FDS Mini Project Report

Armaan C Rao
185001023

Mobile Price Range Prediction

April 15, 2021

Problem Statement

To predict if the mobile with given features will be economical or expensive by finding some relation between its features and selling price.

Literature Survey

1. **Predicting the Price of Used Cars using Machine Learning Techniques:** Sameerchand-Pudaruth predicts the prices of second hand cars in Mauritius [1]. He implemented many techniques like Multiple linear regression, k-nearest neighbors(KNN), Decision Tree, and Naïve Bayes to predict the prices. Sameerchand-Pudaruth got Comparable results from all these techniques. During research it was found that most popular algorithms i.e Decision Tree and Naïve Bayes are unable to handle, classify and predict Numerical values. Number of instances for his research was only 97(47 Toyota+38 Nissan+12 Honda). Due to less number of instances used, very poor prediction accuracies were recorded.
2. **Introduction to Multiple Regression: How Much Is Your Car Worth?:** Shonda Kuiper has also worked in the same field. Kuiper used a multivariate regression model to predict the price of 2005 General Motor cars. He collected the data from an online source www.pakwheels.com [2]. The main part of this research work is "Introduction of suitable variable selection techniques, which helped to find which variables are more suitable and relevant for inclusion in models. This (His research) helps students and future researchers in many fields to understand the conditions under which studies should be conducted and gives them the knowledge to discern when appropriate techniques should be used.

3. **Support Vector Regression Analysis for Price Prediction in a Car Leasing Application:**
   Support Vector Machine(SVM) concept is used by one another researcher Mariana Listiani for the same work [3]. Listiani predicted prices of leased cars using the above mentioned technique. It was found in this research that SVM technique is far more better and accurate for price prediction as compared to others like multiple linear regression when a very large data set is available. The researcher also showed that SVM also handles high dimensional data better and avoids both the underfitting and overfitting issues. To find important features for SVM Listiani used Genetic Algorithms . However, the technique failed to show in terms of variance and mean standard deviation why SVM is better than simple multiple regression.
4. **House Price Prediction: Hedonic Price Model vs. Artificial Neural Network:**
   Neural Networks (NN) are more better in estimating house, this was concluded in the research of Limsombunchai [4]. By comparing with the hedonic method his method was more accurate.Operation of both the methods are the same, but in NN the model is trained first and then tested for prediction. Using both the methods NN produced higher R-sq and smaller root mean square error (RMSE), while hedonic produced lower values. This research was limited because the actual house prices were missing and only estimated prices were used for the research work.
5. **Vehicle Price Prediction System using Machine Learning Techniques:**
   K Noor and Saddaqat J also worked to predict the price of Vehicles using different techniques [5]. The researchers achieved highest accuracy using multiple linear regression. This paper proposes a system where price is a dependent variable which is predicted, and this price is derived from factors like vehicle's model, make, city, version, color, mileage, alloy rims and power steering.

Scope

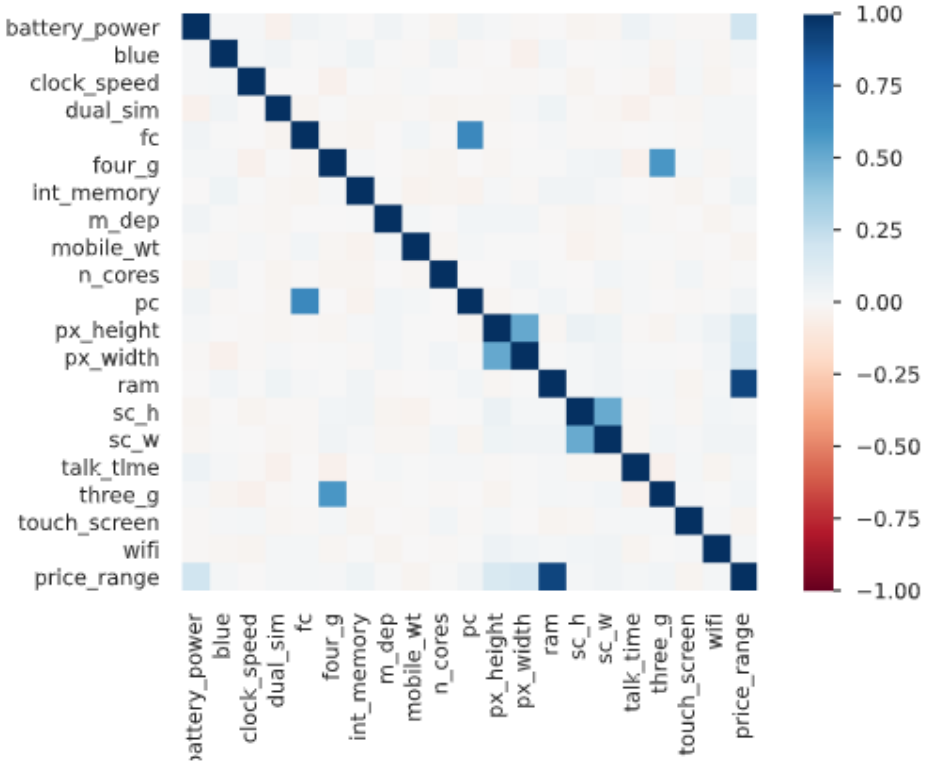
Every day new mobiles with new versions and more features are launched. Hundreds and thousands of mobile phones are sold and purchased on a daily basis. This work can be used in any type of marketing and business to find an optimal product with minimum cost and maximum features.

Dataset Description

The dataset was taken from Kaggle datasets and contains information about 3000 records of mobiles and its features. The records consist of 21 columns, ID, battery power, bluetooth capability, clock speed, dual sim, front camera megapixels, primary camera megapixels, 4-G, 3-G, internal memory in GB, mobile depth, mobile weight, number of cores, pixel resolution height, pixel resolution width, RAM in MB, screen height, screen width, talk-time, touch-screen, and wifi capability.

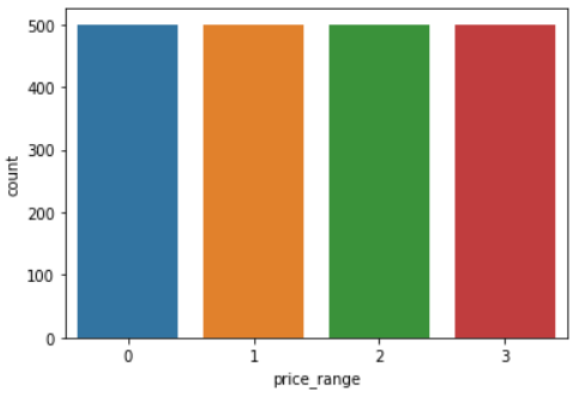
1. **Statistics:**
   There are 21 columns, and 2000 rows. There are no missing values or duplicate rows. The total size of the dataset is 328.2KB and the average record size is 168.1B.

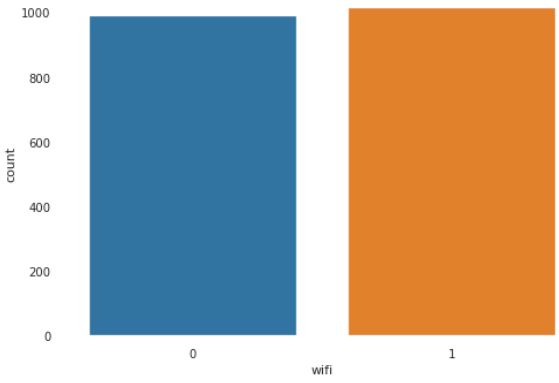
2. **Pearson's Correlation:**

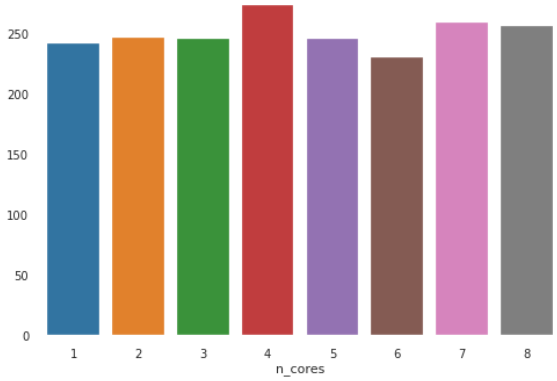


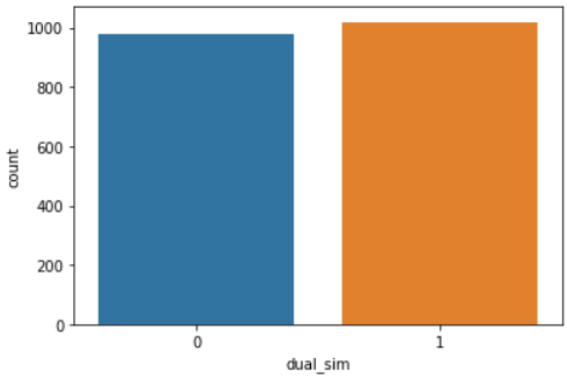
It can be inferred that RAM has the highest correlation with the target variable price_range.

3. **Distribution:**



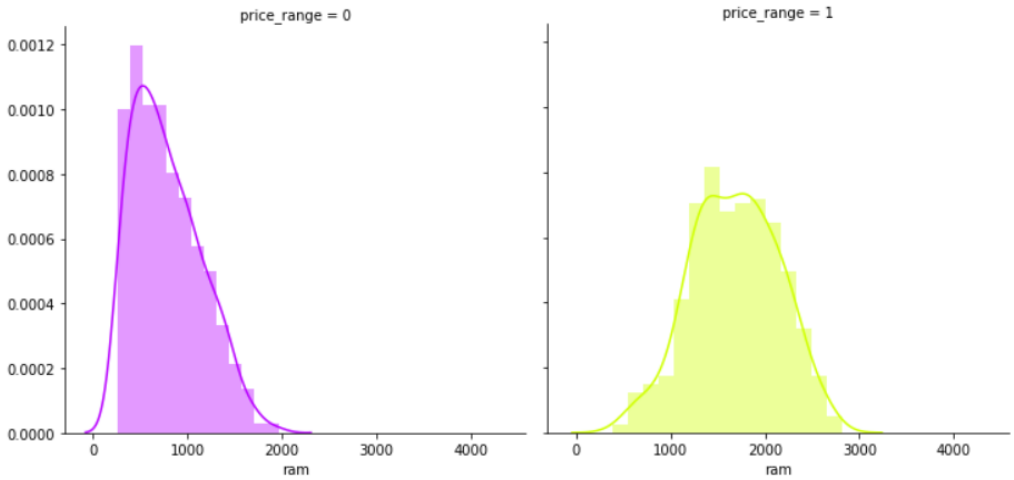


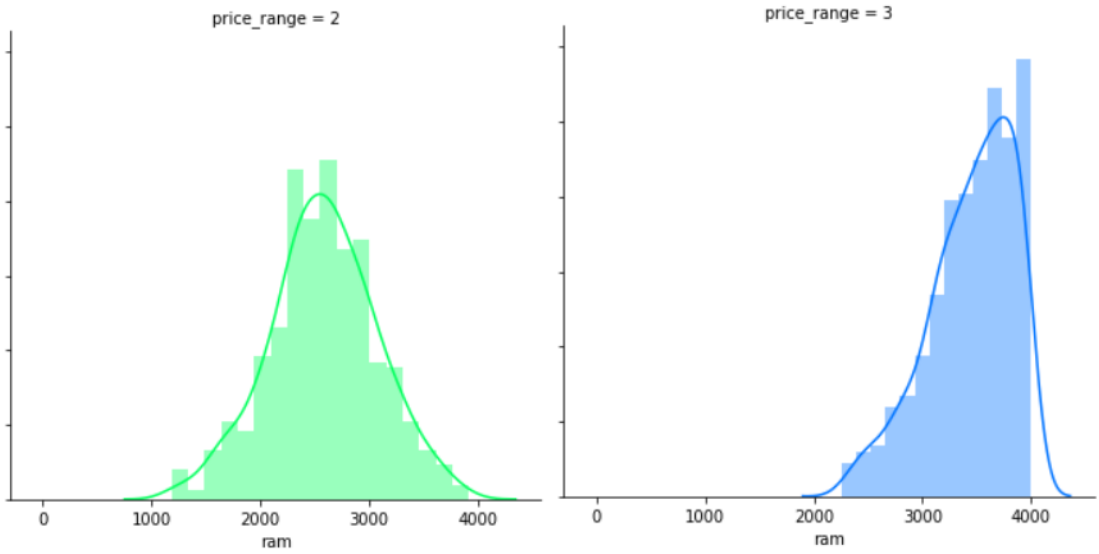




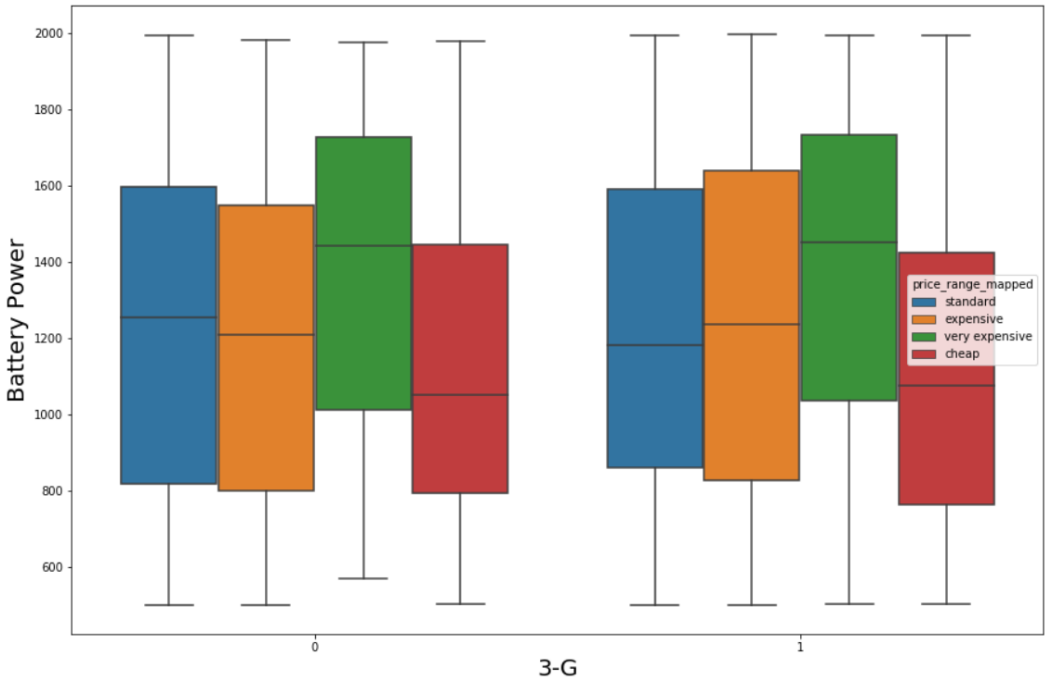
The dataset is evenly distributed.

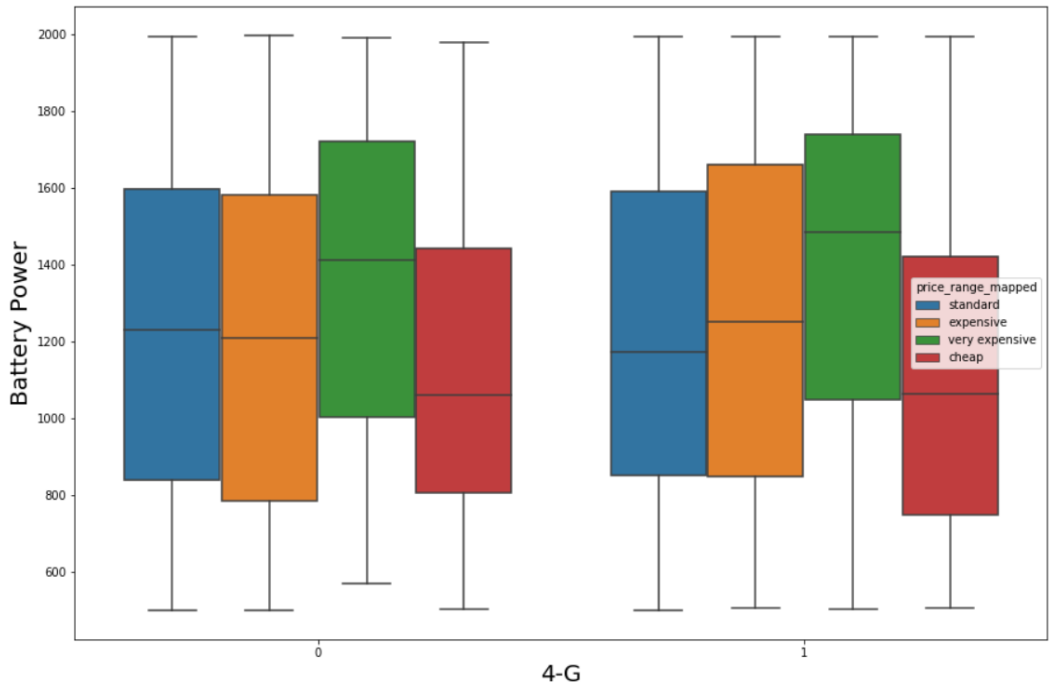
4. **Correlation Between RAM and price range:**





5. **Battery vs. 3G and 4G**





Data Preprocessing

The following steps were performed to prepare the data for model construction.

1. **Feature Selection:**
   From the dataframe, 'battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g', 'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g', 'touch_screen', and 'wifi' are selected as features.

2. **Target Selection:**
   price_range is selected as the target variable.

3. **Train-test splitting:**
   The dataset is split as train and test data. The train data is used for training the model and the test data is used for testing to evaluate the model. The train-test split is done in the 75:25 ratio using the train_test_split method from sklearn.model_selection.
4. **Scaling:** Data is scaled using StandardScaler method from sklearn.preprocessing.

Methodology

The problem statement was solved with three different models and was compared.

1. **Logistic Regression:**

   Justification: Logistic regression tries to find the probabilities of class membership as a multilinear function of the features. Then depending on the probabilities, the classification is done. Since the problem of identifying the price range of a mobile as cheap, standard, expensive or very expensive is a classification problem, logistic regression has been chosen.

   Algorithm:

   ```
   lr = LogisticRegression()
   lr.fit(X_train, y_train)
   lr_predict = lr.predict(X_test)
   lr_conf_matrix = confusion_matrix(y_test, lr_predict)
   lr_acc_score = accuracy_score(y_test, lr_predict)
   ```

   Confusion Matrix:

135	1	0	0
1	120	2	0
0	8	116	3
0	0	2	112

2. **Naive Bayes Classifier:**

   Justification: Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. Naive Bayes classifiers have worked quite well in many real-world

classification problems. They require a small amount of training data to estimate the necessary parameters. It predicts the class with the highest probability.

Algorithm:

```
nb = GaussianNB()
nb.fit(X_train,y_train)
nbpred = nb.predict(X_test)
nb_conf_matrix = confusion_matrix(y_test, nbpred)
nb_acc_score = accuracy_score(y_test, nbpred)
```

Confusion Matrix:

119	16	1	0
5	94	24	0
0	24	88	15
0	0	15	99

3. **Decision Tree Classifier:**

Justification: The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree. Since the problem is to determine the price range of mobile phones based on features, the decision tree classifier can be used.

Algorithm:

```
dt = DecisionTreeClassifier(criterion = 'entropy',random_state=0,max_depth = 6)
dt.fit(X_train, y_train)
dt_predicted = dt.predict(X_test)
dt_conf_matrix = confusion_matrix(y_test, dt_predicted)
dt_acc_score = accuracy_score(y_test, dt_predicted)
```

Confusion Matrix:

121	15	1	0
8	99	16	0
0	24	91	12
0	0	14	100

Tools Used

The following libraries were used to visualise, and preprocess the data, train, and evaluate the model.

1. **numpy**
2. **pandas**
3. **matplotlib.pyplot**
4. **Seaborn**
5. **StandardScaler from sklearn.preprocessing**
6. **Train_test_split from sklearn.model_selection**
7. **DecisionTreeClassifier from sklearn.tree**
8. **confusion_matrix, accuracy_score, classification_report from sklearn.metrics**
9. **LogisticRegression from sklearn.linear_model**
10. **GaussianNB from sklearn.naive_bayes**
11. **DecisionTreeClassifier from sklearn.tree**

Performance Metrics

Accuracy:

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

ACCURACY = Number of correct predictions / Total number of prediction

Precision:

Precision tells what proportion of positive identifications was actually correct.

PRECISION = TP/(TP+FP)

Recall:

Recall tells what proportion of actual positives was identified correctly.

RECALL = TP/(TP+FN)

F1-score:

F1 score is the harmonic mean of precision and recall.

F1 SCORE = 2 / (1 / Precision + 1 / Recall).

Results

Accuracy:

- **Logistic Regression:**

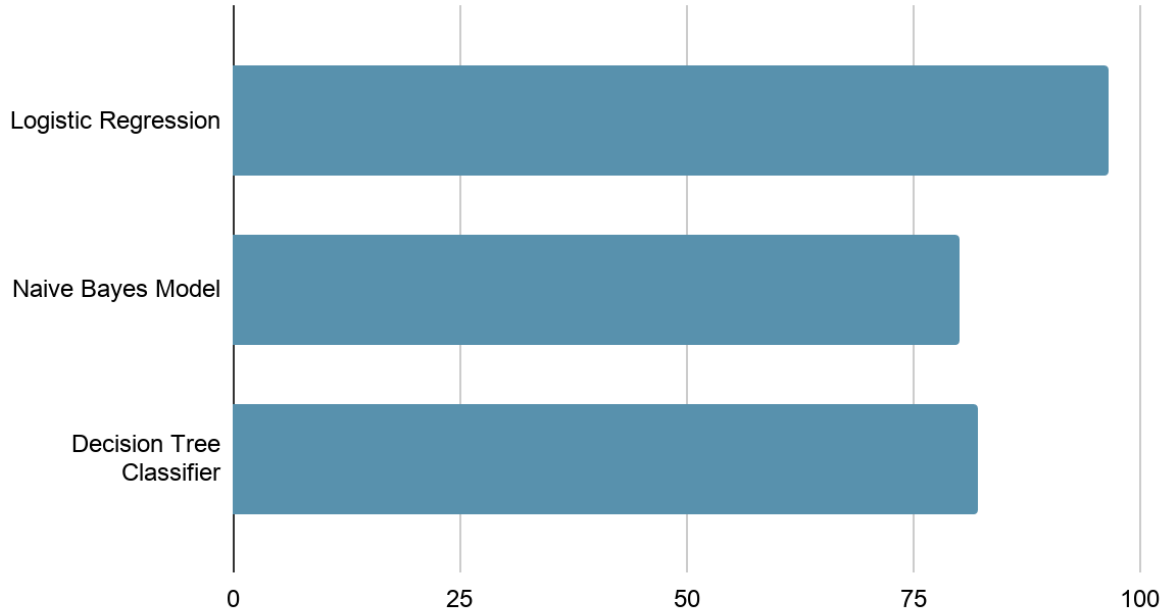
  Accuracy = 96.6%

- **Naive Bayes Classifier:**

  Accuracy = 80.0%

- **Decision Tree Classifier:**

Accuracy = 80.2%



Logistic regression has the highest accuracy, followed by Decision tree classifier, and then Naive Bayes classifier.

Precision, Recall, F1-score:

- **Logistic Regression:**

	precision	recall	f1-score	support
0	0.99	0.99	0.99	136
1	0.93	0.98	0.95	123
2	0.97	0.91	0.94	127
3	0.97	0.98	0.98	114
accuracy			0.97	500
macro avg	0.97	0.97	0.97	500
weighted avg	0.97	0.97	0.97	500

- **Naive Bayes Classifier:**

	precision	recall	f1-score	support
0	0.96	0.88	0.92	136
1	0.70	0.76	0.73	123
2	0.69	0.69	0.69	127
3	0.87	0.87	0.87	114
accuracy			0.80	500
macro avg	0.80	0.80	0.80	500
weighted avg	0.81	0.80	0.80	500

- **Decision Tree Classifier:**

	precision	recall	f1-score	support
0	0.94	0.89	0.91	136
1	0.72	0.80	0.76	123
2	0.75	0.72	0.73	127
3	0.89	0.88	0.88	114
accuracy			0.82	500
macro avg	0.83	0.82	0.82	500
weighted avg	0.83	0.82	0.82	500

Precision, recall, and f1-score decreases in the following order: logistic regression, decision tree classifier, naive bayes classifier.

Conclusion

For the problem statement, experiments were carried out for three different models - Logistic regression, Naive Bayes model, and Decision tree classifier. For the collected dataset, logistic regression showed 96.6%, naive Bayes showed 80.0%, and decision tree showed 82.2% accuracy. From the experiment results, it can be inferred that logistic regression works best for the selected problem statement and dataset.

References

[1] Pudaruth, Sameerchand. (2014). Predicting the Price of Used Cars using Machine Learning Techniques. International Journal of Information & Computation Technology. 4. 753-764.

[2] Shonda Kuiper (2008) Introduction to Multiple Regression: How Much Is Your Car Worth?, Journal of Statistics Education, 16:3, DOI: 10.1080/10691898.2008.11889579

[3] Listiani, M., Möller, R., Morlock, M., Lessmann, S., & Hamburg, G. (2009). Support Vector Regression Analysis for Price Prediction in a Car Leasing Application.

[4] Limsombunchai, Visit & Gan, Christopher & Lee, Minsoo. (2004). House Price Prediction: Hedonic Price Model vs. Artificial Neural Network. American Journal of Applied Sciences. 1. 10.3844/ajassp.2004.193.201.

[5] Noor, Kanwal & Jan, Sadaqat. (2017). Vehicle Price Prediction System using Machine Learning Techniques. International Journal of Computer Applications. 167. 27-31. 10.5120/ijca2017914373.