Phone Price Range Estimation using different Machine learning methods

Jorge Eduardo Hernández Pérez

Tecnológico de Monterrey Campus Querétaro

November 17, 2022

Abstract

The main purpose of the following paper is to document and explain the Python code developed for the Intelligent Systems course so anyone old or new can understand what is happening and if they wish to modify or use it in future ventures. There are 3 main Machine learning algorithms deployed in said code, linear regression, decision trees and K-mean clusters which along with the dataset can make really accurate predictions of the price range of a cellphone with some defining characteristics.

Disclaimer. The Dataset isn´t really all that new so some characteristics may have not kept up to date with modern technology and thus the acceptable operational range is declared before doing any prediction.

Introduction

Technology is always changing, always adapting itself to the needs of the user, and as a future engineer often the question is asked, ¿How can I value what I do? ¿How much is it worth? And the easy answer is as much as you want to charge, however this decision may not make you competitive in the market, and it may lead to your downfall.

That's why we need to rely on statistics, observing the world logically through math may present the solution to problems we did not know could be solved this way, and almost everything can be simplified to raw data, where mathematics do not lie.

So, in this case we are going to be simplifying phones to their main components and characteristics and with the help of statistics observe which of these aspects have the biggest repercussion/relation with the price of mobile phones.

However, statistics are based on probability and probability is never 100% certain and therefore we need to tackle the problem from different angles so we can have the best chance of having the best probable output.

Dataset

The dataset was obtained from Kaggle and was provided by Abhishek Sharma, owner of this dataset, it contains 21 columns with more than 1800 instances (the train dataset), each column represents a characteristic of the phone in the dataset, going all the way from battery power all the way to 4G compatibility.

In the following image all characteristics can be seen:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  2000 non-null   int64
 1   blue           2000 non-null   int64
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64
 4   fc             2000 non-null   int64
 5   four_g         2000 non-null   int64
 6   int_memory     2000 non-null   int64
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64
 9   n_cores        2000 non-null   int64
 10  pc             2000 non-null   int64
 11  px_height      2000 non-null   int64
 12  px_width       2000 non-null   int64
 13  ram            2000 non-null   int64
 14  sc_h           2000 non-null   int64
 15  sc_w           2000 non-null   int64
 16  talk_time      2000 non-null   int64
 17  three_g        2000 non-null   int64
 18  touch_screen   2000 non-null   int64
 19  wifi           2000 non-null   int64
 20  price_range    2000 non-null   int64
```

Fig.

Into the Code

The code of the program is broken into 5 different sections, the first section "*Setup*" is dedicated to importing all the libraries we may need, and to make a direct connection amongst Google Drive and Google Colab, as we are going to develop our program in said software and keeping the dataset in Drive.

The second section "*Understanding the data*" is dedicated to importing the dataset and breaking it down so it is more manageable and more friendly to our processor and to us. After importing the dataset, we observe its classifications and check for null values.

Once we have established there are no null values, we also need to eliminate impossibilities in the data, such as negative or equal to 0 physical dimensions. Once we have debugged our data we need to see what we have otherwise we may as well continue with a blind over our eyes, as the data may not be well appreciated.

For this we plotted a correlation matrix which is going to tell us which values have a strong correlation with our desired output.

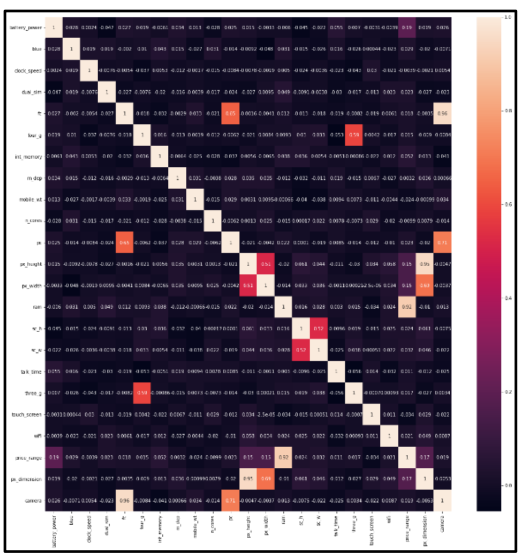


Fig.

Analyzing the matrix, we can observe that the biggest relation with the price range is the ram characteristic, and non-surprisingly other values with strong correlation are those of the dimensions, the 3G with 4G and the Pixels of the front and primary camera.

To make our dataset smaller for making predictions and making our software more efficient we are going to choose the main characteristics and drop those columns we really don't have a use for. (RAM, Pixel, dimensions, battery Power, internal memory and Camera).

For this we are going to combine the pixel Dimensions and camera values as they had a strong correlation, and after that we are going to be dropping all the other characteristics, so our new dataset classifications look something like this:

```
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  1819 non-null   int64
 1   int_memory     1819 non-null   int64
 2   ram            1819 non-null   int64
 3   price_range    1819 non-null   int64
 4   px_dimension   1819 non-null   int64
 5   camera         1819 non-null   int64
```

Fig.

Once done this we need to break the dataset into our desired output and the characteristics which are going to lead us there. After doing so the final step of this section is training the model so we can start doing some machine learning.

Different Models with different Success Rates

The third Section "*Decision trees*" explores the dataset applying the machine learning algorithm of decision trees to make the prediction we want the software to make.

However, ¿what are decision trees? According to Sickit "Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation." (Sickit, n.d.)

Or in mortal words is a mathematical procedure which with the aid of a dataset can make specific predictions based on the information previously gathered.

Proceeding with the code we use already established frameworks to build the decision tree and find its score.

```
y_pred = clf.predict(x_test)
print("Score of Decision Tree: {}".format(accuracy_score(y_test, y_pred)))

Score of Decision Tree: 0.8598901098901099
```

Fig.

As we can see from the previous figure our decision tree has a pretty decent score, being 1 the highest achievable.

, we allowed our tree to have a max depth of 10 as the bigger the depth the more accurate it will be but it is going to exponentially consume more computing power.

The decision tree can be observed in the following link:

https://colab.research.google.com/drive/1h9yRgVgU8zDNu4kdQJSjcueGo5LSv84x#scrollTo=rEvY_wUWL8a6&line=1&uniqifier=1

The 4th section of our code, the "*Clustering*" section deals with K-mean clustering, because between DB scanning and K-mean, I realized that K-mean was the way to go. In here we applied the PCA method so we could reduce the complexity of the dataset. After doing this and plotting our dataset looks something like this:

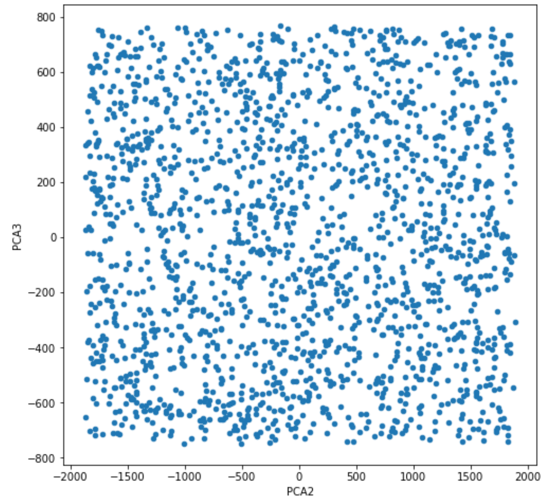


Fig.

So as we know that we have 4 classifications we establish 4 clusters in which the data is going to be classified, giving us the following array of information.

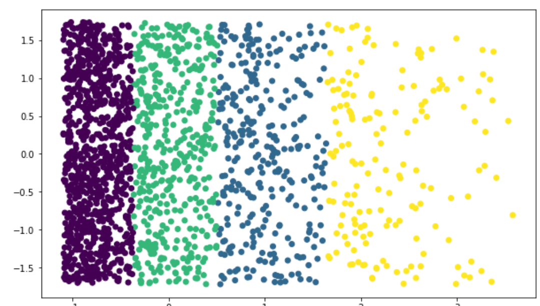


Fig.

With this data organized and our previous entry we can make a prediction to indicate us according to K-mean clustering, to which price range our cellphone belongs to.

The final section of the code "*Linear Regression*" is the simplest of the machine learning algorithms previously studied.

As it is a simpler model we need to only use the main characteristic which has the strongest relation to our price range, however because of our data analysis we know that the characteristic with the strongest relation is the RAM capacity and thus plotting these 2 values we obtain the following graph.

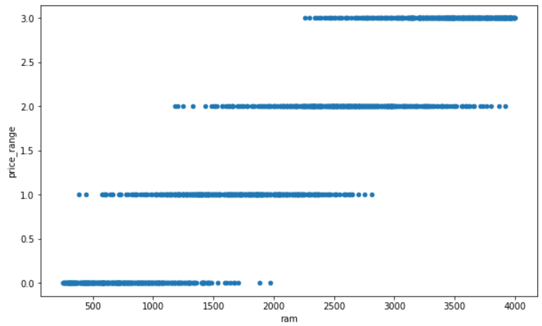
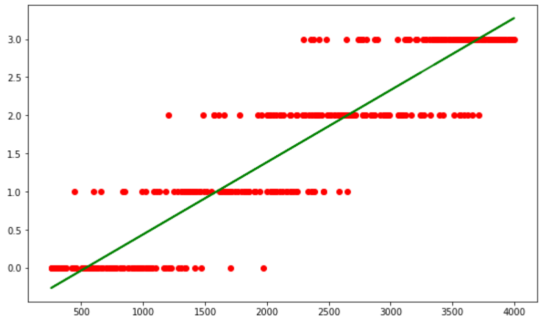


Fig.

At first glance we can very clearly see the classification, however, we still need to apply the framework to determine the linear regression capable of predicting the price range.

After applying the framework, we are left with the following graph:



The line will help us make predictions directly with the RAM characteristic.

However, our classifications are whole values and no fractions, so this Machine learning algorithm indicates us something that the 2 others didn't, and it is how solely considering a characteristic, it gives us more dimensions in which our phone can be valued, its not only black and white, but there is some gray in that mix.

Conclusions

Machine learning has as many uses as numbers in the universe and thus I think its really important to give us a chance to explore it, as even if we are not going to be able to fully comprehend it, we will at least gain something from it.

It is in the engineering nature to explore multiple possibilities, and I hope that this project is an example of that, as many machine learning algorithms were applied with the objective of making certain we achieved the same result.

And in the end we didn't quite got the same result in the 3 methods, each have its strengths and weaknesses, but as everything in life is by complementing our strengths that we can get to a more accurate solution.

This was one of the most difficult projects I have ever done and thus I think it is one of which I have learnt the most.

Bibliography

- 1.10. Decision Trees. (n.d.). scikit-learn. https://scikit-learn.org/stable/modules/tree.html
-