**PRÁCTICA 1**

**INTRODUCCIÓN**

En esta práctica realizaremos una introducción a la aplicación de modelos de regresión en ejemplos reales de manera que se estudiarán todo tipo de datos con tal de realizar un estudio de los mismos con el objetivo de evaluar errores y validar resultados.

Un modelo de regresión es un modelo matemático que busca determinar la relación entre una variable independiente, con respecto a otras variables. Este se utiliza con el fin de determinar si existe o no una relación entre una variable independiente y las demás variables así mismo buscando determinar cuál es el impacto sobre estas.

Para realizar estos modelos, es necesario el uso de técnicas de regresión, estas son un tipo de técnicas estadísticas usadas para el modelado predictivo y la minería de datos.

Para aplicar estas técnicas se usará un notebook con lenguaje python que usa librerías como pueden ser numpy, scikit-learn, matplotlib o spicy, las cuales serán útiles para evaluar el comportamiento de las variables del modelo y extraer conclusiones sobre este.

**OBJETIVOS**

Como objetivos principales de esta práctica encontramos conocer la aplicación de modelos de regresión, sobre todo haciendo énfasis en analizar los atributos para seleccionar los más representativos y normalizarlos, evaluar correctamente el error en el modelo, visualizar los datos y el modelo resultante y saber aplicar el proceso de descenso de gradiente.

Además, también ser capaz de aplicar técnicas de regresión en casos reales, validar los resultados con datos reales, y fomentar la capacidad de presentar resultados técnicos de aprendizaje computacional de forma adecuada delante de otras personas.

**PRESENTACIÓN DE LA BASE DE DATOS**

El caso kaggle que nos ha tocado estudiar y analizar es el siguiente dataset de abalone:

<https://www.kaggle.com/rodolfomendes/abalone-dataset>

Para entender un poco mejor la base de datos explicaremos que es un abalone. El término “abalone” o también conocido como haliótidos es una familia de moluscos gasterópodos conocidos como orejas de mar.

La base de datos contiene diferentes muestras con los siguientes atributos físicos para cada muestra: Sex, Length, Diameter, Height, Whole weight, Shucked weight, Viscera weight, Shell weight, Rings.

Mediante estas muestras deberíamos ser capaces de predecir la edad del abalone.

**APARTADO C**

En este apartado se procede a cargar todos los datos relacionados con la base de datos de nuestro grupo de prácticas, en este caso, nuestra base de datos como se ha comentado anteriormente contiene muestras de moluscos gasterópodos. Mediante la carga de todos estos valores se procede a manipular-los de forma que podamos entender todo lo que significa y a partir de aquí se fijara un atributo objetivo a predecir mediante los otros atributos de la base de datos.

Para realizar esto primeramente se crea un nuevo notebook al que llamaremos Abalone\_Notebook.ipynb, localizado en la carpeta Code de nuestro proyecto en Github.

En nuestro caso, ya teníamos instalados el entorno necesario para ejecutar este notebook, y las librerías a utilizar a procesar estos datos, debido a prácticas realizadas en otras asignaturas. Así, de esta manera comenzamos importando las siguientes librerías:

**Sklearn:** Esta librería sirve para realizar el aprendizaje automático, y nos ayudara a realizar el análisis predictivo. Incluye varios algoritmos de clasificación, regresión y análisis de grupos.

**Numpy:** Esta librería nos da soporte para crear vectores y matrices de una dimensión considerable y multidimensional, está creada precisamente para procesar grandes cantidades de datos de la forma más óptima posible y contiene una larga colección de funciones matemáticas de alto nivel para operar con ellas.

**Pandas:** Esta librería es una extensión de la librería de numpy, hecha para realizar la manipulación y el análisis de datos en python, ofreciéndonos así estructuras de datos y operaciones para manipular tablas numéricas y series temporales, de manera que con esta librería podremos importar de forma sencilla nuestra base de datos.

**Matplotlib:** Esta librería consigue la generación de gráficos a partir de datos contenidos en listas, los cuales habrán sido importados en nuestro caso previamente con uso de las funciones de la librería pandas, de manera que podamos representar los datos deseados de manera gráfica usando varios tipos de técnicas.

**Scipy:** Esta librería nos será útil para aplicar módulos de optimización, algebra lineal, e integración de datos entre otras, cabe destacar que es parte del conjunto de la biblioteca numpy y extiende bibliotecas de computación científica.

Con estas librerías importadas en nuestro notebook, el primer paso es cargar la base de datos mediante un archivo csv, con la función read\_csv de la librería pandas, a la cual le indicaremos por parámetros que nuestros valores están separados por comas en el archivo origen.

Esta carga de la base de datos que se acaba de mencionar sera guardada en una variable llamada dataset, la cual ahora contiene una matriz de datos en la cual las filas corresponden a los valores de la misma base de datos, y las columnas corresponden a cada atributo de la misma.

Para hacer una prueba de que la carga ha funcionado correctamente podemos aplicar la función head() la cual nos devolverá las primeras 5 filas guardadas en nuestro dataset.

Gracias a la aplicación de esta función podemos apreciar los diferentes atributos que contiene nuestra base de datos, los cuales explicamos a continuación así respondiendo a la pregunta también de **Cuál es el tipo de cada atributo**:

**“Sex”**: Se refiere al género que puede tener el molusco en cuestión, cogiendo los valores de M (Para el sexo masculino), F (Para el sexo femenino) y I (Para los infantiles). Como podemos ver el tipo de este atributo es un carácter de tipo letra con longitud 1.

**“Length”**: Se refiere a la medición del caparazón del molusco, y esta medido en milímetros, de manera que su tipo de dato para ser representado es un numero flotante o decimal.

**“Diameter”:** Se refiere al diámetro del caparazón del molusco, o lo que es lo mismo la medida perpendicular a la longitud de su caparazón, este atributo también se expresa en milímetros y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Height”**: Se refiere a la altura total del molusco teniendo en cuenta su cuerpo a parte del caparazón en milímetros, y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Whole Weight”:** Se refiere al peso total de todo el molusco, en gramos, y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Shucked weight”:** Se refiere al peso del molusco sin contar el caparazón, es decir, la carne del mismo, también expresado en gramos y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Viscera weight”**: Se refiere al peso intestinal del molusco, expresado también en gramos y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Shell weight”**: Se refiere al peso del caparazón del molusco, expresado también en gramos y su tipo de dato para ser representado es también un numero flotante o decimal.

**“Rings”:** Anillos que tiene el molusco, representado como un numero entero positivo, cabe destacar que sabiendo el número de anillos, se puede obtener la edad del molusco sumando 1,5 al número de anillos que tiene.

Una vez conocidos los datos con los que trataremos, observaremos como se representan estos en su totalidad y que relación guardan con otros atributos, con el fin de eliminar atributos redundantes y establecer relaciones para predecir una variable objetivo la cual tendremos que especificar también a partir de la representación de estos datos.

Para comenzar a observar cómo se representan los datos, usaremos funciones de la librería pandas primeramente.

El primer paso es saber la dimensión total de los datos obtenidos en nuestra variable dataset el cual podremos saber con la función shape(), así obtenemos que nuestro dataset contiene un total de 4177 filas.

Posteriormente necesitamos saber si existen valores nulos en alguna de las filas de nuestro dataset debido a que estos valores no nos servirán posteriormente para el análisis de datos, así que se hace uso de la función isnull combinada con la función sum() de manera que obtenemos los valores nulos que existen por cada columna de nuestro dataset, dando así como resultado que no contiene valores nulos, de manera que en este apartado no descartaremos datos todavía.

Después de haber realizado estos dos simples pasos, pasamos a utilizar la función describe() la cual nos da datos relevantes sobre cada columna de nuestro dataset, tal como la media de valores, los máximos, los mínimos, los percentiles, etc.

De manera que así podemos observar posibles inconsistencias en los datos para seguir realizando la criba de estos. En este caso vemos que los valores representados con estas funciones parecen correctos a priori, pero nos llama la atención que la altura mínima encontrada es igual a 0, cosa que no tiene sentido, así que se procede a la búsqueda de las filas del dataset que contengan este valor filtrando por el atributo “Height” e igualándolo a 0.

Esta consulta nos devuelve dos filas, las cuales habrá que descartar ya que aportan inconsistencias. Para descartar estas filas hacemos uso de la función drop() a la cual como parámetros le pasaremos en una lista los índices de las filas con altura igual a 0.

Posteriormente se comprueba de nuevo la misma consulta para ver que no encuentra resultados y se comprueba de nuevo la dimensión del dataset para ver que contiene dos filas menos.

Siguiendo estos procedimientos, sabemos que se puede calcular la edad de un molusco sumando 1,5 al número de sus anillos, así que se crea una variable llamada age que sea igual al resultado de esta suma, utilizando la siguiente sentencia:

dataset[‘age’] = dataset.Rings +1.5

Como paso extra, podemos decir que renombramos el nombre de las variables para evitar problemas luego de consultas como pueden ocurrir a partir de espacios entre los nombres, o el uso de mayúsculas. Así que se cambian los espacios por “\_”, y se sustituyen las mayúsculas por minúsculas usando la función rename de la librería pandas pasando por parámetro las columnas con su nombre antiguo y especificando el nuevo.

Para comprobar que estos cambios han sido ejecutados correctamente podremos utilizar de nuevo la función head() para ver mismamente los primeros nuevos valores y así comprobar que los datos han sido actualizados y además aparece el nuevo atributo **“age”**.

2-¿Qué atributos tienen una distribución Guasiana?

3-¿Cuál es el atributo objetivo? ¿Por qué?

**APARTADO B**

**APARTADO A**