Logotipo

Descripción generada automáticamenteInforme de practica de Aprendizaje computacional:

Regresión lineal

Luis Fernando Paz Galeano 1567369

Miguel Del Arco Márquez 1566698

Contenido

[Introducción 3](#_Toc86168377)

[Objetivos 4](#_Toc86168378)

[Apartado C: Análisis de los datos 5](#_Toc86168379)

[Información sobre el dataset 5](#_Toc86168380)

[Carga del Dataset 5](#_Toc86168381)

[Atributos del dataset 5](#_Toc86168382)

[Valores nulos 7](#_Toc86168383)

[Estadística de los datos 9](#_Toc86168384)

[Visualización de los ‘n’ elementos 9](#_Toc86168385)

[Resolución de las preguntas 14](#_Toc86168386)

# Introducción

En el presente informe se detallan los pasos seguidos para el desarrollo de la practica 1 de la asignatura de Aprendizaje computacional, basada en regresión lineal.

La practica consiste en tres aparatados, los cuales se exponen en el informe de la siguiente manera, apartado C ´*Análisis de los datos*´, apartado B ‘*Regresión lineal*’ y apartado A ‘*Gradient descent’*.

Para el apartado C, se ha procedido a analizar la base de datos inicial, con el objetivo de poder entender la distribución, naturaleza y relación de cada uno de los atributos del conjunto de datos, para ello se han utilizado funciones propias de las diferentes librerías de análisis de datos proporcionadas por Python, entre ellas *Matplotlib,Seaborn,Pandas y NumPy.*

Para el apartado B se implemento un modelo de aprendizaje, utilizando, regresión simple y múltiple, así también se hicieron comparativas entre diferentes aplicaciones de técnicas de análisis de datos.

La regresión es una técnica de modelado estadístico que se emplea para describir una variable de respuesta continua como una función de una o varias variables predictoras. Puede ayudar a comprender y predecir el comportamiento de sistemas complejos o a analizar datos experimentales.

El aprendizaje consiste en encontrar cuáles son los mejores parámetros (coeficientes) para los datos que tengamos. Los mejores coeficientes serán los que minimicen alguna medida de error. Para la regresión lineal usaremos el *error cuadrático medio.*

Finalmente, en el apartado ‘A’ se implementó el descenso de gradiente, que es un algoritmo de optimización iterativo de primer orden para encontrar un mínimo local de una función diferenciable, el objetivo de usar dicho algoritmo es con el fin de minimizar la función de coste, y así encontrar el modelo optimo como solución al problema.

# Objetivos

* Familiarizarse con el uso de las principales librerías de Python para *Machine learning.*
* Aprender a analizar un conjunto de datos.
* Aprender a identificar la relación entre atributos de un conjunto de datos.
* Asimilar conceptos teóricos por medio de su implementación.
* Desarrollar modelos básicos de entrenamiento.
* Desarrollar modelos con regresión lineal simple y múltiple.
* Resolución de conflictos e incongruencia en el tratamiento de datos.

# Apartado C: Análisis de los datos

## Información sobre el dataset

El dataset que se nos fue asignado fue *‘*[*Deodorant Instant Liking Data’*](https://www.kaggle.com/ramkumarr02/deodorant-instant-liking-data), que consiste en un dataset que contiene una serie de encuesta, sobre un conjunto de productos, en este caso desodorantes, los cuales constan de varios atributos calificativos, y donde para cada uno de los productos se recoge una repuesta por parte de un entrevistado.

En este apartado hemos analizado el contenido de los datos con los que tratamos, esto con el objetivo de tener una primera percepción sobre la composición de los diferente atributos o elementos que conforman el conjunto de datos, así también poder identificar posible irregularidades o elementos nulos, esto para organizar de mejor manera la información que disponemos.

Para ello pasamos ahora a explicar como se han analizado los datos, antes que nada, deberemos de cargar los datos, para poder trabajar con ellos, así también se muestran la función utilizada para cargar los datos.

### Carga del Dataset

El primer paso es cargar nuestro dataset, ello la hacemos con la ayuda de la función *‘load\_dataset’* , que carga el contenido de nuestra base de datos, esto lo hacemos con la librería panda y la función ‘*read\_csv’*, y luego indicamos la ruta donde se encuentra el archivo.

|  |
| --- |
| def load\_dataset(path):      dataset = pd.read\_csv(path, header=0, delimiter=',')      return dataset |

### Atributos del dataset

Una vez que hemos cargado nuestro dataset, ya estamos listos para analizar el contenido del mismo, para ello nos auxiliamos del función *‘.info()*’, la cual nos devuelve un listado de las columnas , es decir cada uno de los diferente atributos del conjunto de datos, a su vez también nos indica el tipo de dato de cada uno de los elementos, así como el espacio que este ocupa en memoria.

|  |
| --- |
| Texto  Descripción generada automáticamente  Fig. 1 Ejemplo de salida 1 |

Como podemos observar en el caso anterior disponemos de un total de 53 columnas, es decir un total de 53 atributos, de los cuales 2 son elementos de tipo flotante y 52 de tipo integer.

### Valores nulos

Ahora procederemos a limpiar aquellos atributos que nos pueden generar problemas, es decir aquellos atributos que son nulos, y por lo no nos aportan ninguna información útil, para ello lo podemos hacer uso de la siguiente instrucción que nos devuelve una lista con el número de elementos vacíos por atributo.

|  |
| --- |
| Fig.2 Lista de elementos nulos |

Para tener una mejor percepción de lo elementos nulos, creamos una gráfica, desde donde tendremos una visión más global.

Después de ello lo que procederemos a hacer es a eliminar aquellos atributos que contengan elementos nulos o bien estos vacíos, sin información que nos sea útil.

Para ello implementamos la siguiente función.

|  |
| --- |
| Def columns\_with\_na(dframe):      temp = dframe.isna().sum()      temp = temp[temp >0]      print(f”Columns containing nan values:{temp.index}”)      return temp.index |

Esta función nos devuelve una lista con el índice de aquella columnas que contienen elementos vacíos, luego con la ayuda de la función ‘*.drop()’*, pasamos la lista de índices para eliminarlos del dataset.

|  |
| --- |
| Fig. 3 Visualización de elementos nulos 1 |

|  |
| --- |
| Imagen que contiene Gráfico  Descripción generada automáticamente  Fig. 4 Elementos no nulos 1 |

### Estadística de los datos

Podemos ver fácilmente una estadística de los datos para ellos simplemente hacemos un *‘dataset.describe()’* esto puede ser útil para analizar el comportamiento de los datos.

|  |
| --- |
| Fig. 5 Estadística de los datos |

### Visualización de los ‘n’ elementos

|  |
| --- |
| Fig. 6 Muestra de n filas 1 |

Correlación de los datos: Matriz de confusión

La matriz de confusión es una herramienta muy útil para valorar cómo de bueno es un modelo clasificación basado en aprendizaje automático. En particular, sirve para mostrar de forma explícita cuándo una clase es confundida con otra, lo cual nos, permite trabajar de forma separada con distintos tipos de error.

Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

En nuestro caso obtenemos la siguiente matriz de confusión, por temas de simplicidad mostramos solo los primeros 20 atributos de los 53 disponibles, ya que, si bien es necesario comprobar cada una de las posibles correlaciones, por la naturaleza de los datos de nuestro dataset, la correlación existente entre cada uno de los elementos es muy baja.

|  |
| --- |
| Fig 7 Matriz de confusión |

Como podemos observar en la matriz de confusión anterior, la correlación entre la mayoría de los atributos es muy baja, a excepción de dos atributos que son *‘Instant\_Liking’* y *‘q1\_1.personal.opinion.of.this.Deodorant’.*

Como vemos existe una correlación negativa entre ambos atributos, por lo cual hemos decidido utilizarlos como datos de entrada y salida de nuestro modelo de regresión lineal.

Hay que mencionar que los datos contenidos en *‘Instant\_Liking’* son categóricos, es decir, guardan una repuesta de Si o No, representado por 0 y 1.

#### Representación de Instant Liking en base a producto

|  |
| --- |
| Fig. 8 Instant Liking |

*Numero de Entrevistados por cada producto*

|  |
| --- |
| Fig. 9 Entrevistado por producto |

#### Representación de media de respuestas por producto.

Hemos calculado la media de las respuestas de cada uno de los atributos, si bien este valor no resulta determinante para determinar o entrenar algún modelo, a nivel conceptual nos ayuda a hacernos una idea de las repuestas medias de cada uno de los productos, y así determinar, por ejemplo, cuál de los productos tiene una cualificación mayor, que como recordemos, no tiene que ser precisamente el producto mejor valorado, ya que los atributos son muy variados.

|  |
| --- |
| Fig. 10 Media de respuesta por productos |

*Histograma de todos los elementos*

Podemos visualizar el rango de cada uno de los atributos, estos nos ayudan a entender mejor el comportamiento de los datos del dataset.

|  |
| --- |
| Fig. 11 Histograma de cada atributo |

## Resolución de las preguntas

**¿Cuál es el tipo de cada atributo?**

Para determinar el tipo de atributo, podemos hacer uso de la función dataset.info () que nos muestra el tipo de valor de cada uno de los elementos de los que está compuesto nuestro dataset.

Por ejemplo, en nuestro caso, al ejecutar dicho comando nos devuelve una lista de cada una de las columnas, y al final del todo nos proporciona la siguiente información:

dtypes: float64(10), int64(53), object(1)

**¿Qué atributos tienen una distribución Gaussiana?**

En nuestro caso para poder determinar que atributo tenía dicha distribución, hemos hecho uso de la comanda dataset.hist(figsize= (20,20)), esto nos devuelve un histograma para cada uno de los atributos del dataset, y hemos identificado que el atributo q7 y la media tienen una distribución Gaussiana.

**¿Cuál es el atributo objetivo? ¿Por qué?**

Después del análisis de los datos que disponemos, llegamos a la conclusión que lo ideal sería predecir el tipo de producto en base al conjunto de 'Xs' de entrada, ya que en este caso disponemos de un sin número de entradas con valores calificativos de diferentes características del producto.

# Apartado B: Regresión lineal

## Regresión lineal simple

En el apartado C, después de analizar los datos, nos dimos cuenta de una correlación negativa entre los atributos ‘Instant\_Liking’ y ‘q1\_1.personal.opinion.of.this.Deodorant', por lo cual hemos entrenado un modelo de regresión lineal simple, teniendo como entrada ‘Instant\_Liking’ y elemento de salida ‘q1\_1.personal.opinion.of.this.Deodorant'.

Queremos mencionar que, por la relación y distribución de los datos, para este dataset, la regresión lineal, no es la mas apropiada, ya que lo ideal sería implementar una clasificación en su lugar, por la manera en que está planteada la base de datos, es decir contamos con un conjunto de valores que son categóricos, calificativos y que en su mayoría son subjetivos.

|  |
| --- |
| Fig. 12 Relacion de Instant\_Liking y personal.opinion.of.this.Deodorant' |

Hemos separado los datos en entramiento y prueba, decidimos designar un 20% para la parte de pruebas, así también se definió el tipo de algoritmo como *linearRegresion.*

|  |
| --- |
| X\_train,X\_test,y\_train,y\_test = train\_test\_split(X\_train,y\_train,test\_size=0.2,random\_state=15)  #definimos el algoritmo a utilizar  lr = linear\_model.LinearRegression()  #Entreno de modelo  lr.fit(X\_train, y\_train)  #Realizar una prediccion  Y\_pred = lr.predict(X\_test) |

El resultado obtenido es el siguiente:

|  |
| --- |
| Fig. 13 Regresion Lineal Simple |

### Error cuadrático medio y r2

El coeficiente R2 determina la capacidad de un modelo para predecir futuros resultados. El mejor resultado posible es 1.0, y ocurre cuando la predicción coincide con los valores de la variable objetivo.

Por otro lado, el Error cuadrático medio es el estimador que mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima.

Las mediciones sobre el Error cuadrático medio y el porcentaje de r2 obtenidos fueron los siguientes