

Price range forecasting of mobile phones

BENCHAREF OMAR (1) , ZARBAG MOHAMED ELMEHDI (2)

Department of Computer Sciences.Faculty of Sciences and Techniques.Cadi Ayyad University, Marrakesh. Morocco.

o.bencharef@uca.ma (1) , mohamedelmehdi.zarbag@edu.uca.ma (2)

Abstract

In this paper we examine the price range of mobile phones using multiple specifications and characteristics, using Random Forest classification & KNeighbors classification due the fact that there are four price range (0,1,2,3), we also used linear regression to analyze the price range, at the end we use the cross validation system & the stacking classification to combine the two classification systems to bring more value to our model .

I. INTRODUCTION

Mobile phones market has grown in the last twelve years, And the challenge between big companies like Apple, Samsung has increased and involved other Chinese companies.

Companies estimate price of the created mobiles in this difficult mobile phone market by not assuming things. but solving this problem needs sales data of mobile phones of various companies to be collected and examined.

After collecting data, companies study some relations between features of a mobile phone (RAM, Internal Memory, Camera) and its selling price. And combine those results with the original product value form the factory

Sometimes just predicting the price range can help companies define the necessity of making a mobile with certain specifications, or forecasting its success in the market.

II. DATA DESCRIPTION & VALUE

Data contain specifications about mobile phones, gathering most of phones characteristics.

Subject Area	Mobile phones
Type of Data	Numerical values
Data format	Excel sheet
Data accessibility	Within this article
Data source	Kaggle

This the list of the multiple variables that our dataset includes:

battery_power	Total energy a battery can store in one time measured in mAh
blue	Has bluetooth or not
clock_speed	speed at which microprocessor executes instructions
dual_sim	Has dual sim support or not
fc	Front Camera mega pixels
four_g	Has 4G or not
int_memory	Internal Memory in Gigabytes
m_dep	Mobile Depth in cm
mobile_wt	Weight of mobile phone
n_cores	Number of cores of processor
pc	Primary Camera mega pixels
px_height	Pixel Resolution Height
px_width	Pixel Resolution Width
ram	Random Access Memory in Megabytes
sc_h	Screen Height of mobile in cm
sc_w	Screen Width of mobile in cm
talk_time	longest time that a single battery charge will last when you are
three_g	Has 3G or not
touch_screen	Has touch screen or not
wifi	Has wifi or not

The output target of our dataset define the range or the level of the price of the mobile phone. There is four different range defined by a numeric number, the next graph shows the distribution of the price range in our dataset.

la distribution par cas:

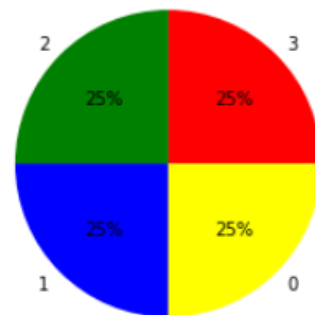


Figure 1 Distribution of Price range

Our dataset is distributed equally over the four type of price range.

As the development of mobile networks extremely increased we need to examine what are the major networks used by those phones in terms of support

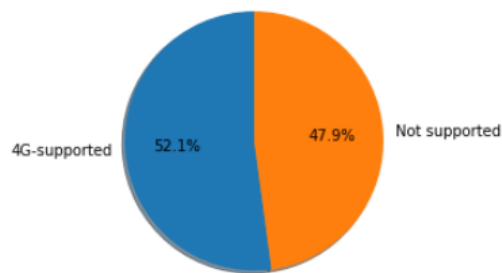


Figure 2 4 Generation distribution

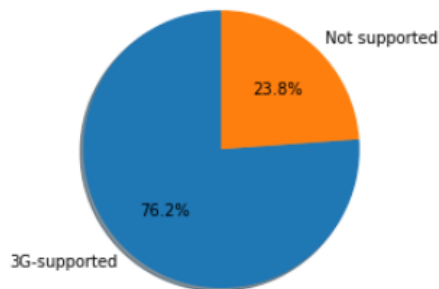


Figure 3 3 Generation distribution

As we already mentioned, our output variable is the price range, that is why we need to examine its relation with other variables.

We choose to examine how the Ram is affected by the price range.

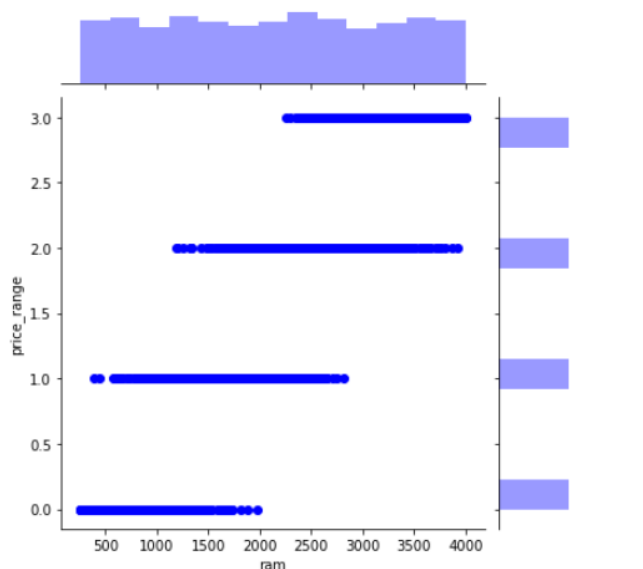


Figure 4 How does ram is affected by price

We observe that ram is increasing by going up in the price range, also the second price range has the biggest mixture of ram types.

We observe also the distribution of battery power by the price range using pox plots.

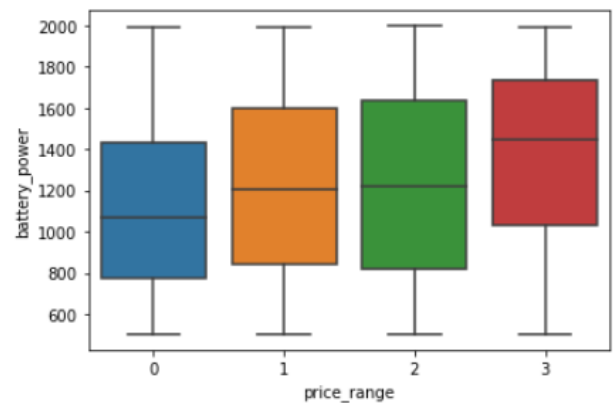


Figure 5 battery power vs price range

There no difference between the 1 & 2 price range in terms of battery power, a customer who is looking for phone with a battery power in the [800,1600] range can buy a phone in the 1 price range to save some money.

The main role of the mobile phone is to make calls .

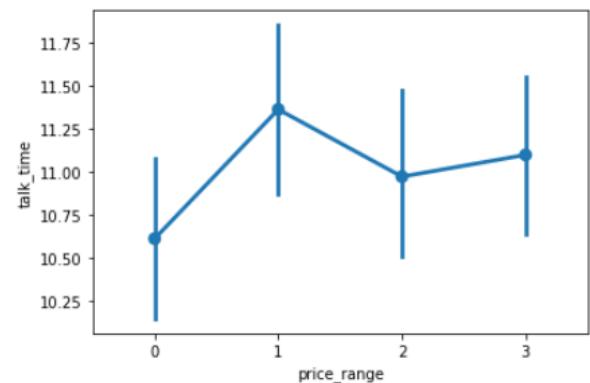


Figure 6 price range vs talk time

The price range 1 has a better interval for the number of hours of calling. it supports more calls time.

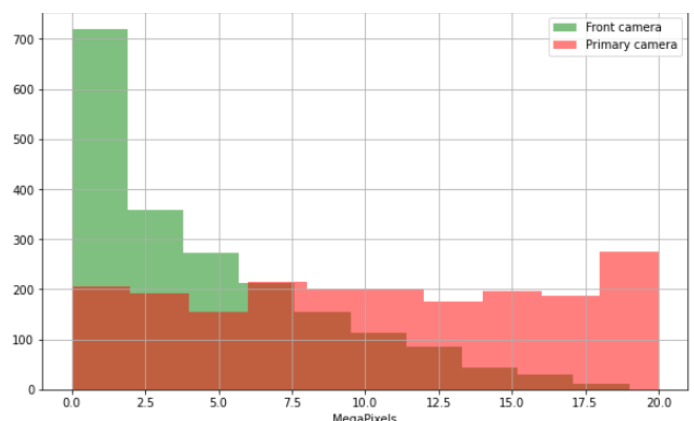


Figure 7 Mégapixels & cameras

More than 1000 phone has a front camera with small amount of megapixel, yet we can find phone with multiple value of megapixels in terms of the primary camera.

III. CLEANING DATA

We have tried to use correlation to find the variables that affects the value of our output price range yet, this method removed many variables that we need to continue the building of our model and it's not really efficient with small amount of data.

We choose to pursuit the cleaning with the backward elimination to detect and remove the variables that does not affect the output by choosing a p-value.

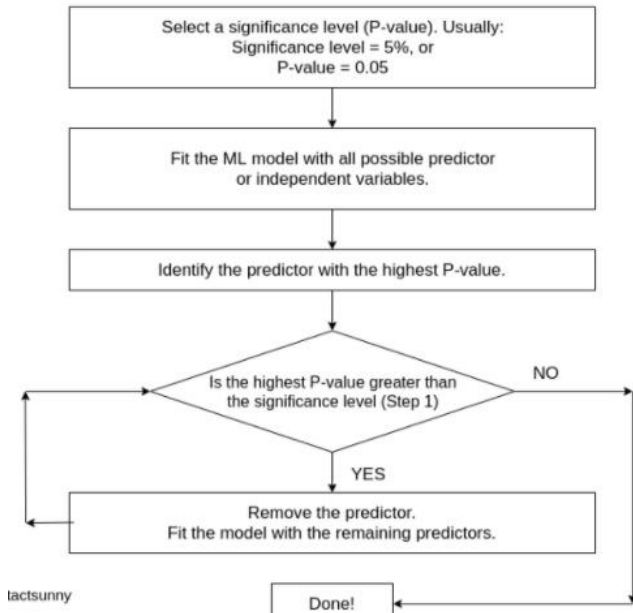


Figure 8 Algorithm of backward elimination

After running the algorithm, we have found many variables deleted and conclude those in the image below.

```
['battery_power', 'clock_speed', 'dual_sim', 'int_memory', 'mobile_wt', 'px_height', 'px_width', 'ram', 'three_g', 'wifi']
```

Figure 9 Variables affecting the output

At the cleaning process, we checked all of the nullity of all variables .

IV. ALGORITHMS

A. Introduction

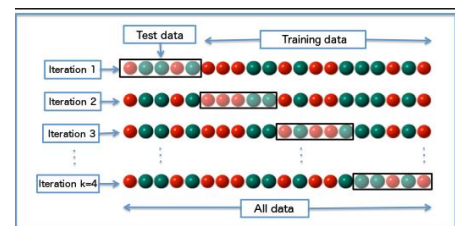
The model that we want to realize can be done using both of the known technics in machine learning as classification and regression. Due the nature of our dataset which include numeric variables and the limited values of our output “price range”.

B. Algorithms & validation used

- **linear regression** was developed in the field of statistics and is used as a model for emphasizing the relationship between input and output numerical variables, but has been used by machine learning. It is both a statistical algorithm and a machine learning algorithm. The equation below shows the relation between two variables x & y using scalars B0 & B1.

$$y = B0 + B1*x$$

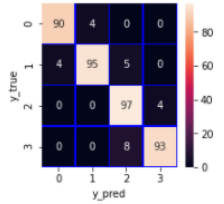
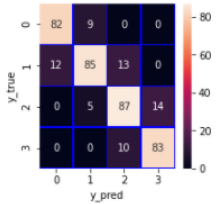
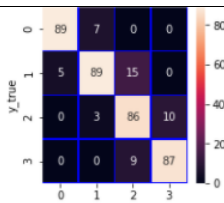
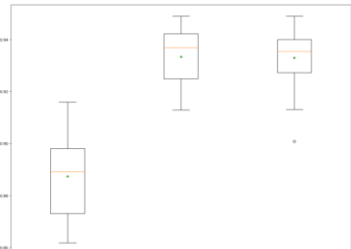
- The principle behind nearest neighbor method or the **KNeighborsClassifier** is to find a predefined training samples closest in distance to the tested point, and predict the price range . The number of samples can be defined (k-nearest neighbor impair value is better). The distance can be any metric measure.
- A decision tree is used by the **DecisionTreeClassifier** it is a tree structure where an internal node represents a specification, the branch represents a decision term, and each leaf node represents the result. The topmost node in a decision tree is called the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This structure can manage the decision making.
- **Stacked generalization** consists in stacking the output of individual estimator and use a classifier to compute the final prediction. Stacking allows to use the strength of each individual estimator by using their output as input of a final estimator.
- In **Train_Test Split** we randomly split the complete data into training and test sets. Then Perform the model training on the training set and use the test set for validation purpose, ideally split the data into 70:30 or 80:20. With this approach there is a possibility of high bias if we have limited data, because we would miss some information about the data which we have not used for training. then this approach is acceptable when the data is huge.
- **K-Folds** technique is a common and easy to understand, it generally results in a less biased model compare to other methods. Because it ensures that every value from the original dataset has the opportunity to appear in training and test set.



1. Split the entire data randomly into K folds .The higher value of K leads to less biased model (but large variance might lead to over-fit), whereas the lower value of K is similar to the train-test split approach we saw before.
2. Then fit the model using the K-1 (K minus 1) folds and validate the model using the remaining Kth fold. Note down the scores/errors.
3. Repeat this process until every K-fold serve as the test set. Then take the average of your recorded scores. That will be the performance metric for the model.

VI. REFERENCES

C. Results

Algorithm	Score & Matrix
Linear Regression	0.90
KNeighborsClassifier	 <p>0.9375</p>
LogisticRegression	0.58
DecisionTreeClassifier	 <p>0.8425</p>
RandomForestClassifier	 <p>0.8775</p>
StackingClassifier with KNeighborsClassifier & RandomForestClassifier as level 1 & using cross validation	 <p>0.933</p>

Cross-validation, Daniel Berrar, Data Science Laboratory, Tokyo Institute of Technology, 2-12-1-S3-70 Ookayama, Meguro-ku, Tokyo 152-8550, Japan.

A fuzzy random forest, Piero Bonissonea, José M. Cadenasb,*, M. Carmen Garridob, R. Andrés Díaz-Valladaresca, GE Global Research, One Research Circle, Niskayuna, NY 12309.

V. OPTIMISATION & CONCLUSION

We have tried to optimize the score of the KNeighborsClassifier by choosing the k-neighbour as an impair value due to the fact that pair values can mess up the classification procedure.

We have tried also to make RandomForestClassifier better by increasing the number of estimator, that can really help with a small dataset.

We linked each classification result with a confusion matrix to show the difficulties related to each price range's prediction.