Project Report

on

Mobile Price Prediction Using KNN Classifier

(A dissertation submitted in partial fulfillment of the requirements of Bachelor of Technology in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology, West Bengal)

Submitted by

Debjit Ganguli
Debdyuti Saha
Chhandak Patra
Protyush Mukherjee
Aniruddha Poddar

Under the guidance of

Shri Debayan Ganguly

Professor,
Dept. of Computer Science and Engineering

Government College of Engineering and Leather Technology

(Affiliated to MAKAUT, West Bengal)

Kolkata - 700106, WB

2020-2021

Certificate of Approval

This is to certify that the project report on “Mobile Price Prediction using KNN Classifier” is a record of bonafide work, carried out by Debjit Ganguli, Debdyuti Saha, Chhandak Patra, Protyush Mukherjee and Aniruddha Poddar under my guidance and supervision

In my opinion, the report in its present form is in conformity as specified by Government College of Engineering and Leather Technology and as per regulations of the Maulana Abul Kalam Azad University of Technology, West Bengal. To the best of my knowledge the results presented here are original in nature and worthy of incorporation in project report for the B.Tech. Program in Computer Science and Engineering.

Signature of
Supervisor/ Guide

Signature of
Head, Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure, I would like to express my profound gratitude and indebtedness to Prof. Debayan Ganguly, Department of Computer Science and Engineering, Government College of Engineering and Leather Technology, W.B. for his continuous guidance, valuable advice and constant encouragement throughout the project work. His valuable and constructive suggestions at many difficult situations are immensely acknowledged. I am in short of words to express his contribution to this thesis through criticism, suggestions and discussions.

I would like to take this opportunity to thank Dr. Santanu Halder, Project Coordinator and Prof. Kalyan Mahata, HOD, Department of Computer Science & Engineering, Government College of Engineering and Leather Technology.

1. Debjit Ganguli – 11200116057
2. Debdyuti Saha – 11200116058
3. Chhandak Patra – 11200116059
4. Aniruddha Poddar – 11200116065
5. Protyush Mukherjee – 11200116040

ABSTRACT

When we think about purchasing a new phone, the most important aspect is usually the budget. The evolution of smartphone has brought a plethora of options and features for us, which can often become overwhelming while looking for a new phone. This is why mobile price classification is so important. It will not only help you in finding a phone with desired features but also within the desired price range. This is the main motive behind the project.
We use a dataset that contains the features of a mobile that affect its price, namely processor clock speed, RAM capacity, front camera, primary camera, internal memory and so on. The dataset can be found at www.kaggle.com [1]
20 features were used as input with the output being the price range.
We code different models with different feature selection algorithms like Linear Regression, Logistic Regression, Random Forest Classifier and KNN Classifier. Ultimately, KNN Classifier is chosen because it has the most accuracy among all the models. The value of k is chosen to be 13 because it provides us the most accuracy.
Test data evaluation shows that the KNN Classifier model with k = 13 is able to predict the price range with an accuracy of 92.62%.

Table of Contents

1. Introduction 1
1.1. Motivation 1
1.2. Background 1
1.3. Summary of previous work 2
1.4. Software used 3
2. Preprocessing 4
2.1. Loading the dataset and cleaning 4
2.2. Number of phones in each price range 5
3. Effect of features on price 6
3.1. Percentage of 4G phones in each price range 6
3.2. Percentage of phones with Bluetooth in each price range 7
3.3. Average Internal Memory of phones at different price ranges 7
3.4. Average Battery Power of phones at different price ranges 8
3.5. Average Processor Clock Speed of phones at different price ranges 9
3.6. Average RAM capacity of phones at different price ranges 10
4. Creating models and finding the best accuracy 12
4.1. K Nearest Neighbours Classifier 12
4.2. Selecting a model 13
4.3. Confusion Matrix of KNN Classifier model 13
5. Predicting price range in test data 15
5.1. Number of phones in each price range in test dataset 15
5.2. Average RAM capacity of phones at different price ranges in test dataset 16
6. Future Scope 18
7. REFERENCES 18

1. Introduction

1.1. Motivation

Mobile phone is a device which was invented as a compact alternative to telephones. It has since evolved to the point that we not only have smartphones, but we also cannot imagine life without a mobile. With time, many companies have started manufacturing mobiles, each with their own set of features and price. With so many mobile brands, models and features to choose from, we are left confused to which phone we must buy. We are torn in anxiety, are we receiving the right value for our money? This is where this project comes in, we take a dataset of mobile phones and their features along with a price range. We create a Machine Learning model to predict the price range of a mobile based on its feature, using the data of the previous mobile phones in our dataset.
The models we use are: Linear Regression, Logistic Regression, Random Forest Classifier and KNN Classifier. We check the accuracy for each and choose the model with the most accuracy.

1.2. Background

In our dataset, we have a train file and a test file. But the test file has no data for price range, so we have to split the training data into training data and testing data to find the accuracy of the models. We will use the test dataset to predict the price range of those mobiles and compare some features with the training dataset to check if our model works as expected. Let's take this opportunity to explore the columns in the dataset: [2]

1. battery_power - The total energy that can be stored by the mobile battery, in mAh
2. blue - Indicates whether the phone has bluetooth or not(0 for no, 1 for yes)
3. clock_speed - The speed at which the mobile processor runs
4. dual_sum - Indicates whether the phone has dual sim support or not(0 for no, 1 for yes)
5. fc - The megapixels of the front camera
6. four_g - Indicates whether the phone has 4G support or not(0 for no, 1 for yes)
7. int_memory - Internal memory of the phone, in GB
8. m_dep - Depth of mobile, in cm
9. mobile_wt - Weight of mobile, in grams
10. n_cores - Number of cores of mobile processor
11. pc - The megapixels of the primary or back camera
12. px_height - Pixel resolution height
13. px_width - Pixel resolution width
14. ram - RAM capacity, in MB
15. sc_h - Screen height
16. sc_w - Screen width
17. talktime - The longest time that one battery charge will last
18. three_g - Indicates whether the phone has 3G support or not(0 for no, 1 for yes)
19. touch_screen - Indicates whether the phone has touchscreen or not(0 for no, 1 for yes)

20. wifi - Indicates whether the phone has WiFi support or not(0 for no, 1 for yes)
21. price_range - Indicates the price range the mobile belongs to(0 for cheap, 1 for average, 2 for expensive, 3 for very expensive)

The training dataset has 2000 rows, with 21 features. The testing dataset has 1000 rows, with 20 features. The testing dataset does not include price_range, so we will use our model to predict price_range.

The first few rows of the training dataset are:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	t
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	

5 rows × 21 columns

]:

lual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	0	1	1
1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	1	0	2
1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	1	0	2
0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	0	0	2
0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	1	0	1

Fig. 1.1: First 5 rows of the dataset using head()

1.3. Summary of previous work

The previous work on this dataset can be found at www.medium.com [3]
In our project, we try to use simpler models that can be explained easily and have similar accuracy as the models used earlier. We also try to find a feature that has a high impact on price. For the feature, we find average values in every price range in the training dataset and keep it noted. We predict the price range for every phone in testing dataset, and find average values for the feature in every price range and compare the values. We also try to explain some features that appear ambiguous for different price ranges.

1.4. Software used

We use Python 3.7 as our base programming language. The packages we import in Python are:

1. numpy – To deal with statistics
2. pandas – To read the csv files, store it and perform various operations on it
3. matplotlib and seaborn – To create graphs
4. sklearn.model_selection – To split data into training data and testing data
5. sklearn.linear_model – To create Linear Regression and Logistic Regression models
6. sklearn.ensemble – To create Random Forest Classifier model
7. sklearn.neighbors – To create K Nearest Neighbours model
8. sklearn.metrics – To find the confusion matrix for the model we select

2. Preprocessing

2.1. Loading the dataset and cleaning

Since the dataset is a csv file, we use read_csv() from pandas to read the training dataset into a DataFrame. Cleaning a dataset is removing rows where one or more columns have null values, so we need to find if any column has null values. We do that by using info() on our DataFrame. We get the following:

```
<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
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

Fig. 2.1: Information about the columns using info()

All the columns have 2000 non-null values, so the dataset does not need any cleaning. Hence we can proceed without much preprocessing.

2.2. Number of phones in each price range

We count the number of phones in each price range, 0 to 3, and graph them to have an initial idea of how well distributed the data is. We use pandas function shape() and the appropriate conditions to graph it with matplotlib barh().

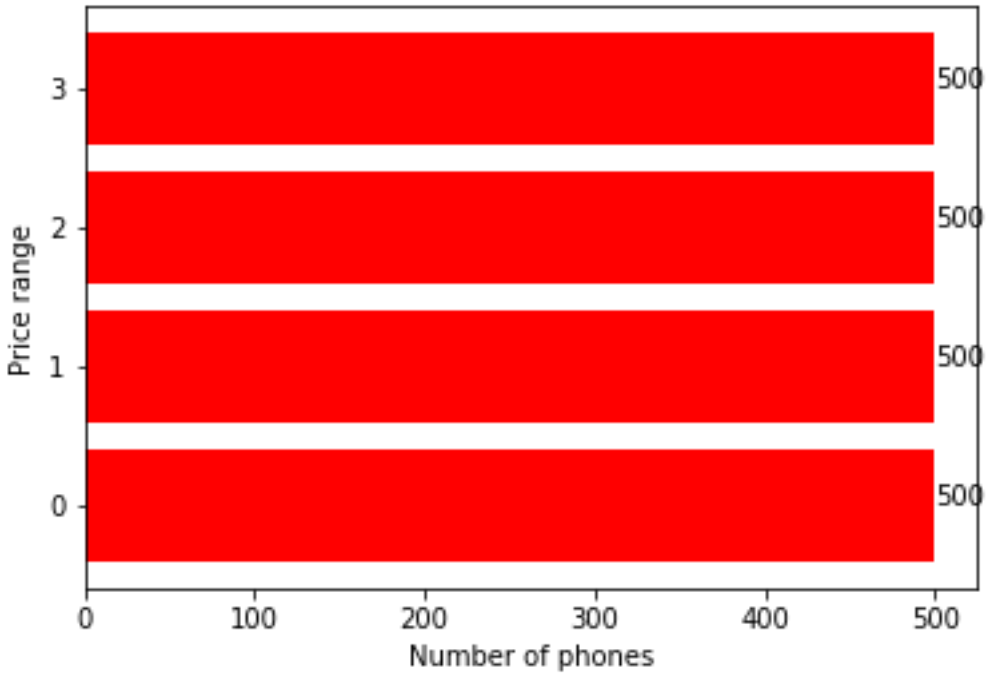

Fig. 2.2: Number of phones in each price range

Each price range has 500 mobiles. The dataset is very evenly distributed. We do not any preprocessing, we proceed to analyzing the features.

3. Effect of features on price

In this section, we try to find which feature has a significant effect on the price of a mobile. We find the average value of a feature in every price range and graph it, and then we try to see if the difference is significant or not. We also try to explain some ambiguous feature effects on price.

3.1. Percentage of 4G phones in each price range

Since phones that support 4G require better hardware, it makes sense for phones with 4G support to be more expensive than other phones. We find the percentage of 4G phones in each price range using shape() from pandas and barh() from matplotlib.

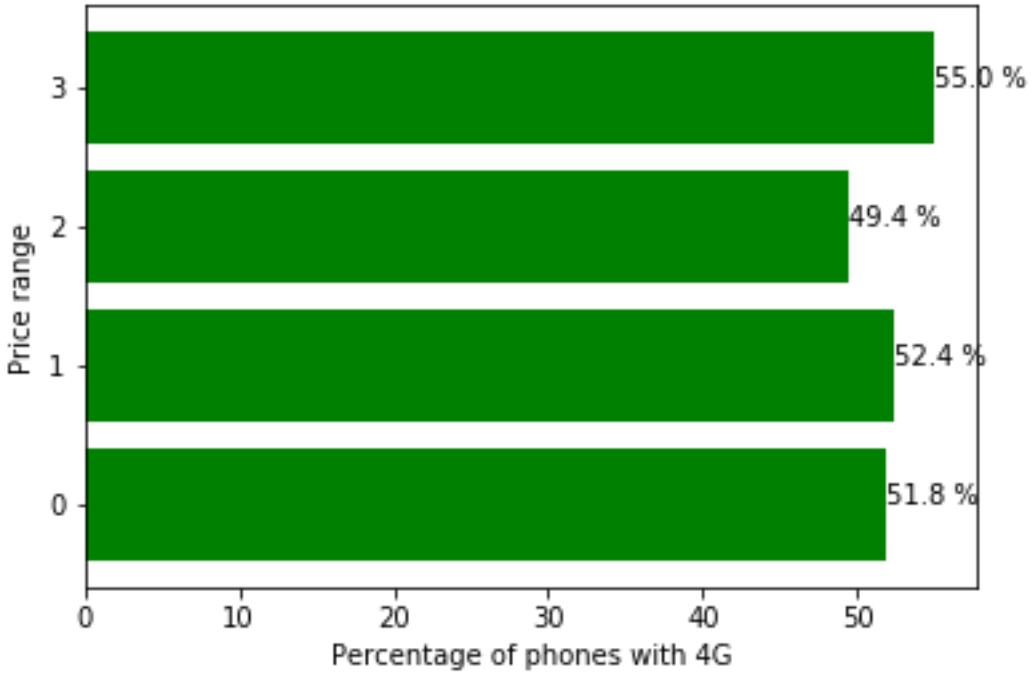

Fig. 3.1: Percentage of 4G phones in each price range

The graph makes sense, since expensive phones are more likely to support 4G than cheaper phones. However, we see that the expensive price range has lower number of 4G phones than average and cheap price range. This might be owing to the fact that some 3G phones with

expensive features were released before 4G was available for use and some inexpensive 4G phones were released once 4G was available for use.

3.2. Percentage of phones with Bluetooth in each price range

Bluetooth is another very important feature for mobiles, since it enables media sharing across mobiles. We find the percentage of phones with Bluetooth in each price range using shape() from pandas and barh() from matplotlib.

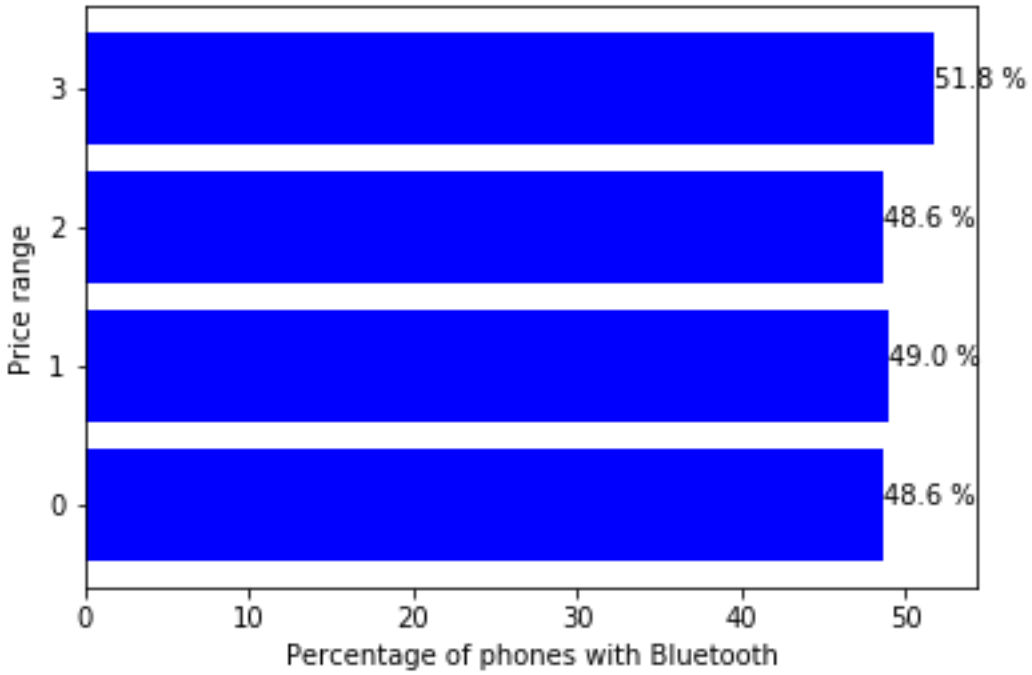

Fig. 3.2: Percentage of phones with Bluetooth in each price range

The graph makes sense since phones that support Bluetooth are expected to be more expensive.

3.3. Average Internal Memory of phones at different price ranges

Phones of the same model have different prices for different internal memory capacities. We find the average internal memory of phones in each price range using mean() from pandas and barh() from matplotlib.

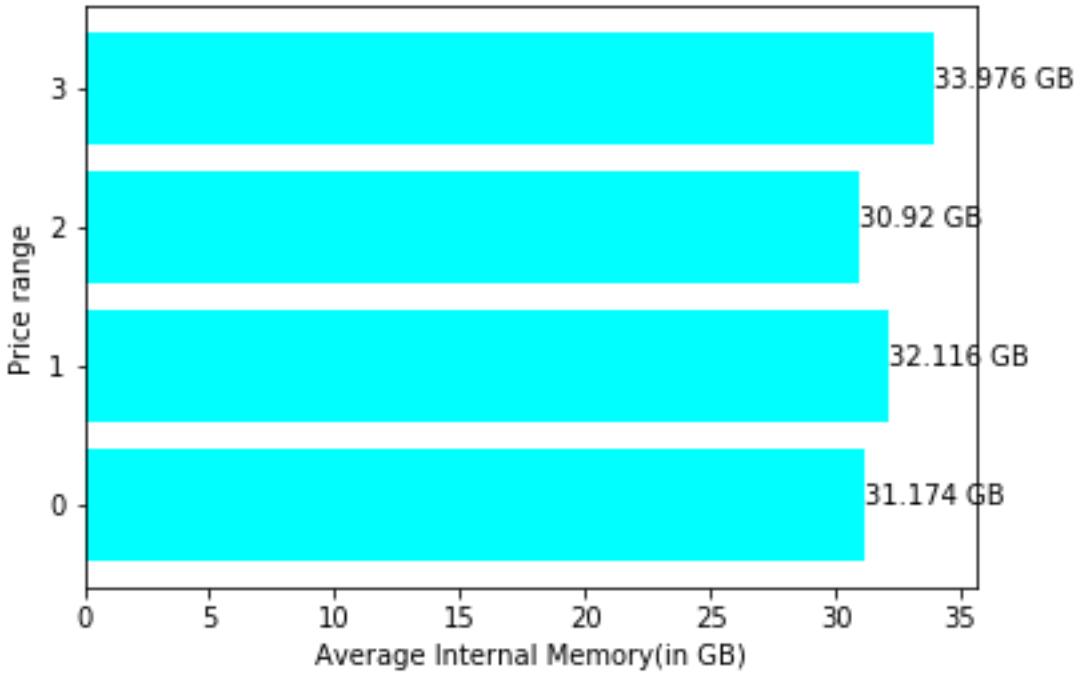

Fig. 3.3: Average internal memory of phones in each price range

Internal memory is almost the same for all price ranges, since most mobiles will have the same internal memory but different features at different price ranges. However, very expensive phones are more likely to have more internal memory than cheaper phones.

3.4. Average Battery Power of phones at different price ranges

The battery power of a phone will determine how long the phone can be used before it needs to be charged again. We find the average battery power of phones in each price range using mean() from pandas and barh() from matplotlib.

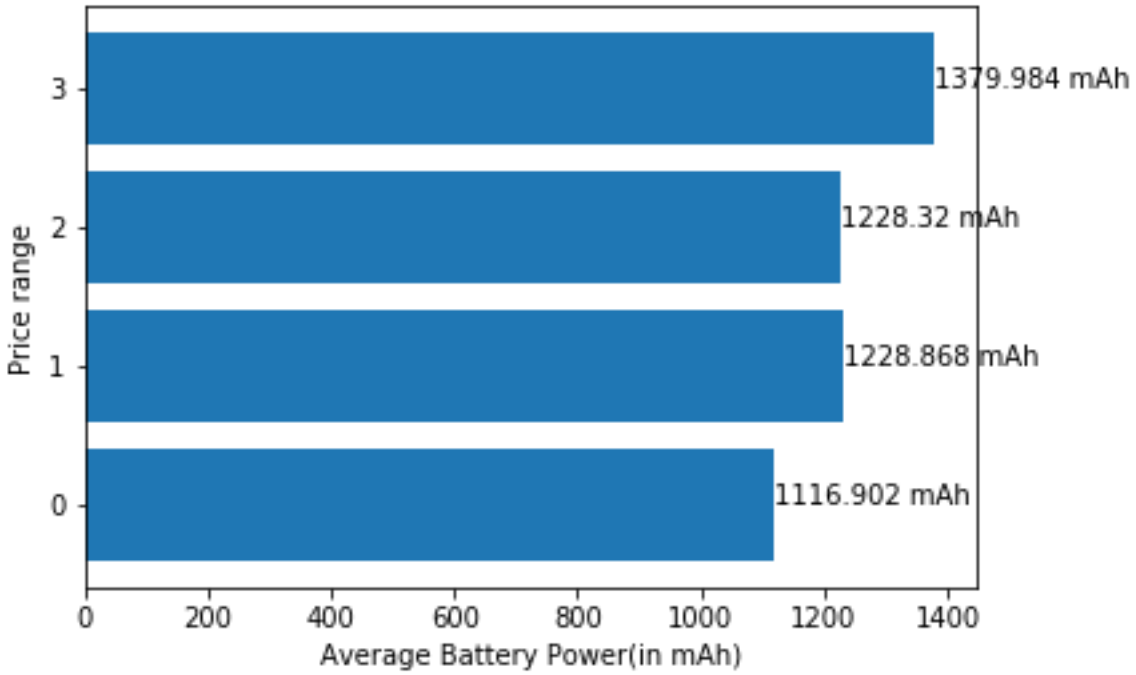

Fig. 3.4: Average battery power of phones in each price range

Phones at higher price range are likely to have more battery power, which makes complete sense.

3.5. Average Processor Clock Speed of phones at different price ranges

The clock speed of a processor determines how fast it can execute instructions. However, since clock speed of a processor does not directly indicate better performance, we might see a different result than expected. We find the average processor clock speed of phones in each price range using mean() from pandas and barh() from matplotlib.

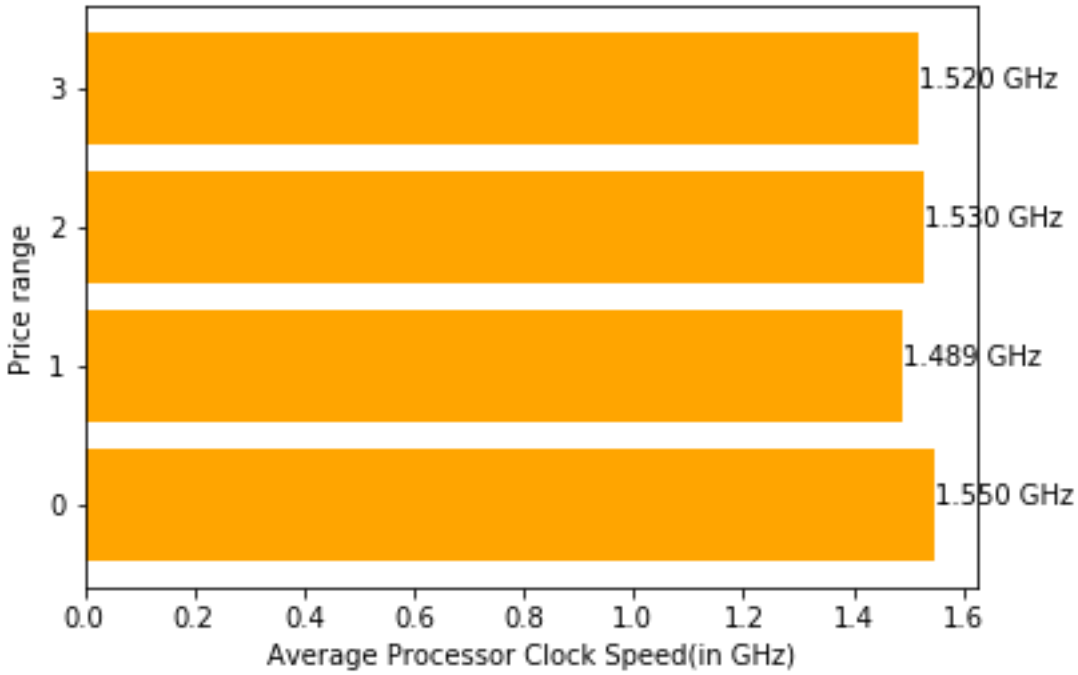

Fig. 3.5: Average processor clock speed of phones in each price range

As we can see, phones in very cheap category have a higher average clock speed processor than phones at higher price ranges. This is owing to the fact that the more clock speed of processors does not directly equal better performance. Performance of a processor depends on the number of cores, the technology used in heat dissipation and so on. In fact, more expensive processors usually have lower clock speed but the other factors make up for the performance.

3.6. Average RAM capacity of phones at different price ranges

The RAM of a phone is very important for people who want to multitask. A little change in RAM can mean a huge boost in performance. We find the average RAM capacity of phones in each price range using mean() from pandas and barh() from matplotlib.

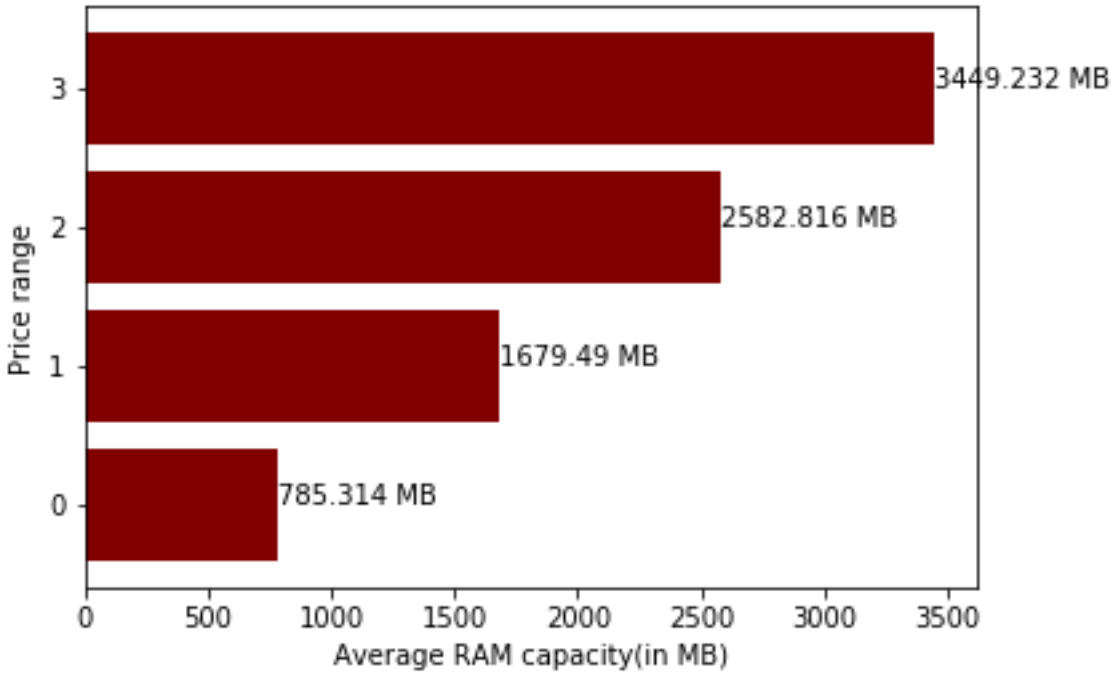

Fig. 3.6: Average RAM capacity of phones in each price range

The difference in RAM capacity in each price range is drastic. RAM is a very important feature in determining the price of a phone. We will use this feature to check if our predictions are accurate later.

4. Creating models and finding the best accuracy

As we discussed before, the test file has no data for price range, so we have to split the training data into training data and testing data to find the accuracy of the models. We split them on a 60:40 basis. We set random state to a constant so that the result is same every time the code is run.
We create 4 different models. They are:

1. Linear Regression
2. Logistic Regression
3. Random Forest Classifier
4. K Nearest Neighbours Classifier

Models 1-3 were simple and straight forward, however K Nearest Neighbours Classifier differs with the value of k set.

4.1. K Nearest Neighbours Classifier

We create a different model for different values of k and compare the accuracies. We select k to be a minimum of 4, since there are 4 classes to be classified in price range. We barh() from matplotlib to graph the data.

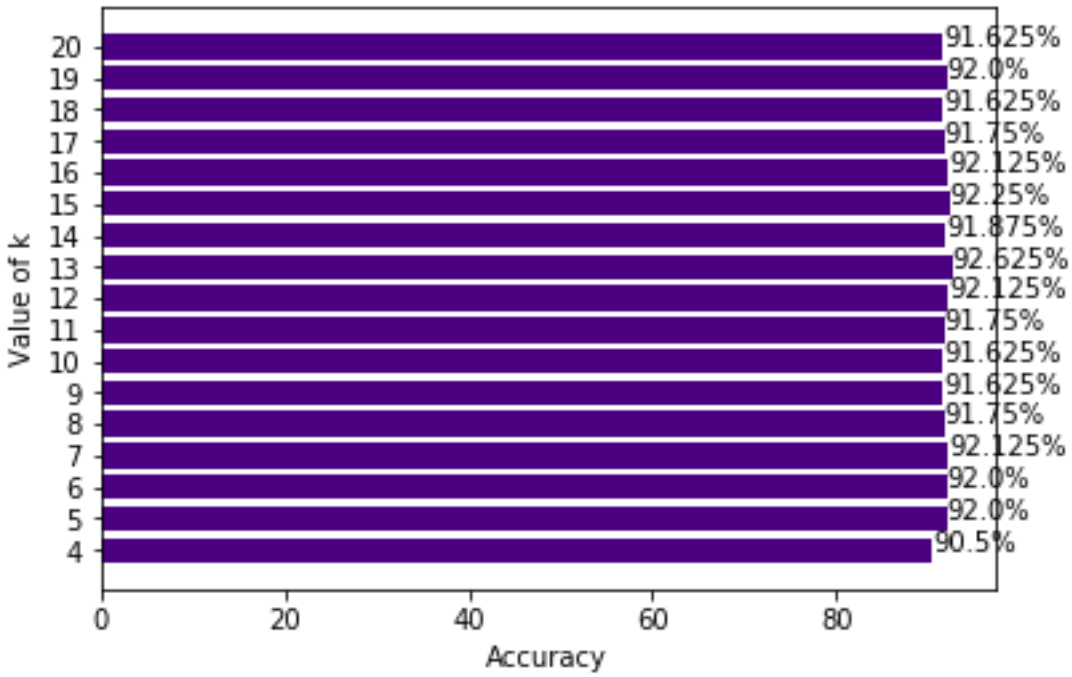

Fig. 4.1: Accuracy of KNN classifier for different values of k

Since accuracy is maximum for k=13, we select KNN classifier with k=13.

4.2. Selecting a model

We select a model from the models we create by graphing the accuracies. We use barh() from matplotlib to graph the accuracies.

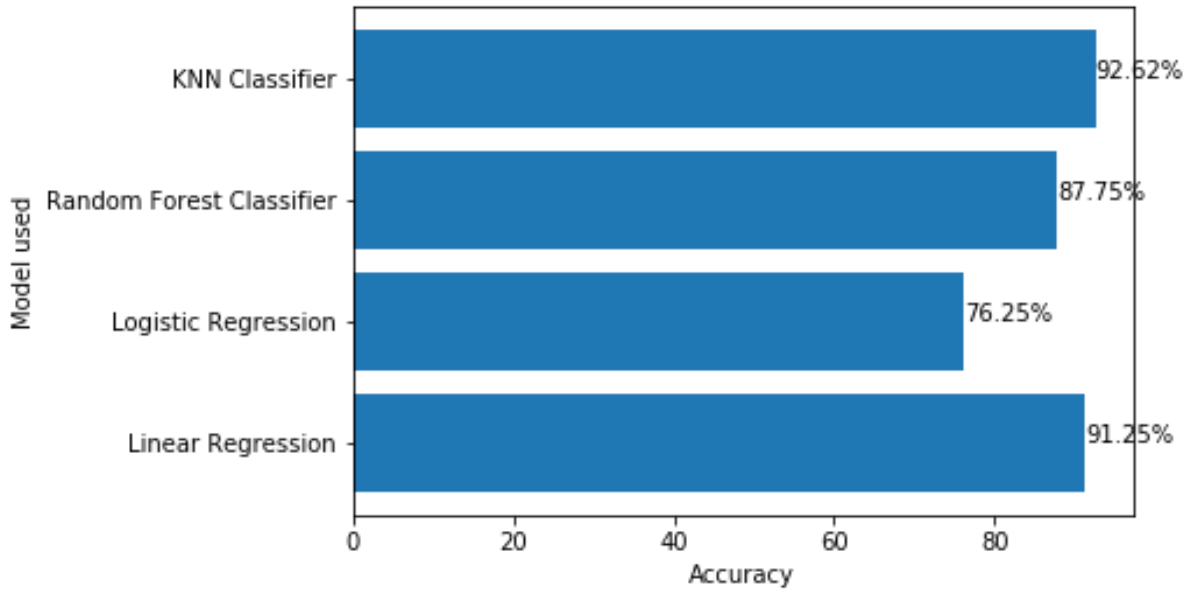

Fig. 4.2: Accuracy of the different models

Linear Regression and KNN Classifier have almost the same accuracy. We choose KNN Classifier since we are dealing with classification here of discrete values, and Linear Regression is a better choice for regression with continuous values.

4.3. Confusion Matrix of KNN Classifier model

We produce the confusion matrix using confusion_matrix from sklearn.metrics by comparing the predicted values to the actual values of price range. We use heatmap() from seaborn to graph it.

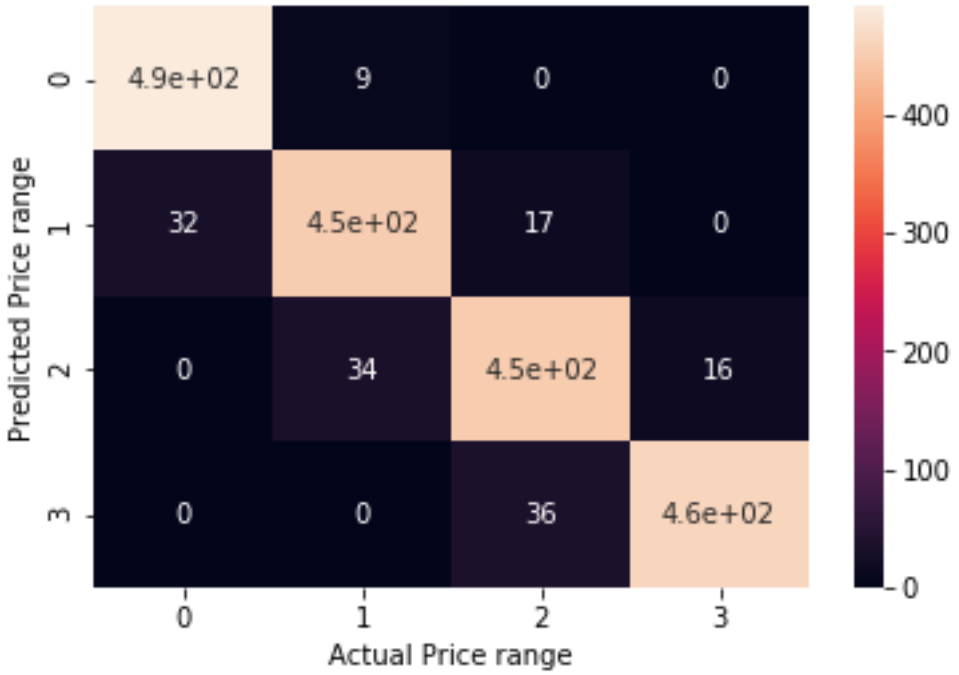

Fig. 4.3: Confusion Matrix for KNN Classifier with k=13

The Confusion Matrix is heavier in the diagonals, suggesting a pretty accurate model. The few inaccuracies are 1 price range apart, since sometimes companies can choose to overprice their phone based on brand value while some companies lower prices to eliminate the competition.

5. Predicting price range in test data

As we discussed earlier, the test dataset has no column for price_range. So, we use our model to predict the price_range for the test dataset. There is, however, no way of knowing if the results are correct. As we covered earlier, the average RAM capacity is pretty significant at different price ranges, so we can compare the average RAM capacities for the test dataset to check if our model is working as expected. We read the test dataset into a DataFrame using read_csv() from pandas. Then we use predict() from our KNN Classifier model to get the predicted values. We add it to the DataFrame and display the first few rows using head().

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	226	1412	3476	12	7	2	
1	2	841	1	0.5	1	4	1	61	0.8	191	...	746	857	3895	6	0	7	
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	1270	1366	2396	17	10	10	
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	295	1752	3893	10	0	7	
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	749	810	1773	15	8	7	

5 rows × 22 columns

_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
1.8	1	14	0	5	0.1	193	...	226	1412	3476	12	7	2	0	1	0	3
0.5	1	4	1	61	0.8	191	...	746	857	3895	6	0	7	1	0	0	3
2.8	0	1	0	27	0.9	186	...	1270	1366	2396	17	10	10	0	1	1	2
0.5	1	18	1	25	0.5	96	...	295	1752	3893	10	0	7	1	1	0	3
1.4	0	11	1	49	0.5	108	...	749	810	1773	15	8	7	1	0	1	1

Fig. 5.1: First 5 rows of the test dataset using head()

5.1. Number of phones in each price range in test dataset

We count the number of phones in each price range, 0 to 3, and graph them to check how distribute the price ranges are. We use pandas function shape() and the appropriate conditions to graph it with matplotlib barh().

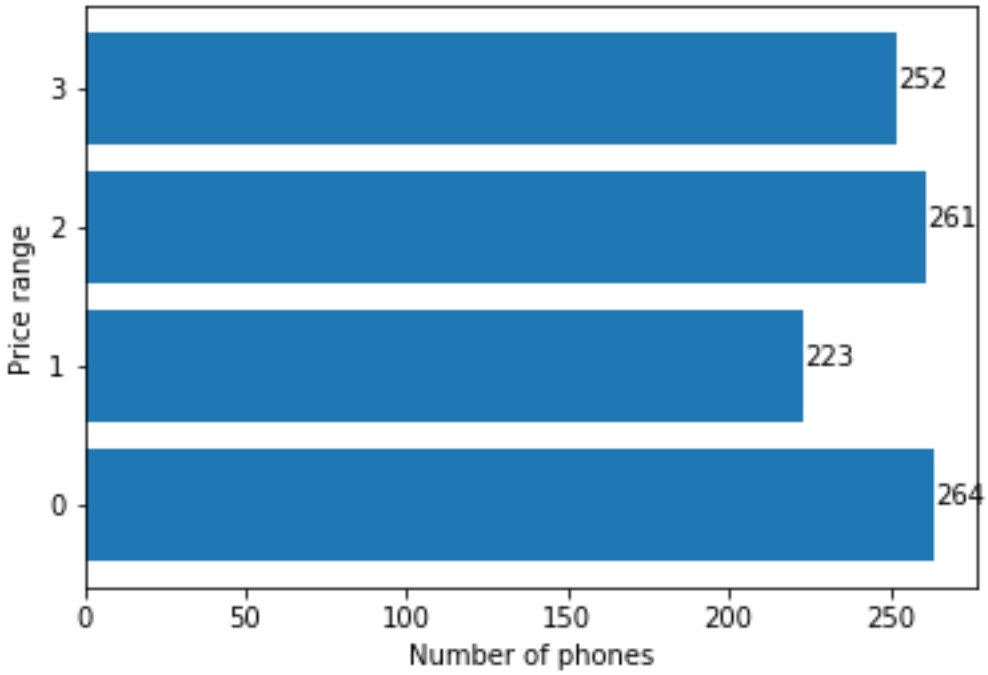

Fig. 5.2: Number of phones in each price range in test dataset

Except for price range 1, all the other ranges are very well distributed again.

5.2. Average RAM capacity of phones at different price ranges in test dataset

As we have seen earlier, RAM capacity is one of the features that affects price the most. The difference in average RAM capacity at different price ranges is pretty solid. As earlier, we find the average RAM capacity of phones in each price range using mean() from pandas and barh() from matplotlib.

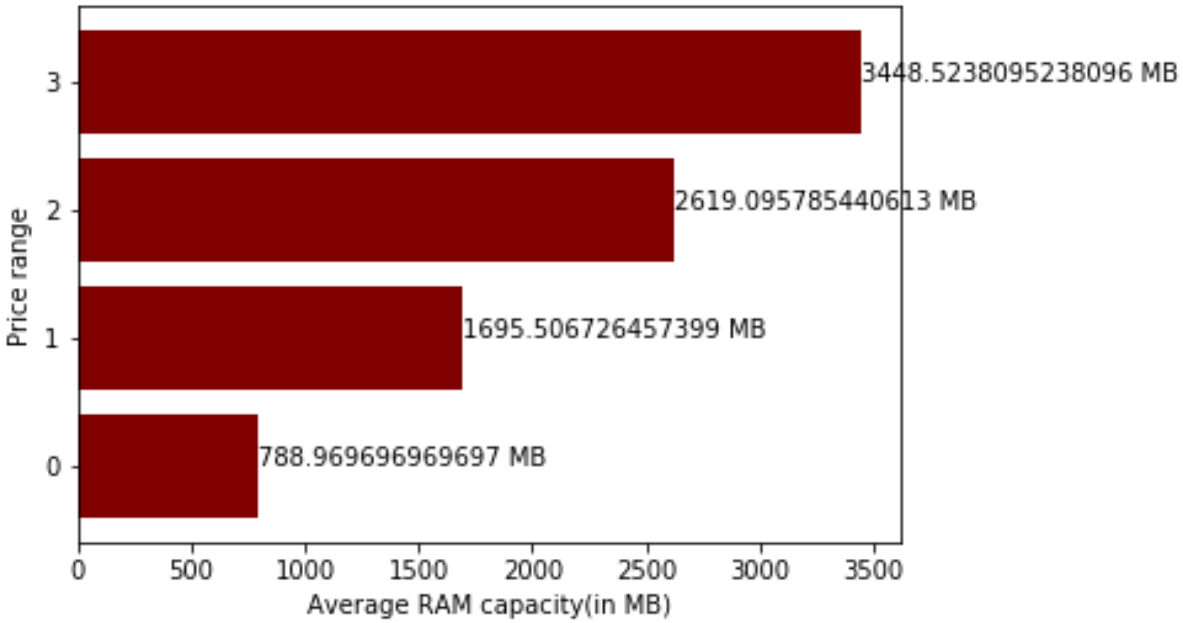

Fig. 5.3: Average RAM capacity of phones in each price range in test dataset

The average RAM capacity for each price range in test dataset is almost the same as that of testing dataset, so we can safely conclude that our model is working pretty well.

6. Future Scope

The accuracy of our KNN classifier is 92.62%, but the accuracy can be improved by using advanced AI tools such as Convolution Neural Network (CNN). To achieve maximum accuracy and predict more accurate, more and more instances should be added to the data set. And selecting more appropriate features can also increase the accuracy. So data set should be large and more appropriate features should be selected to achieve higher accuracy. [4]

7. REFERENCES

[1] https://www.kaggle.com/iabhishekofficial/mobile-price-classification, Abhishek Sharma
[2] ANN for Predicting Mobile Phone Price Range, Ibrahim M. Nasser and Mohammed Al-Shawwa, IJAISR, Pg. 2
[3] https://medium.com/@sai.teja667edu/mobile-price-prediction-using-machine-learning-classification-techniques-1893262704e6, Saiteja
[4] Mobile Price Class prediction using Machine Learning Techniques, Muhammad Asim and Zafar Khan, IJCA, Pg. 11