CSE422: Artificial Intelligence

Mobile Price Prediction

Determining Market Prices of Mobile Phones using ML

Mobile Price Prediction

Submitted By

Md. Ishrak Khan

2010151

Alina Hasan

20101301

Tahmidul Karim Takee

20101609

&

Nishat Zerin

20101136

Date of Submission

30.08.2022

BRAC University

Introduction

Mobile phones are available in a wide range of costs, features such as internal memory or battery power, and other criteria. A key component of consumer strategy is the estimate and forecast of prices. An article by *Remington Hall* states forecasting allows for data-driven strategy development and well-informed corporate decision-making, achievable via Artificial Intelligence, specifically Business intelligence or BI.

Hence why our study focuses on training machine learning models using readily accessible information on the numerous features and price ranges of mobile phones in the market to make the prices of newer products determinable.

Methodology

1. Dataset Description

Our dataset *train.csv* is a Comma Separated Values file, a plain text file containing a list of data. It has been published on the data publishing site *Kaggle* under *Mobile Price Classification* by *Abhishek Sharma,* a *Maropost* Data Scientist. The set consists of 2000 mobile phones and 21 variables including

1. Battery Power: Total energy a battery can store, measured in mAh
2. Bluetooth: 1 means 'has' and 0 otherwise
3. Clock Speed: Instruction execution speed of the microprocessor
4. Dual Sim Support: 1 means 'has' and 0 otherwise
5. Front Camera Megapixels
6. 4G: 1 means 'has' and 0 otherwise
7. Internal Memory: Measured in Gigabytes
8. Mobile Depth: Measured in centimeters
9. Mobile Phone Weight
10. Number of Core Processors
11. Primary Camera Megapixels
12. Pixel Resolution Height
13. Pixel Resolution Width
14. Random Access Memory: Measured in Megabytes

15. Mobile Screen Height: Measured in centimeters
16. Mobile Screen Width: Measured in centimeters
17. Talk Time: Longest time a single battery charge lasts on call
18. 3G: 1 means ‘has’ and 0 otherwise
19. Touch Screen: 1 means ‘has’ and 0 otherwise
20. WiFi: 1 means ‘has’ and 0 otherwise
21. Price Range: Target feature; Ranges from 0 to 3, with 0 lowest and 3 highest

2. Pre-Processing Techniques

2.1 Libraries & Tools

Pandas, a data analysis library that utilizes two-dimensional data structures similar to excel spreadsheets called a data frame, imports and reads the *train.csv* file. Another library, Scikit-Learn, provides learning algorithms and other associated functions such as *.StandardScaler()* for predictive data analysis. Throughout our project, we will apply several of its modules. Finally, MatPlotLib, a two-dimensional plotting library, provides the visuals of our data.

2.2 Checking for Null Values

Applying *.null()* on the imported dataset returns only *False* values in each row. For a more thorough view, applying *.null().sum()* returns a count of 0 for each feature indicating that null values are absent. Therefore, there are no missing values in the dataset.

2.3 Checking Data Types

Applying *.info()* on the dataset returns the data type of the domain of each attribute (or feature). All data types are either in *int64* or *float64*. They are all in numbers so further classification is not required.

```
In [212]: data_train.info() #checking data types

<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
```

Figure: Data types of our used data

2.4 Standardization

Standardization is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Furthermore, it can be helpful in cases where the data follows a Gaussian distribution. However, this is not necessarily true. Unlike normalization, standardization does not have a bounding range. So, even if there are outliers in data, they will not be affected by standardization.

3. Models

3.1 K-Nearest Neighbor

KNN calculates the Euclidean distance (diagonal distance) between the query point and the k-number of the nearest neighboring points. Then chooses the class label based on the highest frequency label. It is a supervised learning classifier that analyzes proximity to classify or anticipate how a single data point is grouped.

In our program, we import *KNeighborsClassifier* from *sklearn.neighbors*, a Scikit-Learn module, and model train with the default value of k being 5. This produces a poor accuracy score of 52.0% on the dataset.

3.2 Logistic Regression

Logistic Regression, a supervised learning classifier, is a statistical model which predicts the probability of a binary event occurring on a given input dataset of independent variables. Therefore the output of the dependent variable is a probability between 0 and 1 (inclusive).

From *sklearn.linear_model*, another Scikit-Learn module, we import *LogisticRegression* to train the model. Doing so gives an accuracy of 96.0% on the dataset, a score significantly higher than the KNN model.

3.3 Naive Bayes

A subset of classification algorithms built on the Bayes Theorem is known as naive Bayes. It is a group of algorithms created under the assumption that each set of features being categorized stands alone from the others.

The *sklearn.naive_bayes* Scikit-Learn module allows us to import *GaussianNB* that trains the model using Naive Bayes, producing an accuracy rate of 76.0% which is higher than the KNN model but not as high as the Logistic Regression model.

3.4 Decision Tree

Classification and regression issues can be resolved using a decision tree, a supervised learning approach. It is a tree-structured classifier in which each leaf node represents the classification outcome, each internal node represents a feature and each branch is for decision-making.

From the final Scikit-Learn module *sklearn.tree*, we import *DecisionTreeClassifier* to train the model using the decision tree learning algorithm. This results in an accuracy score of 86.0% which is higher than the Naive Bayes model but not as high as the Logistic Regression model.

Conclusion

As per demonstrations of the four models, it is evident that the Logistic Regression model produces the highest accuracy score (96.0%), whereas the KNN model produces the lowest accuracy score (52.0%). The Naive Bayes model generates a mediocre score of 76%. And lastly, the Decision Tree model generates a score of 86.0%, which sits between the Logistic and the Naive Bayes models. In conclusion, the logistic regression model can best predict the price of mobile phones relative to the given dataset. The visual representation of the final results is shown below through a bar chart which gives a better overview of the outcomes.

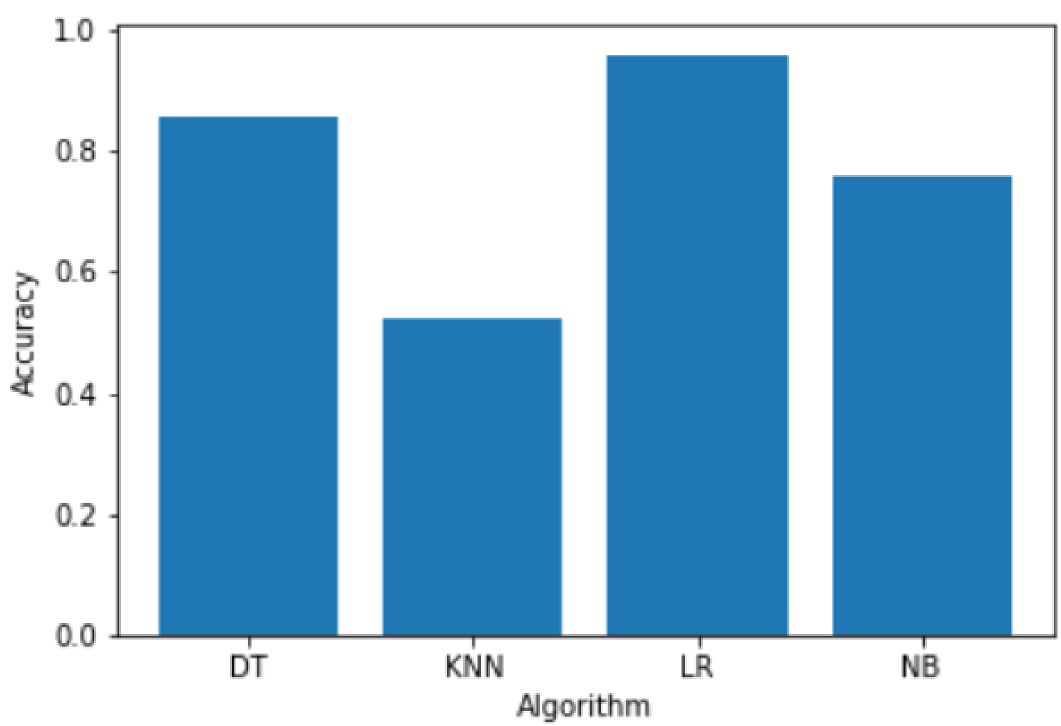

Figure: Accuracy vs Algorithm chart

Conclusion

We can conclude that our machine model can predict a phone price quite accurately depending on the given data. Therefore, budget reduction and important business decisions for newer mobile phone products can both be aided by this learning model.

References

1. Remington Hall (2020) *Why Forecasting is Important for Business Success* [Online]. Available at: https://www.baass.com/blog/why-forecasting-is-important-for-business-success (Accessed: 29 August 2022)
2. Abhishek Sharma (2017) *Mobile Price Classification* [Online]. Available at: https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification?select=train.csv (Accessed: 17 July 2022)