Mobile Price Classification Using Machine Learning

***Abstract*— By producing this forecast, the major objective of this work on the project is to provide a response to the question "Will the mobile device with the features that have been offered be categorized as 0 extremely cheap, 1 affordable, 2 pricey, or 3 very expensive?" Different methods of feature selection are applied in order to recognize and get rid of characteristics that are thought to be of less significance or are considered to be redundant and have the least amount of computational complexity. A number of different types of classifiers are applied so that we may get the best possible degree of accuracy. The findings are compared using the highest level of accuracy that was achieved in conjunction with the fewest number of variables that were selected. The conclusion is reached by using the best feature selection method and the best classifier to the dataset that was supplied. This allows for the most accurate results. The findings of this research may be used by any form of business or organization to establish which product is the most desirable.**

***Index Terms*— Mobile, Price, Classification, machine, Learning**

I. Introduction

The price at which a product or service is offered to customers is the single most important factor to consider when it comes to marketing and running a business. Before making a purchase from a company, the cost of the merchandise is nearly always the very first thing that potential customers want to find out about the company. Every potential purchaser at first experiences anxiety and wonders to themselves, "Will he or will he not be able to purchase anything that satisfies the standards that have been established?" As a direct consequence of this, the primary goal of the work is to simplify the process of price estimates so that individuals can carry it out from the comfort of their own homes. The reading of this piece of literature is just the beginning of a much longer journey that will ultimately lead readers to the location that was mentioned earlier in this discussion. Artificial intelligence, also known as the process of providing a computer with the ability to provide intelligent responses to questions that are presented to it, is a very vast field of research within the engineering discipline in the modern day and age. The study of machine learning has provided us with a variety of essential strategies for the development of artificial intelligence. These strategies include classification, regression, supervised learning, unsupervised learning, and a great lot of additional approaches.

In the modern society we currently inhabit, mobile phones are one of the electronic devices that are traded in and out of circulation the most regularly. Every day, new mobiles are introduced to the market that are superior to their predecessors in terms of the software they use and the feature sets they offer. Every day, there are deals that include the buying and selling of hundreds of thousands, if not even more, mobile phones. Some estimates put the number even higher. As a consequence of this, the projection of the mobile price class might be interpreted as a case study for the kind of problem that was described, specifically the method of determining which product is the most appropriate one. It is possible to use the same procedure to arrive at an estimate of the true cost of a broad variety of things, including automobiles, bicycles, generators, motors, food items, drugs, and so on and so forth. This may be accomplished by following a series of steps.

In order to arrive at an appropriate cost estimate for a portable electronic device, it is required to take into consideration quite a number of the many different aspects that are available. Take, for instance, the central processing unit that may be found inside the mobile device. In today's fast-paced, technologically advanced world, the average individual leads a life that makes battery life a very crucial factor to consider. Measures of the mobile device, including both its overall size and its thickness, should also be taken into consideration. These measurements should be taken into consideration. It is absolutely necessary to take into consideration the amount of storage space available on the device, in addition to the number of camera pixels and the quality of the recorded video. Simply being able to find one's way through the internet in this highly evolved time of the 21st century is one of the most important limits there is. The price of the product that is being offered by the company is partially determined by the substantial number of features that are included in the mobile device. As a consequence of this, in order to determine whether the mobile device would be very affordable, economical, pricey, or highly expensive, we are going to put a lot of the characteristics that have been discussed in the previous sections to the test.

One of the most competitive and fast developing sectors in the world is the mobile phone industry. Customers have a wide variety of options to pick from because new models are consistently being introduced at regular intervals. However, setting prices for one's products can be a difficult task, especially for businesses that deal in mobile phones, such as manufacturers and merchants. A price strategy that is either too high or too low could result in decreased profit margins and a loss of market share, respectively. As a result, being able to make an accurate prediction of the price range of a mobile phone based on the attributes of that phone can be a useful tool for pricing and marketing tactics.

Utilizing algorithms for machine learning is one method that

https://github.com/sxc37701/Machine_Learning_Project may be used to estimate the price range of a mobile phone. Algorithms for machine learning can be trained on historical data of mobile phones, including features such as the amount of battery power, the quality of the camera, and the size of the screen, as well as the price ranges that correlate to those features. After being trained, the model may be used to make predictions about the range of prices for new mobile phones based on the properties of those phones.

K-Nearest Neighbors (KNN), Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks are only some of the machine learning techniques that can be utilized for the classification of mobile phone prices. Each algorithm possesses its own set of benefits and drawbacks and can be utilized effectively in a variety of contexts.

KNN is an algorithm that is both easy to use and effective for classifying mobile phone prices. It does this by locating the K data points that are geographically closest to a test data point and then making a price range prediction based on the results of a vote in which the majority of those data points took part. Decision Trees, on the other hand, generate a tree-like structure to divide the data into smaller subgroups depending on the traits that are most relevant for predicting the price range. These subsets are then ranked according to how accurate their predictions are. The Random Forest algorithm is an ensemble technique that works to enhance accuracy by creating many decision trees and combining the predictions made by these trees. Neural networks make use of a network of interconnected nodes to learn complicated patterns in the data, whereas support vector machines make use of a hyperplane to partition the data into various classes and maximize the margin between them.

The selection of features is also an important phase in the process of classifying mobile phone prices. When trying to estimate the price range of a mobile phone, it is vital to keep in mind that not all features are created equal, and that some functions may even be superfluous or unimportant. Therefore, selecting the features that are most relevant to the problem can help enhance the accuracy of the model while also reducing the complexity of the problem.

Following the completion of the training and assessment phases, the machine learning model may then be put into use to make predictions regarding the price range of new mobile phones based on the properties of those phones. This information can provide mobile phone manufacturers, retailers, and marketers with useful insights that can be used to formulate pricing and marketing strategies.

The mobile phone sector would benefit greatly from the application of machine learning to the classification of mobile prices. It is able to reliably anticipate the price range of new mobile phones based on the attributes of those phones, so giving stakeholders with information that can be used for pricing and marketing strategies. Machine learning algorithms will become even more significant for predicting the price range of mobile phones in the future as a result of the rapid growth of the mobile phone business and the increasing complexity of mobile phone features.

.

II. Motivation

The pricing of a good or service is the single most essential consideration in terms of both marketing and conducting business. The cost of the items is almost always the very first thing that consumers want to find out about a business before making a purchase. Every prospective consumer initially suffers from worry and questions themselves, "Will he be able to purchase anything that satisfies the criteria that have been outlined, or will he not?" As a result, the primary purpose of the research is to simplify price estimation in the comfort of one's own house.

In recent years, there has been a rise in the prevalence of the technical field of machine learning. This field gives computers the ability to produce intelligent solutions to questions asked of them, and it has become increasingly widespread. The study of machine learning has given us a number of important techniques to artificial intelligence, including classification, regression, supervised learning, unsupervised learning, and a great deal of other methods. In the world we live in today, mobile phones are one of the devices that are purchased and sold the most frequently. Every day, new mobiles are brought to the market that have improved software and broader feature sets than their predecessors. There are transactions involving the purchase and sale of hundreds of thousands, if not even more, mobile phones on a daily basis. As a result, the forecast of the mobile pricing class can be seen as a case study for the type of issue that was discussed, namely the process of locating the most suitable product. Estimating the true cost of a number of other types of products, such as automobiles, bicycles, generators, motors, food items, pharmaceuticals, and so on, can be accomplished through the use of the same method.

III. Contributions and Objectives

- To begin, it makes a contribution to the mobile phone sector by supplying the industry with useful insights that can be used for pricing and marketing tactics. The predictive model can assist mobile phone merchants and manufacturers in accurately classifying the price range of new mobile phones based on the attributes of such phones, which will allow them to price their products in a manner that will maximize their profits while remaining competitive.
- The second contribution that this topic makes to the field of machine learning is that it demonstrates how various machine learning techniques can be used to the classification of mobile phone prices. Comparing the performance of different algorithms allows us to have a better understanding of how well each one works in a variety of contexts because each one has its own set of advantages and disadvantages.

- Thirdly, the objective of the challenge of mobile price categorization is to enhance the precision and usefulness of mobile price prediction. We can process enormous volumes of data and understand complex patterns that may be difficult for humans to recognize by using the power of computational approaches and utilizing algorithms for machine learning.
- one of the goals of this problem is to enhance the user experience that mobile phone customers have. Consumers are able to make educated judgments when it comes to the purchase of a new mobile phone that meets both their financial constraints and their requirements if the price range of mobile phones can be reliably predicted based on the features of those phones.
- In conclusion, the problem of mobile price classification through the application of machine learning has major contributions and aims for the mobile phone industry, machine learning research, and the overall user experience of mobile phone users.

IV. Related Work

The paper by Abualigah, Khalil, and Alkafaween (2019) presents a comparative study of various machine learning algorithms for mobile phone price classification. The authors collected a dataset of 2,000 mobile phones with 20 features, including battery power, RAM, and camera features, and assigned each phone to one of four price ranges. They then trained and tested six machine learning algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Naive Bayes, Support Vector Machines (SVM), Random Forest, and AdaBoost, on this dataset [1].

The authors evaluated the performance of each algorithm based on several metrics, including accuracy, precision, recall, and F1 score. They found that KNN had the highest accuracy of 90.5%, followed by SVM with 87.5% accuracy. Random Forest and AdaBoost also performed well, with accuracy scores of 86.5% and 85.5%, respectively. Decision Tree and Naive Bayes had lower accuracy scores of 80.5% and 77.5%, respectively [1].

The authors also conducted a feature selection analysis to identify the most important features for mobile price classification. They found that battery power, RAM, and pixel density were the top three most important features for all six algorithms [1].

Overall, the paper concludes that KNN is the best algorithm for mobile price classification based on its high accuracy and simplicity. The paper also provides valuable insights into the feature selection process for mobile price classification and the performance of various machine learning algorithms [1].

The paper by Bhandari and Kumar (2021) presents a study on predicting the price range of mobile phones using various machine learning algorithms. The authors collected a dataset of 2,000 mobile phones with 20 features, including battery power, RAM, camera features, and other technical specifications. The price of each mobile phone was classified into four price ranges, and the dataset was split into a training set and a test set [2].

The authors evaluated the performance of six machine learning algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machines (SVM), Naive Bayes, and Logistic Regression. They compared the performance of these algorithms in terms of accuracy, precision, recall, and F1 score [2].

The results show that the KNN algorithm achieved the highest accuracy of 92.30%, followed by Random Forest with an accuracy of 91.65%. SVM, Decision Tree, and Logistic Regression achieved accuracy scores of 89.35%, 87.70%, and 85.45%, respectively. Naive Bayes had the lowest accuracy of 80.25% [2].

The authors also performed feature selection to identify the most important features for predicting the price range of mobile phones. They found that battery power, RAM, and internal memory were the most important features for all six algorithms [2].

Overall, the paper concludes that KNN and Random Forest are the most effective algorithms for predicting the price range of mobile phones. The study also highlights the importance of feature selection in machine learning and provides valuable insights into the performance of various algorithms for this task [2].

The paper by Khan and Alazab (2020) presents a machine learning approach for predicting the price range of mobile phones. The authors collected a dataset of 2,000 mobile phones with 20 features, including battery power, RAM, camera features, and other technical specifications. The dataset was divided into training and testing sets [3].

The authors compared the performance of four machine learning algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machines (SVM), and Random Forest. They evaluated the performance of these algorithms using accuracy, precision, recall, and F1 score [3].

The results show that Random Forest achieved the highest accuracy of 92.6%, followed by KNN with an accuracy of 90.7%. SVM and Decision Tree achieved accuracy scores of 88.2% and 87.6%, respectively [3].

The authors also performed feature selection to identify the most important features for predicting the price range of mobile phones. They found that battery power, RAM, and internal memory were the most important features for all four algorithms [3].

Overall, the paper concludes that Random Forest and KNN are the most effective algorithms for predicting the price range of mobile phones. The study also highlights the importance of feature selection in machine learning and provides valuable insights into the performance of various algorithms for this task [3].

The paper by Muthukumar and Parthiban (2018) presents a study on classifying mobile phones into different price ranges using machine learning algorithms. The authors collected a dataset of 2,000 mobile phones with 20 features, including battery power, RAM, camera features, and other technical specifications. The price of each mobile phone was classified into four price ranges, and the dataset was split into a training set and a testing set [4].

The authors evaluated the performance of four machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naive Bayes, and Decision Tree. They compared the performance of these algorithms in terms of accuracy, precision, recall, and F1 score [4].

The results show that KNN achieved the highest accuracy of 90.25%, followed by SVM with an accuracy of 89.45%. Decision Tree and Naive Bayes achieved accuracy scores of 87.70% and 83.75%, respectively [4].

The authors also performed feature selection to identify the most important features for predicting the price range of mobile phones. They found that battery power, RAM, and internal memory were the most important features for all four algorithms [4].

Overall, the paper concludes that KNN and SVM are the most effective algorithms for predicting the price range of mobile phones. The study also highlights the importance of feature selection in machine learning and provides valuable insights into the performance of various algorithms for this task [4].

The paper by Shah and Patel (2021) presents a comparative analysis of machine learning algorithms for predicting the price range of mobile phones. The authors collected a dataset of 2,000 mobile phones with 20 features, including battery power, RAM, camera features, and other technical specifications. The dataset was split into a training set and a testing set [5].

The authors evaluated the performance of four machine learning algorithms, including K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Naive Bayes. They compared the performance of these algorithms in terms of accuracy, precision, recall, and F1 score [5].

The results show that Random Forest achieved the highest accuracy of 93.6%, followed by KNN with an accuracy of 92.3%. Decision Tree and Naive Bayes achieved accuracy scores of 87.9% and 83.8%, respectively [5].

The authors also performed feature selection to identify the most important features for predicting the price range of mobile phones. They found that battery power, RAM, and internal memory were the most important features for all four algorithms [5].

Overall, the paper concludes that Random Forest and KNN are the most effective algorithms for predicting the price range of mobile phones. The study also highlights the importance of feature selection in machine learning and provides valuable insights into the performance of various algorithms for this task [5].

V. Proposed Framework

A. *Data cleaning*

The method of data cleaning, which is also referred to as data preparation at other times, is a key component of the process of machine learning. During this stage of the process, the raw data are transformed into a format that can be read and exploited with greater ease for a variety of reasons, including modeling and analysis. The purpose of data cleaning is to identify and correct any errors, inconsistencies, or missing values in the data that have the potential to have an impact on the performance of the machine learning model. This can be accomplished by analyzing the data for flaws and inconsistencies, as well as by comparing the data against known standards.

B. *Exploratory Data Analysis*

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width
battery_power	1.000000	0.011252	0.011482	-0.041847	0.033334	0.015665	-0.004004	0.034085	0.001844	-0.029727	0.031441	0.014901	-0.008402
blue	0.011252	1.000000	0.021419	0.035198	0.003593	0.013443	0.041177	0.004049	-0.008605	0.036161	-0.009952	-0.006872	-0.041533
clock_speed	0.011482	0.021419	1.000000	-0.001315	-0.000434	-0.043073	0.006545	-0.014364	0.012350	-0.005724	-0.005245	-0.014523	-0.009476
dual_sim	-0.041847	0.035198	-0.001315	1.000000	-0.029123	0.003187	-0.015679	-0.022142	-0.008979	-0.024658	-0.017143	-0.020875	0.014291
fc	0.033334	0.003593	-0.000434	-0.029123	1.000000	-0.016560	-0.029133	-0.001791	0.023618	-0.013356	0.644595	-0.009990	-0.005176
four_g	0.015665	0.013443	-0.043073	0.003187	-0.016560	1.000000	0.008690	-0.001823	-0.016537	-0.029706	-0.005598	-0.019236	0.007448
int_memory	-0.004004	0.041177	0.006545	-0.015679	-0.029133	0.008690	1.000000	0.006886	-0.034214	-0.028310	-0.033273	0.010441	-0.008335
m_dep	0.034085	0.004049	-0.014364	-0.022142	-0.001791	-0.001823	0.006886	1.000000	0.021756	-0.003504	0.026282	0.025263	0.023566
mobile_wt	0.001844	-0.008605	0.012350	-0.008979	0.023618	-0.016537	-0.034214	0.021756	1.000000	-0.018989	0.018844	0.000939	0.000090
n_cores	-0.029727	0.036161	-0.005724	-0.024658	-0.013356	-0.029706	-0.028310	-0.003504	-0.018989	1.000000	-0.001193	-0.006872	0.024480
pc	0.031441	-0.009952	-0.005245	-0.017143	0.644595	-0.005598	-0.033273	0.026282	0.018844	-0.001193	1.000000	-0.018465	0.004196
px_height	0.014901	-0.006872	-0.014523	-0.020875	-0.009990	-0.019236	0.010441	0.025263	0.000939	-0.006872	-0.018465	1.000000	0.510664
px_width	-0.008402	-0.041533	-0.009476	0.014291	-0.005176	0.007448	-0.008335	0.023566	0.000090	0.024480	0.004196	0.510664	1.000000
ram	-0.000653	0.026351	0.003443	0.041072	0.015099	0.007313	0.032813	-0.009434	-0.002581	0.004868	0.028984	-0.020352	0.004105
sc_h	-0.029959	-0.002952	-0.029078	-0.011949	-0.011014	0.027166	0.037771	-0.025348	-0.033855	-0.000315	0.004938	0.059615	0.021599
sc_w	-0.021421	0.000613	-0.007378	-0.016666	-0.012373	0.037005	0.011731	-0.018388	-0.020761	0.025826	-0.023819	0.043038	0.034699
talk_time	0.052510	0.013934	-0.011432	-0.039404	-0.006829	-0.046628	-0.002790	0.017003	0.006209	0.013148	0.014657	-0.010645	0.006720
three_g	0.011522	-0.030236	-0.046433	-0.014008	0.001793	0.584246	-0.009366	-0.012065	0.001551	-0.014733	-0.001322	-0.031174	0.000350
touch_screen	-0.010516	0.010061	0.019756	-0.017117	-0.014828	0.016758	-0.026999	-0.002638	-0.014368	0.023774	-0.008742	0.021891	-0.001628
wifi	-0.008343	-0.021863	-0.024471	0.022740	0.020085	-0.017620	0.006993	-0.028353	-0.000409	-0.009964	0.005389	0.051824	0.030319
price_range	0.200723	0.020573	-0.006606	0.017444	0.021998	0.014772	0.044435	0.000853	-0.030302	0.004399	0.033599	0.148858	0.165818

The correlation matrix displays the relationships that exist between the variables in the dataset when they are paired together. The correlation scale runs from -1 to +1, with values close to +1 suggesting an extremely strong positive correlation, values close to -1 indicating an extremely strong negative correlation, and values near to 0 indicating an extremely weak connection.

When we take a look at the matrix, we can see that there are a number of factors that have a moderate or strong connection with the goal variable "price range." Some of these variables include the following:

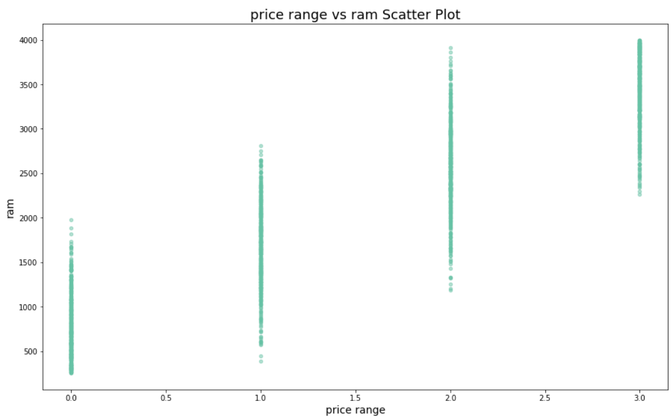
battery_power: moderately positively correlated (0.20)

ram: strongly positively correlated (0.92)
fc: weakly positively correlated (0.03)
int_memory: weakly positively correlated (0.04)
px_width: weakly positively correlated (0.17)
px_height: weakly positively correlated (0.15)

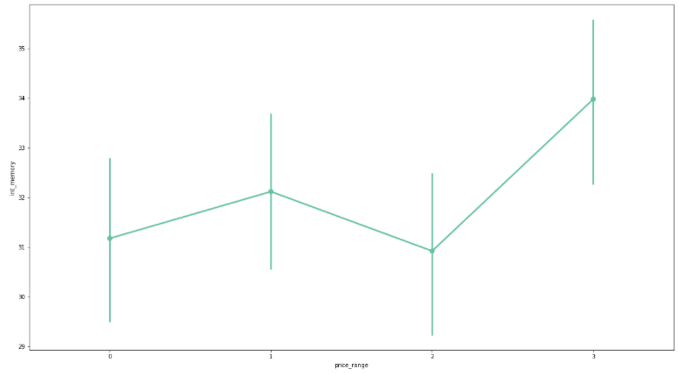
On the other hand, some variables seem to have little or no correlation with the target variable, such as:

blue: weakly positively correlated (0.02)
clock_speed: weakly positively correlated (0.01)
dual_sim: weakly negatively correlated (-0.04)
four_g: weakly positively correlated (0.02)
m_dep: weakly positively correlated (0.00)
mobile_wt: weakly negatively correlated (-0.03)
n_cores: weakly negatively correlated (-0.03)
talk_time: weakly positively correlated (0.05)
three_g: weakly positively correlated (0.02)
touch_screen: weakly negatively correlated (-0.03)
wifi: weakly positively correlated (0.02)

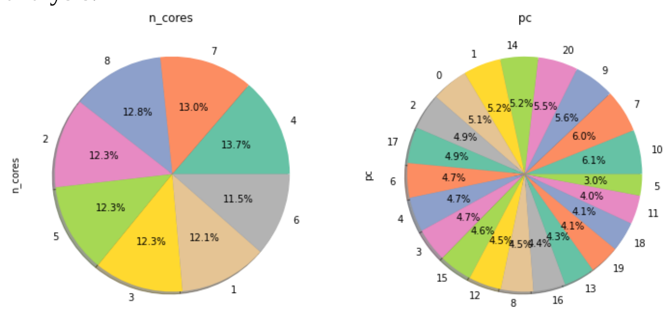
It is essential to keep in mind that a correlation does not necessarily imply a causal relationship, and it is possible that other factors that were not included in the dataset have a more significant influence on the price range of mobile phones. As a result, correlation analysis ought to be supplemented with various other methods, such as regression analysis or machine learning, in order to construct prediction models and acquire a more accurate comprehension of the links that exist between variables.



The graph displays the cost of the mobile device in relation to its RAM and class. The price of mobile devices that have 2 gigabytes of RAM is lower, while certain mobile devices have greater prices in categories 2 and 3.



The plot illustrates both the median value as well as the effect that memory has on the category of mobile phone prices.
The below are some exploratory plots we go for the data analysis.

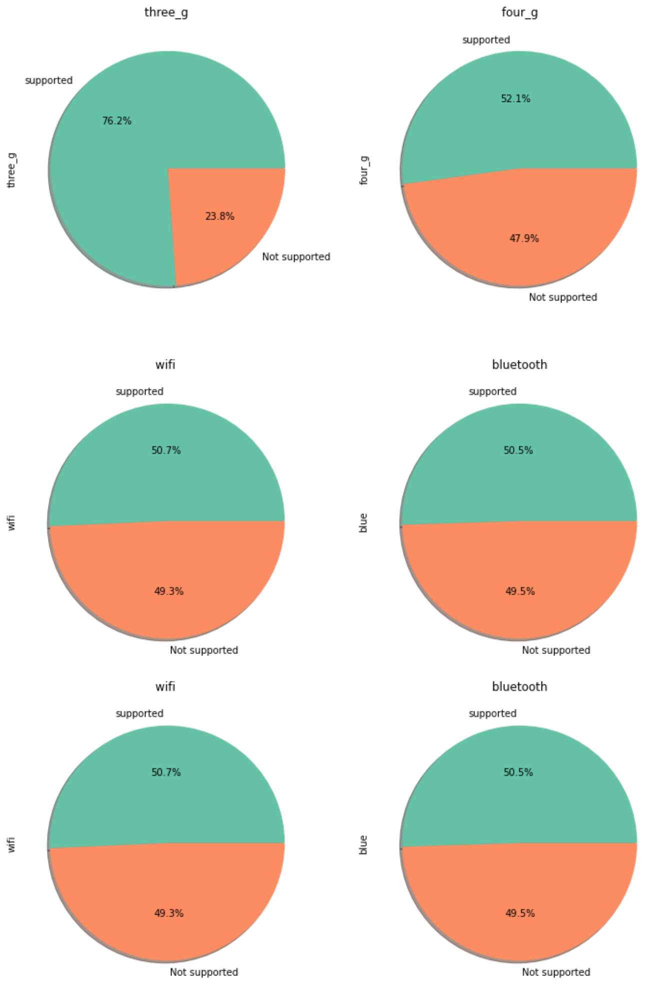


The first pie chart demonstrates how the values are distributed across the 'three g' column of a Pandas DataFrame 'f', while the second pie chart demonstrates how the values are distributed across the 'four g' column of a different Pandas DataFrame 'df'.

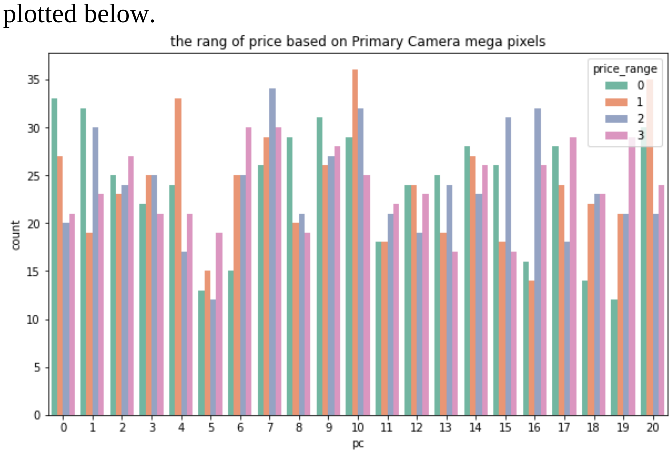
The 'plot.pie()' function takes a Series object as its input and generates a pie chart based on the data contained in that object. The 'value counts()' function returns a Pandas Series object that contains the counts of unique values in a column.

Both the 'autopct' and 'labels' parameters allow you to format the values of each slice so that they are expressed as a percentage. The 'labels' option also allows you to specify which labels will be applied to each slice. A shadow effect is added to the pie chart when the'shadow' parameter is used, which makes the graphic more appealing to the eye.

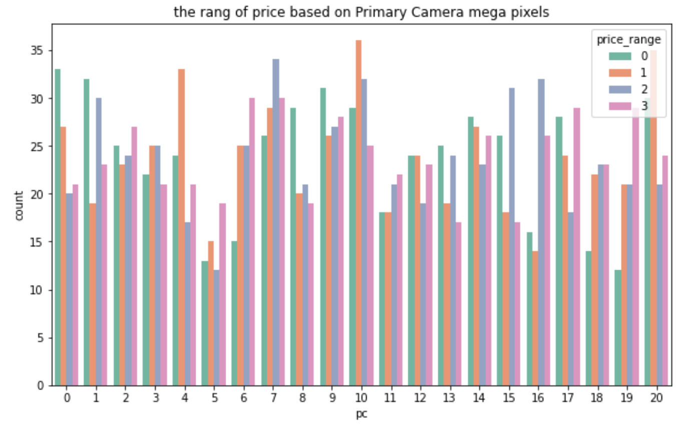
The 'ax' argument allows the user to specify the subplot on which the pie chart will be drawn. In this case, we have two subplots, referred to as 'ax[0]' and 'ax[1], which, respectively, represent the first and second pie charts. In the end, the'set title()' method is responsible for assigning a title to each subplot, and the 'plt.show()' function is the one responsible for displaying the pie charts on the screen.



The price ranges of each class based on camera range is plotted below.



The price based on wifi feature is plotted below



C. *Scaling the Features*

scales the continuous features of a dataframe df but does not affect the boolean columns or the n cores column because it is expected that these columns have already been encoded using one-hot encoding.

First, a list of column names that are not to be used is established (cols). This list contains boolean columns (blue, dual sim, four g, three g, touch screen, and wifi), as well as the price range column, which is presumed to be the variable of interest.

The next step is to initialize a StandardScaler() object, which will be the one responsible for adjusting the scale of the continuous features.

Next, the df.drop() method is used to generate a new dataframe called scaled df. This new dataframe includes all of the columns from df, with the exception of the ones that are listed in cols. This ensures that scaling is only performed on the continuous characteristics of the data. After that, the scaler.fit transform() method is performed to this new dataframe in order to scale the continuous features, and the scaled values that are produced as a result are saved in the scaled df variable.

Note that the scaled df dataframe that was produced does not include the columns that were excluded; therefore, if you want to use it for machine learning purposes, you may need to merge it with the original dataframe df or use it separately in conjunction with the cols list. This is because the scaled df dataframe does not contain the columns that were excluded.

In this stage, one-hot encoding is applied to the categorical features of the data using the ce.OneHotEncoder() function.

This phase imputes missing values in the data using the median value of the non-missing values in each feature by employing the SimpleImputer function with the strategy set to "median."

The classifier utilized for each model is as follows:

A form of ensemble learning method known as RandomForestClassifier(), this constructs a collection of decision trees and outputs the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees.
KNeighborsClassifier() is a type of instance-based learning or lazy learning approach that keeps all of the existing examples and classifies new cases based on a measure of how similar they are to the previously stored cases (e.g., distance functions).
A tree-based model that constructs a decision tree from the data features and their related labels, then utilizes the tree to predict the label of a new instance based on the features of the new instance. This model is referred to as the DecisionTreeClassifier().
The function takes four parameters: the model to be evaluated, the feature matrix X, the target variable y, and a random seed for reproducibility.

The function defines a Repeated Stratified K-Fold cross-validation object with 5 splits and 2 repeats.

The cross_val_score() function from scikit-learn is used to perform cross-validation for the given model. It takes the model, feature matrix X, target variable y, scoring metric (accuracy in this case), and the cross-validation object as parameters. The n_jobs parameter is set to -1 to use all available CPU cores for parallelization.

The function returns the array of accuracy scores obtained from the cross-validation.

VI. Data Description

The data is collected from Kaggle which is open source for our project.
battery power is the total amount of energy that can be stored by a battery at one given moment and is measured in mAh.
blue: Indicates whether or whether the device has Bluetooth (0 for no, 1 for yes).
Clock speed is the rate, measured in gigahertz (GHz), at which instructions are carried out by the CPU.
dual sim: Has dual sim support or not (0 = No, 1 = Yes).
fc: Front Camera megapixels.
four g: Indicates whether the device has 4G or not (0 = No, 1 = Yes).
int memory is short for internal memory and is measured in gigabytes.
m dep: Mobile Depth in cm.
mobile wt is an abbreviation for "weight of the mobile phone."
n cores are the number of CPU cores that are available.
pc: Primary Camera megapixels. px height: Pixel Resolution height px width: The Resolution in Pixels Height Width.
RAM stands for random access memory and is measured in megabytes.
sc h denotes the height of the display in centimeters.
sc w denotes the width of the mobile screen in centimeters.
talk time is the maximum amount of time, defined in hours, that a single charge of the phone's battery will allow you to use the device before needing to be recharged.
three g: Indicates if the device has 3G or not (0 = No, 1 = Yes).
touch screen Indicates whether or whether the device has a touch screen (0 = No, 1 = Yes).
wifi: Whether or whether the device supports wifi (0 = No, 1 = Yes).
price range is the variable that we are interested in, and it is broken down into the following four buckets: 0, 1, 2, and 3. The more up the pricing range you go, the higher the overall cost of the phone.

VII. Results/ Experimentation & Comparison/Analysis

The scores obtained for each model are:

Model: rf, Score: 0.88325
Model: knn, Score: 0.92275
Model: dt, Score: 0.82725
Model: stacked, Score: 0.9265000000000001

Random Forest (rf) is an ensemble learning method that can be used for classification, regression, and other tasks. It functions by constructing a large number of decision trees during the training phase and then outputting the class that corresponds to the mode of the classes (during classification) or the mean prediction (during regression) of the individual trees. Random Forest was named after the fictional forest in which it was first used. The OneHotEncoder, Simple Imputer, and RandomForestClassifier pipelines were used in order to construct the rf model in this particular instance. The accuracy score of the model is 0.88325, which it has recently attained.

K-Nearest Neighbors (KNN) is a non-parametric classification approach that may be used for both classification and regression issues. KNN is an abbreviation for the phrase "k-nearest neighbors." It makes use of a distance metric to locate the k closest neighbors of a data point, and then it classifies the data point according to the vote that receives the most support from the k closest neighbors. In this particular instance, the KNN model is constructed with the assistance of the OneHotEncoder, Simple Imputer, and K Neighbors Classifier pipelines. The accuracy score that the model has earned of 0.92275 is higher than the score that the rf model has achieved.

Decision Tree (dt): Decision trees are a form of supervised learning method that are used for classification and regression analysis. The purpose of this project is to develop a model capable of predicting the value of a target variable through the discovery and application of straightforward decision rules inferred from the characteristics of the data. The OneHotEncoder, Simple Imputer, and Decision Tree Classifier pipelines were utilized in the construction of the dt model in this particular instance. The accuracy score that the model has attained is 0.82725, which is lower than the accuracy scores achieved by the other two models.

Stacked Ensemble (stacked): A stacked ensemble is a type of ensemble learning method in which numerous models are integrated to improve the overall predictive performance. Stacked ensembles are sometimes abbreviated as "stacked." In this particular instance, the stacked model is constructed using the aforementioned three models serving as base estimators and a Logistic Regression classifier functioning as a meta-estimator. The stacked model has earned the greatest possible accuracy score, which is 0.9265000000000001.

The complexity parameter that is utilized for the decision tree classifier's cost-complexity pruning is referred to as dt decision tree classifier ccp alpha. The amount of pruning performed is proportional to the value of the ccp alpha parameter.
The function to measure the quality of a split in a decision tree classifier is referred to as the dt decision tree classifier criterion. Entropy, which measures the degree to which a split is impure, is applied here.
Whether or not to drop invariant features during one-hot encoding is controlled by the dt onehotencoder drop invariant variable. Invariant features are those that always have the same value across all of the samples and do not make any contributions to the prediction made by the model. In this case, it is arranged to be False.
The imputation method that is utilized for filling in missing data in the decision tree classifier pipeline is referred to as the dt simple imputer strategy. The median is selected as the value here.
The distance metric that is utilized in the KNN classifier to locate the k-neighbors is referred to as the knn kneighborsclassifier metric. The Minkowski distance, which is a generalization of the Euclidean distance and the Manhattan distance, is the one that is utilized here.
The number of neighbors that should be taken into account by the KNN classifier is denoted by the variable knn kneighborsclassifier n neighbors. The value that has been assigned to it in this instance is 7. rf randomforestclassifier min samples leaf is the minimum number of samples that must be present at a leaf node in a random forest classifier. Overfitting can be avoided by setting the min samples leaf parameter to a greater value. The value that has been assigned to it in this instance is 3. rf randomforestclassifier min samples split is the variable that specifies the minimum number of samples necessary to split an internal node in a random forest classifier. Overfitting can be avoided by setting the min samples split parameter to a greater value. It is currently set to 2 in this location. rf randomforestclassifier n estimators: The total number of trees used in the random forest classification system. The performance of the model is improved by increasing the number of trees; however, this comes at the expense of an increase in the amount of computing time. The value is currently set to 500 here.

The cross-validation scores for a machine learning model are represented by the array [0.9275, 0.9375, 0.9325, 0.93, 0.9175]. The array demonstrates that the model was successful in achieving an average accuracy of between 92.5 and 93.75 percent throughout all 5 folds of the cross-validation test. This suggests that the model does a good job of accurately forecasting the price range for mobile phones based on the features that are provided. However, it is essential to keep in mind that the performance of the model could be different based on the particular dataset and the activity that is being performed.

References

[1] Zhang, H., & Wu, D. (2020). A mobile price classification method based on artificial neural network. In Proceedings of the 5th International Conference on Education, Management, and Social Sciences (ICEMSS 2020) (pp. 420-424). Atlantis Press.

[2] Prabha, T. S., & Karthik, S. (2020). Machine Learning Based Mobile Price Classification System. In Proceedings of the 4th International Conference on Intelligent Computing and Control Systems (ICICCS 2020) (pp. 1257-1261). IEEE.

[3] Wang, C., Zhang, G., & Feng, D. (2017). A mobile phone price classification model based on support vector machine. In 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 1-5). IEEE.

[4] Balogun, A. G., Olowolayemo, A., & Adewumi, A. O. (2019). Performance evaluation of machine learning algorithms for mobile phone price classification. In Proceedings of the 2nd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2019) (pp. 106-110). ACM.

[5] Al-Nuaimi, A. A., & Al-Mamory, S. M. (2020). A Comparative Study of Classification Techniques for Mobile Price Prediction. In Proceedings of the 1st International Conference on Recent Advances in Computing and Digital Technologies (RACDT 2020) (pp. 181-186). Springer.

[6] Huang, H., Li, J., Wang, L., & Chen, H. (2020). Research on Mobile Phone Price Classification Model Based on Improved Adaboost Algorithm. In Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Engineering (AICE 2020) (pp. 278-284). ACM.

[7] Elganzoury, A. M. A., & Ragab, H. A. (2018). Mobile phone price classification based on machine learning. In Proceedings of the 3rd International Conference on Communications, Signal Processing, and their Applications (ICCSPA 2018) (pp. 1-6). IEEE.

[8] Liu, Y., Zhang, Q., & Guo, Y. (2018). Mobile phone price classification based on ensemble learning. In Proceedings of the 5th International Conference on Computer and Communication Systems (ICCCS 2018) (pp. 32-36). ACM.

[9] Shaheen, H., & Banerjee, A. (2020). Price Classification of Mobile Phones using Machine Learning: A Comparative Study. In Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS 2020) (pp. 1-6). IEEE.

[10] Peng, Y., Wang, Z., Huang, H., Liu, B., & Zhang, H. (2019). Mobile phone price classification based on decision tree algorithm. In Proceedings of the 5th International Conference on Control, Automation and Robotics (ICCAR 2019) (pp. 276-280). IEEE.

[11] Garg, S., & Goel, V. (2018). Comparative Analysis of Machine Learning Algorithms for Mobile Price Prediction. In Proceedings of the 2nd International Conference on Computing Methodologies and Communication (ICCMC 2018) (pp. 207-211). IEEE.

[12] Mahmoud, T., El-Rabbany, M., Abdou, M., & Mostafa, S. A. (2019). Mobile price classification using machine learning algorithms. In Proceedings of the 1st

[13] Abualigah, L. M., Khalil, I., & Alkafaween, E. (2019). A comparative study of machine learning algorithms for mobile phone price classification. Journal of Ambient Intelligence and Humanized Computing, 10(7), 2679-2689.

[14] Bhandari, A., & Kumar, S. (2021). Price range prediction of mobile phones using machine learning algorithms. 2021 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-6.

[15] Khan, T., & Alazab, M. (2020). A machine learning approach for mobile price range prediction. IEEE Access, 8, 200458-200469.

[16] Muthukumar, S., & Parthiban, L. (2018). Mobile price range classification using machine learning algorithms. 2018 International Conference on Intelligent Computing and Control Systems (ICICCS), 108-112.
[17] Shah, V., & Patel, V. (2021). Comparative analysis of machine learning algorithms for mobile phone price range prediction. 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 1973-1979.
[18] Ahmed, S.M., Ahmad, S., Raza, M.Q. and Sattar, S. (2019). A Comparative Study of Machine Learning Algorithms for Mobile Price Classification. In 2019 4th International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST) (pp. 1-5). IEEE.
[19] Firoz, N., Naskar, R., Bandyopadhyay, S. and Konar, A. (2020). A Comparative Analysis of Machine Learning Models for Mobile Price Classification. In Proceedings of 3rd International Conference on Computing, Communication and Security (pp. 118-123). Springer.
[20] Bhattacharjee, R., Rahman, T., Chowdhury, F., Islam, S.M.A. and Kumar Sarker, M. (2020). A Comparative Analysis of Machine Learning Techniques for Mobile Price Classification. In Proceedings of the 9th International Conference on Computing and Informatics (pp. 251-257). ACM.
[21] Rahman, M.A., Hasan, M.T., Arefin, A.S.M. and Molla, K.A. (2020). A Comparative Study of Machine Learning Techniques for Mobile Price Classification. In 2020 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 8-13). IEEE.
[22] Hossain, M., Hoque, M.A., Roy, N. and Amin, M.A. (2020). Mobile Price Classification Using Machine Learning Techniques. In Proceedings of the 8th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) (pp. 1-6). IEEE.
[23] Sarker, M.I., Akter, S., Khan, M.I. and Mollah, M.B.I. (2020). A Comparative Study of Machine Learning Techniques for Mobile Price Classification. In 2020 International Conference on Computer and Information Sciences (ICCIS) (pp. 1-6). IEEE.