Clasificación de celulares

Natalia Guevara Guevara[1], Sebastian De La Cruz Gutierrez[2]
[1]guevara.pnatalia@javeriana.edu.co [2]delacruzsebastian@javeriana.edu.co
Inteligencia artificial - Pontificia Universidad Javeriana
2021-I
Fecha de Entrega : Junio 4 de 2020

***Index Terms*—Python, maching learning, pandas, sklearn, KNN, SVM, regresion logistica.**

I. Introducción

Actualmente en nuestra sociedad es indispensable el uso del celular y para muchas personas son importantes las características y los precios de los celulares. Debido a esto se realizara un clasificador de celulares por precio. Esto se hará comparando tres diferentes clasificadores sobre un dataset que describe celulares a partir de 21 características.

II. OBJETIVOS

OBJETIVO GENERAL:

- Clasificar de celulares por precio, a partir de 21 diferentes características.

OBJETIVOS ESPECÍFICOS:

- Evaluar los clasificadores a partir de la metodología vista.
- Entrenar a partir de histogramas bidireccionales.

III. DESARROLLO Y RESULTADOS

El dataset se compone de 21 columnas y 2000 filas. 20 de estas columnas representan las características.

Se organizo el proceso de desarrollo del proyecto en 5 partes; la primera parte fue el proceso de descripción del dataset lo que nos ayudo a comprender de una mejor manera los datos, las partes 2,3 y 4 fueron de aplicación de los modelos de clasificación, y por ultimo se realizo PCA aplicando un modelo de clasificación con los datos ajustados por PCA.

A continuación se explicará con mas detalle cada una de las partes del desarrollo del proyecto.

III-A. Primera parte: Descripción del Dataset

En esta primera parte, se analizaron los datos diferenciando las características categóricas de las características continuas.

En el siguiente histograma podemos observar las 4 clases que tenemos:

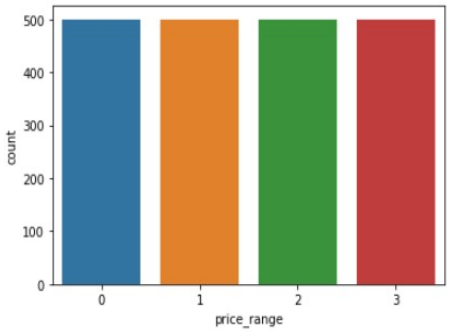

Figura 1. Histograma de clases.

III-A1. Características categóricas: Se gráfico el histograma de cada una de las características categóricas así como se puede observar en las siguientes imágenes.

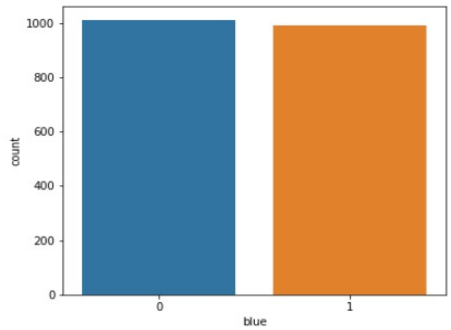

Figura 2. Histograma Blue.

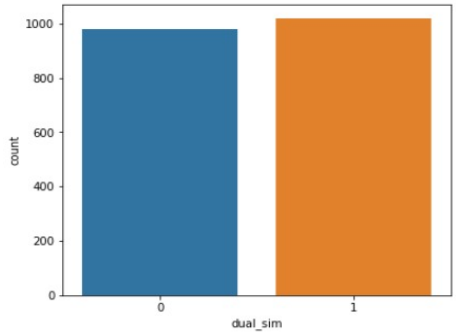

Figura 3. Histograma DualSim.

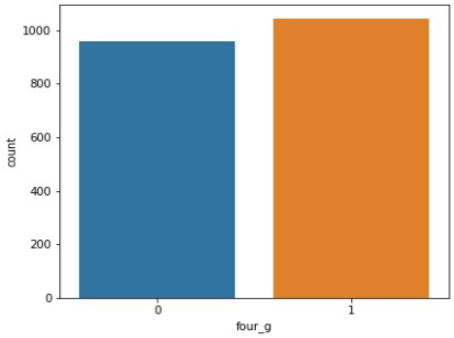

Figura 4. Histograma FourG.

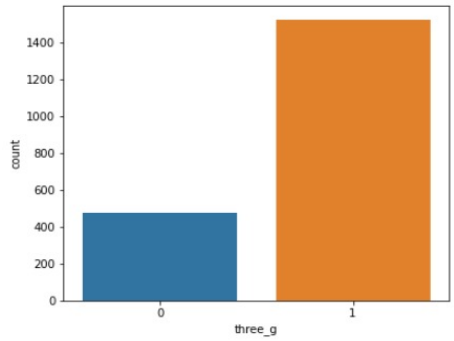

Figura 5. Histograma ThreeG.

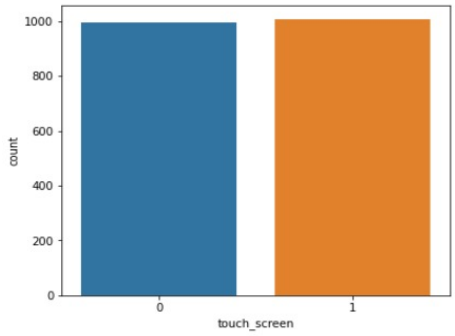

Figura 6. Histograma touchScreen.

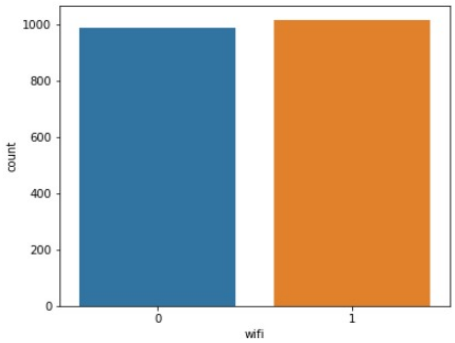

Figura 7. Histograma Wifi.

III-A2. Características continuas: Se gráfico el histograma de cada una de las características continuas así como se puede observar en las siguientes imágenes.

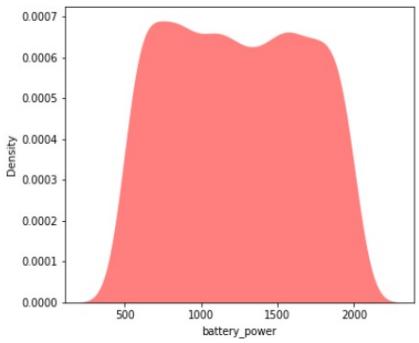

Figura 8. Histograma BatteryPower.

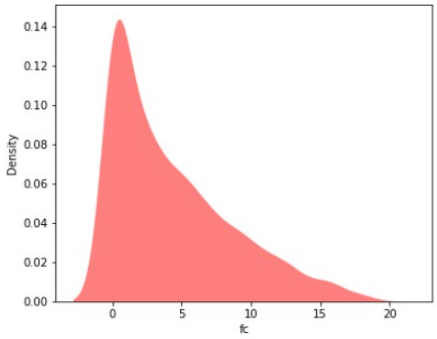

Figura 9. Histograma Fc.

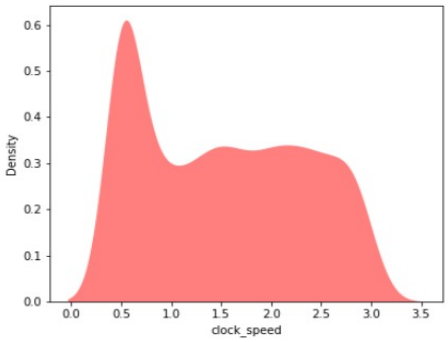

Figura 10. Histograma ClockSpeed.

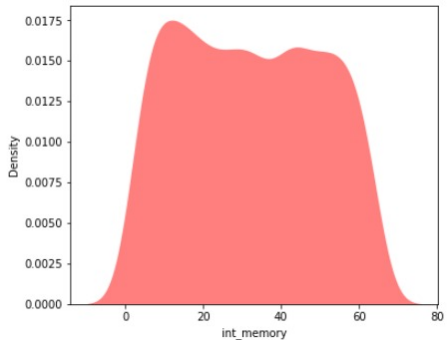

Figura 11. Histograma IntMemory.

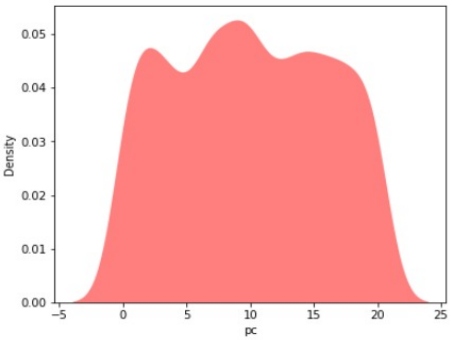

Figura 15. Histograma Pc.

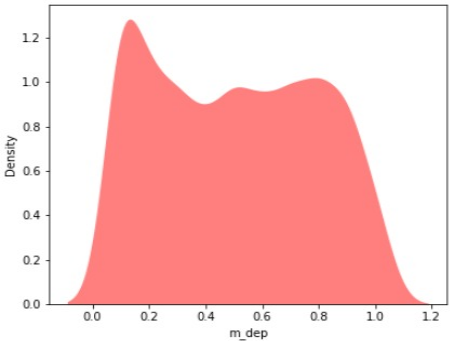

Figura 12. Histograma MDep.

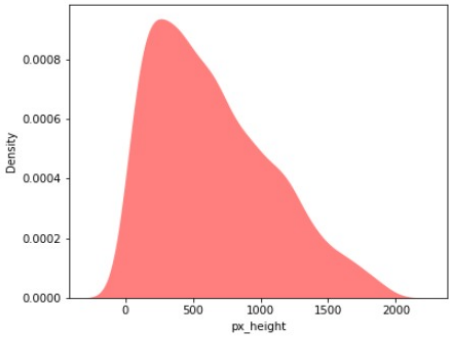

Figura 16. Histograma PxHeight.

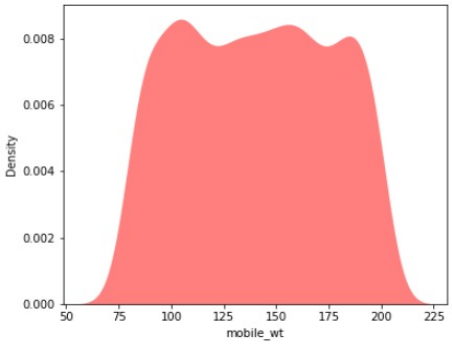

Figura 13. Histograma MobileWt.

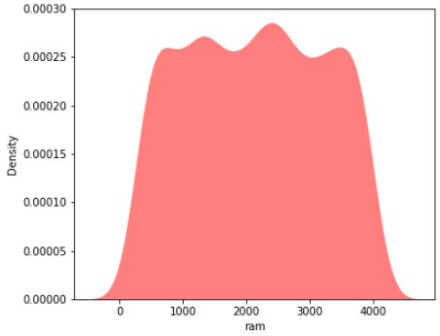

Figura 17. Histograma Ram.

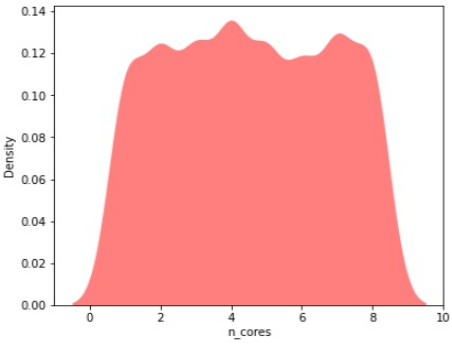

Figura 14. Histograma NCores.

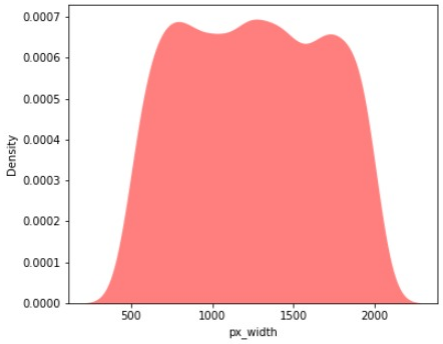

Figura 18. Histograma PxWidth.

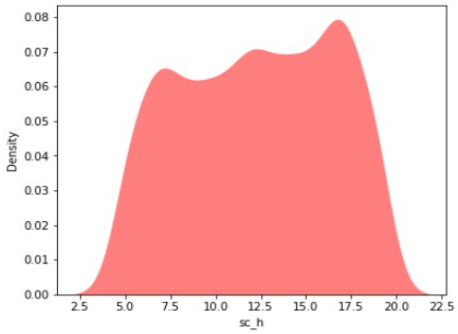

Figura 19. Histograma ScH.

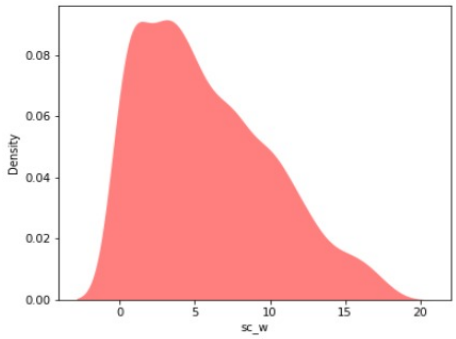

Figura 20. Histograma ScW.

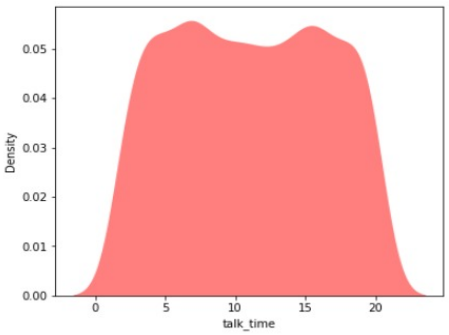

Figura 21. Histograma TalkTime.

III-A3. Analisis 3D: Para continuar comprendiendo el dataset se graficaron una característica categórica vs una característica continua y viendo al mismo tiempo a cuales de nuestras clases pertenecen como se muestra a continuación.

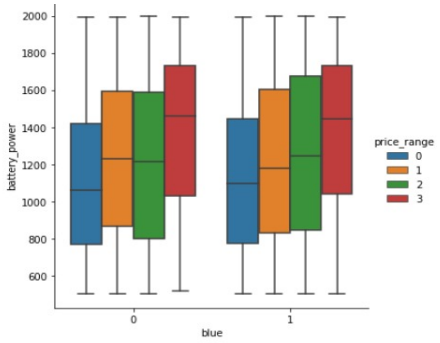

Figura 22. Histograma BatteryPower vs Blue vs PriceRange.

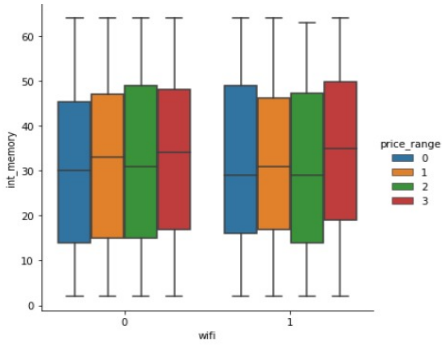

Figura 23. Histograma IntMemory vs Wifi vs PriceRange.

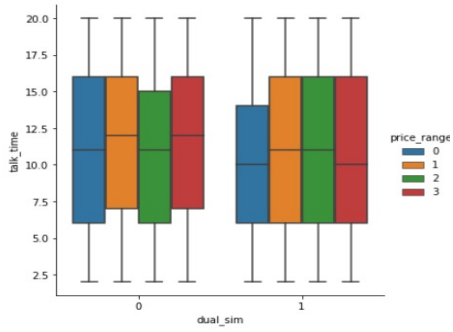

Figura 24. Histograma TalkTime vs DualSim vs PriceRange.

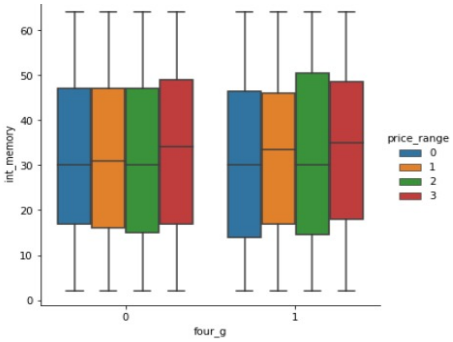

Figura 25. Histograma IntMemory vs FourG vs PriceRange.

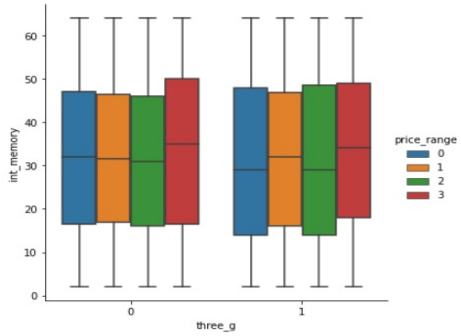

Figura 26. Histograma IntMemory vs ThreeG vs PriceRange.

III-B. Segunda parte: K Nearest Neighbors Classifier

III-B1. Escoger k: Para escoger el mejor k graficamos los resultados de la clasificación haciendo un barrido de los k de lo que obtuvimos lo siguiente.

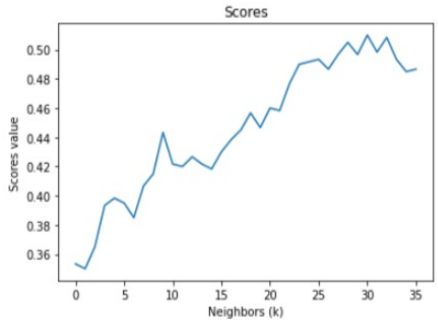

Figura 27. Scores KNN.

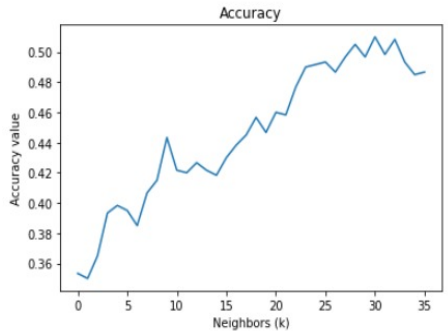

Figura 28. Accuarcy KNN.

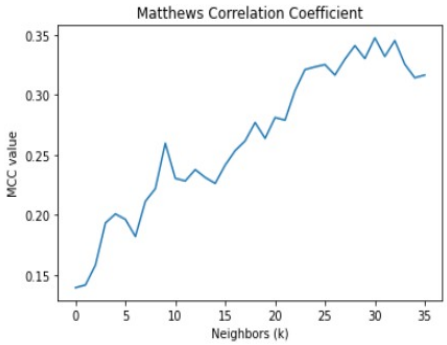

Figura 29. Coeficiente de correlación de Matthews KNN.

Con la ayuda de estas gráficas y de los resultados hallamos el mejor k posible.

```
Best k by Scores: 31 . Value: 0.51
Best k by Accuracy: 31 . Value: 0.51
Best k by MCC: 31 . Value: 0.3475966786667766
```

Figura 30. Mejor K.

III-B2. Clasificación: Luego de encontrar el mejor k posible desarrollamos el clasificador y obtuvimos los siguientes resultados.

```
Train subset
Accuracy of K-NN classifier on training set: 0.58429
MCC of K-NN classifier on training set: 0.44627

Test subset
Accuracy of K-NN classifier on test set: 0.51000
MCC of K-NN classifier on test set: 0.34760

Classification time: 0.05736 s
```

Figura 31. Resultados KNN.

III-C. Tercera parte: Logistic Regression Classifier

Desarrollamos el clasificador por regresión logística y obtuvimos los siguientes resultados

```
Train subset
Accuracy of LR classifier on training set: 0.80429
MCC of LR classifier on training set: 0.74333

Test subset
Accuracy of LR classifier on test set: 0.77500
MCC of LR classifier on test set: 0.70390

Classification time: 0.00097 s
```

Figura 32. Resultados Regresión Logística.

III-D. Cuarta parte: Kernel-SVM classifier

Hicimos bastantes pruebas en la aplicación de este clasificador a nuestro problema, cambiando el kernel luego de estas pruebas encontramos que el kernel polinómico de orden 3 es el que mejor se ajusta a nuestro problema dando los siguientes resultados.

```
Train subset
Accuracy of SVM classifier on training set: 0.99357
MCC of SVM classifier on training set: 0.99143

Test subset
Accuracy of SVM classifier on test set: 0.87833
MCC of SVM classifier on test set: 0.83820

Classification time: 0.01265 s
```

Figura 33. Resultados SVM.

III-E. Quinta parte: PCA

Se realizo el ajuste de los datos con PCA, y luego de esto con los datos ajustados con PCA, usamos el clasificador por kernel-SVM ya que este fue el que mejores resultados nos había dado.

```
Train subset with PCA
Accuracy of SVM classifier with PCA on training set: 0.72933
MCC of SVM classifier with PCA on training set: 0.64611

Test subset with PCA
Accuracy of SVM classifier with PCA on test set: 0.72400
MCC of SVM classifier with PCA on test set: 0.64302

Classification time: 0.00663 s
```

Figura 34. Resultados SVM con PCA.

IV. GITHUB

El proceso de como se realizo el desarrollo de este proyecto se encuentra en:

https://github.com/delacruzsebastian-cpu/Proyect-Cell-phone-classification-ML

V. Conclusiones

- Se marca una diferencia importante en los clasificadores con ayuda del coeficiente de correlación de Matthews.
- En nuestro caso el metodo de clasificacion que mejor se comporto fue el de SVM (Maquina de soporte vectorial).
- El metodo que menos precision tuvo para nuestro caso fue el de KNN.
- KNN clasifica a partir de la distribución de los datos del dataset, calculando la distancia de las K muestras mas cercanas a la muestra que se quiere clasificar. Este proceso permite que el clasificador no dependa de un umbral sino de la selección de un buen k, esto hace que sea mas costoso computacionalmente.
- La matriz de confusión y el accuracy son de gran ayuda para evaluar el modelo

VI. Referencias

[1] Apuntes de la clase de inteligencia artificial

[2] [Online]. Available: https://www.kaggle.com/iabhishekofficial/mobile-price-classification