**MOBILE PRICE PREDICTION**

**Sundar K**

**Alma Better**

|  |  |
| --- | --- |
| **1** | **Abstract** |
| **2** | **Problem Statement** |
| **3** | **Steps Involved** |
| **4** | **Algorithms** |
| **5** | **Models performed** |
| **6** | **Result** |
| **7** | **Conclusion** |
| **8** | **Reference** |

|  |
| --- |
| **CONTENTS** |

**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 Almabetter 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 as higher accuracy as possible. Results are compared in terms of highest accuracy achieved and minimum features selected. Conclusion is made on the base of best feature selection algorithm and best classifier for the given dataset.

**2.Problem statement**

The data contains information regarding mobile phone features, specifications etc and their price range. The various features and information can be used to predict the price range of a mobile phone.

* Price\_range- Price of the mobile
* battery\_power- Battery Power in mAh
* blue- Has BlueTooth or not
* clock\_speed- Microprocessor clock speed
* dual\_sim - The phone has dual sim support or not
* fc- Front Camera Megapixels
* four\_g- Has 4G support or not
* Int\_memory- Internal Memory in GigaBytes
* m\_dep- Mobile Depth in Cm
* mobile\_wt- Weight of Mobile Phone
* n\_cores- Number of cores in the processor
* Pc- Primary Camera Megapixels
* px\_height- Pixel Resolution height
* px\_width- Pixel resolution width
* ram- RAM in MB
* sc\_h- Mobile screen height in cm
* sc\_w- Mobile screen width in cm
* talk\_time- Longest time after a single charge
* three\_g- 3g or not
* touch\_screen- Has touch screen or not
* Wifi- Has wifi or not

**3.Steps involved:**

* **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is price range with other independent variables. This process helped us figuring out various aspects and relationships among 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 contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.

* **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.

* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **KNN**
2. **Decision Tree**
3. **Random Forest Classifier**

**4.Algorithms**

**1.KNN Classifier**

The K Nearest Neighbor method is a type of supervised learning technique that is used for classification and regression. It’s a flexible approach that may also be used to fill in missing values and resample datasets. K Nearest Neighbor examines K Nearest Neighbors (Data points) to forecast the class or continuous value for a new Datapoint, as the name indicates.



Image: https://blakelobato1.medium.com/k-nearest-neighbor-classifier-implement-homemade-class-compare-with-sklearn-import-6896f49b89e

The K-NN method saves all available data and classifies a new data point based on its similarity to the existing data. This implies that fresh data may be quickly sorted into a well-defined category using the K-NN method. The K-NN algorithm is a non-parametric algorithm, which means it makes no assumptions about the underlying data. It’s also known as a lazy learner algorithm since it doesn’t learn from the training set right away; instead, it saves the dataset and performs an action on it when it comes time to classify it.

**2. Decision Tree**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next nod

****

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

**3.Radom forest**

A random forest is a supervised machine learning method built from decision tree techniques. This algorithm is used to anticipate behaviour and results in a variety of sectors, including banking and e-commerce.

A random forest is a machine learning approach for solving regression and classification issues. It makes use of ensemble learning, which is a technique that combines multiple classifiers to solve complicated problems.

A random forest method is made up of a large number of decision trees. The random forest algorithm’s ‘forest’ is trained via bagging or bootstrap aggregation. Bagging is a meta-algorithm ensemble that increases the accuracy of machine learning algorithms.

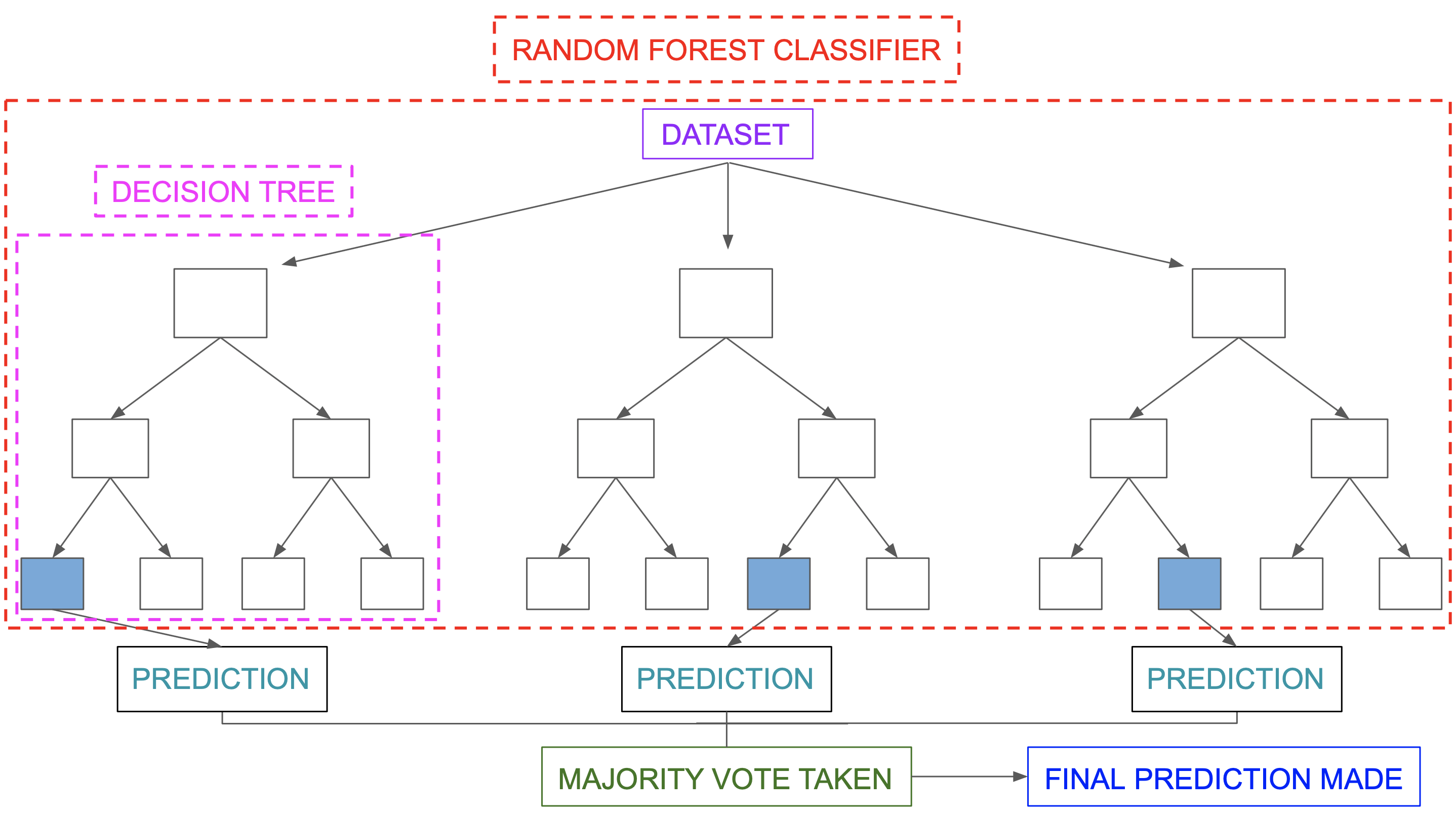
The outcome is determined by the (random forest) algorithm based on the predictions of the decision trees. It forecasts by averaging or averaging the output of several trees. The precision of the outcome improves as the number of trees grows. ****

Image Source: https://miro.medium.com/max/5752/1\*5dq\_1hnqkboZTcKFfwbO9A.png

A random forest system is built on a variety of decision trees. Every decision tree is made up of nodes that represent decisions, leaf nodes, and a root node. The leaf node of each tree represents the decision tree’s final result. The final product is chosen using a majority-voting procedure. In this situation, the output picked by the majority of the decision trees becomes the random forest system’s ultimate output. Let us now implement the random forest algorithm.

1. **Model performance:**

Model can be evaluated by various metrics such as:

**Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification modelis at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

**Precision/Recall**-

Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP

Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

**Accuracy**-

Accuracy is given by the number of correctly classified examples divided by the total number

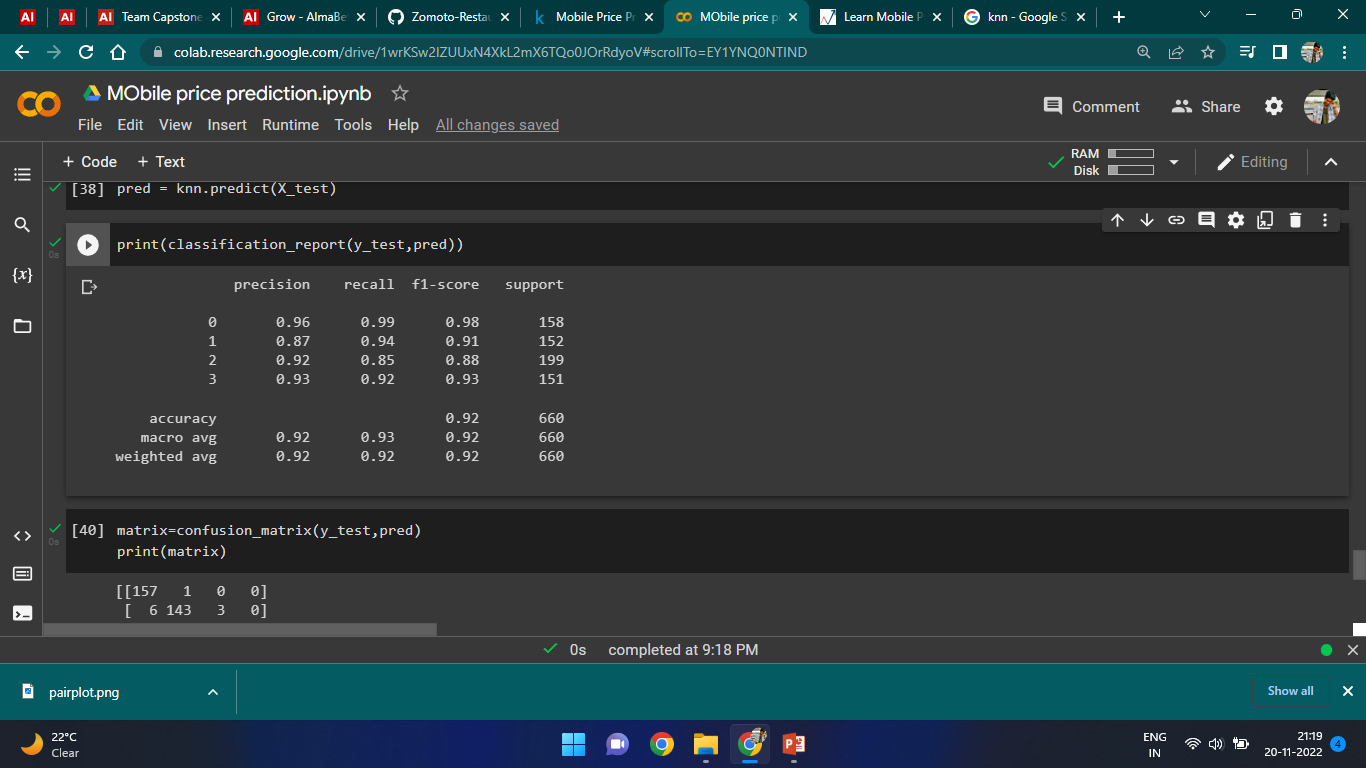
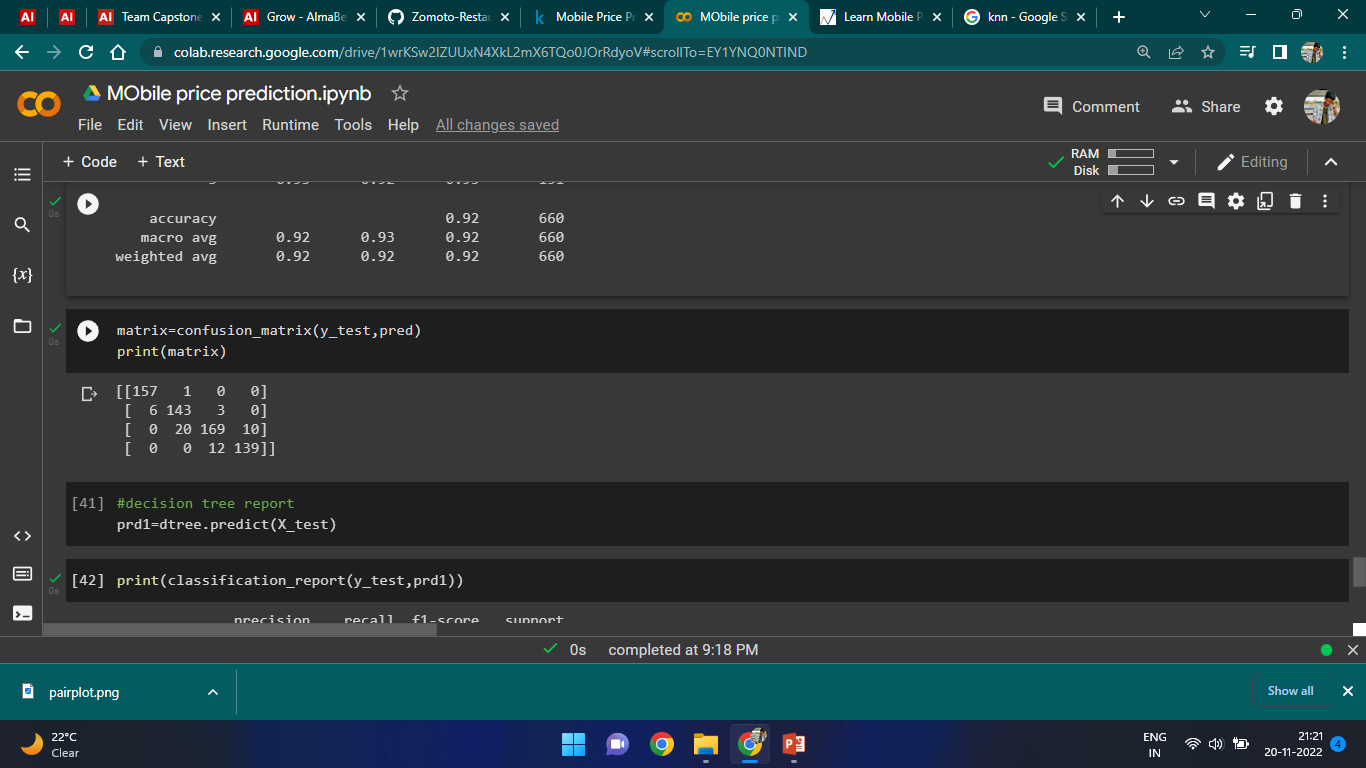
of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

**Area under ROC Curve(AUC)**-

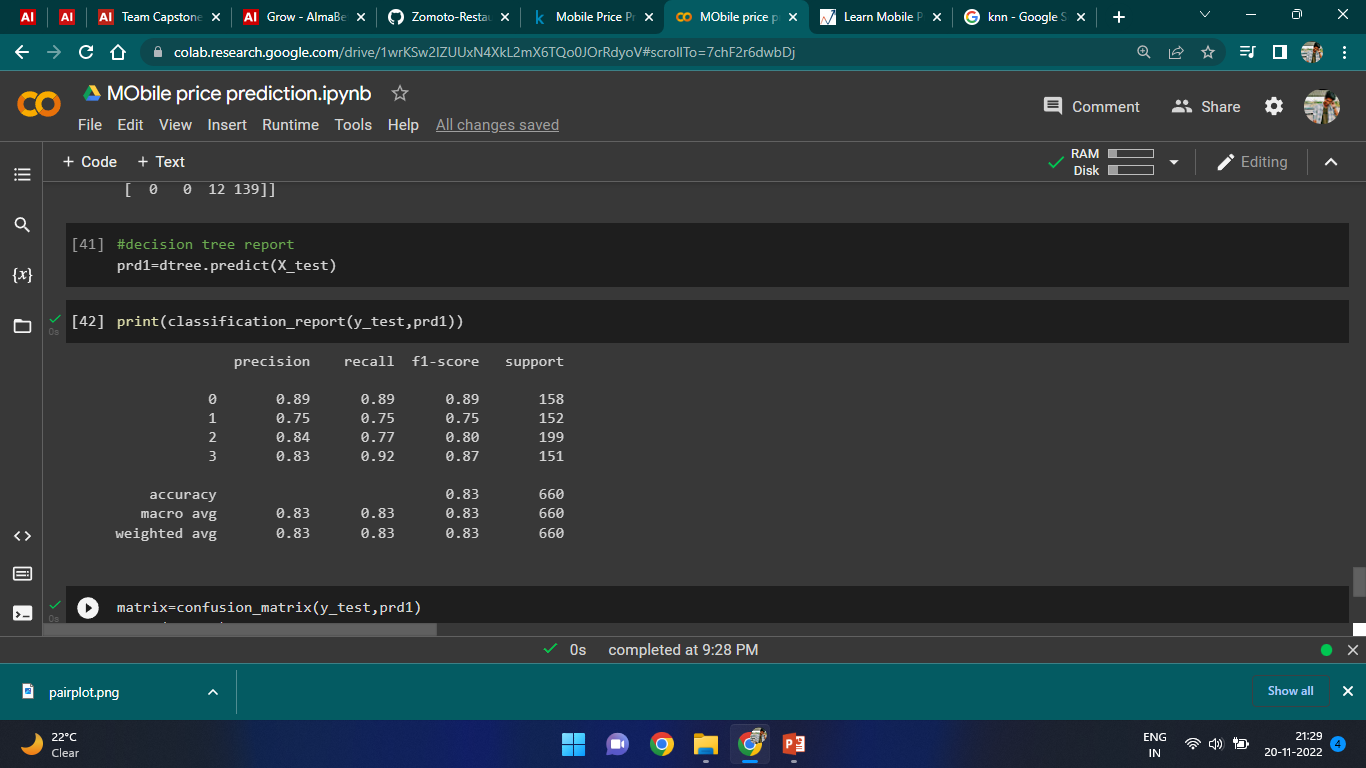
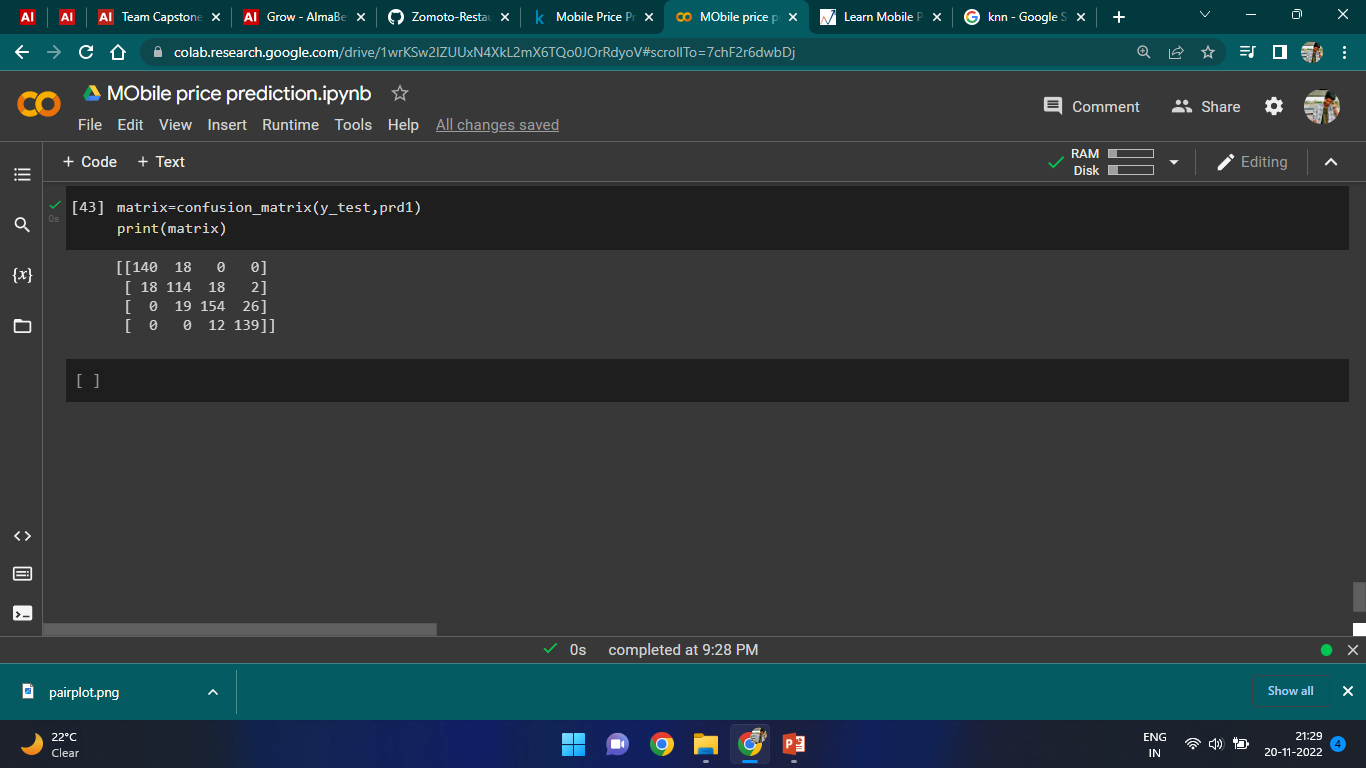
ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

1. **Result**

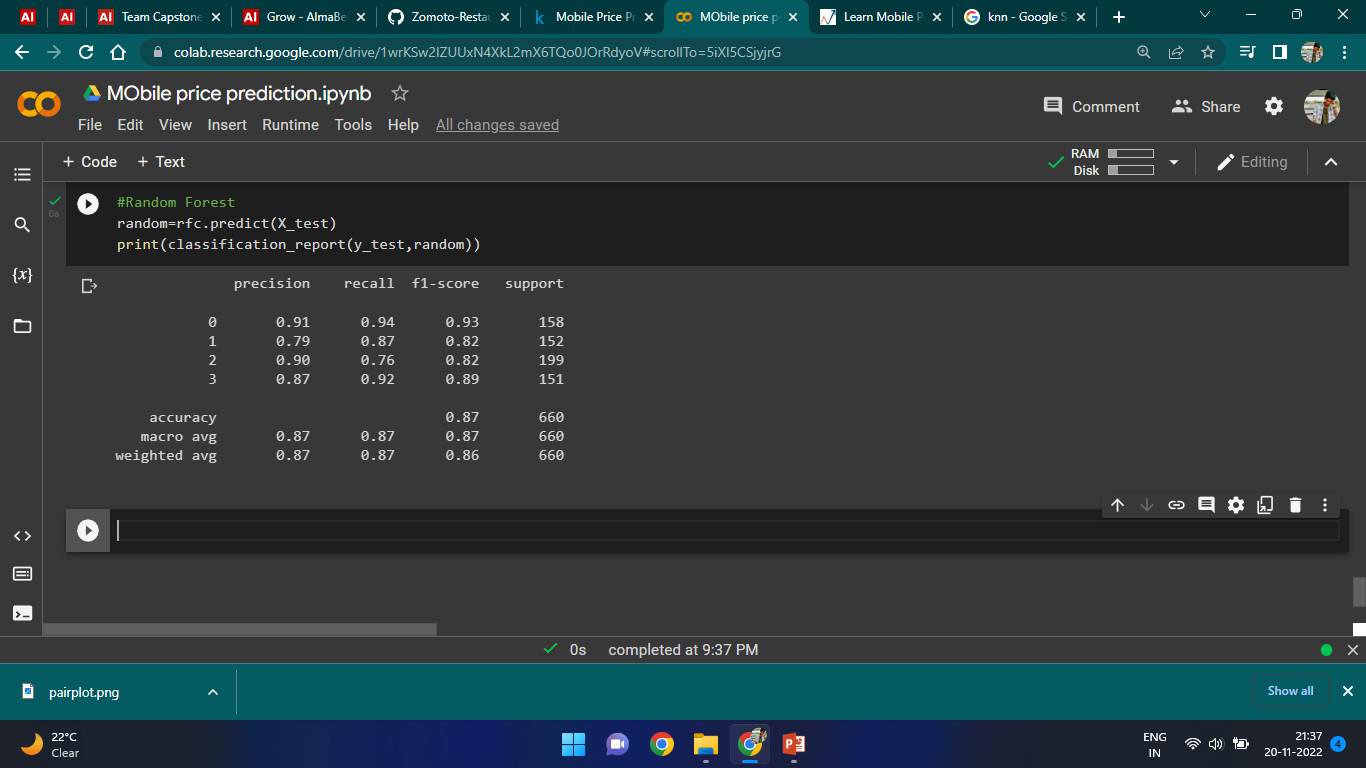
**KNN Classifier**

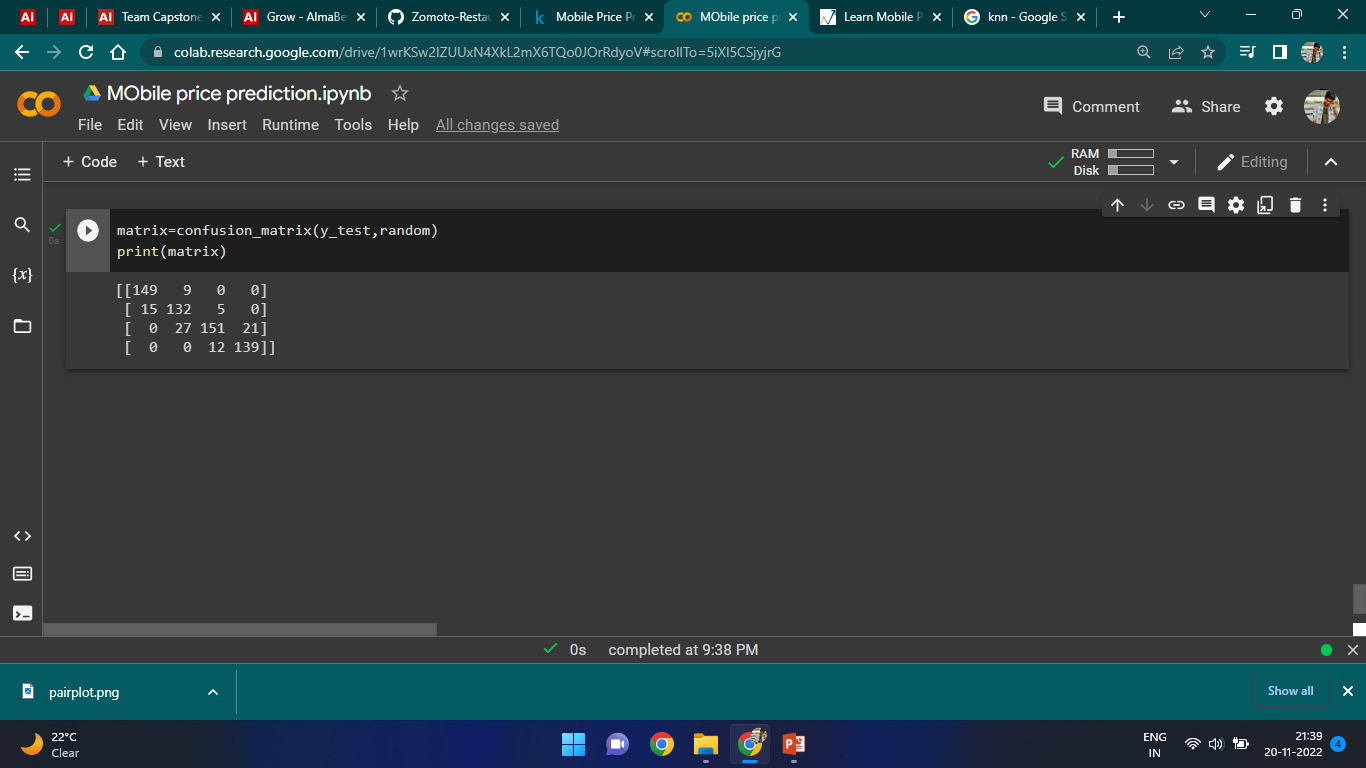


**Decision Tree**

****

**Random Forest**

****

****

1. **Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.

In all of these models our accuracy revolves in the range of 80 to 92%..

So the accuracy of our best model is 92% which can be said to be good for this large dataset.

1. **References-**
2. MachineLearningMastery
3. GeeksforGeeks
4. Analytics Vidhya