MOBILE PRICE RANGE PREDICTION

AJIT KUMAR PATEL

DATA SCIENCE TRAINEE

ALMABATTER, BANGLORE

1. ABSTRACT

To predict “If the mobile with given features will be Economical or Expensive” is the main motive of this research work. Real Dataset is collected from the website Almabatter projects. Different feature selection algorithms are used to identify and remove less important and redundant features and have minimum computational complexity. Different classifiers are used to achieve a higher accuracy as possible. Results are compared in terms of the highest accuracy achieved and minimum features selected. The conclusion is made on the base of the best feature selection algorithm and best classifier for the given dataset. This work can be used in any type of marketing and business to find the optimal products (with minimum cost and maximum features). Future work is suggested to extend this research and find a more sophisticated solution to the given problem and a more accurate tool for price estimation.

This paper provides visualization and prediction tools for Mobile price range prediction. The presented multi-agent system includes an agent that performs data collection and cleaning processes, it is also capable of creating demand forecasting models for each bicycle station. Moreover, the architecture offers API (Application Programming Interface) services and provides a web application for visualization and forecasting. This work aims to make the system generic enough for it to be able to integrate data from different types of Mobile price range predictions. Thus, in future studies, it will be possible to employ the proposed system in different types of Mobile price range prediction. This article contains a literature review, a section on the process of developing the system, and the built-in prediction models. Moreover, a case study validates the proposed system by implementing it in a public Mobile price range prediction. It also includes an outline of the results and conclusions, a discussion on the challenges encountered in this domain, as well as possibilities for future work.

2. Featured application:

The main application of this work is the analysis and prediction of the demand in Mobile price range predictive systems using their open/private data. A multi-agent system is proposed and a case study is conducted using the data of a mobile pricing system from a company.

3. introduction

Price is the most effective attribute of marketing and business. The very first question of the customer user is about the price of items. All costumesmers are first worried think “If he would be able to purchase something with the given specifications or not”. So estimating prices at home is the basic purpose of the work. This paper is only the first step toward the above-mentioned destination. Artificial Intelligence-which makes the chine capable to answer questions intelligently- nowadays is via ry vast engineering field. Machine learning provides us best techniques for artificial intelligence like classification, regression, supervised learning and unsupervised learning, and many more. Different tools are available for machine learning tasks like MATLAB, Python, Cygwin, WEK, A, etc. We can use any of the classifiers like Decision ree, Naïve Ba, yes, and many more. Different types of feature selection algorithms are available to select only the best features and minimize the dataset. This will reduce the computational complexity of the problem. As this is an optimization problem so many optimization techniques are also used to reduce the dimensionality of the dataset. Mobnowadaysdays is one of the most selling and purchasing devices. Every day new mobiles with the new version and more features are launched. Hundreds and thousands of mobile are sold and purchased on daily basis. So here the mobile price\_class prediction is a case study for the given type of problem i.e finding an optimal product. The same work can be done to estimate the real price of all products like cars, bikes, generators, motors, food items, medicine, etc.

price of mobile. For example Processor of the mobile. Battery timing is also very important in today’s busy schedule of a human being. The size and thickness of the mobile are also important decision factors. Internal memory, Camera pixels, and video quality must be under consideration. Internet browsing is also one of the most important constraints in this technological era of the 21st century. And so is the list of many features based upon those, the mobile price is decided. So we will use many of the above-mentioned features to classify whether the mobile would be very economic, economical, expensive, or very\_ expensive. The structure of the paper is as follows. The next section is a review of previous work.3 rd Section contains Methodology and Experimental procedure. Section 4 is the summary of the results. A comparative study is done in section 5. After that paper is concluded in section 6. The outcomes of the work are discussed in section 7. At last in the 8th section, some suggestions about future work are given.

4. Mobile price range prediction:

Price prediction uses an algorithm to analyze a product or service based on its characteristics, demand, and current market trends. Then the software sets a price at a level it predicts will both attract customers and maximize sales. In some circles, the practice is called price forecasting or predictive pricing

In the competitive mobile phone market companies want to understand sales data of mobile phones and the factors which drive the prices. The objective is to find out some relation between the features of a mobile phone(eg:- RAM, Internal Memory, etc) and its selling price. In this problem, we do not have to predict the actual price but a price range indicating how high the price is.

5. Data Description:

\* Battery\_power - Total energy a battery can store in one time measured in mAh

\* Blue - Has Bluetooth or not

\* Clock\_speed – the speed at which 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 the mobile phone

\* N\_cores - Number of cores of a processor

\* Pc - Primary Camera megapixels

\* Px\_height - Pixel Resolution Height

\* Px\_width - Pixel Resolution Width

\* Ram - Random Access Memory in MegaBytes

\* 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 the values of 0(low cost), 1(medium cost),

2(high cost) and 3(very high cost).

6. METHODOLOGY:

* **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is rent\_count with other independent variables. This process helped us figure out various aspects and relationships between the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

* **Null values Treatment**

Our dataset does not contain the null nither which might tend to disturb our accuracy hence we dropped them at the beginning of our project to get a better result.

* **Encoding categorical columns**

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to the numerical format.

* **Feature Selection**

In feature selection, we are interested in finding k of the d dimensions that give us the most information, and we discard the other (d − k) dimension

In feature extraction, we are interested in finding a new set of k dimensions that are combinations of the original d dimensions for example Principal Component Analysis

In these steps, we used algorithms like the extra tree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

Next, we used Chi2 for categorical features and ANOVA for numerical features to select the best feature which we will be using further in our model.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

7. MODEL SELECTION:

This section describes how the MPR dataset was split and the methodology that was used to select the prediction models that will be included in the predictor agent. Like in Kaggle competitions, the data have been split into two datasets; a training dataset and a validation dataset. schematically how the available data were employed in the training, selection, and validation of the models used. In the upper part of the figure, the green part represents the entire dataset. Like in the Kaggle competition, a validation dataset is initially extracted and it is formed from the 20th to the end of each month, in the diagram it is represented in blue. The rest of the dataset, (those from the 1st to the 19th of each month), will be used as training data, represented in violet in the diagram. These data will be one of the inputs of the hyperparameter search technique: GridSearchCV [50]. This technique will use the following as inputs: (1) regression algorithms with their corresponding parameter grid; (2) a scoring function, to evaluate the input models, in this case, RMSLE and R2; finally; (3) a cross-validation method, in this case, TimeSeriesSplit, a method that is intended specifically for time series and which resembles the usual functioning of a system in production. This method makes it possible to progressively use data from the past for training and use future data for validation, a diagram showing its functioning g is situated in the lower part under GridSearchCV.

The GridSearch method will make all the possible combinations for each algorithm with the provided grid parameters; this will be done by employing the cross-validation method (Time Series Split) and evaluating the trained methods with the provided scoring functions. As the output of this method, models for each algorithm with the best results will be obtained and these will be evaluated with the validation dataset that had been split at the beginning, on the right-hand side of Figure 8. A Dummy Regressor has been added to the models used and was established as their prediction strategy to continually predict the average. The regression algorithms, as well as the following parameter girds, have been trained using GridSearchCV

Classification

Now let’s go through the last step that’s classification. As mentioned above separate test set is used to evaluate the classifier and find accuracy. Any classification is correct if it can be judged by calculating the number of correctly identified class samples (true positives), the number of correctly identified samples that are not members of the class (true negative, es), and samples that I was incorrectly allocated to the class (false positives) or that were not identified as class samples (false negatives)[10]. Accuracy tells us a percentage of correctly classified instances. Mathematically ACCURACY = Correctly Classified Samples/ Total Samples \*100 Classifier is trained by the training set. Two classifiers are used i.e Decision Tree (J48) classifier and the Naive Bayes classifier. Classifier output for the first classification is shown below.

1. Logistic regression

2. Naive Bayes

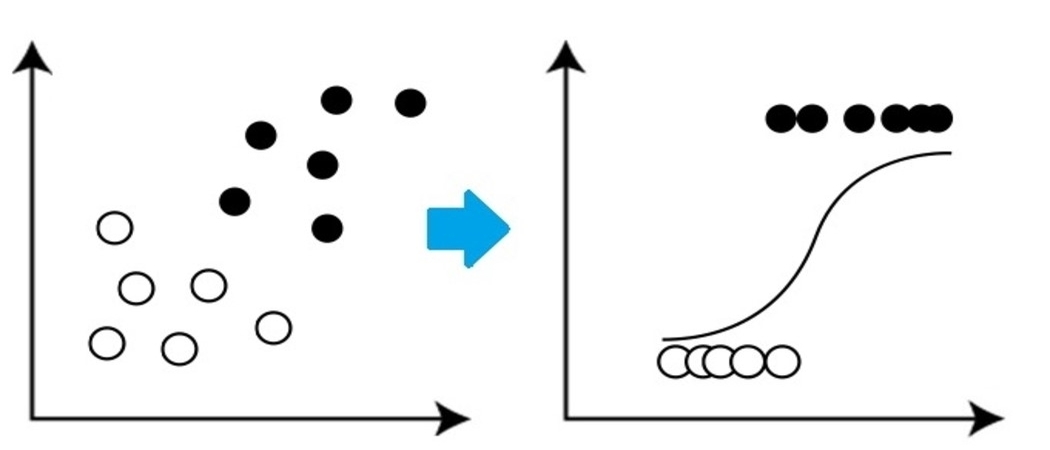
3. K-Nearest Neighbors

4.Support Vector Machine

5. Decision Tree

1. Logistic Regression

It is a very basic yet important classification algorithm in machine learning that uses one or more independent variables to determine an outcome. Logistic regression tries to find a best-fitting relationship between the dependent variable and a set of independent variables. The best-fitting line in this algorithm looks like S-shape as shown in the figure.



## 2. Naive Bayes

Naive Bayes is based on **Bayes’s theorem** which gives an assumption of independence among predictors. This classifier assumes that the presence of a particular feature in a class is not related to the presence of any other  
feature/variable.

Naive Bayes Classifier are of three types: Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussian Naive Bayes.

Pros:

* This algorithm works very fast.
* It can also be used to solve multi-class prediction problems as it’s quite useful with them.
* This classifier performs better than other models with less training data if the assumption of independence of features holds.

Cons:

* It assumes  
  that all the features are independent. While it might sound great in  
  theory, but in real life, anyone can hardly find a set of independent features.

3. K-Nearest Neighbor Algorithm

* KNN works on the very same principle. It classifies the new data points depending upon the class of the majority of data points amongst the K neighbor, where K is the number of neighbors to be considered. KNN captures the idea of similarity (sometimes called distance,  
  proximity, or closeness) with some basic mathematical distance formulas like euclidean distance, Manhattan distance, etc.
* **Choosing the right value for K**
* To select the K that’s right for the data you want to train, run the KNN algorithm several times with different values of K and choose that value of K which reduces the number of errors on unseen data.

Pros:

* KNN is simple and easiest to implement.
* There’s no need to build a model, tuning several parameters, or make additional assumptions like some of the other classification algorithms.
* It can be used for classification, regression, and search. So, it is flexible.

Cons:

* The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

## 4. SVM

SVM stands for Support Vector Machine. This is a supervised machine learning algorithm that is very often used for both classification and regression challenges. However, it is mostly used in classification problems. The basic concept of the Support Vector Machine and how it works can be best understood by this simple example. So, just imagine you have two tags: green and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x,y) coordinates, outputs if it’s either green or blue. Plot labeled training data on a plane and then try to find a plane (hyperplane of dimensions increases) that segregates data points of both colors very clearly.



But this is the case with data that is linear. But what if data is non-linear, then it uses kernel trick. So, to handle this we increase dimension, this brings data in space and now data becomes linearly separable in two groups.

Pros:

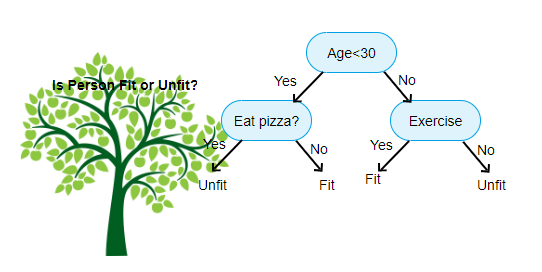
* SVM works relatively well when there is a clear margin of separation between classes.
* SVM is more effective in high-dimensional spaces.

Cons:

* SVM  is not suitable for large data sets.
* SVM does not perform very well when the data set has more noise i.e. when target classes are overlapping. So, it needs to be ha

## 5.Decision Tree

The decision tree is one of the most popular machine learning algorithms used. They are used for both classification and regression problems. Decision trees mimic human-level thinking so it’s simple to understand the data and make some good intuitions and interpretations. They make you see the logic for the data to interpret. Decision trees are not like black-box algorithms like SVM, Neural Networks, etc.



For example, if we are classifying a person as fit or unfit then the decision tree looks like somewhat this above in the image.

So, in short, a decision tree is a tree where each node represents a  
feature/attribute, each branch represents a decision, a rule, and each leaf represents an outcome. This outcome may be a categorical or continuous value. Categorical in case of classification and continuous in case of regression applications.

Pros:

* When compared to other algorithms, decision trees require less effort for data preparation while pre-processing.
* They do not require normalization of data and scaling as well.
* Model made on decision tree is very intuitive and easy to explain to technical teams and stakeholders.

Cons:

* If even a small change is done in the data, that can lead to a significant change in the structure of the decision tree causing instability.
* Sometimes calculation can go far more complex compared to other algorithms.

8. CONCLUSION:

* That's it! We reached the end of our exercise.
* Starting with loading the data so far we have done EDA, encoding of categorical columns, feature selection, and then model building.
* In all of these models our accuracy revolves in the range of 90 to 95%.
* And there is no such improvement in accuracy score even after hyperparameter tuning.
* The performance of the models is measured using an accuracy score (since the data is balanced). From the above table, it can be seen that Random Forest and XGBoost have a good accuracy score yet they are overfitted models. However, stacking(Random Forest, XGBoost, Decision Tree) not only is not overfit but also gives a good accuracy score. We can conclude that the stacking model is a good classifier of the mobile price range.

1. The price range of mobiles is thus determined and the most important features are shown in the image to the right.
2. Most expensive/flagship phones, as well as low-end phones, can be well classified but there are minor overlaps in budget and mid-range phones.