestra de datos de dataset Titanic

Una vez se consigue mostrar los datos, el siguiente paso es dividir el conjunto de datos en training y test y para ello se añade un elemento de introducción de un número para establecer el porcentaje de datos usados para test.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Ilustración . Cuadro de introducción de porcentaje de test

A continuación, se tienen que ajustar los datos. Para ello, se debe normalizar los datos y se establecen tres tipos de escalado de datos, que se escogerá en el selector:

* StandardScaler: estandariza las características eliminando la media y escalando a la varianza de la unidad.
* MinMaxScaler: escala y traduce cada característica individualmente de modo que esté en el rango dado, por ejemplo, entre cero y uno.
* RobustScaler: elimina la mediana y escala los datos de acuerdo con el rango del cuantil.

Además, en caso de haber una variable de tipo string y no sea la variable para predecir se transformará a int (variable numérica); mientras que, si es la variable predictora se transformará a one-hot encoding, un array con tantas posiciones como clases tenga la variable a predecir.

De esta forma, pulsando en el botón de “Transformar datos” se mostrará como se muestra en la “Ilustración 18. Conjuntos de training y de test” los conjuntos de training y de test.

Tabla

Descripción generada automáticamente

Ilustración . Conjuntos de training y de test

Ahora se procederá a mostrar las diferentes gráficas usando el elemento de tabs (o pestañas) de Streamlit. Para ello, dependiendo de si se está realizando una clasificación o predicción se crearán unas pestañas u otras.

Imagen que contiene Texto

Descripción generada automáticamente

Ilustración . Tabs de Clasificación



Ilustración . Tabs de Regresión

Como se observa en las “Ilustración 19. Tabs de Clasificación” y “Ilustración 20. Tabs de Regresión”, se crean tres pestañas para clasificación y regresión donde coincide la matriz de correlación.

En la “Ilustración 21. Fragmento de código de tabs de Clasificación” se muestra cómo se implementan las tabs y las funciones a las que se invocan cuando se selecciona la pestaña correspondiente.

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código de tabs de Clasificación

La matriz de correlación va a indicar cómo la relación existente entre dos atributos, pudiendo el valor estar en el rango entre -1 y 1. Cuanto más negativo y cercano a -1 sea el valor, esos dos atributos se encontrarán relaciones de manera inversa (si uno aumenta, el otro disminuye); por el contrario, cuanto más cercano sea al número 1, más directamente proporcionales serán, es decir, si aumenta uno, aumentará el otro. En la “Ilustración 22. Matriz de correlación de dataset Iris”, se observa un ejemplo de la matriz de correlación en el conjunto de datos de Iris.

Gráfico

Descripción generada automáticamente

Ilustración . Matriz de correlación de dataset Iris

Así mismo, si al construirse la matriz de correlación se produjese un error, se mostraría como en la “Ilustración 23. Error producido al realizar la matriz de correlación”



Ilustración . Error producido al realizar la matriz de correlación

Además, se introduce un multiselector, de tal forma que se seleccionan las variables que se quieren representar en las gráficas.



Ilustración . Código generador del multiselector

Las gráficas disponibles como se ha comentado anteriormente dependerán del tipo de predicción a realizar, de tal forma que regresión se construirá un boxplot o un distplot:

* Boxplot (“Ilustración 25. Boxplot para la variable age de dataset Titanic”): o diagrama de caja, se encarga de representar los datos numéricos a través de sus cuartiles y mediana, pudiendo representarse también, los valores atípicos.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

Ilustración . Boxplot para la variable age de dataset Titanic

* Distplot (“Ilustración 26. Displot para la variable age de dataset Titanic”): muestra cómo se distribuye la densidad de la población.

Gráfico, Histograma

Descripción generada automáticamente

Ilustración . Displot para la variable age de dataset Titanic

En estos casos, se acompañan las gráficas de un selector, con el objetivo de seleccionar la variable a representar.

Para la clasificación, se decide utilizar las siguientes gráficas:

* Facetgrid (“Ilustración 27. Facetgrid para las dos variables del dataset Iris”): se representan todas las muestras en forma de puntos en función de dos variables elegidas en un selector y cada punto (una instancia en el conjunto de datos), es clasificada en función de la variable a predecir

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Facetgrid para las dos variables del dataset Iris

* Violinplot (“Ilustración 28. Violinplot para la variable sepalLengthCm de dataset Iris”): muestra cómo se reparte la densidad de la población respecto a una variable. En este caso, se muestra uno por cada una de las clases de la variable a predecir.

Gráfico, Gráfico de superficie

Descripción generada automáticamente

Ilustración . Violinplot para la variable sepalLengthCm de dataset Iris

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamente  Ilustración . Radio selector de regresión | Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto, Correo electrónico  Descripción generada automáticamente  Ilustración . Radio selector de clasificación |

A la hora de seleccionar el algoritmo a usar para la predicción, también dependerá si es clasificación o regresión.

Texto

Descripción generada automáticamente

Ilustración . Pasos de entrenamiento y resultado de la predicción

Una vez seleccionado el algoritmo, se producirá el entrenamiento del modelo, para que a continuación se muestre el accuracy (precisión) o el error, dependiendo de si es clasificación o regresión respectivamente. A continuación, se mostrarán los cuadros de textos para introducir la instancia a predecir y por último, se dará el resultado. Estos pasos han sido descritos en la “Ilustración 31. Pasos de entrenamiento y resultado de la predicción”

Además, para el algoritmo de SVC se debe escoger el kernel utilizado para la clasificación y para los algoritmos de KNN y KNR, el número de vecinos más cercanos.

Tras esto, es cuando se muestran las métricas correspondientes (“Ilustración 32. Mensaje de accuracy” y “Ilustración 33. Mensajes de métricas de error”).

En el caso de la accuracy, dependiendo del valor, se mostrará usando un fondo verde, amarillo o rojo.

Las métricas de error corresponden con:

* MAE: (mean absolute error) error absoluto medio
* MAPE: (mean absolute percentage error) error absoluto medio porcentual
* MSE: (mean squared error) error cuadrático medio



Ilustración . Mensaje de accuracy

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Mensajes de métricas de error

Tras mostrar las métricas, se mostrarán los cuadros de introducción de los valores de la instancia a predecir.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración . Inputs creados para las variables del dataset Titanic

En la “Ilustración 34. Inputs creados para las variables del dataset Titanic”, se muestran los tres tipos de cuadros de introducción existentes:

* Caso 1: en el caso de que solo existan dos valores numéricos en esa variable, se muestran en el mensaje superior.
* Caso 2: que la clase sea numérica y tenga más de dos valores, donde no se mostrarían estos.
* Caso 3: que la clase sea de tipo string, donde se mostrarían los posibles valores para posteriormente ser transformados de manera consecuente.

En la “Ilustración 35. Fragmento de código generador de los inputs” se muestra el código usado para la generación de los respectivos cuadros de introducción de valores.

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código generador de los inputs

Por último, se mostrará el resultado de la predicción y uso globos subiendo a modo de felicitación.Gráfico, Gráfico de burbujas

Descripción generada automáticamente

Ilustración . Pantalla final de la aplicación.

# Problemas y soluciones encontradas

Como ya se ha comentado anteriormente, Streamlit se trata de una potentísima librería que permite generar interfaces para la solución de problemas de análisis de datos; sin embargo, durante el desarrollo, se han encontrado diversos problemas que han dificultado el mismo. En este punto, se van a explicar los problemas encontrados y las diferentes soluciones propuestas.

## Tamaño de las imágenes.

Uno de los problemas encontrados ha sido el tamaño de las gráficas al ser pintadas en Streamlit, las cuáles ocupaban toda la pantalla haciendo que no existiera sintonía con el resto de los elementos de la interfaz y resultando ser costoso de ver todo el gráfico.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Imagen demasiado grande en Streamlit

Como se puede observar en “Ilustración 37. Imagen demasiado grande en Streamlit”, este gráfico no se muestra de manera correcta, al ser el tamaño el mismo superior, quedando sin poderse contemplar el título del gráfico y el nombre del eje X.

Streamlit no permite de una forma sencilla establecer un tamaño de la imagen (o gráfico) mediante un parámetro en el método, como sería lógico.

Por ello, se fueron explorando diversas formas de ajustar el tamaño de la imagen a lo querido. Se piensa en usar las “tabs” (pestañas en español) para que, por cada combinación de las variables seleccionadas por el usuario, se mostrará una tab con su respectiva imagen, (desembocando en el problema 2), resultando en una imagen considerablemente más pequeña, pero que no centrada en la página.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración . Imagen demasiado pequeña en Streamlit

En este caso, se contempla como la “Ilustración 38. Imagen demasiado pequeña en Streamlit” muestra el gráfico considerablemente más pequeño que en la “Ilustración 37. Imagen demasiado grande en Streamlit”. Además, se observan las diferentes tabs con los nombres (posibles combinaciones de las variables del conjunto de datos Iris).

Para conseguir que la imagen fuera más pequeña, se optó por generar el gráfico, guardarlo en un archivo y posteriormente cargarlo de nuevo, siendo finalmente descartado al no tratarse de una solución eficiente

Finalmente, la solución encontrada es usar las “columns”. Este elemento de Streamlit permite dividir la pantalla, como su propio nombre indica, en columnas (que como se ha explicado anteriormente, cabe la posibilidad de asignarle un tamaño dentro de la página). De esta manera, se construyen 3 columnas que presenta un tamaño mayor la columna central con el objetivo de mostrar la imagen en un tamaño adecuado.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Resultado final de la gráfica en aplicación

En la “Ilustración 39. Resultado final de la gráfica en aplicación” se observa el resultado final de la gráfica.

Buscando cómo resolverse esta situación, se llegaron a los siguientes problemas: “6.2. Tabs y obtención de los nombres.” y “6.3. Columnas”.

## Tabs y obtención de los nombres.

Como ya se ha podido ver en el punto 1, existe un momento del desarrollo en el que se plantea la posibilidad de introducir las tabs con su respectiva imagen. Esta idea, aunque como se observa en la “Ilustración 38. Imagen demasiado pequeña en Streamlit” se consiguió, quedó descartada.

El problema reside en que se generan las diferentes combinaciones de las variables del conjunto de datos, las cuales van a ser los nombres de las tabs. La importancia del nombre se debe a la hora de construir el gráfico, puesto que se necesitan los nombres de las variables que se encuentran separado por un guion en el nombre de la tab. Por ello, habría resultado mucho más sencillo obtener el nombre de la tab en cuestión y así usar los nombres de las variables.

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código para solucionar tabs

Para la solución, como se muestra en la “Ilustración 40. Fragmento de código para solucionar tabs”, se opta por usar un bucle for para recorrer las diferentes combinaciones posibles y así obtener el nombre usado para montar el gráfico.

## Columnas

Las columnas a pesar de ser la solución a uno de los problemas en los que se ha invertido más tiempo también han presentado muchos inconvenientes relacionados con las imágenes. Otra de las opciones que se plantea al mostrar los gráficos, es usar una misma ventana para poder mostrar todas las imágenes de las gráficas de las posibles combinaciones de las variables del conjunto de datos.

El objetivo era encadenar dos grupos de columnas, de tal modo que en la columna central se creara otro grupo de columnas para así que se mostraran las imágenes de una forma ordenada y fácil de ver y comparar.

## Reinicio de la aplicación

Este problema hace referencia a que por cada botón que se quiera crear hay que introducir una entrada en el diccionario de “session\_state” indicando el valor en “true” para así una vez se seleccione una opción dentro de un “selectbox” o al introducir un valor en un cuadro de texto, no se produzca un reinicio de la aplicación teniendo que comenzar desde el principio.

Texto

Descripción generada automáticamente

Ilustración . Fragmento de código de botones y estados

Para evitar este inconveniente, se introducen las líneas de código mostradas en la “Ilustración 41. Fragmento de código de botones y estados”, permitiendo tener guardado el estado del botón y así no se producirá el reinicio de la aplicación

# Resultados

En este apartado, se van a mostrar los resultados obtenidos para el conjunto de datos de iris, aunque, los resultados como tal, serían el conjunto de imágenes mostradas durante el desarrollo de la aplicación.

Lo primero que se va a ver es la matriz de correlación obtenida. Para ello, será necesario ver la “Ilustración 22. Matriz de correlación de dataset Iris”.

La gráfica de Facetgrid y de Violinplot, se pueden ver en la “Ilustración 27. Facetgrid para las dos variables del dataset Iris” y la “Ilustración 28. Violinplot para la variable sepalLengthCm de dataset Iris” respectivamente.

Además, se van a mostrar las restantes gráficas de Facetgrid.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Clasificación según sepal-length y sepal-width

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Clasificación segun sepal-length y petal-width

Gráfico, Gráfico de dispersión, Gráfico de burbujas

Descripción generada automáticamente

Ilustración . Clasificación según sepal-length y petal-length

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Clasificación según sepal-width y petal-width

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración . Clasificación según petal-width y petal-length

A continuación, se mostrarán los restantes violinplots:

Gráfico, Gráfico de superficie

Descripción generada automáticamente

Ilustración . Violinplot para la variable sepalWidthCm de dataset Iris

Gráfico, Gráfico de superficie

Descripción generada automáticamente

Ilustración .Violinplot para la variable petallLengthCm de dataset Iris

Gráfico, Gráfico de superficie

Descripción generada automáticamente

Ilustración . Violinplot para la variable petalWidthCm de dataset Iris

También, se mostrará la accuracy obtenida usando los diferentes modelos.

* Árbol de decisión: 97% de accuracy
* SVC (depende del kernel): 95.1% de accuracy (media de 10 intentos)

La que ha obtenido mejor accuracy es el kernel ‘rbf’

* KNN (depende del número de vecinos): con 5 se obtiene un valor cercano al 100%

# Conclusiones

Durante este trabajo he comprendido que Streamlit es un potentísimo framework con un amplio abanico de elementos a tu disposición para generar una aplicación capaz de cumplir prácticamente cualquier objetivo referente al análisis de datos, pero también he descubierto ciertos aspectos los cuales intentaría mejorar para poder hacer la experiencia al desarrollador más sencilla.

Respecto al resultado final de la aplicación, pienso que se ha conseguido el objetivo principal, puesto que cualquier persona que tenga un conjunto de datos apoyándose del manual de usuario será capaz de realizar predicciones.

# Futuras líneas y mejoras

## Selector para variables

Actualmente, en la aplicación el usuario debe introducir su conjunto de datos únicamente con las variables que desee realizar su predicción. Una de las posibles futuras mejoras sería incorporar un selector múltiple, para que, una vez el usuario introduzca los datos, realice una selección de las diferentes variables que desea incorporar, de tal modo que no debería estar realizando cambios constantes en el conjunto de datos inicial.

## Implementación de más algoritmos de predicción

Una posible futura mejora o línea de trabajo sería continuar añadiendo algoritmos de predicción.

Además de incorporar otros algoritmos referentes al aprendizaje supervisado como sería la regresión logística, se podría implementar el **aprendizaje no supervisado** que al contrario que el aprendizaje supervisado y su naturaleza predictiva, este tipo de aprendizaje se basa en buscar las relaciones ocultas existentes entre los datos que permitan describirlos. Para el aprendizaje no supervisado no existe una variable de salida a predecir, ya que intenta agrupar el conjunto de datos en diferentes subconjuntos. Este aprendizaje se encuentra compuesto por las técnicas de **clustering**.

Otra posible implementación sería la incorporación de **Deep Learning** mediante las redes neuronales. Las **redes neuronales artificiales** (RNA) se originan en función de las redes neuronales biológicas. El modelo más simple se trata del **perceptrón**, compuesto por las diferentes señales de entrada, una función de activación y la salida a predecir:

Diagrama

Descripción generada automáticamente

Ilustración . Imagen Perceptrón Simple (Gutiérrez)

La ventaja de formar amplias redes neuronales conectando diferentes perceptrones en paralelo son:

* **Aprendizaje adaptativo**. Las RNA realizan predicciones a partir del conocimiento extraído de los datos
* **Auto-organización**. Se encuentran compuestas por una organización característica propia.
* **Operación en tiempo real** ya que se encuentran conectadas en paralelo.
* **Tolerancia a fallos parciales**. Esto se debe a que, si el sistema neuronal es dañado parcialmente, no es destruido por completo.

Diagrama

Descripción generada automáticamente

Ilustración . Red Neuronal Artificial (Wikipedia, s.f.)

Como se observa en la imagen “Ilustración 43. Red Neuronal Artificial (Wikipedia, s.f.)” una red neuronal se encuentra compuesta por:

* la **capa de entrada**, que presentará tantas entradas como de variables disponga el conjunto de datos
* un número variable de **capas ocultas** en el que también varía el número de neuronas de la que se compone la capa y la función de activación de estas
* la **capa de salida** que variará de las variables a predecir

Las RNA serán los más complicado de implementar ya que va a depender del número de capas ocultas que se quieran añadir, además de escoger todas las diferentes opciones que se presentan a la hora de construir una RNA y el tiempo de entrenamiento (que dependerá del tamaño del conjunto de datos)

## Estudio de los outliers

Un **outlier** o valor atípico se trata de una observación anormal y extrema en una muestra estadística o serie de datos que pueda afectar a la estimación de los parámetros del algoritmo, es decir, es un valor que se encuentra distante al resto del conjunto de datos.

En este proyecto no se encuentra incorporado, pero sería una buena práctica añadir el estudio de outliers con el fin de afinar los modelos de predicción y conseguir las mejores métricas posibles.

# Referencias bibliográficas

*¿Qué es un outlier o atípico?* (s.f.). Obtenido de https://aprendeconeli.com/que-es-un-outlier-atipico/#:~:text=Un%20outlier%20es%20una%20observaci%C3%B3n,de%20los%20par%C3%A1metros%20del%20mismo

Amazon. (s.f.). Obtenido de aws.amazon.com

Correoso, K. (9 de Septiembre de 2020). *Haciendo un Dashboard con Streamlit en Python*. Obtenido de https://pybonacci.org/2020/02/20/haciendo-un-dashboard-con-streamlit-en-python/

Cortés, G. A. (2018). *APRENDIZAJE SUPERVISADO.* Universidad Internacional de Valencia.

Das, T. (17 de Noviembre de 2022). *Las 8 mejores herramientas de pronóstico y predicción de datos impulsadas por IA para empresas*. Obtenido de https://geekflare.com/es/ai-powered-business-forecasting-tools/

Gutiérrez, J. M. (s.f.). *Introducción a las Redes Neuronales.* Universidad de Cantabria. Obtenido de https://personales.unican.es/gutierjm/docs/tut\_RedesNeuro.pdf

Heras, J. M. (9 de Septiembre de 2020). *Árboles de Decisión con ejemplos en Python*. Obtenido de https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/

Hernández-González, J. (2019). *APRENDIZAJE NO SUPERVISADO.* Universidad Internacional de Valencia.

*How to Display Multiple Images in One Figure Correctly in Matplotlib?* (2 de Febrero de 2021). Obtenido de geeksforgeeks: https://www.geeksforgeeks.org/how-to-display-multiple-images-in-one-figure-correctly-in-matplotlib/

IBM. (s.f.). Obtenido de https://www.ibm.com/es-es

JetBrains. (s.f.). Obtenido de https://www.jetbrains.com/

Kaggle. (s.f.). *Titanic - Machine Learning from Disaster*. Obtenido de https://www.kaggle.com/competitions/titanic/overview

Kömürcü, İ. (7 de Noviembre de 2022). *Streamlit App for Data Science Projects*. Obtenido de https://heartbeat.comet.ml/streamlit-app-for-data-science-projects-cd569d934497

*Matplotlib*. (s.f.). Obtenido de https://matplotlib.org/

Pandas. (s.f.). Obtenido de https://pandas.pydata.org/

PERERA, P. (s.f.). *The Boston Housing Dataset*. Obtenido de https://www.kaggle.com/code/prasadperera/the-boston-housing-dataset

*Scikit-Learn*. (s.f.). Obtenido de https://scikit-learn.org/stable/

*SeaBorn*. (s.f.). Obtenido de https://seaborn.pydata.org/index.html

StackOverflow. (s.f.). *StackOverflow*. Obtenido de https://stackoverflow.com/

*StreamlLit*. (s.f.). Obtenido de https://streamlit.io/

UCI Machine Learning. (s.f.). *Kaggle*. Obtenido de Iris Species: https://www.kaggle.com/datasets/uciml/iris

Watt, J., Borhani, R., & K. Katsaggelos, A. (2016). *Machine Learning Refined: Foundations, Algorithms, and Applications.* Cambridge University Press.

*Wikipedia*. (s.f.). Obtenido de es.wikipedia.org

# Anexos

## Datos Iris

El conjunto de datos de Iris se trata de un dataset balanceado (todas las clases tienen el mismo número de datos) de clasificación de flores. Cada fila se trata de las medidas del sépalo y pétalo de una flor y su respectiva etiqueta. Se encuentra compuesto por 150 instancias.

Los atributos que se utilizan en el conjunto de datos son:

* Id: se trata del id del dato. No aporta ninguna información relevante
* SepalLengthCm: un dato numérico sobre la longitud del sépalo
* SepalWidthCm: un dato numérico sobre la anchura del sépalo
* PetalLengthCm: un dato numérico sobre la longitud del pétalo
* PetalWidthCm: un dato numérico sobre la anchura del pétalo
* Species: tipo de flor a predecir

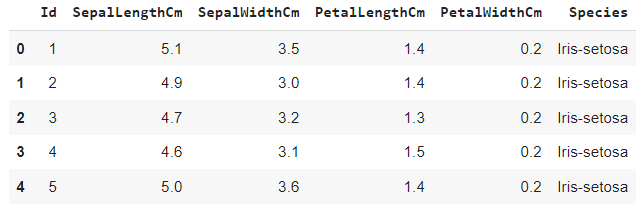


Ilustración . Muestra dataset Iris (UCI Machine Learning, s.f.)

## Datos Titanic

Todo el mundo conoce la tragedia del barco Titanic. Este conjunto de datos son las muestras de los diferentes pasajeros del barco que pudieron sobrevivir y los que no. Las variables que se utilizan en el dataset:

* PassengerId: no aporta conocimiento a la predicción
* Survived: variable numérica con valor 0 o 1, que indica si el pasajero sobrevivió o no. Se puede realizar una clasificación o una regresión.
* Pclass: variable numérica que indica el tipo de servicio del pasajero
* Name: nombre del pasajero que no aporta conocimiento a la predicción
* Sex: género del pasajero, es una variable de tipo string
* Age: edad del pasajero, es una variable numérica
* SibSp: variable numérica que especifica el número de hermanos a bordo
* Parch: variable numérica que especifica el número de padres e hijos a bordo.
* Ticket: variable de tipo string que indica el ticket (puede no aportar conocimiento)
* Fare: variable numérica que indica el precio del ticket (puede no aportar conocimiento)
* Cabin: variable string que indica la cabina de la persona
* Embarked: variable string que indica el puerto de embarque

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración . Muestra conjunto de datos Titanic (Kaggle, s.f.)

## Datos Boston

Se trata de un conjunto de datos de regresión que tiene como objetivo predecir el valor de una casa en Boston. Las variables que componen el conjunto de datos son:

* CRIM: variable numérica que indica la tasa de criminalidad per cápita
* ZN: variable numérica que indica la proporción de terreno residencial zonificado para lotes de más de 25,000 pies cuadrados
* INDUS: variable numérica que indica la proporción de acres comerciales no minoristas por ciudad.
* CHAS: variable ficticia del río Charles (1 si el tramo limita con el río; 0 en caso contrario)
* NOX: variable numérica que indica la concentración de óxidos nítricos
* RM: variable numérica que indica el número promedio de cuartos por vivienda
* AGE: variable numérica que indica la proporción de unidades ocupadas por propietarios construidas antes de 1940
* DIS: variable numérica que indica las distancias ponderadas a cinco centros de empleo de Boston
* RAD: variable numérica que indica el índice de accesibilidad a las carreteras radiales
* TAX: variable numérica que indica la tasa de impuestos sobre la propiedad de valor total por cada $10,000
* PTRATIO: variable numérica que indica el ratio alumno-profesor por localidad
* B - 1000(Bk - 0.63)^2 donde Bk es la proporción de negros por ciudad
* LSTAT: variable numérica que indica el % estado más bajo de la población
* MEDV: variable numérica que indica el valor medio de las viviendas ocupadas por sus propietarios en miles de dólares

Imagen que contiene Texto

Descripción generada automáticamente

Ilustración . Muestra datos de Boston (PERERA, s.f.)

## Manual de usuario

* Antes de comenzar el usuario deberá instalar Python 3.9, así como las dependencias utilizadas en el desarrollo del trabajo.
* Una vez hecho esto, el usuario deberá abrir una terminal en la carpeta en la que se encuentre el fichero .py e introducir el siguiente comando:

“streamlit run tfm.py”

* El primer paso que debe seguir el usuario es realizar una limpieza de las diferentes variables que considere que no será de importancia en la predicción del modelo.
* El usuario deberá asegurarse que el conjunto de datos introducido está en formato .csv y las columnas se encuentran separadas. Para ello tendrá los datos como en la “Ilustración 47. Datos sin separar por columnas”:

Tabla

Descripción generada automáticamente

Ilustración 55. Datos sin separar por columnas

Y usará la funcionalidad de Excel para separar los datos en columnas, resultando como en la “Ilustración 48. Datos separados por columnas“

Tabla

Descripción generada automáticamente con confianza baja

Ilustración . Datos separados por columnas

Además, en esta última imagen se observa una variable que no aporta conocimiento para la predicción como es el Id, por lo que el usuario deberá eliminar esta variable como se comenta en el paso 1.

* El usuario deberá introducir el conjunto de datos usando el desplegable de la “Ilustración 5. Desplegable para seleccionar datos” o arrastrando los datos a la zona correspondiente.
* Una vez introducidos los datos, el usuario elegirá el tipo de predicción a realizar en función del conjunto de datos cargado, como se muestra en la “Ilustración 7. Radio selector de tipo de predicción”
* En el caso de ser necesario, deberá pulsar el botón de “Imputar valores”.
* Una vez, hayan sido imputados (si hiciera falta), el usuario seleccionará la variable sobre la que se realizará la predicción.
* A continuación, se pulsará el botón “Mostrar datos” para ver el conjunto de datos.
* El siguiente paso, será seleccionar cómo desea escalar los datos.
* Al seleccionar el tipo de escalado de datos, se activará el cuadro de introducción del porcentaje de datos para test, transformando y dividiendo los datos en training y test.
* Luego, el usuario podrá ver las respectivas gráficas dependiendo de si se encuentra en clasificación o regresión.
* Ahora, deberá pulsar el botón de “Realizar predicciones” para así poder seleccionar el algoritmo que desee.
* Tras seleccionar el algoritmo, si es necesario, deberá introducir alguna característica propia del algoritmo y luego se entrenará el modelo, mostrando la accuracy o el error del modelo.
* El último paso, será introducir todas las características de la instancia a predecir y pulsar el botón de “Predecir resultado”