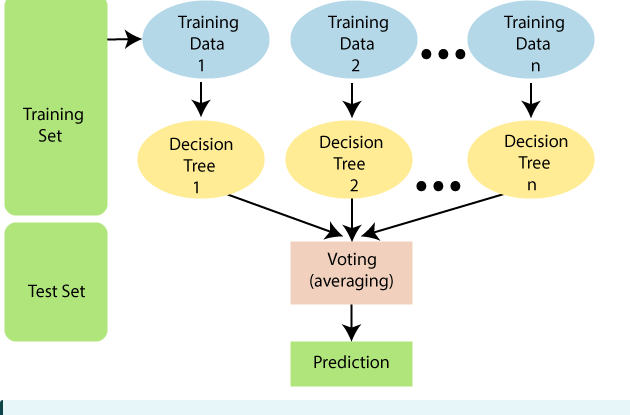
**Q. What is Random Forest ?**

**A. Random Forest** is a popular and powerful machine learning algorithm used for both **classification** and **regression** tasks. It is an ensemble learning method that combines multiple decision trees to make more accurate and robust predictions.

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

****

**Why use Random Forest?**

Below are some points that explain why we should use the Random Forest algorithm.

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

### ****How Random Forest Works:****

1. **Ensemble of Decision Trees**:

Random Forest builds multiple decision trees during training. Each tree is trained on a random subset of the training data and features (hence "random").

* + For **classification**, each tree makes a prediction (votes), and the majority vote among the trees becomes the final prediction.
  + For **regression**, the final prediction is the average of the predictions from all the trees.

1. **Bagging (Bootstrap Aggregating, eg—xgboost )**:
   * Random Forest uses a technique called **bagging**, where each tree is trained on a random subset of the data (sampled with replacement).
   * This ensures that the trees are different from each other, which helps in reducing variance and overfitting.
2. **Random Feature Selection**:
   * During the tree-building process, Random Forest randomly selects a subset of features at each split, which introduces more randomness and prevents trees from becoming too similar.

**The Working process can be explained in the below steps and diagram:-------------------------------------------------------------**

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data

point(Subsets).

**Step-3:** Choose the number N for decision trees that you want to

build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision

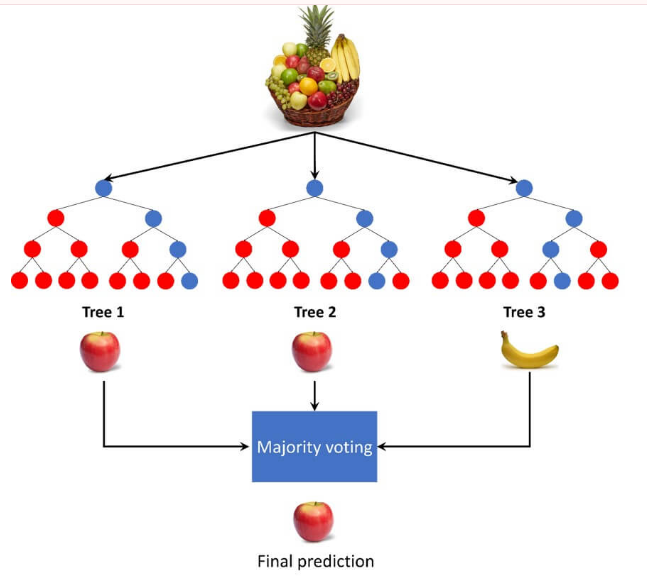
tree, and assign the new data points to the category that wins the

majority votes.

The working of the algorithm can be better understood by the below example:

**Example:**

 Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



**Applications of Random Forest**

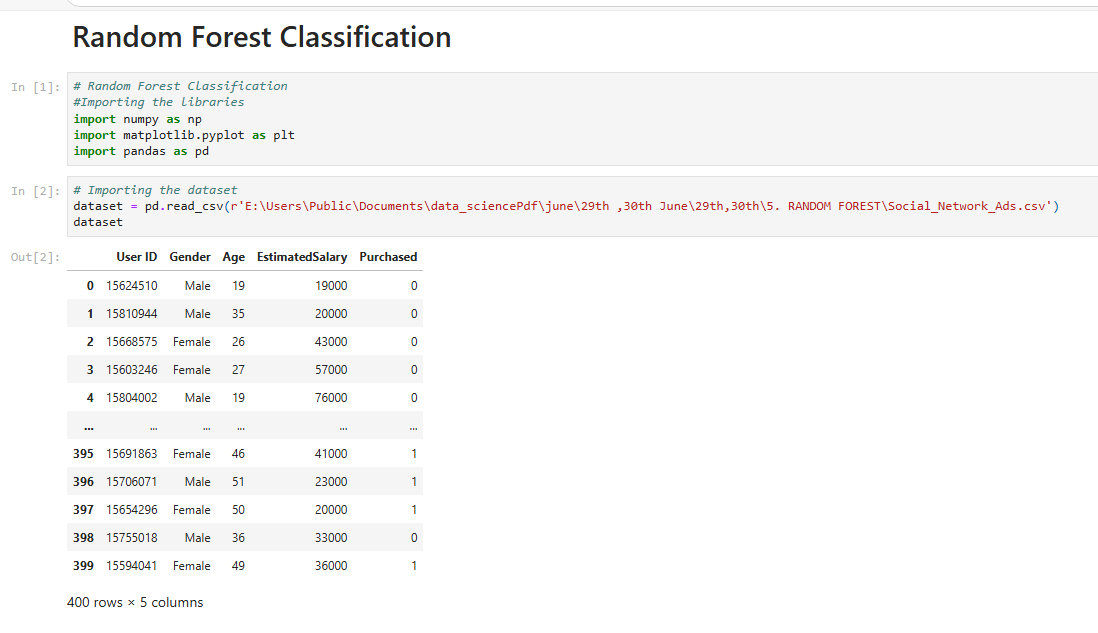
There are mainly four sectors where Random forest mostly used:

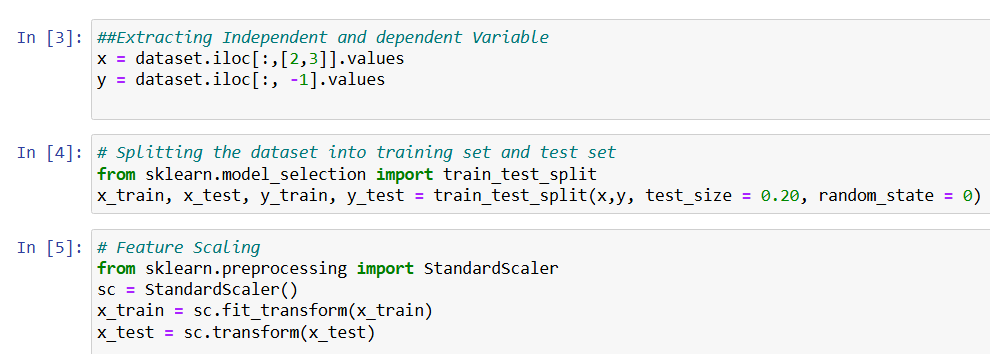
1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.(eg—House price prediction)
4. **Marketing:** Marketing trends can be identified using this algorithm.

**Implementation Steps are given below:**

* Data Pre-processing step
* Fitting the Random forest algorithm to the Training set
* Predicting the test result
* Test accuracy of the result (Creation of Confusion matrix)
* Visualizing the test set result.

**Data Preprocessing------------------------------------------------------------------**

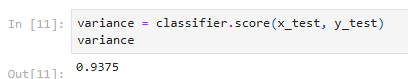




### 2. Fitting the Random Forest algorithm to the training set:

Now we will fit the Random forest algorithm to the training set. To fit it, we will import the**RandomForestClassifier**class from the **sklearn.ensemble** library. The code is given below:





* **n\_estimators=** The required number of trees in the Random Forest. The default value is 10. We can choose any number but need to take care of the overfitting issue.
* **criterion=** It is a function to analyze the accuracy of the split. Here we have taken "entropy" for the information gain.

**3.Predicting the Test Set result**

Since our model is fitted to the training set, so now we can predict the test result. For prediction, we will create a new prediction vector y\_pred. Below is the code for it:

#Predicting the test set result

y\_pred= classifier.predict(x\_test)

### 4. Creating the Confusion Matrix

Now we will create the confusion matrix to determine the correct and incorrect predictions. Below is the code for it:

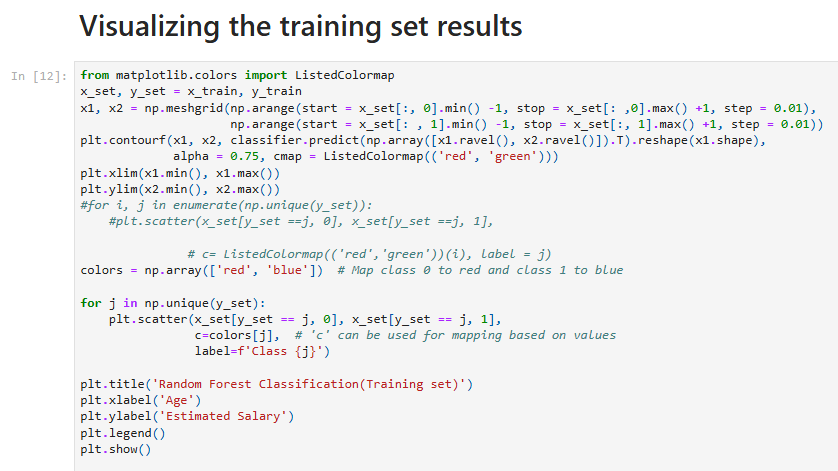
#Creating the Confusion matrix

from sklearn.metrics **import** confusion\_matrix

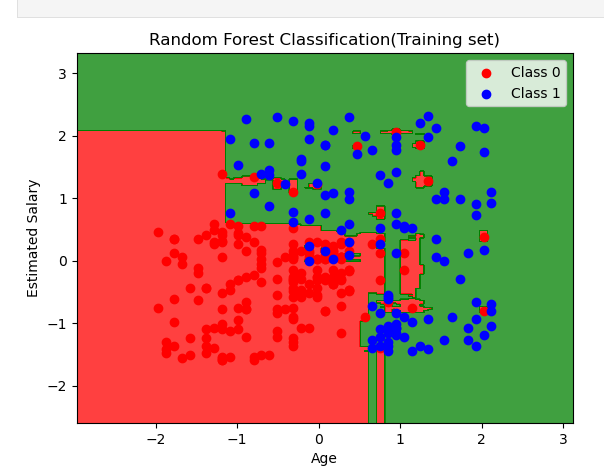
cm= confusion\_matrix(y\_test, y\_pred)

### 5. Visualizing the training Set result

Here we will visualize the training set result. To visualize the training set result we will plot a graph for the Random forest classifier. The classifier will predict yes or No for the users who have either Purchased or Not purchased the SUV car as we did in [Logistic Regression.](https://www.javatpoint.com/logistic-regression-in-machine-learning) Below is the code for it:



Output:------



The above image is the visualization result for the Random Forest classifier working with the training set result. It is very much similar to the Decision tree classifier. Each data point corresponds to each user of the user\_data, and the purple and green regions are the prediction regions. The purple region is classified for the users who did not purchase the SUV car, and the green region is for the users who purchased the SUV.

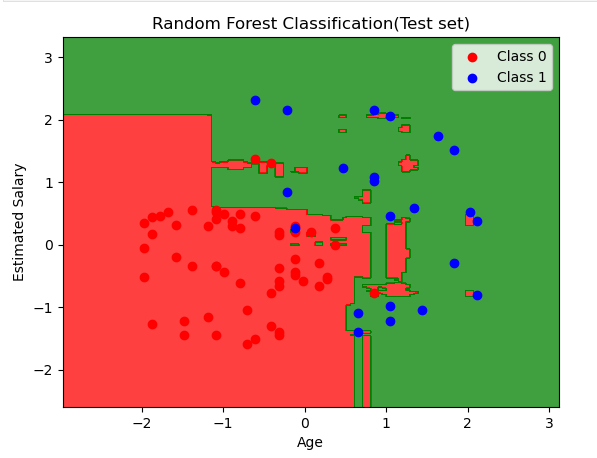
So, in the Random Forest classifier, we have taken 10 trees that have predicted Yes or NO for the Purchased variable. The classifier took the majority of the predictions and provided the result.

### Visualizing the test set result

**Now we will visualize the test set result. Below is the code for it:**

### Screenshot (262).png

Output: -------------------------------



The above image is the visualization result for the test set. We can check that there is a minimum number of incorrect predictions (8) without the Overfitting issue. We will get different results by changing the number of trees in the classifier.

### ****Advantages of Random Forest****:

1. **High Accuracy**: Due to the combination of multiple decision trees, Random Forest tends to have high accuracy and performs well on many types of data.
2. **Works Well with Missing Data**: It can handle datasets with missing values well.
3. **Robust to Overfitting**: It is less likely to overfit compared to individual decision trees.
4. **Handles Both Classification and Regression**: It can be applied to a wide variety of tasks.

### ****Disadvantages of Random Forest****:

1. **Slower Prediction**: Since it involves multiple trees, predictions can be slower, especially with large datasets and many trees.
2. **Less Interpretable**: Unlike a single decision tree, a random forest is harder to interpret because it aggregates many trees.

### ****When to Use Random Forest****:

* When you want high accuracy and have enough computing power.
* When you have a complex dataset with many features and possibly missing values.
* When you want to reduce overfitting but still capture complex patterns.
* When you need to estimate feature importance.

### ****Summary****:

* **Random Forest** is an ensemble learning algorithm that builds multiple decision trees and averages their results.
* It is robust, reduces overfitting, and works for both classification and regression tasks.
* It’s commonly used because of its ability to produce high-quality predictions while being resistant to overfitting and handling missing data effectively.