Predictive Analysis of Mobile Device Pricing: A Machine Learning Approach

Muhammed Shibil C V
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, India
muhammed4shibil@gmail.com

***Abstract*— This research delves into the predictive analysis of mobile device pricing based on a diverse set of features. The dataset comprises key attributes such as battery power, Bluetooth capability, clock speed, dual SIM support, front and primary camera specifications, 4G and 3G connectivity, internal memory, processor details, screen dimensions, and other relevant parameters. The primary aim is to develop a robust predictive model capable of estimating the price range of mobile devices. To achieve this, various machine learning algorithms, including but not limited to Linear Regression, K-Nearest Neighbors, Logistic Regression, Decision Tree, and Random Forest, are employed. Additionally, feature selection techniques are applied to streamline the dataset and enhance computational efficiency.**

Keywords—Linear Regression, K-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest.

1. INTRODUCTION

Price is a pivotal element in the realm of marketing and business, acting as a primary concern for industries. The initial query that resonates across sectors is the feasibility of potential buyers acquiring items based on specific specifications. In this context, machine learning emerges as a potent ally, offering robust techniques such as supervised and unsupervised learning. Notably, versatile tools like Python serve as indispensable resources for executing machine learning tasks.

Within the machine learning framework, a variety of classifiers, including but not limited to Linear Regression and K-Nearest Neighbors (KNN), present themselves as formidable options. Concurrently, feature selection algorithms prove invaluable, enabling the identification of optimal features while minimizing the dataset and, consequently, reducing computational complexity.

The journey of machine learning prediction involves a systematic seven-step process, encompassing data gathering, data preparation, model selection, training, evaluation, hyperparameter tuning, and ultimately, prediction.

In the contemporary landscape, mobile devices stand out as one of the most ubiquitous and sought-after commodities. The market witnesses a continuous influx of new mobiles featuring advancements and enhancements. Numerous factors play a crucial role in estimating the price of a mobile device, including the processor, battery capacity, size, thickness, internal memory, camera specifications, video quality, and internet browsing capabilities.

2. LITERATURE REVIEW

Price prediction, particularly in the context of mobile devices, has garnered significant attention in the machine learning domain. Several studies have contributed valuable insights into this realm.

The work titled "Mobile Price Prediction by its Features Using Predictive Model of Machine Learning" [1] is a notable exploration of predictive modeling techniques for mobile price estimation. Emphasizing the importance of features, the paper implements multiple techniques such as multiple linear regression, K-nearest neighbors, decision tree, and Naive Bayes.

In a similar vein, the study "Mobile Phone Price Class Prediction Using Different Classification Algorithms with Feature Selection and Parameter Optimization" [2] delves into the application of various classification algorithms for mobile price classification. The research highlights the significance of feature selection methods, such as ANOVA f-test and mutual information score, as well as the impact of preprocessing and parameter optimization on model accuracy.

Aiming to develop a model for predicting mobile phone prices, the research [3] utilizes supervised machine learning algorithms, including Naive Bayes, K-NN, and Random Forest. The study evaluates the models using metrics such as confusion matrix, classification report, and accuracy score.

Beyond the realm of mobile devices, research on price prediction extends to other domains. Notably, "House Resale Price Prediction Using Classification Algorithms" [4] and [5] explore diverse approaches for predicting house resale prices. Techniques such as fuzzy logic, artificial neural networks, K-nearest neighbors, and data-driven fuzzy rule extraction are employed to forecast housing values.

In the context of housing price direction, the paper "Predicting the Housing Price Direction using Machine Learning Techniques" [6] addresses the classification problem of whether house prices will rise or fall. The study employs various feature selection and data transformation techniques, measuring the performance of machine learning models through parameters like accuracy, precision, specificity, and sensitivity.

Lastly, the study "Predicting the Price of Used Cars using Machine Learning Techniques" [7] fills a gap in predicting used car prices. It reviews various methodologies, including support vector machines, hybrid car valuation, and artificial neural networks, demonstrating the growing interest in predicting prices for second-hand vehicles.

These studies collectively underscore the significance of predictive modeling, encompassing various machine learning algorithms and techniques, in estimating prices across diverse domains.

3. MACHINE LEARNING MODELS

LINEAR REGRESSION

Linear Regression is a fundamental machine learning model employed for predictive analysis. In the context of mobile price prediction, this model establishes a linear relationship between the input features and the target variable, which, in this case, is the mobile price. The model assumes a linear correlation, allowing it to predict the price based on a weighted sum of the input features.

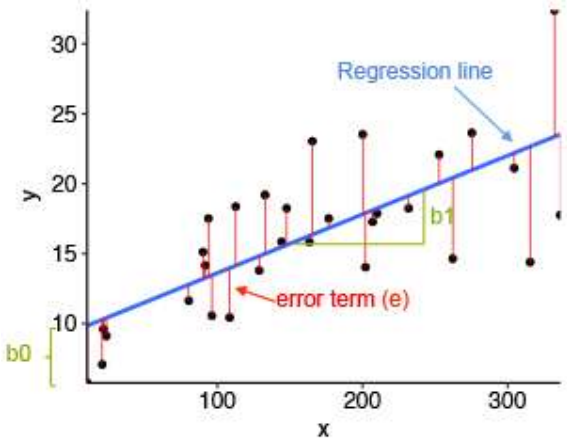

Fig. 1. Linear Regression
Analyticsvidhya

Despite its simplicity, Linear Regression provides valuable insights into the linear dependencies within the dataset, serving as a foundational benchmark for more complex models.

K-NEAREST NEIGHBORS (KNN)

K-Nearest Neighbors, or KNN, is a versatile algorithm used for both classification and regression tasks. In mobile price prediction, KNN operates by classifying or estimating the price of a mobile device based on the majority class or average of its nearest neighbors in the feature space. This model is particularly effective when considering local patterns and dependencies in the dataset. The choice of an optimal 'k' value, representing the number of neighbors considered, is crucial in fine-tuning the accuracy of the predictions.

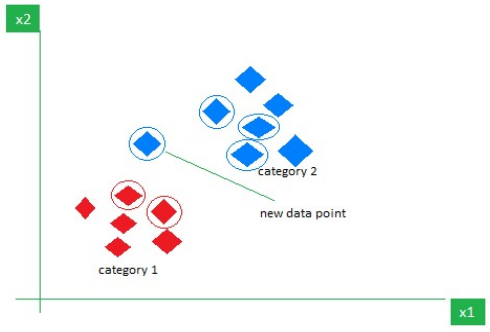

Fig. 2. K
geeksforgeeks

LOGISTIC REGRESSION

Contrary to its name, Logistic Regression is primarily utilized for binary classification problems. In the context of mobile price prediction, it can be adapted to classify mobile devices into discrete price categories. Logistic Regression models the probability of an instance belonging to a particular class, making it a valuable tool for understanding the likelihood of a mobile falling into predefined price ranges. Despite its name, Logistic Regression is a classification algorithm, not a regression algorithm.

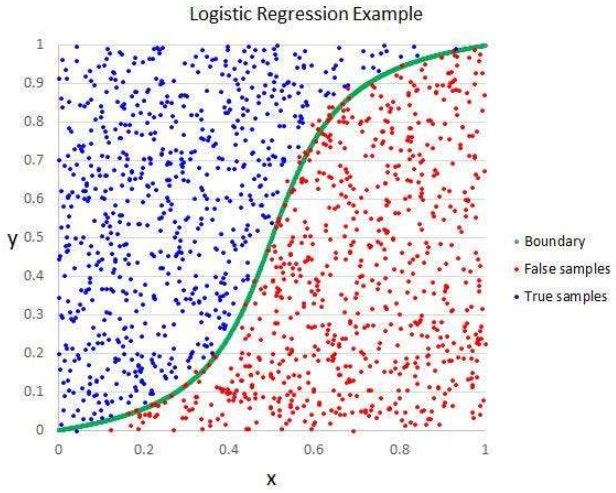

Fig. 3. Logistic Regression
medium.com

DECISION TREE

A Decision Tree is a tree-like model that recursively splits the dataset into subsets based on the most significant features. Each internal node of the tree represents a decision based on a specific feature, leading to a final prediction at the leaf nodes. In mobile price prediction, Decision Trees can capture complex relationships between features, providing interpretability and insights into the decision-making process. However, they are prone to overfitting, and strategies like pruning are often employed to enhance generalization.

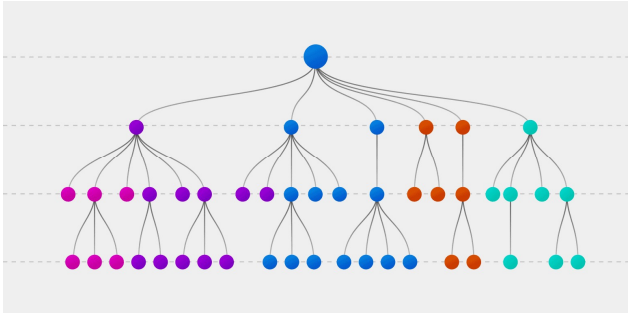

Fig. 4. Decision Tree
datascienceprophet.com

RANDOM FOREST

Random Forest is an ensemble learning technique that combines multiple Decision Trees to improve predictive accuracy and control overfitting. In the context of mobile price prediction, a Random Forest aggregates the predictions of numerous Decision Trees, each trained on different subsets of the data.

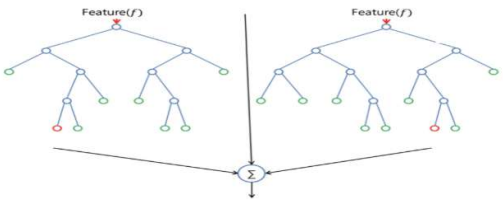

Fig. 5. Random Forest
builtin.com

4. UNDERSTANDING THE DATASET

The predictive model was trained using the Mobile Price Class dataset obtained from the Kaggle data science community website at https://www.kaggle.com/iabhishekofficial/mobile-price-classification. This dataset classifies mobile devices into distinct price ranges, leveraging a total of 21 attributes. Among these attributes, 20 serve as features, encompassing battery capacity, RAM, weight, camera pixels, and more. The remaining attribute is the class label, denoting the price range. The class label consists of ordinal data with four values – 0, 1, 2, and 3. These values represent increasing degrees of price, signifying economical, mid-range, flagship, and premium categories, respectively.

Despite the traditional numeric nature of pricing problems, the machine learning approach chosen is classification rather than regression. This choice is influenced by the discrete values in the class label, where each value corresponds to a specific price range category. The discrete nature of the class labels proves advantageous, especially when utilizing algorithms like Naïve Bayes and Decision Trees, which typically excel in handling categorical data rather than continuous numeric data.

```
In [57]: dataset.columns
Out[57]: Index(['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', 'wifi', 'price_range'],
               dtype='object')
```

Columns in Dataset

```
In [6]: dataset.describe()
```

Out[6]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000

5. DATA VISUALIZATION & ANALYSIS

Explore the relationship between internal memory and price range through insightful data visualization and analysis. Uncover how variations in internal memory influence mobile device pricing. Gain valuable insights into pricing dynamics based on internal memory capacities.

Internal Memory vs Price Range:

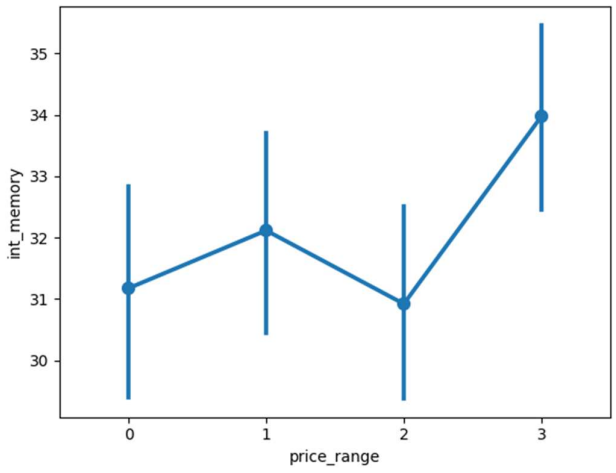

Fig. 6. Internal Memory vs Price Range

Percentage of phones that support 3G:

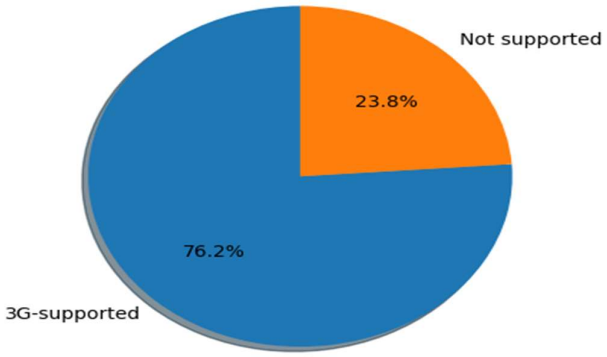

Fig. 7. Percentage of phones that support 3G

Percentage of phones that support 4G:

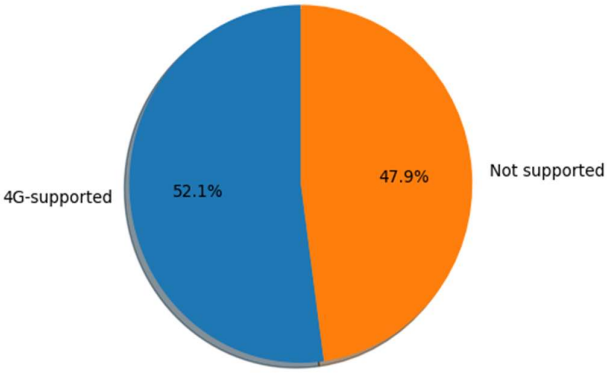

Fig. 8. Percentage of phones that support 4G

Battery power vs Price Range:

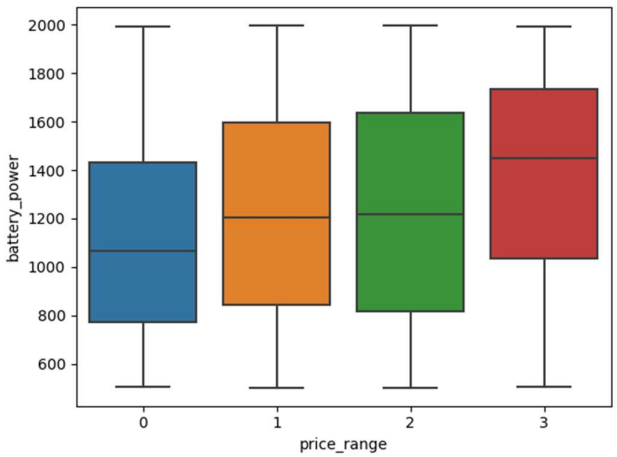

Fig. 8. Battery power vs Price range

No of Phones vs Camera megapixels of front and primary camera:

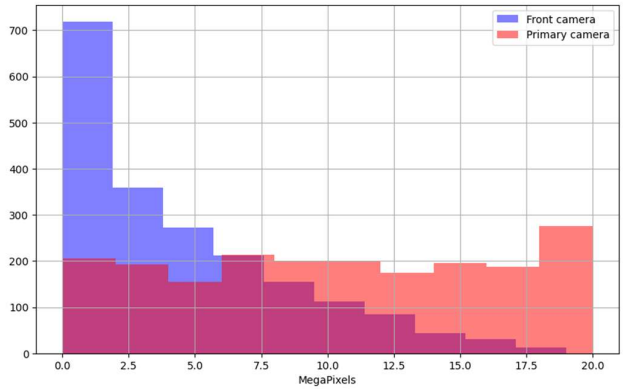

Fig. 9. No. of Phones vs Camera megapixels

Mobile Weight vs Price range:

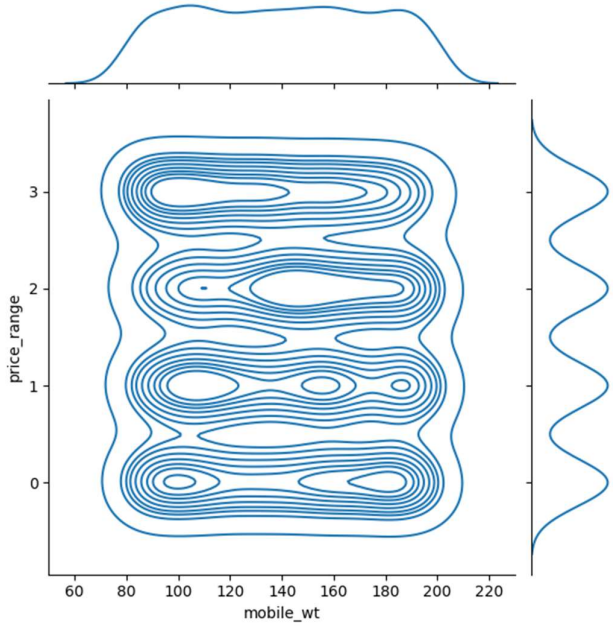

Fig. 10. Mobile weight vs Price range

Talk time vs Price range:

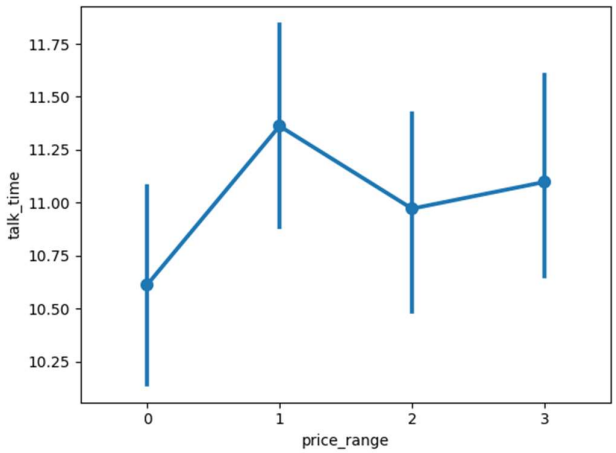

Fig. 11. Talk time vs Price range

6. TRAINING THE PREDICTION MODEL

Splitting the data

```
In [19]: from sklearn.model_selection import train_test_split
In [20]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=101)
```

The dataset is split into two segments, with a test size of 0.2 indicating that 80% of the data is allocated for training the prediction model, while the remaining portion is employed to evaluate the efficacy of the developed model.

Creating & Training Linear Regression Model

```
In [21]: from sklearn.linear_model import LinearRegression
         lm = LinearRegression()
In [22]: lm.fit(X_train,y_train)
Out[22]: LinearRegression()
In [23]: lm.score(X_test,y_test)
Out[23]: 0.913280148185277
```

Linear Regression was used here to train the prediction model. And got 91% Accuracy.

Creating & Training KNN Model

```
In [24]: from sklearn.neighbors import KNeighborsClassifier
         knn = KNeighborsClassifier(n_neighbors=10)
         knn.fit(X_train,y_train)
Out[24]: KNeighborsClassifier(n_neighbors=10)
In [25]: knn.score(X_test,y_test)
         C:\Users\user\Anaconda2022\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning:
         Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically
         preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `k
         eepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the
         value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
           mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
Out[25]: 0.9212121212121213
```

KNN was used here to train the prediction model with the accuracy of 92%.

Elbow Method For optimum value of K

```
In [26]: error_rate = []
         for i in range(1,20):

             knn = KNeighborsClassifier(n_neighbors=i)
             knn.fit(X_train,y_train)
             pred_i = knn.predict(X_test)
             error_rate.append(np.mean(pred_i != y_test))
```

Elbow method used for finding the optimum value of K

Creating & Training Logistic Regression Model

```
In [28]: from sklearn.linear_model import LogisticRegression
         logmodel = LogisticRegression()
```

Logistic Regression also used here to train the prediction model. It got the accuracy of 61%. Comparatively weak.

Creating & Training Decision Tree Model

```
In [31]: from sklearn.tree import DecisionTreeClassifier
         dtree = DecisionTreeClassifier()
In [32]: dtree.fit(X_train,y_train)
Out[32]: DecisionTreeClassifier()
In [33]: dtree.score(X_test,y_test)
Out[33]: 0.8242424242424242
```

Decision Tree also applied and got 82% accuracy.

Creating & Training Random Forest Model

```
In [37]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier(n_estimators=200)
         rfc.fit(X_train, y_train)
Out[37]: RandomForestClassifier(n_estimators=200)
In [38]: rfc.score(X_test,y_test)
Out[38]: 0.8651515151515151
```

Random forest model also applied and got 86% of accuracy.

7. TRAINING THE PREDICTION MODEL

The evaluation of algorithms in this paper involves utilizing key metrics such as the confusion matrix, classification report, and accuracy score. The confusion matrix displays the count of correctly classified instances across its diagonal and the count of misclassified instances elsewhere. With 4 class values, the resulting matrix is a 4x4 configuration. A comprehensive classification report provides detailed insights into parameters like recall, precision, and f1-score. The accuracy score

quantifies the performance of the trained model, determined by evaluating it with test data, representing 20% of the dataset.
Conclusion: KNN & Linear Regression performed the best

RESULT: Linear Regression

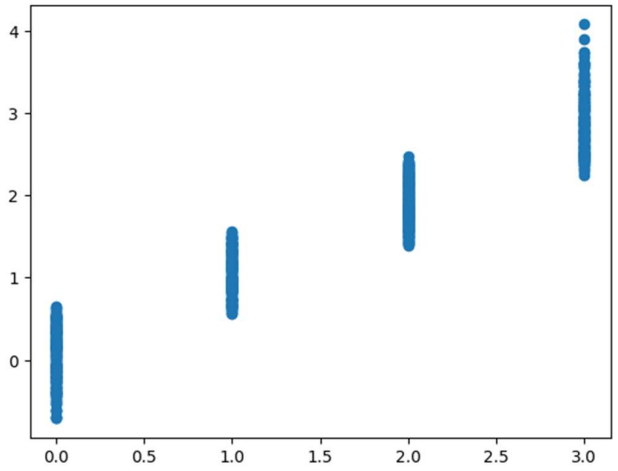

Fig. 12. Linear regression result

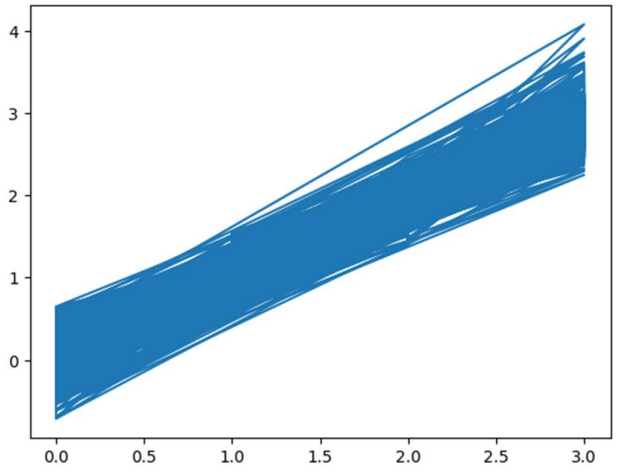

Fig. 13. Linear regression result

RESULT: KNN

```
In [44]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.96	0.99	0.97	158
1	0.88	0.95	0.91	152
2	0.93	0.85	0.89	199
3	0.92	0.93	0.93	151
accuracy			0.93	660
macro avg	0.93	0.93	0.93	660
weighted avg	0.93	0.93	0.93	660

```
In [45]: matrix=confusion_matrix(y_test,pred)
         print(matrix)

[[156   2   0   0]
 [  6 144   2   0]
 [  0  17 170  12]
 [  0   0  10 141]]
```

Among the tested algorithms, KNN demonstrated the highest accuracy in classifying instances, achieving a noteworthy 92.75%. Following closely is the Linear Regression model, which secured an accuracy of 91%. In contrast, Logistic Regression exhibited comparatively weaker performance with an accuracy of 61%. The Decision Tree classifier attained an accuracy of 82%, and the Random Forest model performed well with an accuracy of 86%.

8. CONCLUSION

The model trained using KNN demonstrated the highest accuracy in predicting mobile price classes (92.75%). Linear Regression achieved an accuracy of 91%, while Logistic Regression exhibited comparatively weaker performance with 61%. The Decision Tree classifier attained an accuracy of 82%, and the Random Forest model performed well with an accuracy of 86%. To further enhance model accuracy, implementing data preprocessing steps such as normalization and standardization can be beneficial. Additionally, the application of feature selection and extraction algorithms can help eliminate unsuitable and redundant features, resulting in improved outcomes. The methodology employed in this study is transferable and can be applied to predict prices for various products, including cars, bikes, houses, etc., utilizing archival data with relevant features. This approach facilitates more informed decision-making for both organizations and consumers in the realm of pricing.

9. REFERENCES

[1] Gupta, A. A., & Vijaykumar, S. (2020). Mobile Price Prediction by its Features Using Predictive Model of Machine Learning. UGC Care Journal, 40(35), 906–913. Retrieved from https://www.researchgate.net/publication/362491098

[2] Cetin, M., & Koc, Y. (2021). Mobile Phone Price Class Prediction Using Different Classification Algorithms with Feature Selection and Parameter Optimization. In ISMSIT 2021 - 5th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings (pp. 483–487). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ISMSIT52890.2021.9604550Cortes, Ivan (2017). Automatic Positioning System for Inductive Wireless Charging Devices and Application to Mobile Robot. Master's thesis, Texas A & M University. Available electronically from https://doi.org/10.1109/ISMSIT52890.2021.9604550

[3] Varun Kiran, A. (2022). Prediction of Mobile Phone Price Class using Supervised Machine Learning Techniques. International Journal of Innovative Science and Research Technology, 7(1), 248–251. Retrieved from www.ijisrt.com248

[4] Aboul Ella Hassanien Oscar Castillo Sameer Anand Ajay Jaiswal Editors International Conference on Innovative Computing and Communications Proceedings of ICICC 2023, Volume 3.

[5] Durganjali, P., & Pujitha, M. V. (2019). House Resale Price Prediction Using Classification Algorithms. In 6th IEEE International Conference on "Smart Structures and Systems", ICSSS 2019. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICSSS.2019.8882842

[6] Banerjee, D., & Dutta, S. (2018). Predicting the housing price direction using machine learning techniques. In IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017 (pp. 2998–3000). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICPCSI.2017.8392275

[7] Pudaruth, S. (2014). Predicting the Price of Used Cars using Machine Learning Techniques. International Journal of Information & Computation Technology (Vol. 4, pp. 753–764). Retrieved from http://www.irphouse.com