# Mobile Price Prediction using Support Vector Machine (SVM)

UNIVERSITAT POLITECNICA
DE CATALUNYA


Asha Seif
*Universitat Politècnica de Catalunya (UPC)*
asha.seif@estudiantat.upc.edu

Hang Yu
*Universitat Politècnica de Catalunya (UPC)*
hang.yu1@estudiantat.upc.edu



**ABSTRACT**

The motivation behind this project is to predict the price range of a mobile phone, based on its features such as Random Access Memory(RAM), internal memory, battery power, and others by using the art of machine learning algorithm. The aim of this work is not about predicting the actual price of a mobile phone, (i.e. it is not a regression problem) but it predicts the price range which is either low (0), medium(1), high(2) and very high (3). Five machine learning algorithms including Random Forest Classifier, Support Vector Machine (SVM), Naïve Bayes, and others were tested to find the one with the highest prediction accuracy that will be used as a final prediction machine learning model for this problem. Finally, the SVM attain the best estimator score of approximately 97% prediction accuracy.



**General term:** Machine Learning

**Keywords:** Machine Learning, Prediction, Support Vector Machine(SVM)


## 1. INTRODUCTION

The most difficult challenge many business companies face is to predict the price of a product when they want to introduce a new product to the market or increase the price of their existing products. Of course, every company desires to have the best price which will bring the benefit to their company as well as be affordable to their target customer. There are several methods that can be used to predict the price as the one suggested by Symson [9] are cost-based, demand-based, competition-based, and value-based. But all of these methods need some data that can be used to generate valuable information for the success of business companies. There are a lot of websites such as Kaggle, UCI, Google Dataset Search, and many others that make the data available for free. Aside from data nowadays machine learning provides us with different tools such as Decision Tree, Naïve Bayes, and Support Vector Machine that specifically make use of this data to generate useful insights. As our main goal of this research work is to predict the price range of the mobile phone we want to use the benefits and advantages of machine learning on solving this problem. Therefore we will use a different machine learning algorithm and compare them and choose the one which is suitable according to the nature of our datasets. The main objective is to use different phone features such as RAM, internal memory, battery power, Bluetooth, and other features to predict the price range of the mobile phone. Since the nature of this problem falls under supervised machine learning, especially in the classification kind of

problem, we will use an algorithm that works well for this kind of problem. Therefore we consider using Random Forest Classifier, Support Vector Machine, Naïve Bayes, Extreme Gradient Boost, and Decision tree and evaluate them based on their learning skills and prediction score. After that, we will choose the one with the highest prediction score as our main algorithm for predicting the price range.

## 2. LITERATURE REVIEW

The work of predicting the price range of different mobile prices is previously done by many people. There is a lot of contribution and research conducted on this dataset that we are working with, researchers implement different classification algorithms such as Random Forest Classifier**(RFC),** K-Nearest Neighbour **(KNN ),** and Naïve Bayes, Support Vector Machine **(SVM),** and other models. For example, Prateek(2022) in his machine learning work [8] implement three machine learning algorithms which are Random Forest Classifier, KNN, Naïve Bayes, and Support Vector Machine. The prediction accuracy attained by three algorithms are in approximately above 90% with SVM being the highest at approximately 96%, and the only Naïve Bayes got the accuracy of 85%. Also, the article published by Kalaivani [6] with his co-author uses algorithms like SVM, RFC, and Logistic Regression with the prediction accuracy of 97%, 87%, and 81% respectively after applying features selection. Hassanali[7] uses a Neural network to solve this problem and got an accurate prediction of 94%. Compared to many other authors that publish on the website and blog in working with this data set among all the algorithms they use Support Vector machine seems to get the highest prediction score. Through this finding, we can conclude that the SVM algorithm is the best model for this dataset, however, we cannot make that conclusion to see other algorithms and check the results because the prediction of algorithms' performance depends on the data pre-processing and features selection as well as hyperparameter tuning of the same algorithms. This previous work enables us to keep SVM among the candidate set of algorithms that we consider to use in our research work. However, we cannot just use this algorithm as our final selected model before going some steps to prove that SVM is a suitable model for our problem

## 3. METHODOLOGY

The methodology used to conduct this research work includes the process of **data collection**, **data exploration** to get an understanding of the data and its distribution, pre-processing and feature selection cleaning the data by removing duplicates, checking for incorrect values, and filtering most important features, **machine learning training** of five classification algorithms and evaluate their performance, building the final model that will predict the price range of the phone and analysis the results obtained.

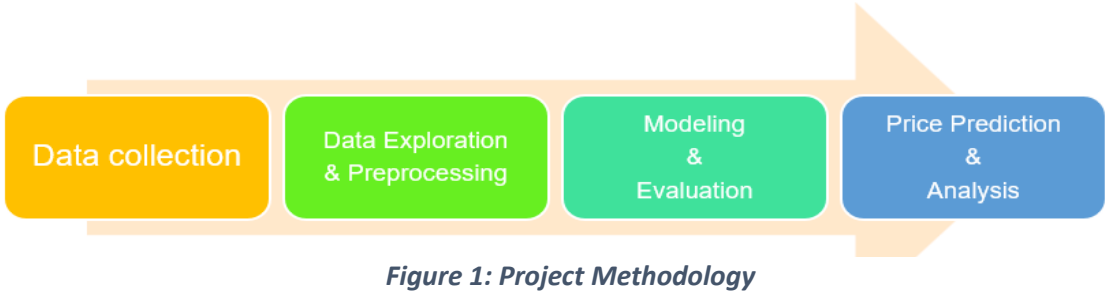



*Figure 1: Project Methodology*

### 3.1 Data Collection

The data used in this research work was collected from Kaggle [1], a data repository website. The data consists of 2000 samples with 21 attribute features which are in the format of .csv, the whole data samples are present in a single file named train.csv The nature of the dataset was mainly with many categorical features such as Bluetooth, Wi-Fi, 3G, 4G with some numeric types features such size of

internal memory, RAM, battery power and others. **Table 1** below summarizes the features of the datasets

**Table of Summary of Data Attributes Descriptions in Dataset**

| Attribute Name | Description | Type of Data | Units |
|---|---|---|---|
| battery_power | Total energy a battery can store | Numeric | mAh |
| blue | Has Bluetooth (1) or not (0) | Categoric | NA |
| clock_speed | The speed at which the microprocessor executes instructions | Numeric | GHz |
| dual_sim | Has dual sim support(1) or not (0) | Categoric | NA |
| Int_memory | Internal memory storage | Numeric | GB |
| fc | Front camera | Numeric | pixel |
| four_g | Has 4G(1) or not(0) | Categoric | NA |
| m_dep | Mobile depth | Numeric | cm |
| mobile_wt | Weight of the mobile phone | Numeric | gram |
| n_cores | Number of cores of the processor | Numeric | NA |
| pc | Primary Camera | Numeric | megapixel |
| px_height | Pixel Resolution Height | Numeric | cm |
| px_width | Pixel Resolution Width | Numeric | cm |
| ram | Random Access Memory | Numeric | mb |
| sc_h | Screen Height of mobile | Numeric | cm |
| sc_w | Screen Width of mobile | Numeric | cm |
| talk_time | The longest time that a single battery charge will last when you are talking | Numeric | seconds |
| sc_w | Screen Width of mobile | Numeric | cm |
| three_g | Has 3G (1) or not (0) | Categoric | NA |
| touch_screen | Has touch screen (1) or not (0) | Categoric | NA |
| wifi | Has wifi(1) or not(0) | Categoric | NA |
| price_range | The price range of the phone where of 0(low), 1(medium), 2(high) and 3 (very high) | Categoric | NA |

***Table 1: Features Description of The Dataset***

The nature of this problem is a multi-classification where the algorithm needs to predict more than two classes, the target variable is **price_range** which is categorized by a number 0 as **low**, 1 as a **medium**, 2 as **high** and **3** as very high cost. Since we want to use a machine learning algorithm we need to change the data type of any text type to a numeric format so that it could be easy for a machine learning to learn and predict accurate, our luck is that these data are in the numeric format and therefore we did not change the data type of any of these features.

## 3.2 Exploration Data Analysis and Pre-processing

Exploratory Data Analysis (EDA) was conducted to explore and analyze different features of the data by using visual techniques, which are more readable compared to reading them in a table format. Through this process, we have been able to perform different tasks like checking for missing values, data distribution among them, and how the features depend on each other, after getting an insight into the nature and features of the data we perform data pre-processing work so that we can have a clean data for making accuracy prediction.

### *3.2.1 Data Visualization and Distribution*

Different graphs and diagrams have been created to see the distribution of the data and get the answer to some observation questions. Data were categorized into continuous and categoric features where we made an assumption that categoric features are those with a unique value of less than 20. Most of the categorical features show the good distribution of the data where data values seem to be equally distributed among the classes, however, the only feature of the **three_g** as shown in **figure 2** has an imbalance of data where phones with 3G are **75%** and only **25%** are those with no 3G features. But for this issue of 3G, we did not do any normalization techniques to solve this problem, because in the reality most the phones now are having 3G as minimum network technology features, therefore we did not see any important reasoning for balancing the data value and losing some important value from it.

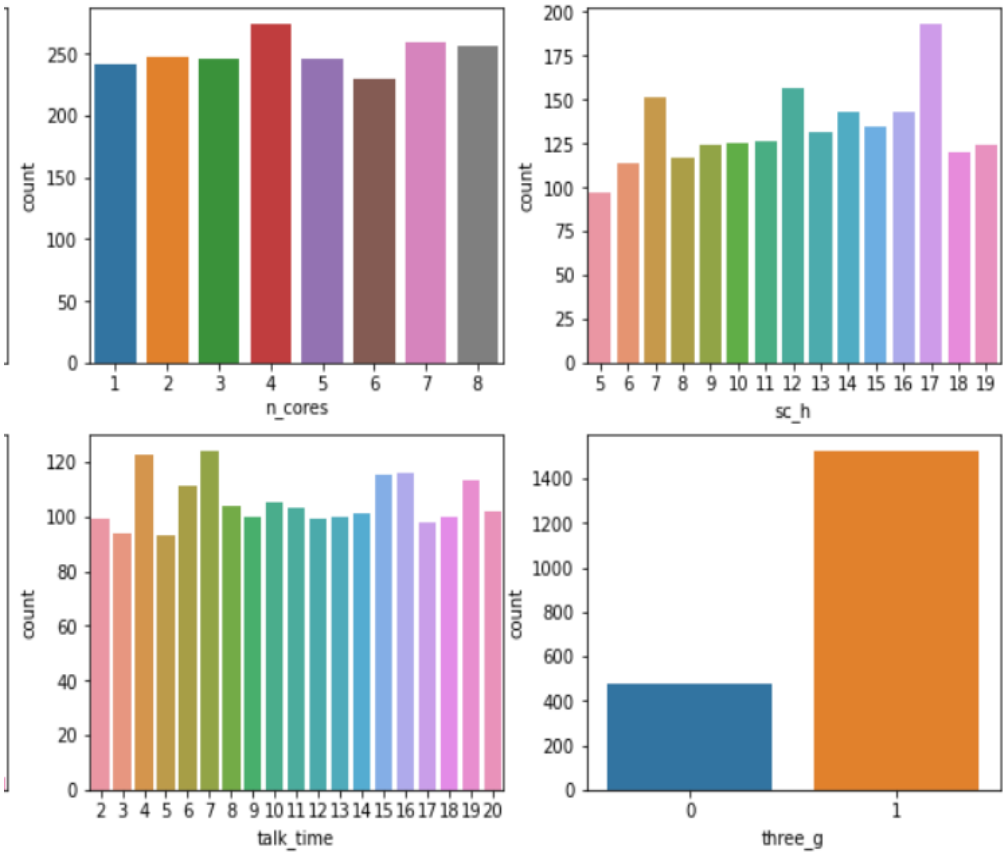



*Figure 2: Data Distribution of Categoric Features*

On continuous features, most of the data seem to be well-distributed kind of gaussian and some features like Screen Width **(sc_w),** Front Camera **(fc),** Pixel resolution height **(px_height)** are slightly skewed at the right side (i.e Positively skewed distribution ) as shown in **Figure 3.** Apart from data distribution through this process, we have been able to get the answer to some questions which people like to ask when they want to buy a mobile phone to look for some specifications like RAM, internal memory, and pixel of camera. Therefore we have formulated some questions like how does the price range affected by RAM, internal memory, the number of the core of the processor, and other features? The answers to these questions are visually presented in **Figure 4**.

**Observation of EDA**

- The price of the mobile phone increases as the size of RAM increases
- The price of the phone increases as the battery power increases from the price range of 0 to 1 and 2 to 3, however, the phone price in the range of 1 and 2 seem to have the same battery power, meaning that the size of battery power in this kind of phone can be the same while the price range is different.
- The weight of the phone does not have any impact on the price range
- The phones which support 4G and 3G share the same price range
- The number of cores and clock speed does not have much correlation with the price range, i.e there are phones with the highest price range but low clock speed and a minimum number of cores of the processor.
- The price range increases with an increase in pixel resolution height and width.

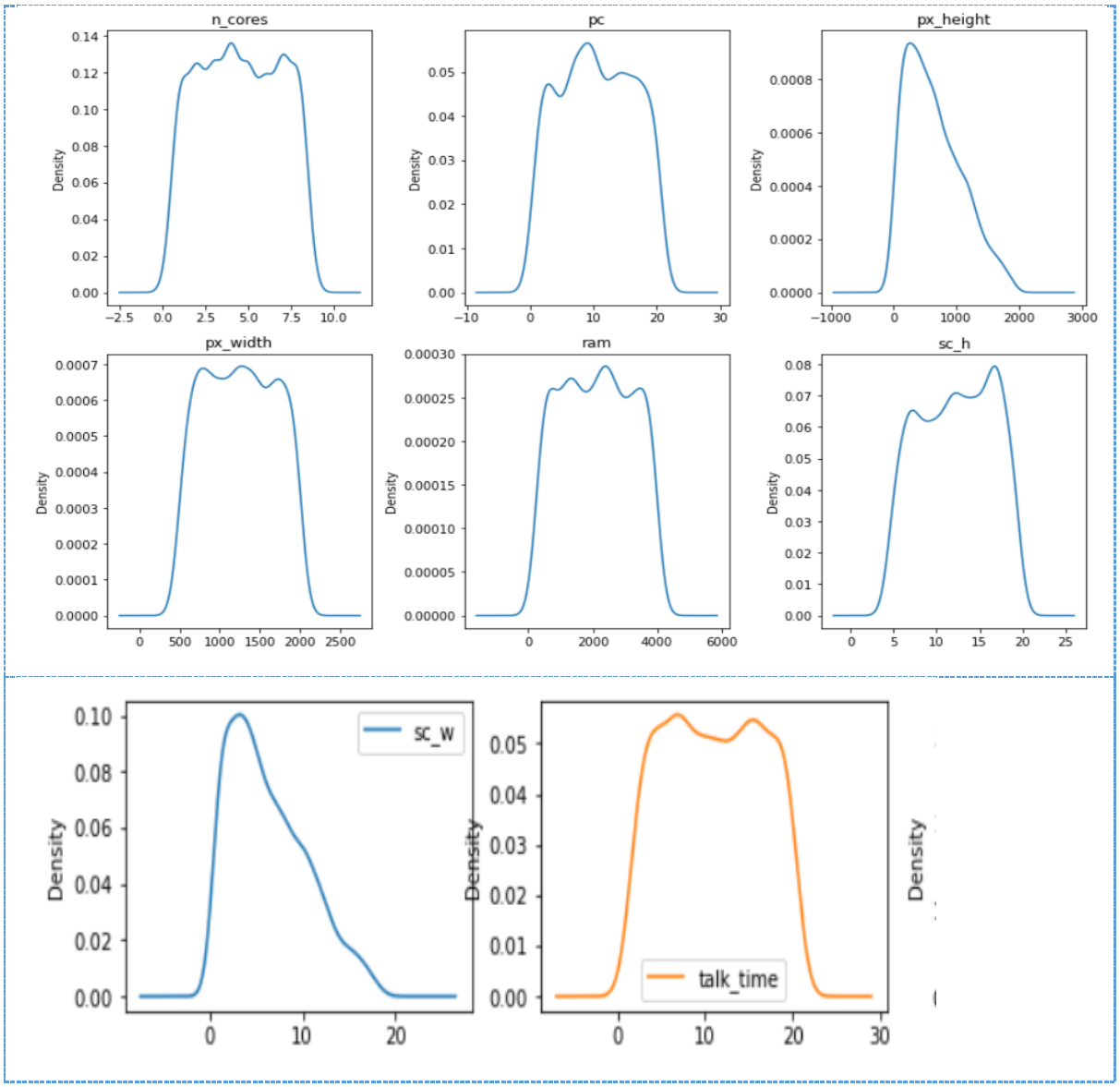



*Figure 3: Data Distribution in Continuous Features*

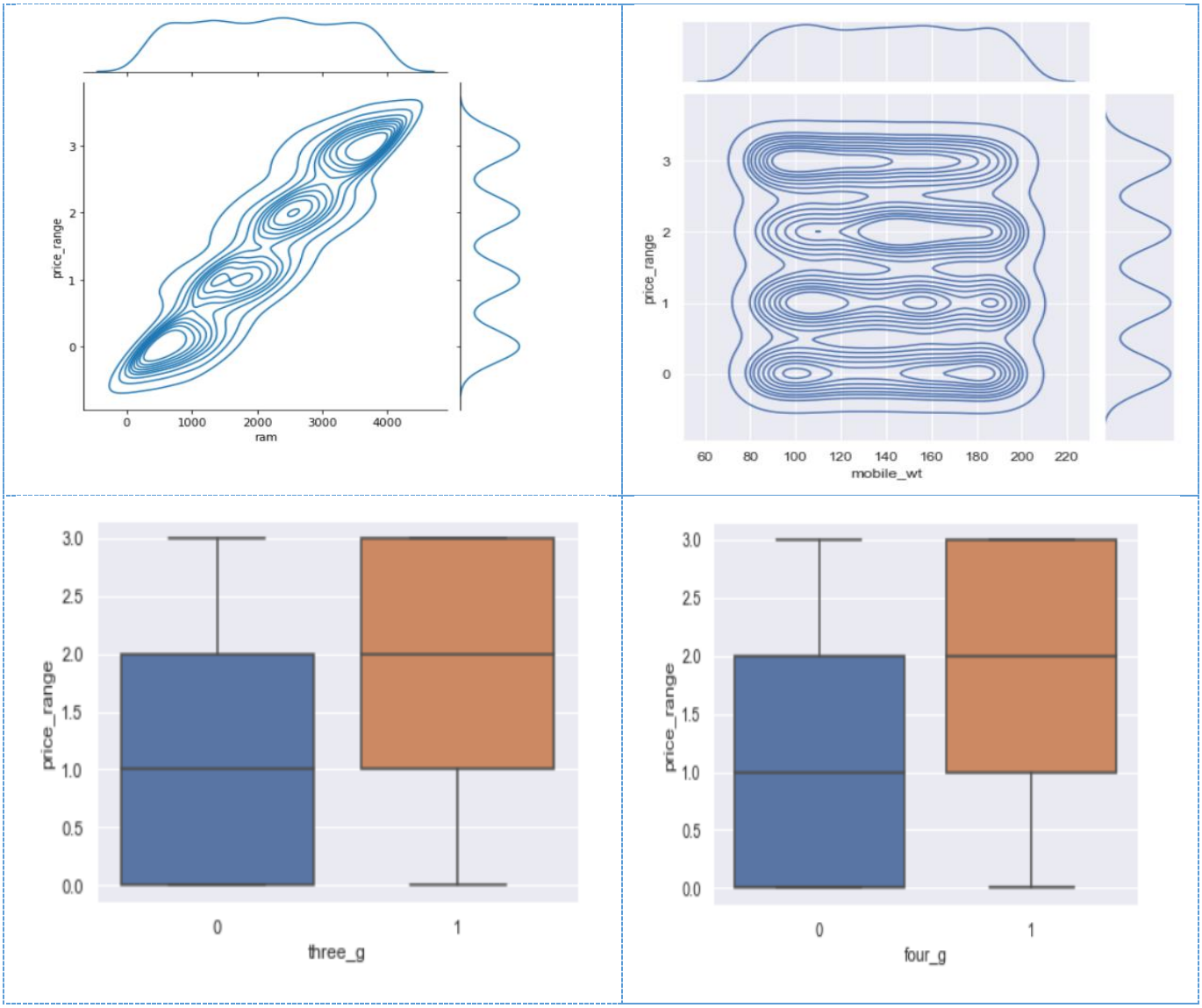



*Figure 3: Relationship of Common Attribute and Price Range*

### *3.2.2 Data Cleaning and Pre-processing*

After getting an insight into the features of the data distribution we continue with data cleaning and preparation. In this part, we check for any missing values, duplicates, and incorrect data values, outliers and feature selection.

#### *3.2.2.1 Dealing with Missing Value and Duplicate*

There were no missing values and duplicates of the data, and the dataset is almost clean however we have noticed some incorrect values in continuous features have a value of 0 and according to the nature of these data there are not supposed to have a value of 0 and therefore we treat this data as a missing value. Since these data are many in number we could not simply remove them instead we perform some imputation process to add the value by comparing the features of other attributes with the same characteristics. The method used is **KNN** referred to as **K-Nearest Neighbour** and the value of k we choose is 1. The reason for using KNN is after getting the idea as suggested by Jason [5] that the KNN algorithm has proven to be a generally effective method of computing the missing values. **Figure 5** show those data shape before the imputation method and after the imputation method and the total number of the incorrect value presented before imputation and after doing imputation there were no features with a value of 0.

**Before Imputation**

```
In [15]: (mobile.fc==0).value_counts()
         (mobile.pc==0).value_counts()
         (mobile.sc_w==0).value_counts()
         (mobile.px_height==0).value_counts()

Out[15]: False    1526
         True      474
         Name: fc, dtype: int64

Out[15]: False    1899
         True      101
         Name: pc, dtype: int64

Out[15]: False    1820
         True      180
         Name: sc_w, dtype: int64

Out[15]: False    1998
         True        2
         Name: px_height, dtype: int64
```

**After Imputation**

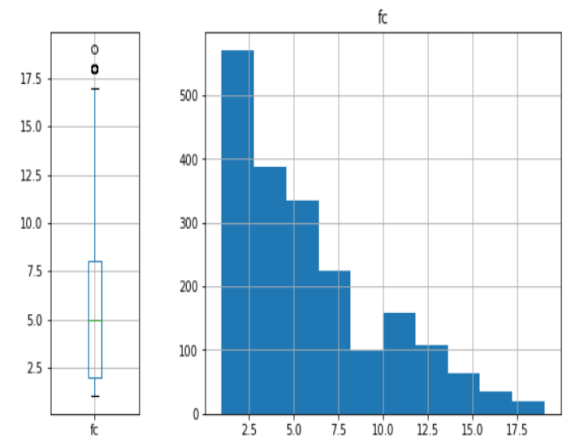
```
In [66]: (mobile.fc==0).value_counts()
         (mobile.pc==0).value_counts()
         (mobile.sc_w==0).value_counts()
         (mobile.px_height==0).value_counts()

Out[66]: False    2000
         Name: fc, dtype: int64

Out[66]: False    2000
         Name: pc, dtype: int64

Out[66]: False    2000
         Name: sc_w, dtype: int64

Out[66]: False    2000
         Name: px_height, dtype: int64
```

*Figure 4: Data Information before & After Imputation*

### *3.2.2.2 Detecting and Removing Outliers*

After that, we continue through the process of finding outliers that are present in the data, to detect outliers we **Inter-Quartile Range** (**IQR) Method** and plot the data in a boxplot, which enables us to draw inference from it and tells us about the various metrics of a data arranged in ascending order. These metrics include minimum and maximum values, and the median. Those values that seem to be above the maximum value or below the minimum value are considered as upper bound and lower bound outliers respectively. Two features; **fc** and **px_height** were detected to have upper bound outliers, even though these numbers of outliers are small in magnitude and we could have simply deleted them but we did not want to lose any information and our aim is to get a machine learning model which can predict the price range accurately, therefore we perform median value imputation to deal with these outliers. **Figure 6** shows data with an outlier (on the LHS) median imputation and without outliers(on the RHS)

**Data With outliers**




**Data after Median Imputation**

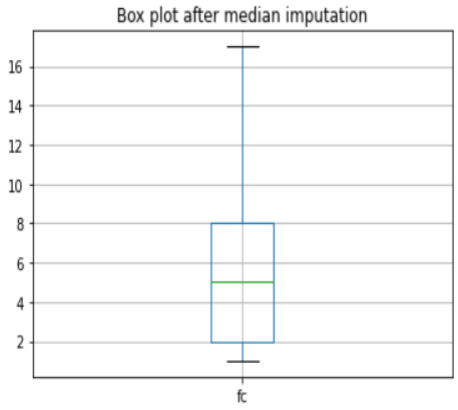



*Figure 5:Data with and without outlier After Median Imputation*

### *3.2.2.3 Feature Selection*

It is not a good idea to fit all the features that do not have any impact on the predictable features because it will make the machine learning algorithms keep considering those features and therefore may cause bias in the prediction. Therefore before training any model we need to do features selection on the data and choose only the important features that have a relationship with the data. To find correlation among the features we use **corr() of** with '**pearson**' method which returns the value between **-1** and **1**, where 1 represent positive correlation, meaning that either increasing or decreasing the value will impact other attributed to going in the same direction, -1 represent negative correlation in which when one the value of one attribute increases the on other correlated attributes will decrease and the value of **0** represents neutral correlations, **Figure 7** shows all the data in the yellow, green and blue cell are correlated features; therefore in top 10 of the features RAM, battery power, pixel resolution width, and height show a positive correlation with a price range and a number of cores of process, mobile depth, clock speed, mobile weight, and touch screen are negatively correlated features.

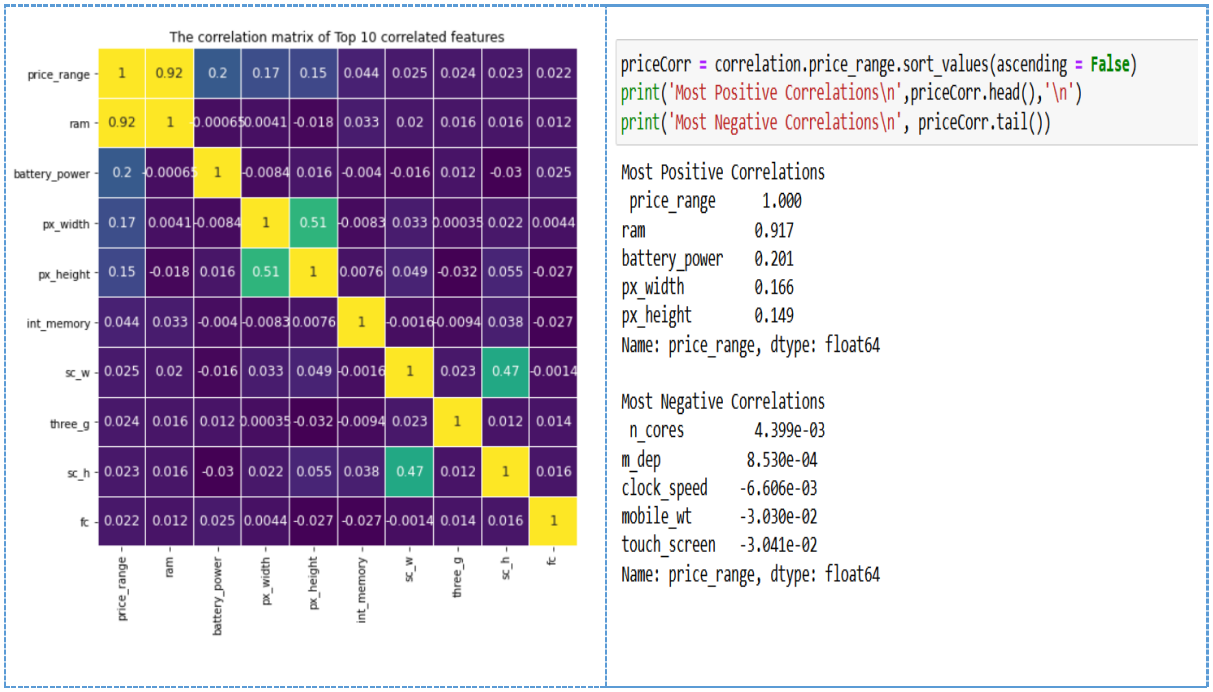



*Figure 6: Correlation between Features*

Also, we use the feature importance method from the sklearn library to select the 10 most important features for predicting the price range of the phone. The results also are the same as those obtained in the correlation analysis as represented in **Figure 7.** During the data preparation work, we have not applied any filtering however we just got an idea of the existence of the features which are not important to predict our target value. These best 10 features will be used during the process of training our final model. Then we finalize the data pre-processing task by dividing the data set randomly into training and testing sets in 80% to 20% respectively. The training set will be used for the whole process of making the algorithms learn and the testing set will be used as a final test for measuring the performance of the prediction of the chosen model for unseen data. Therefore as the total number of samples is 2000, the 1600 samples will be used as training and the remaining 400 samples will be used as a testing set.

### 3.3 Machine Learning Model and Evaluation

After finishing the part of pre-processing now it is time to select the best algorithm that will work well in our datasets, as mentioned earlier the five classification algorithms will be evaluated, and choose the winner as a final model to predict the price range of the phone. These algorithms are Random Forest Classifier **(RFC),** which is ensemble algorithms that combine multiple decision trees to solve a complicated classification problem. Support Vector Machine **(SVM) is a supervised learning model that** analyses data for classification, regression analysis, and outlier detection. SVM algorithm creates the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. SVM is proven to be the best algorithm for multi-class problems[3]. **Extreme Gradient Boost (Xgboost)** is a scalable, distributed gradient-boosted decision tree **(GBDT)** machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. **Naïve Bayes(NB)** is a probabilistic machine learning model that's used for classification tasks based on the Bayes theorem which has proven to work well in the data with Gaussian distribution. **Decision Tree Classifier(DTC)** is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems

#### *3.3.1 Model Selection and Validation using 5-Fold Cross-Validation*

The five algorithms were trained and evaluated by using k-fold cross-validation techniques with the combination of GridSearchCV for finding the best parameters for each algorithm. The value of k used is 5, the reason why we use k-5 is that we have a quite large number of sample which is about 1600 samples and the GridSearchCV took a much time while find the best parameters combination. This is proven when we first use a value of k=10 the process took a lot of time just to complete checking the best estimator for a single algorithm and therefore finally we decide to use k=5.

| Algorithms | Best Estimator | Best Score |
|---|---|---|
| RFC | `RandomForestClassifier(criterion='entropy', max_features=15, min_samples_leaf=3, min_samples_split=6, n_estimators=30` | 0.906875 |
| SVM | `SVC(C=1, gamma=0.1, kernel='linear')` | 0.970625 |
| XGBOOST | `XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1,colsample_bytree=0.8, enable_categorical=False, gamma=0, gpu_id=-1,importance_type=None, learning_rate=0.01, max_delta_step=0, max_depth=4,min_child_weight=6, missing=nan, monotone_constraints='()', n_estimators=5000, n_jobs=8, num_parallel_tree=1, objective='multi:softprob', predictor='auto', random_state=0, reg_alpha=0.005, reg_lambda=1, scale_pos_weight=None, subsample=0.8, tree_method='exact', validate_parameters=1, verbosity=None` | 0.906875 |
| Naives Bayes | `GaussianNB(var_smoothing=1e-05)` | 0.801875 |
| DCT | `DecisionTreeClassifier(max_features='auto', min_samples_leaf=3, min_samples_split=12, random_state=123)` | 0.703125 |

***Table 2: Prediction Score & Best Estimator***

The results of the model validation are summarized in **Table 2** which shows the best estimator of each algorithm along with the best parameters and the best score on the prediction of the price range. Among all the algorithms SVM got the highest best score of about approximately 97%, followed by RFC with 91%, XgBoost with 91%, Naïve Bayes with 80,% and the lowest score is 70% of the DCT. Therefore we have decided to choose SVM as our final model that will be used for the final testing and predictions of mobile price. The comparison among these scores is best visualized in **Figure 8**.

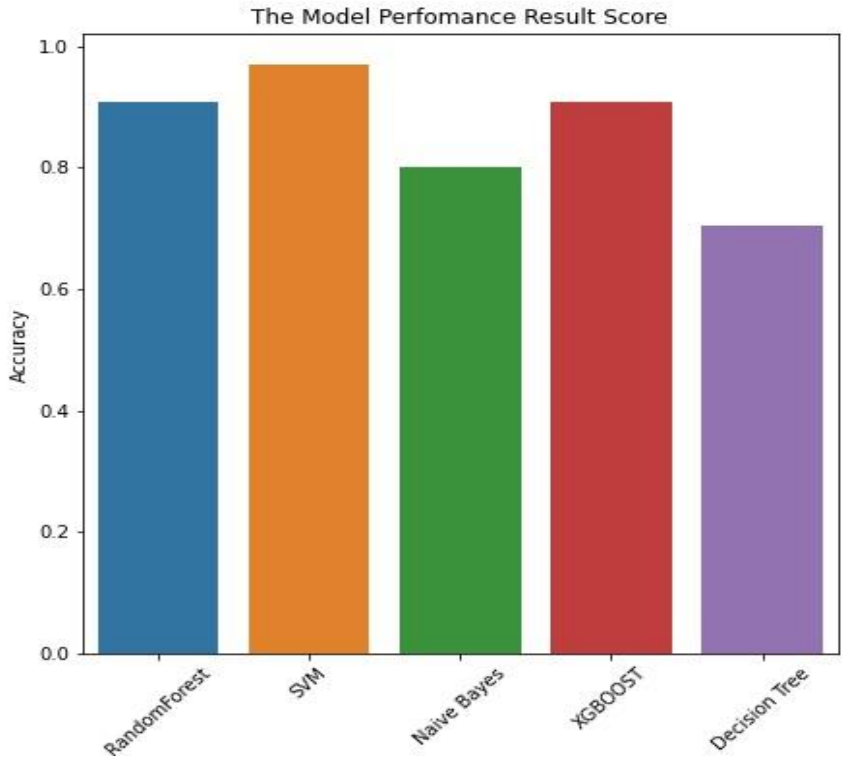



*Figure 7: Performance Prediction Score Results*

### *3.3.2 Training of The Final Model and Validation Analysis*

Support Vector Machine (SVC) is used to build the final model for predicting the price range of the mobile with parameters configuration of **(kernel='linear', gamma=0.1, C=1).** The total training set uses **1072** samples of the data and the remaining **528** samples are used as a testing set where Class (0) consists of **129** samples, Class 1 consists of **130** samples, Class 2 consists of **137,** and Class 3 consists of **132** samples. After fitting the final model to the training set the prediction performance attained by SVM is **96%** with a mean prediction error of approximately **4%.** The prediction result is summarized in the confusion matrix in **Figure 9** which is interpreted as follows:

- **Class 0 (low): 125** are correctly predicted and only **4** samples are classified in **Class 1**
- **Class 1 (medium): 126** are correctly predicted, **3** samples placed on **Class 0** and **1** in **Class 2**
- **Class 2 (high ):125** are correctly predicted, **6** samples placed in **Class 1** and another **6** in **Class 3**
- **Class 3 (very high): 130** are correctly predicted and only **2** samples are placed in **Class 2**

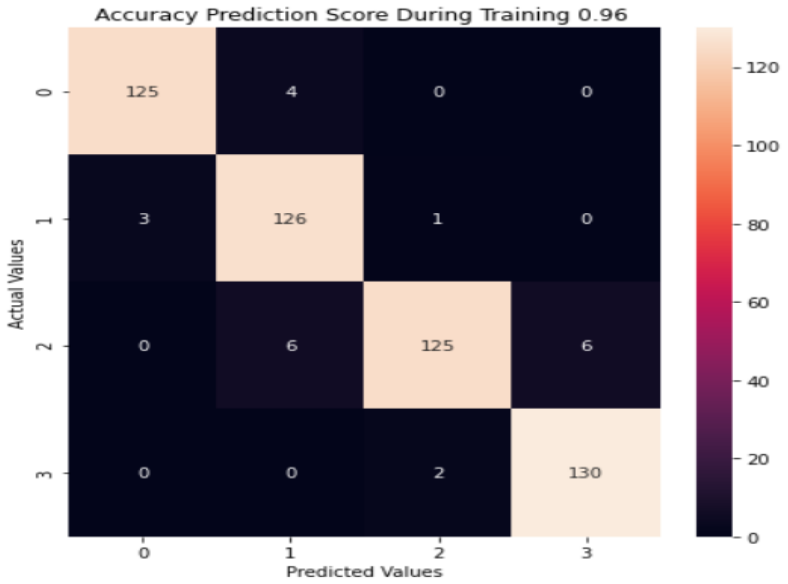



*Figure 8: Prediction Result During Training*

### 3.4 Price Prediction and Analysis

After completing the process of training our final model, then it is time to do a prediction and final test by using unseen data that were not used during the training process. The data consists of **400** samples which were equally distributed as **100** samples per class. The model predicts the results with an accuracy of **98%** with a mean error of **2%,** the prediction report is a summarized in **Figure 10** which is interpreted as follows:

- **Class 0 (low):** All **100** samples are correctly predicted
- **Class 1(medium): 97** are correctly predicted,**2** samples are placed in **Class 0** and 1 in **Class 2**
- **Class 2(high): 96** samples are correctly predicted,**2** samples are placed in **Class 1,** and the remaining **2** in **Class 3**
- **Class 3(very high):** All **100** samples are correctly predicted

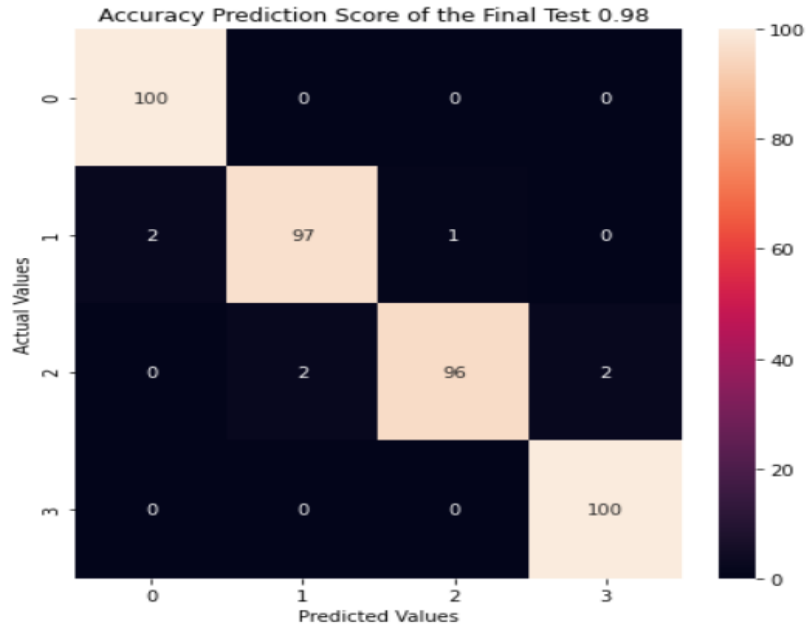



*Figure 9: Prediction Results of The Final Test*

# 4. CONCLUSION AND FUTURE WORK

## 4.1 Conclusion

Among all the selected machine learning algorithms SVM is the best and most suitable model for our dataset which attain the best estimator score of approximately 97%, followed by Random Forest Classifier and XGBoost with a score of approximately 91%, Gaussian Naïve Bayes with 80% and Decision Tree with 70%. Therefore the Support Vector Machine is the machine learning algorithm used for predicting the price range of mobile phones in this research work.

During the final testing phase, 400 samples are used which are not present during the training process, among them, only 7 samples are incorrectly classified and the remaining samples are correctly classified to their respective classes. SVM show a prediction accuracy of 98% with a mean error of 2%. However, it couldn't predict well in Class 1 (medium cost) and Class 2 (high cost) which consists of most related features which are, in general, true when compared to the nature and features of these kinds of phones.

As our main goal is to predict the price range of the mobile phone and see how the price is affected by the features of the phone, the analysis shows that the price range of the phone is highly related to features like RAM, battery power, pixel resolution height, and width. But the analysis couldn't show any impact of the internal memory with respect to the price of the mobile phone, while normally in life we know that the internal memory storage of the phone is the most important feature which affects the price of the phone.

## 4.2 Future Work

- More machine learning techniques can be used to maximize the prediction accuracy of the price
- More features can be added to the datasets that could differentiate the phone of Class 1 and Class 2 so that machine algorithms could be accurately distinguished among the phones of these classes
- Data can be improved to see how the internal memory affects the price of the phone as used in relation to normal life.

# References


[1] A.Sharma. (2018). *Mobile Price Classification*. Retrieved from Kaggle: https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification?select=train.csv

[2] Brownlee, J. (2018, April 27). *How to Calculate Correlation Between Variables in Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

[3]J.Brownlee. (2020, April 8). *4 Types of Classification in Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[4] J.Brownlee. (2020, July 31). *How to configure k-fold-cross-validation*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/

[5] J.Brownlee. (2020, September 19). *Hyperparameter Optimization With Random Search and Grid Search*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/

[6]Kalaivan, K. (2021, November 1). *Predicting The Price range of Mobile Phones using Machine Learning Techniques*. Retrieved from AIP Conference Proceedings: https://aip.scitation.org/doi/epdf/10.1063/5.0068605

[7] M.Hassanali. (2022, June 05). *Kaggle*. Retrieved from 95% accuracy using Neural Networks Only: https://www.kaggle.com/code/mohamedhassanali/94-25-accuracy-using-neural-networks-only

[8] Majumder, P. (2022, February 23). *Learn Mobile Price Prediction Through Four Classification Algorithms*. Retrieved from Analyticsvidhya: https://www.analyticsvidhya.com/blog/2022/02/learn-mobile-price-prediction-through-four-classification-algorithms/

[9]*The 8 Biggest Pricing Challenges*. (n.d.). Retrieved from Symson: https://www.symson.com/blog/the-8-biggest-pricing-challenges