IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Estimating Mobile Phone Categorical Prices Using Machine Learning: A Study On Feature Selection And Algorithmic Approaches

[1]Dhruv Jore, [2]Amey Dubey, [3]Shubhangi Jore, [4]Vikas Khare

[1]Undergraduate Student, [2] Undergraduate Student, [3]Associate Professor, [4]Associate Professor
[1]School of Technology Management and Engineering,
[1] SVKM's NMIMS Deemed to be University, Indore, India

Abstract: The price of mobile phones significantly impacts their market success and sustainability. It is crucial to estimate the price of a mobile phone before its release to ensure optimal marketing and competitiveness with other products in the market. This prediction requires the collection of data on existing mobile phones and the application of different algorithms to reduce complexity and identify the most critical selection features for comparison within the data. Using machine learning techniques like forward and backward selection helps to find the best price with maximum specifications. In this study, we collected data on mobile phones' specifications and features like camera, video, processor quality, and material quality. The dataset is complex to collect due to the thousands of mobile phone releases each year. Therefore, we selectively choose features to reduce the dataset's complexity and get an estimate of the price to release the product in the market. The Logistic Regression predicts the mobile phone's price in three categories (low-priced, moderately priced, and highly priced).

***Index Terms* - Exploratory Analysis, Price Categorization, ML Modelling.**

I. INTRODUCTION

Mobile phones have become an essential part of modern life, with an estimated 5 billion people owning and using them worldwide. With new features, specifications, and designs being released every year, the mobile phone market has become highly competitive, with manufacturers vying for consumers' attention and money.

One of the critical factors in a mobile phone's market success is its price. Price affects not only the development and sustainability of the product but also its popularity and competitiveness with other products in the market. Consumers are more likely to purchase a mobile phone that fits their desired specifications at a reasonable price point. Therefore, it is essential to estimate a mobile phone's price before its release to ensure optimal marketing and competitiveness in the market.

In this context, the use of machine learning techniques like logistic regression, decision trees, and KNN models can be highly useful in predicting a mobile phone's optimal price point. These algorithms can help to identify the critical selection features for comparison within the data, allowing manufacturers and marketers to estimate the price of a mobile phone with maximum specifications.

This study focuses on collecting data on mobile phones' specifications and features like camera, video, processor quality, and material quality. Due to the thousands of mobile phone releases each year, collecting this data can be complex. Therefore, selective feature selection techniques like forward selection and backward selection are employed to reduce the dataset's complexity.

The logistic regression algorithm predicts the mobile phone's price in three categories (low-priced, moderately priced, and highly priced. The decision tree algorithm is employed to partition the data based on specific features to predict a mobile phone's price. The KNN model algorithm is used to calculate the distances between K models and test the dataset, with accuracy computed using both the KNN and training models.

Overall, this study highlights the importance of considering price in mobile phone development and the utility of machine learning algorithms in predicting a mobile phone's optimal price point. The combination of these machine learning techniques provides more accurate predictions and insights into the market, enabling manufacturers and marketers to make informed decisions about product features and competitive pricing.

II. LITERATURE REVIEW

Previous research has focused on using machine learning for price prediction and recommendation systems in various domains. The use of prior data to estimate the pricing of available and new launch products is an intriguing study background for machine-learning researchers.

Sameerchand-Pudaruth used multiple linear regression, k-nearest neighbors (KNN), decision trees, and Naive Bayes algorithms to predict the price of used automobiles in Mauritius [1]. The research revealed that the most used algorithms, decision tree, and Naive Bayes, are unsuitable for processing and forecasting numerical data. Due to the restricted number of accessible examples, the accuracy of the predictions was subpar.

Shonda Kuiper built a multivariate regression model to forecast the pricing of General Motors automobiles [2]. The study employed appropriate strategies for variable selection to identify the pertinent variables for inclusion in the model. This allowed students and researchers from diverse professions to assess the optimal settings for conducting experiments.

Mariana Listiani predicted the costs of leased automobiles using the support vector machine (SVM) technique [3]. The study determined that SVM is more accurate than multiple linear regression for price forecasting, especially when dealing with huge datasets. The SVM method was also excellent at managing high-dimensional data and avoiding under and over-fitting issues.

Comparing neural networks and hedonic methods for predicting property values, Limsombunchai [4] contrasted neural networks and hedonic approaches. The neural network strategy produced greater R-square and lower root mean square error (RMSE) values than the hedonic method, according to the study. Nevertheless, the study was hampered using predicted housing values rather than actual ones.

K Noor and Saddaqat J used multiple linear regression to predict the price of automobiles based on variables like vehicle type, manufacturer, region, edition, colour, mileage, alloy wheels, and power steering [5]. The research reached the highest level of precision and provided a technique for forecasting prices based on factors.

Previous research has utilized a variety of machine-learning approaches to forecasting the pricing of mobile devices based on their attributes. The research has emphasized the significance of selecting variables appropriately and the applicability of classification algorithms for ordinal data types. SVM has been demonstrated to be more accurate than multiple linear regression in predicting prices, although neural networks have shown promise in forecasting property values.

III. UNDERSTANDING DATASET

The focus of this research paper is on "Mobile Price Categorization Using Machine Learning: A Study on Feature Selection and Algorithmic Approaches". The dataset is sourced from the Kaggle data science community website [6].

The researchers used the Mobile Price Class dataset from the Kaggle data science community website to train the prediction model. This dataset categorizes mobile phones into different price ranges based on their features, such as battery capacity, RAM, weight, camera pixels, and more. The dataset consists of 3000 customer records, 21 attributes, including 20 features of mobile phones, and a class label that represents the price range. The class label is an ordinal data type with four values, ranging from 0 to 3, indicating an increasing degree of price. These values can be interpreted as economical, mid-range, flagship, and premium. The class label is the target variable indicating whether a customer will purchase a low, moderate, high, or very high-priced mobile phone. 20 features of the mobile phone are considered as input variables. Even though price is typically a numeric problem, the researchers used a classification approach in this study, as the class label contains discrete values. This approach was advantageous for using algorithms such as Naïve Bayes and Decision Tree, which are not well-suited for numeric data.

IV. RESEARCH METHODOLOGY

The present research attempts to predict mobile phone prices based on features as described in Table 1 (b).

A. *Data Collection*

The dataset of mobile prices along with different attributes was collected for the reason of variability in the prices of the mobile phones. The availability of the data in good volume with similarity in terms of attributes of mobile phones is the other reason for choosing the current dataset.

The dataset contained various attributes of mobile phones. Some of the prominent attributes of mobile phones such as Battery, Front Camera, Back Camera, RAM, Number of SIM slots, etc. are closely related to the classification and categorizing of mobile phones to provide justification for the analysis.

Table I: Dataset Collected

Feature	***Minimum***	***Maximum***
Battery power	501	1998
Front camera	0 px	19 px
Internal memory	4 gb	64 gb
RAM	256 mb	4gb

Table II: Description of attributes of Mobile Phones

Attribute	***Description***
battery power	The total energy a battery can store at one time is measured in mAh
blue	Has bluetooth or not
clock_speed	speed at which the microprocessor executes instructions
dual_sim	Has dual sim support or not
fc	Front Camera megapixels
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 Mega Bytes
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
price_range	This is the target variable with values of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).

B. Data Analysis

Exploratory Data Analysis is done after the Data collection. All the measures of central tendency are calculated, and the range is taken into consideration. After the exploratory data analysis, the data is split into test train splits. The price value is divided into 4 categories as shown below:

Table III: Price Range Category

Price Range	***Category***
0	low-priced
1	moderately priced
2	high-priced
3	very-high-priced

C. Data Visualization

Data visualization is done with the help of Matplotlib and Seaborn. Scatter plots, box plots, and violin plots are plotted. Scatter plots are plotted to check whether the attribute is affecting the price range. Box plots and Violin plots are plotted for outlier detection.

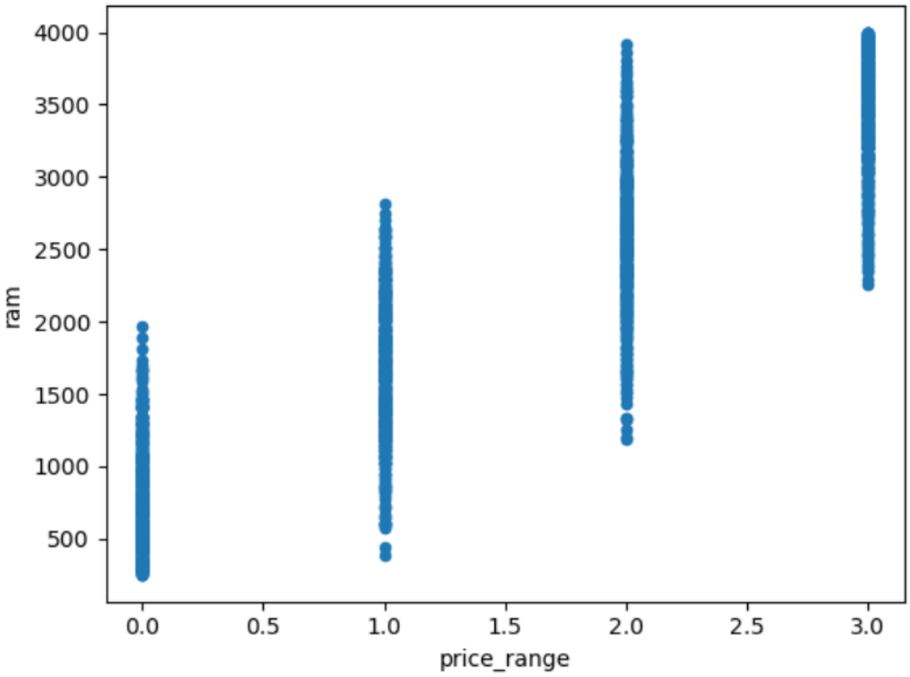


Fig. 1. Price Range vs Ram.

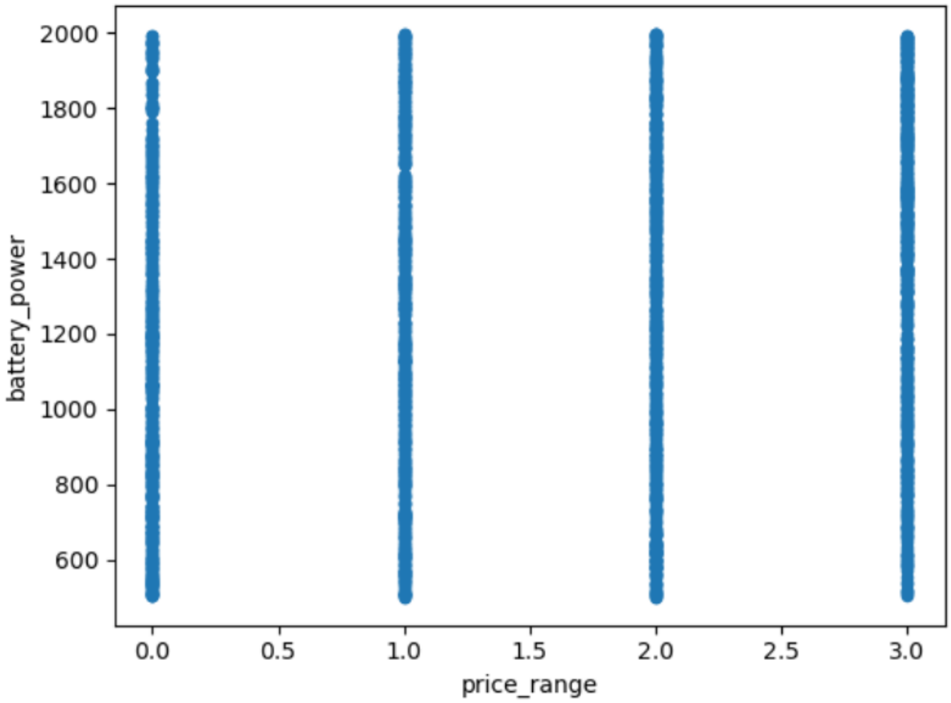


Fig. 2. Price Range vs Battery Power.

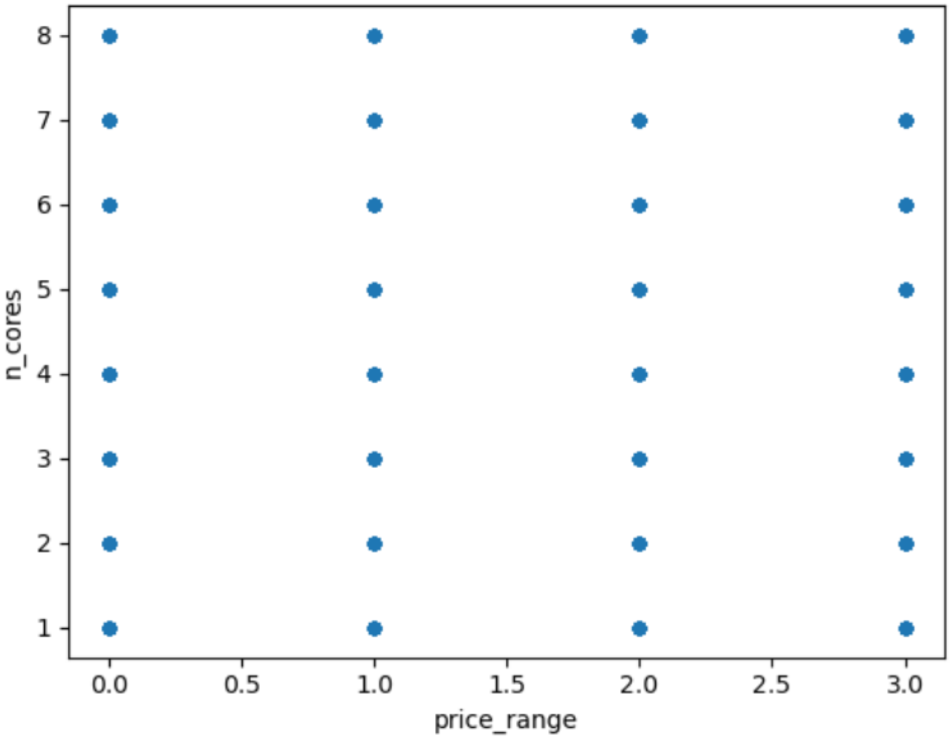


Fig. 3. Price Range vs N Cores.

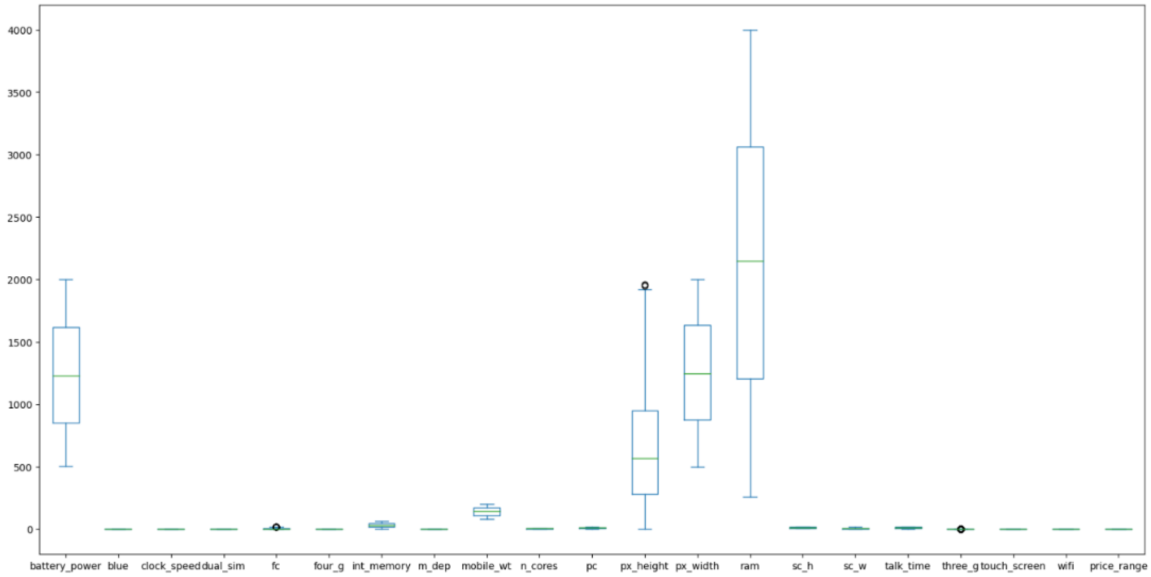
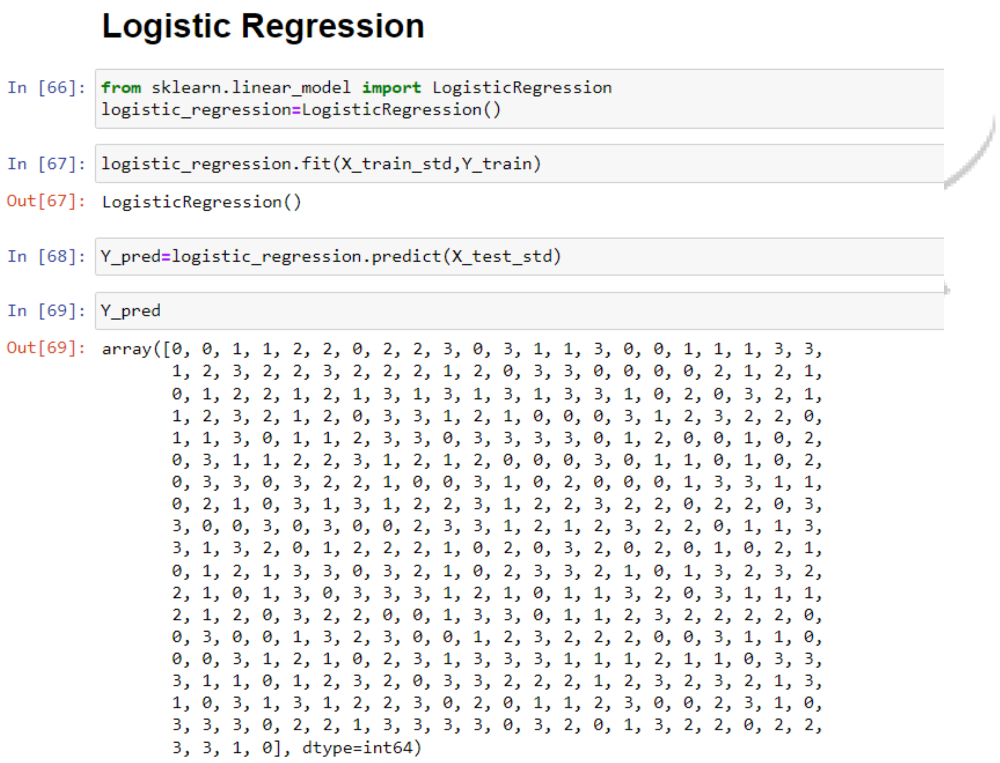


Fig. 4. Box plot for outlier detection.

D. Model Training

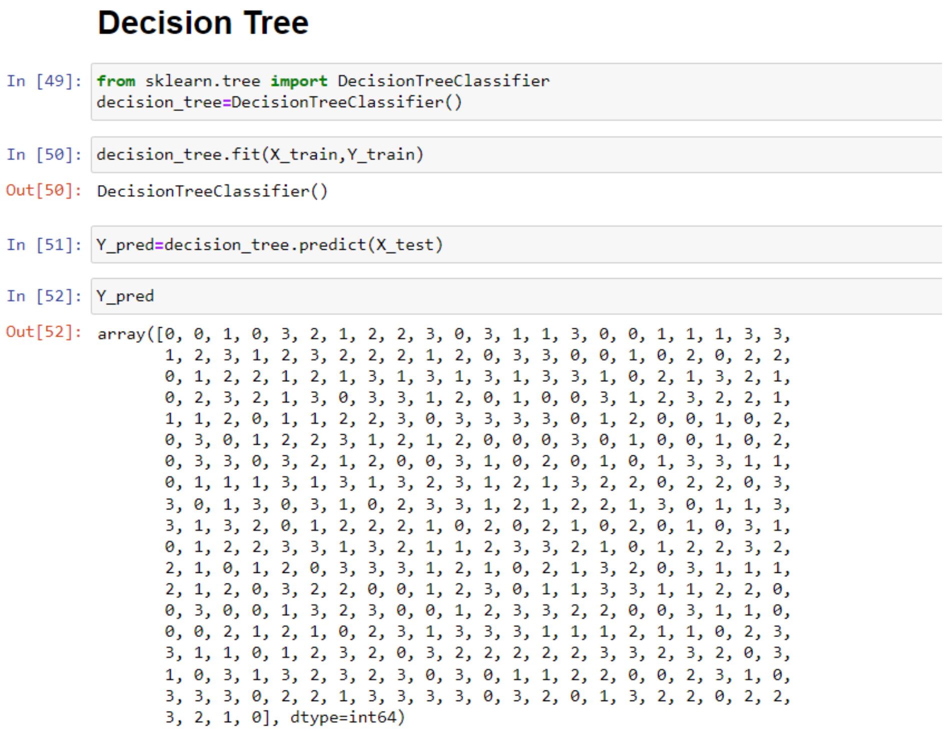
After the data visualization, we will train three different algorithms:



Logistic Regression

```
In [66]: from sklearn.linear_model import LogisticRegression
         logistic_regression=LogisticRegression()

In [67]: logistic_regression.fit(X_train_std,Y_train)

Out[67]: LogisticRegression()

In [68]: Y_pred=logistic_regression.predict(X_test_std)

In [69]: Y_pred

Out[69]: array([0, 0, 1, 1, 2, 2, 0, 2, 2, 3, 0, 3, 1, 1, 3, 0, 0, 1, 1, 1, 3, 3,
                1, 2, 3, 2, 2, 3, 2, 2, 2, 1, 2, 0, 3, 3, 0, 0, 0, 0, 2, 1, 2, 1,
                0, 1, 2, 2, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3, 3, 1, 0, 2, 0, 3, 2, 1,
                1, 2, 3, 2, 1, 2, 0, 3, 3, 1, 2, 1, 0, 0, 0, 3, 1, 2, 3, 2, 2, 0,
                1, 1, 3, 0, 1, 1, 2, 3, 3, 0, 3, 3, 3, 3, 0, 1, 2, 0, 0, 1, 0, 2,
                0, 3, 1, 1, 2, 2, 3, 1, 2, 1, 2, 0, 0, 0, 3, 0, 1, 1, 0, 1, 0, 2,
                0, 3, 3, 0, 3, 2, 2, 1, 0, 0, 3, 1, 0, 2, 0, 0, 0, 1, 3, 3, 1, 1,
                0, 2, 1, 0, 3, 1, 3, 1, 2, 2, 3, 1, 2, 2, 3, 2, 2, 0, 2, 2, 0, 3,
                3, 0, 0, 3, 0, 3, 0, 0, 2, 3, 3, 1, 2, 1, 2, 3, 2, 2, 0, 1, 1, 3,
                3, 1, 3, 2, 0, 1, 2, 2, 2, 1, 0, 2, 0, 3, 2, 0, 2, 0, 1, 0, 2, 1,
                0, 1, 2, 1, 3, 3, 0, 3, 2, 1, 0, 2, 3, 3, 2, 1, 0, 1, 3, 2, 3, 2,
                2, 1, 0, 1, 3, 0, 3, 3, 3, 1, 2, 1, 0, 1, 1, 3, 2, 0, 3, 1, 1, 1,
                2, 1, 2, 0, 3, 2, 2, 0, 0, 1, 3, 3, 0, 1, 1, 2, 3, 2, 2, 2, 2, 0,
                0, 3, 0, 0, 1, 3, 2, 3, 0, 0, 1, 2, 3, 2, 2, 2, 0, 0, 3, 1, 1, 0,
                0, 0, 3, 1, 2, 1, 0, 2, 3, 1, 3, 3, 3, 1, 1, 1, 2, 1, 1, 0, 3, 3,
                3, 1, 1, 0, 1, 2, 3, 2, 0, 3, 3, 2, 2, 2, 1, 2, 3, 2, 3, 2, 1, 3,
                1, 0, 3, 1, 3, 1, 2, 2, 3, 0, 2, 0, 1, 1, 2, 3, 0, 0, 2, 3, 1, 0,
                3, 3, 3, 0, 2, 2, 1, 3, 3, 3, 3, 0, 3, 2, 0, 1, 3, 2, 2, 0, 2, 2,
                3, 3, 1, 0], dtype=int64)
```

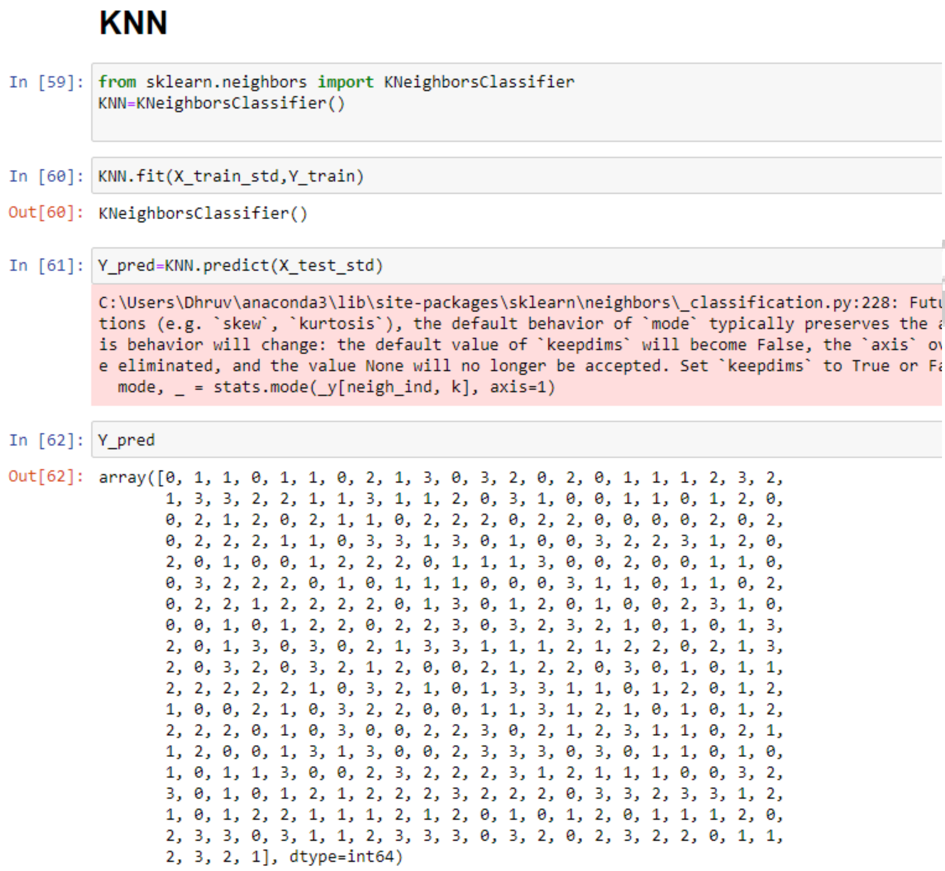
Fig. 5. Logistic Regression Model Training.



Decision Tree

```
In [49]: from sklearn.tree import DecisionTreeClassifier
         decision_tree=DecisionTreeClassifier()

In [50]: decision_tree.fit(X_train,Y_train)

Out[50]: DecisionTreeClassifier()

In [51]: Y_pred=decision_tree.predict(X_test)

In [52]: Y_pred

Out[52]: array([0, 0, 1, 0, 3, 2, 1, 2, 2, 3, 0, 3, 1, 1, 3, 0, 0, 1, 1, 1, 3, 3,
                1, 2, 3, 1, 2, 3, 2, 2, 2, 1, 2, 0, 3, 3, 0, 0, 1, 0, 2, 0, 2, 2,
                0, 1, 2, 2, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3, 3, 1, 0, 2, 1, 3, 2, 1,
                0, 2, 3, 2, 1, 3, 0, 3, 3, 1, 2, 0, 1, 0, 0, 3, 1, 2, 3, 2, 2, 1,
                1, 1, 2, 0, 1, 1, 2, 2, 3, 0, 3, 3, 3, 3, 0, 1, 2, 0, 0, 1, 0, 2,
                0, 3, 0, 1, 2, 2, 3, 1, 2, 1, 2, 0, 0, 0, 3, 0, 1, 0, 0, 1, 0, 2,
                0, 3, 3, 0, 3, 2, 1, 2, 0, 0, 3, 1, 0, 2, 0, 1, 0, 1, 3, 3, 1, 1,
                0, 1, 1, 1, 3, 1, 3, 1, 3, 2, 3, 1, 2, 1, 3, 2, 2, 0, 2, 2, 0, 3,
                3, 0, 1, 3, 0, 3, 1, 0, 2, 3, 3, 1, 2, 1, 2, 2, 1, 3, 0, 1, 1, 3,
                3, 1, 3, 2, 0, 1, 2, 2, 2, 1, 0, 2, 0, 2, 1, 0, 2, 0, 1, 0, 3, 1,
                0, 1, 2, 2, 3, 3, 1, 3, 2, 1, 1, 2, 3, 3, 2, 1, 0, 1, 2, 2, 3, 2,
                2, 1, 0, 1, 2, 0, 3, 3, 3, 1, 2, 1, 0, 2, 1, 3, 2, 0, 3, 1, 1, 1,
                2, 1, 2, 0, 3, 2, 2, 0, 0, 1, 2, 3, 0, 1, 1, 3, 3, 1, 1, 2, 2, 0,
                0, 3, 0, 0, 1, 3, 2, 3, 0, 0, 1, 2, 3, 3, 2, 2, 0, 0, 3, 1, 1, 0,
                0, 0, 2, 1, 2, 1, 0, 2, 3, 1, 3, 3, 3, 1, 1, 1, 2, 1, 1, 0, 2, 3,
                3, 1, 1, 0, 1, 2, 3, 2, 0, 3, 2, 2, 2, 2, 2, 3, 3, 2, 3, 2, 0, 3,
                1, 0, 3, 1, 3, 2, 3, 2, 3, 0, 3, 0, 1, 1, 2, 2, 0, 0, 2, 3, 1, 0,
                3, 3, 3, 0, 2, 2, 1, 3, 3, 3, 3, 0, 3, 2, 0, 1, 3, 2, 2, 0, 2, 2,
                3, 2, 1, 0], dtype=int64)
```

Fig. 6. Decision Tree Model Training.



KNN

```
In [59]: from sklearn.neighbors import KNeighborsClassifier
         KNN=KNeighborsClassifier()

In [60]: KNN.fit(X_train_std,Y_train)

Out[60]: KNeighborsClassifier()

In [61]: Y_pred=KNN.predict(X_test_std)

         C:\Users\Dhruv\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: Futu
         tions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the a
         is behavior will change: the default value of `keepdims` will become False, the `axis` ov
         e eliminated, and the value None will no longer be accepted. Set `keepdims` to True or Fa
           mode, _ = stats.mode(_y[neigh_ind, k], axis=1)

In [62]: Y_pred

Out[62]: array([0, 1, 1, 0, 1, 1, 0, 2, 1, 3, 0, 3, 2, 0, 2, 0, 1, 1, 1, 2, 3, 2,
                1, 3, 3, 2, 2, 1, 1, 3, 1, 1, 2, 0, 3, 1, 0, 0, 1, 1, 0, 1, 2, 0,
                0, 2, 1, 2, 0, 2, 1, 1, 0, 2, 2, 2, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2,
                0, 2, 2, 2, 1, 1, 0, 3, 3, 1, 3, 0, 1, 0, 0, 3, 2, 2, 3, 1, 2, 0,
                2, 0, 1, 0, 0, 1, 2, 2, 2, 0, 1, 1, 1, 3, 0, 0, 2, 0, 0, 1, 1, 0,
                0, 3, 2, 2, 2, 0, 1, 0, 1, 1, 1, 0, 0, 0, 3, 1, 1, 0, 1, 1, 0, 2,
                0, 2, 2, 1, 2, 2, 2, 2, 0, 1, 3, 0, 1, 2, 0, 1, 0, 0, 2, 3, 1, 0,
                0, 0, 1, 0, 1, 2, 2, 0, 2, 2, 3, 0, 3, 2, 3, 2, 1, 0, 1, 0, 1, 3,
                2, 0, 1, 3, 0, 3, 0, 2, 1, 3, 3, 1, 1, 1, 2, 1, 2, 2, 0, 2, 1, 3,
                2, 0, 3, 2, 0, 3, 2, 1, 2, 0, 0, 2, 1, 2, 2, 0, 3, 0, 1, 0, 1, 1,
                2, 2, 2, 2, 2, 1, 0, 3, 2, 1, 0, 1, 3, 3, 1, 1, 0, 1, 2, 0, 1, 2,
                1, 0, 0, 2, 1, 0, 3, 2, 2, 0, 0, 1, 1, 3, 1, 2, 1, 0, 1, 0, 1, 2,
                2, 2, 2, 0, 1, 0, 3, 0, 0, 2, 2, 3, 0, 2, 1, 2, 3, 1, 1, 0, 2, 1,
                1, 2, 0, 0, 1, 3, 1, 3, 0, 0, 2, 3, 3, 3, 0, 3, 0, 1, 1, 0, 1, 0,
                1, 0, 1, 1, 3, 0, 0, 2, 3, 2, 2, 2, 3, 1, 2, 1, 1, 1, 0, 0, 3, 2,
                3, 0, 1, 0, 1, 2, 1, 2, 2, 2, 3, 2, 2, 2, 0, 3, 3, 2, 3, 3, 1, 2,
                1, 0, 1, 2, 2, 1, 1, 1, 2, 1, 2, 0, 1, 0, 1, 2, 0, 1, 1, 1, 2, 0,
                2, 3, 3, 0, 3, 1, 1, 2, 3, 3, 3, 0, 3, 2, 0, 2, 3, 2, 2, 0, 1, 1,
                2, 3, 2, 1], dtype=int64)
```

Fig. 7. K-Nearest Neighbour Model Training.

Then by using these algorithms, we will classify the given mobile's price range.

E. Model Evaluation

We train three different models to decide which of the models is predicting the price range correctly. The Confusion Matrix is the metric that is used for model evaluation. Logistic Regression is the most appropriate model to be used in the prediction with 96% accuracy.

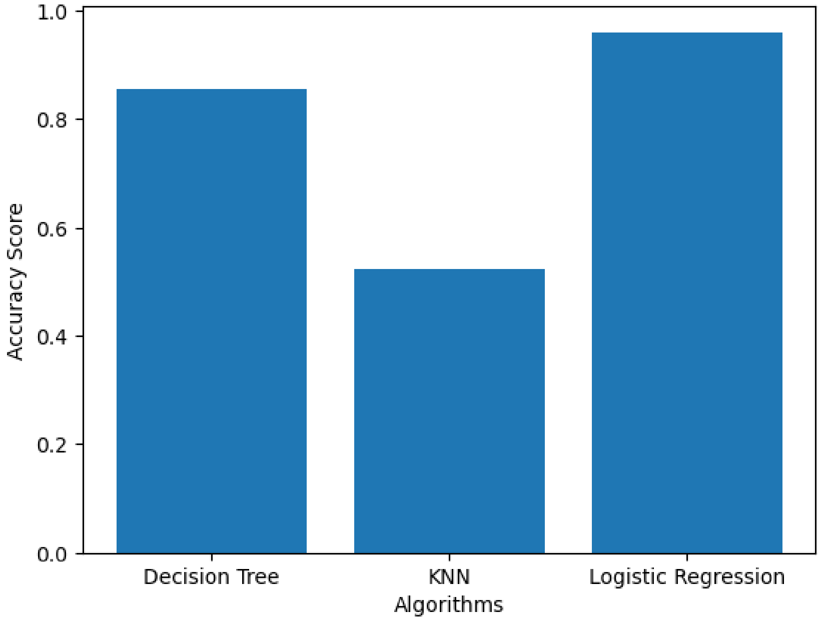
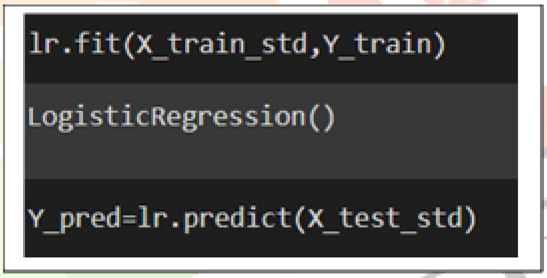


Fig. 8. Accuracy Scores for the trained algorithm.

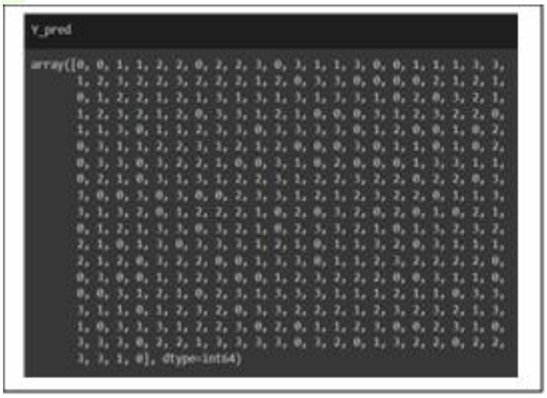
F. Data Classification

Using Logistic Regression, we classify the mobile price ranges.



```
lr.fit(X_train_std,Y_train)

LogisticRegression()

Y_pred=lr.predict(X_test_std)
```

Fig. 9. Fitting the data set for the Logistic Regression model and using it for train and test data for prediction.



```
Y_pred

array([0, 0, 1, 1, 2, 2, 0, 2, 2, 3, 0, 3, 1, 1, 3, 0, 0, 1, 1, 1, 3, 3,
       1, 2, 3, 2, 2, 3, 2, 2, 2, 1, 2, 0, 3, 3, 0, 0, 0, 0, 2, 1, 2, 1,
       0, 1, 2, 2, 1, 2, 1, 3, 1, 3, 1, 3, 1, 3, 3, 1, 0, 2, 0, 3, 2, 1,
       1, 2, 3, 2, 1, 2, 0, 3, 3, 1, 2, 1, 0, 0, 0, 3, 1, 2, 3, 2, 2, 0,
       1, 1, 3, 0, 1, 1, 2, 3, 3, 0, 3, 3, 3, 3, 0, 1, 2, 0, 0, 1, 0, 2,
       0, 3, 1, 1, 2, 2, 3, 1, 2, 1, 2, 0, 0, 0, 3, 0, 1, 1, 0, 1, 0, 2,
       0, 3, 3, 0, 3, 2, 2, 1, 0, 0, 3, 1, 0, 2, 0, 0, 0, 1, 3, 3, 1, 1,
       0, 2, 1, 0, 3, 1, 3, 1, 2, 2, 3, 1, 2, 2, 3, 2, 2, 0, 2, 2, 0, 3,
       3, 0, 0, 3, 0, 3, 0, 0, 2, 3, 3, 1, 2, 1, 2, 3, 2, 2, 0, 1, 1, 3,
       3, 1, 3, 2, 0, 1, 2, 2, 2, 1, 0, 2, 0, 3, 2, 0, 2, 0, 1, 0, 2, 1,
       0, 1, 2, 1, 3, 3, 0, 3, 2, 1, 0, 2, 3, 3, 2, 1, 0, 1, 3, 2, 3, 2,
       2, 1, 0, 1, 3, 0, 3, 3, 3, 1, 2, 1, 0, 1, 1, 3, 2, 0, 3, 1, 1, 1,
       2, 1, 2, 0, 3, 2, 2, 0, 0, 1, 3, 3, 0, 1, 1, 2, 3, 2, 2, 2, 2, 0,
       0, 3, 0, 0, 1, 3, 2, 3, 0, 0, 1, 2, 3, 2, 2, 2, 0, 0, 3, 1, 1, 0,
       0, 0, 3, 1, 2, 1, 0, 2, 3, 1, 3, 3, 3, 1, 1, 1, 2, 1, 1, 0, 3, 3,
       3, 1, 1, 0, 1, 2, 3, 2, 0, 3, 3, 2, 2, 2, 1, 2, 3, 2, 3, 2, 1, 3,
       1, 0, 3, 1, 3, 1, 2, 2, 3, 0, 2, 0, 1, 1, 2, 3, 0, 0, 2, 3, 1, 0,
       3, 3, 3, 0, 2, 2, 1, 3, 2, 3, 3, 0, 3, 2, 0, 1, 3, 2, 2, 0, 2, 2,
       3, 3, 1, 0], dtype=int64)
```

Fig. 10. Used scalar transformation and predicted the range based on the transformed array values.

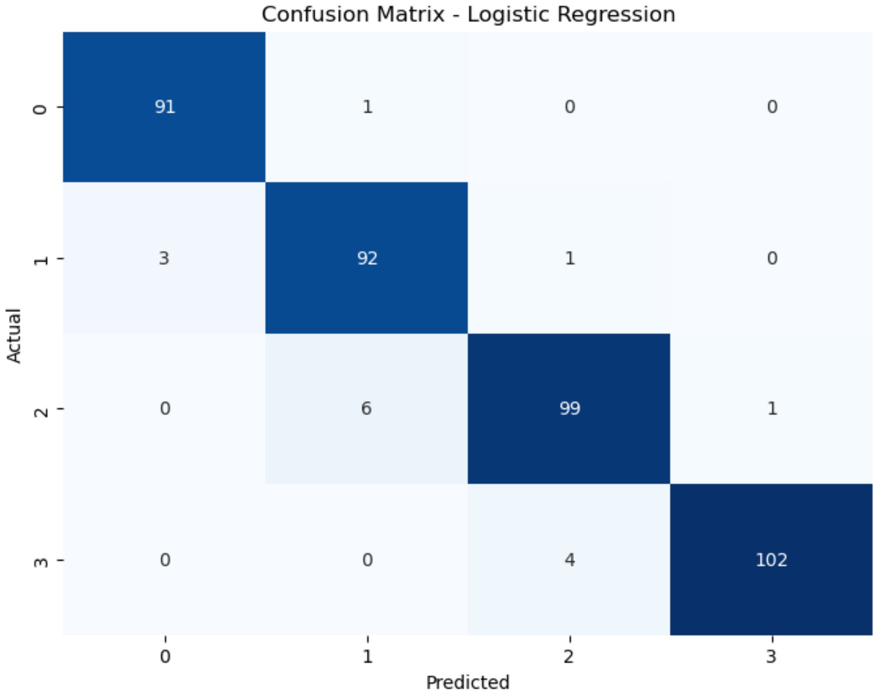


Fig. 11. Confusion Matrix for Logistic Regression

The Logistic Regression model has demonstrated high effectiveness in classifying the instances correctly, as indicated by the Precision, Recall, and F1 Score of 0.96.

Precision of 0.96: This indicates that 96% of the instances that were predicted as positive by the model were actually positive. It implies that the model has a very low false positive rate, making it highly reliable in its positive predictions.

High precision is crucial in situations where the cost of false positives is high, and it demonstrates that the model is highly specific in its predictions.

Recall of 0.96: This denotes that the model has correctly identified 96% of all actual positive instances. It signifies that the model has a low false negative rate, capturing most of the positive instances accurately. High recall is essential in situations where identifying all actual positive instances is critical, reflecting that the model is highly sensitive to the positive class.

F1 Score of 0.96: The F1 Score is the harmonic mean of Precision and Recall, and it is particularly useful when the class distribution is imbalanced. An F1 Score of 0.96 indicates a well-balanced model, with both Precision and Recall being high.

It implies that the model has achieved a balance between specificity (Precision) and sensitivity (Recall), making it robust and reliable in various scenarios.

The Logistic Regression model exhibits excellent predictive performance, with high reliability and accuracy in identifying the correct classes. The high Precision, Recall, and F1 Score indicate that the model is robust, making very few false predictions, and is adept at identifying the positive instances correctly. This model can be considered highly trustworthy for making predictions on unseen data, given its balanced and high Precision, Recall, and F1 Score.

V. CONCLUSION

In conclusion, the mobile price classification project was successful in accurately predicting the price range of mobile phones using a linear regression algorithm. The dataset was divided into four price ranges ranging from 0 to 3, and the logistic regression model was found to be the most accurate among the ones tested, achieving an accuracy of 96%. This demonstrates the effectiveness of machine learning algorithms in predicting the prices of products based on various features. The results of this project can be used by businesses to make informed decisions about pricing strategies and provide customers with better recommendations based on their budgets and needs. Overall, this project highlights the potential of machine learning in solving real-world problems and improving business operations.

VI. DEVELOPMENT AND FUTURE WORK

Predicting costs is a crucial aspect of marketing and business. For all sorts of things, including automobiles, food, medicine, laptops, etc., the same method may be used to estimate their costs.

The most effective marketing technique is to identify the product with the lowest price and highest quality. So, items may be compared based on their specs, prices, manufacturers, etc. By identifying a customer's budget, it is possible to offer a suitable product. With more complex artificial intelligence approaches, it is possible to maximize accuracy and accurately estimate the prices of items. It is possible to construct software or a mobile application that can anticipate the market price of any newly announced goods.

To reach maximum accuracy and make more accurate predictions, it is necessary to add more and more cases to the data collection. Moreover, picking more pertinent characteristics might improve accuracy. To attain more precision, it is necessary to choose more pertinent characteristics from a larger data collection.

REFERENCES

[1] Sameerchand Pudaruth . “Predicting the Price of Used Cars using Machine Learning Techniques”, International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 753-764.

[2] Shonda Kuiper, "Multiple Regression Introduction: How much is your car worth? "November 2008, Journal of Statistics Education

[3] Mariana Listiani , 2009. “Support Vector Regression Analysis for Price Prediction in a Car Leasing Application”. Master Thesis. Hamburg University of Technology.

[4] Limsombunchai, V. 2004. “House Price Prediction: Hedonic Price Model vs. Artificial Neural Network”, New Zealand Agricultural and Resource Economics Society Conference, New Zealand, pp. 25-26. 2004

[5] Kanwal Noor and Sadaqat Jan, “Vehicle Price Prediction System using Machine Learning Techniques”, International Journal of Computer Applications (0975 – 8887) Volume 167 – No.9, June 2017.

[6] https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification/code?resource=download

[7] Jianglin Huang, Yan-Fu Li, Min Xie, “An empirical analysis of data preprocessing for machine learning-based software cost estimation”, Information and Software Technology, Volume 67, 2015, Pages 108-127, ISSN 0950-5849, doi.org/10.1016/j.infsof.2015.07.004.