MOBILE PRICE CLASSIFICATION



**Proyecto final de Aprendizaje Automático y Minería de Datos**

Presentado por

**Sergio Gavilán Fernández**  [sgavil01@ucm.es](mailto:sgavil01@ucm.es)

**Alejandro Villar Rubio** [alvill04@ucm.es](mailto:alvill04@ucm.es)

Facultad de Informática, Universidad Complutense de Madrid

Grado en Desarrollo de Videojuegos

Madrid, 2019/2020

Contenido

[Resumen 2](#_Toc28004352)

[Objetivos 2](#_Toc28004353)

[Datos usados 2](#_Toc28004354)

[Regresión Logística 4](#_Toc28004355)

[Redes Neuronales 4](#_Toc28004356)

[Conclusiones 4](#_Toc28004357)

# Resumen

La elección de un teléfono móvil puede ser en algunas ocasiones un quebradero de cabeza para algunas personas, ya sea por su edad o simplemente porque no conocen la industria lo suficiente. Además, a este desconocimiento hay que añadir que cada día hay más dispositivos en el mercado por lo que la dificultad de su compra aumenta. Esto puede provocar la llamada **paradoja de la elección** (“La paradoja de la elección: por qué nos cuesta decidir,” n.d.), nuestra tendencia a estar menos satisfechos con nuestras adquisiciones mientras más alternativas existan.

En este proyecto se planteará una herramienta con la cuál cualquier persona podrá crear un teléfono móvil a su gusto y comprobar en qué rango de precio se encuentra. Con esto, el usuario podrá orientarse para su posterior compra.

# Objetivos

El objetivo principal de este proyecto es, como se ha nombrado anteriormente, proporcionar una herramienta con la que poder comprobar el rango de precio de un determinado teléfono móvil.

Para ello se aplicarán diversas técnicas de clasificación sobre la base de datos que se dispone y realizar una selección para usar la que proporcione una mayor precisión.

Las técnicas que se usarán son:

* Regresión logística.
* Redes neuronales.
* Support Vector Machines (SVM)

# Datos usados

Se dispone de una base de datos adquirida de la plataforma *Kaggle* (“Mobile Price Classification | Kaggle,” n.d.)*.* Esta proporciona dos archivos, **test.csv** y **train.csv**, con un total de 21 características que conforman un teléfono móvil.

|  |  |  |
| --- | --- | --- |
| Capacidad batería | Bluetooth | Velocidad del procesador |
| Dual Sim | Resolución cámara frontal | 4G |
| Memoria interna | Profundidad del teléfono móvil (cm) | Peso |
| Número de núcleos del procesador | Resolución de la cámara principal | Resolución de la pantalla, altura |
| Resolución de la pantalla, anchura | RAM | Altura |
| Anchura | Autonomía de la batería | 3G |
| Pantalla táctil | Wifi | Rango de precio / ID |

Hay una pequeña diferencia entre estos dos archivos, ambos tienen el mismo número de características, pero *train* tiene una columna con “price\_range” y *test* una con “id”. Posiblemente este último no tiene la información del precio porque en alguna ocasión se ha usado este conjunto de datos para alguna competición de **machine learning**. Esto hace que solo se pueda usar el archivo *train* porque es el que contiene lo que será la futura “Y”.

El conjunto de datos *train* está formado por 21 características y 2000 ejemplos. Esto hace que se pueda dividir en 3 tipos de conjuntos y aun así tener un gran número de ejemplos en cada uno. Las diferentes divisiones son: **train** (60%, 1200 ejemplos)**, validation** (20%, 400 ejemplos)y **test** (20%, 400 ejemplos). Esto se ha realizado a través de:

from pandas.io.parsers import read\_csv

import csv

import numpy as np

def carga\_csv(file\_name):

    valores = read\_csv(file\_name, header=None).values

    # suponemos que siempre trabajaremos con float

    return valores[1:]

def main():

    datos = carga\_csv("train.csv")

    # Eliminamos la primera fila con los atributos del dataset

    datosLen = len(datos)  # 2001 casos de entrenamiento

    # Cogemos el 60% de los datos set de entrenamiento

    n\_training = int(datosLen \* 0.6)

    training\_set = datos[:n\_training]

    # Por otro lado el 20% para el set de validacion

    n\_validation = int(datosLen \* 0.2)

    validation\_set = datos[n\_training:n\_training+n\_validation]

    # Por ultimo el 20% restante para el conjunto de prueba

    n\_test = datosLen - n\_training - n\_validation

    test\_set = datos[-n\_test:]

    with open('ProcessedDataSet/train.csv', mode='w', newline='') as processedTraining:

        processedTrainingWriter = csv.writer(

            processedTraining, delimiter=',')

        processedTrainingWriter.writerows(training\_set)

    with open('ProcessedDataSet/validation.csv', mode='w', newline='') as processedValidation:

        processedValidationWriter = csv.writer(processedValidation, delimiter=',')

        processedValidationWriter.writerows(validation\_set)

    with open('ProcessedDataSet/test.csv', mode='w', newline='') as processedTest:

        processedTestWriter = csv.writer(processedTest, delimiter=',')

        processedTestWriter.writerows(test\_set)

main()

# 

# Regresión Logística

# Redes Neuronales

# Support Vector Machines

# Conclusiones

# Bibliografía

La paradoja de la elección: por qué nos cuesta decidir. (n.d.). Retrieved December 24, 2019, from https://hipertextual.com/2015/07/paradoja-eleccion

Mobile Price Classification | Kaggle. (n.d.). Retrieved December 24, 2019, from https://www.kaggle.com/iabhishekofficial/mobile-price-classification