Agricultural Crop Yield Analysis

Akash Konda
Bindu Kambam
Ganesh Guram
Jabili Adepu

Outline

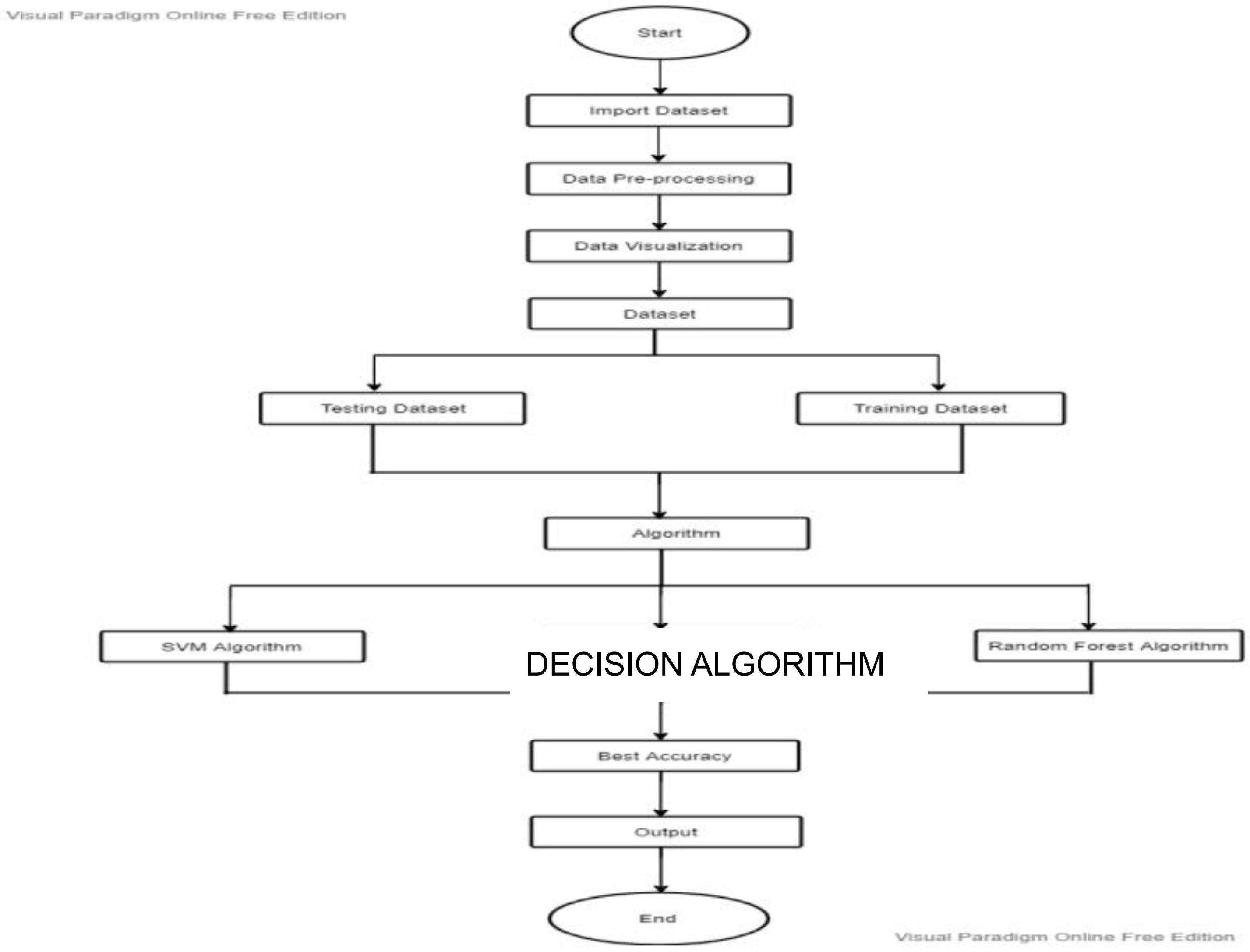
- System Design
- System Implementation
- Results
- Conclusion

System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.

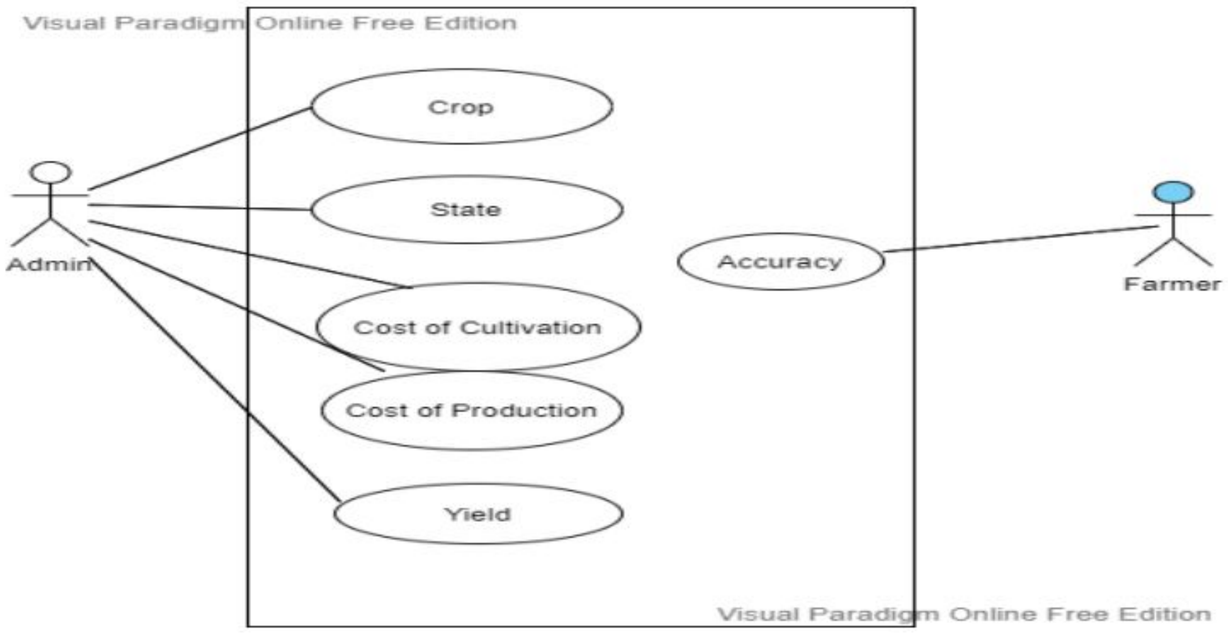
- Data Flow Diagaram
- UML Digaram

Data Flow Diagram



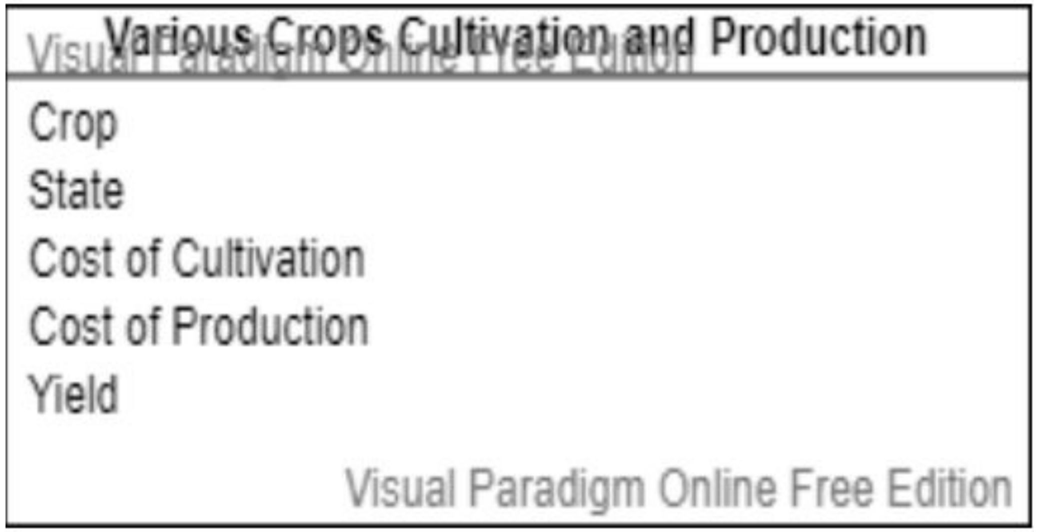
UML Diagrams

Use case Diagram

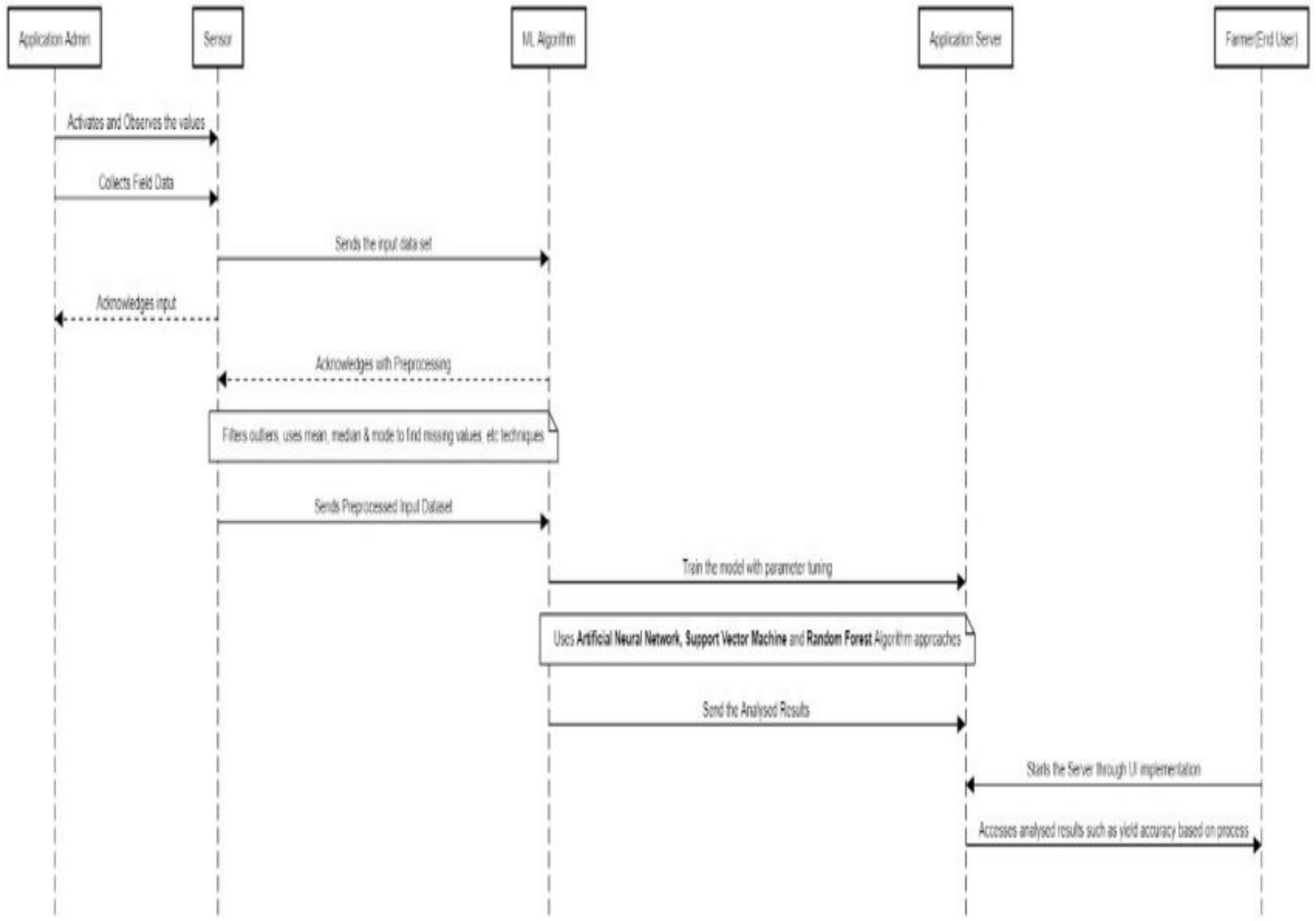


Visual Paradigm Online Free Edition
Crop
State
Accuracy
Cost of Cultivation
Cost of Production
Yield
Admin
Farmer
Visual Paradigm Online Free Edition

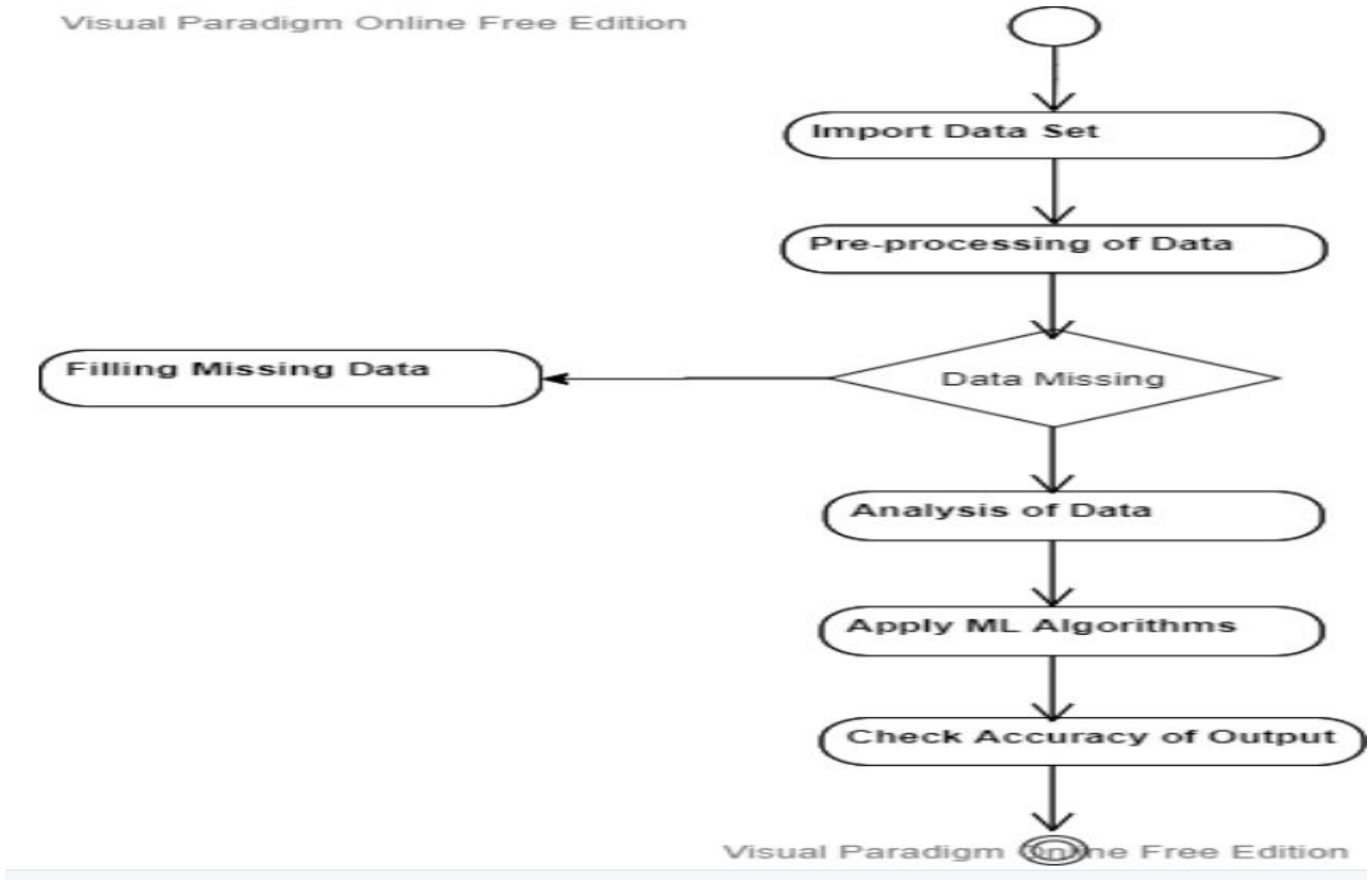
Class Diagram



Sequence Diagram


Application Admin
Sensor
ML Algorithm
Application Server
Farmer(End User)
Activates and Observes the values
Collects Field Data
Sends the input data set
Acknowledges input
Acknowledges with Preprocessing
Filters outliers, uses mean, median & mode to find missing values, etc techniques
Sends Preprocessed Input Dataset
Train the model with parameter tuning
Uses **Artificial Neural Network, Support Vector Machine** and **Random Forest** Algorithm approaches
Send the Analysed Results
Starts the Server through UI implementation
Accesses analysed results such as yield accuracy based on process

Activity Diagram



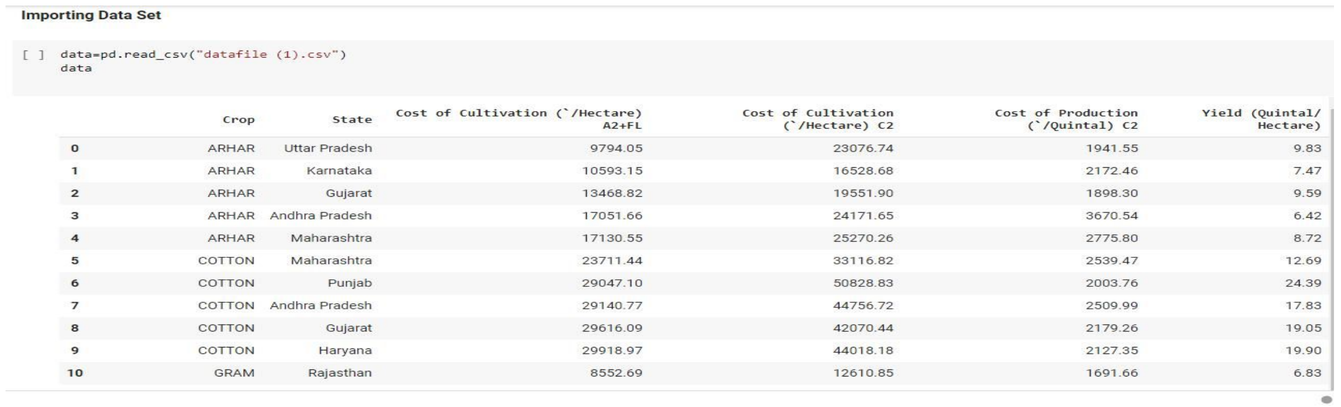
System Implementation

The implementation stage of any project is a true display of the defining moments that make a project a success or a failure.The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements.

- **DATA SET COLLECTION**
- **DATA PRE-PROCESSING**
- **DATA SPLITTING**
- **SUPPORT VECTOR MACHINE ALGORITHM**
- **DECISION TREE ALGORITHM**
- **RANDOM FOREST ALGORITHM**

DATA SET COLLECTION

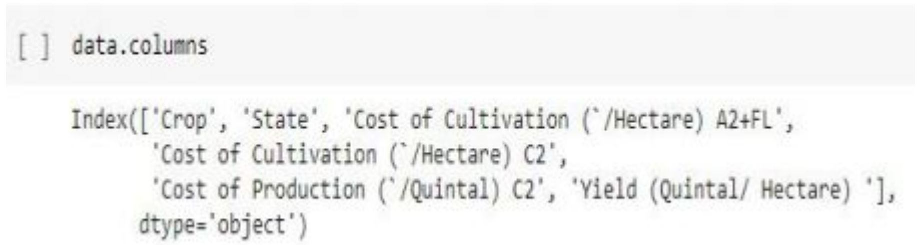
Dataset Collection plays a major role on the prediction model, if the availability of the dataset is less than the predicted model will not work fine because of the less data will lead to bias of the model. If the dataset is more in no with high records or tuples it will work well and the model will predict well, the dataset should contain less no of missing values and less no of strings and less no of dependent features.In our dataset it consists of 6900 records which indicates that the model developed is well and the prediction will be more accurate, and the loss margin/error margin will be less.



Importing Data Set

```
[ ]  data=pd.read_csv("datafile (1).csv")
     data
```

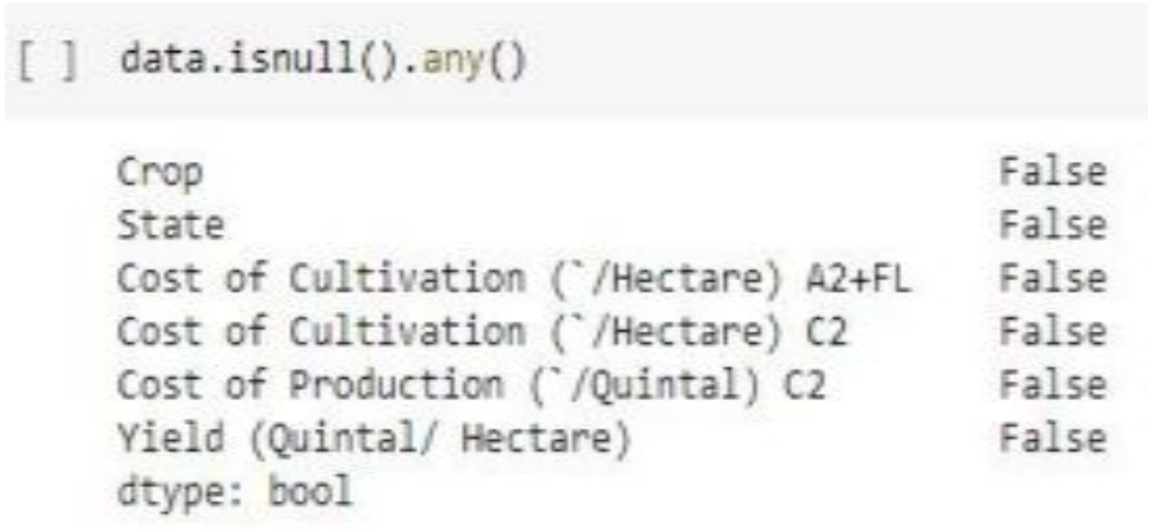
	Crop	State	Cost of Cultivation (`/Hectare) A2+FL	Cost of Cultivation (`/Hectare) C2	Cost of Production (`/Quintal) C2	Yield (Quintal/ Hectare)
0	ARHAR	Uttar Pradesh	9794.05	23076.74	1941.55	9.83
1	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
2	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59
3	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
4	ARHAR	Maharashtra	17130.55	25270.26	2775.80	8.72
5	COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69
6	COTTON	Punjab	29047.10	50828.83	2003.76	24.39
7	COTTON	Andhra Pradesh	29140.77	44756.72	2509.99	17.83
8	COTTON	Gujarat	29616.09	42070.44	2179.26	19.05
9	COTTON	Haryana	29918.97	44018.18	2127.35	19.90
10	GRAM	Rajasthan	8552.69	12610.85	1691.66	6.83



```
[ ]  data.columns

     Index(['Crop', 'State', 'Cost of Cultivation (`/Hectare) A2+FL',
            'Cost of Cultivation (`/Hectare) C2',
            'Cost of Production (`/Quintal) C2', 'Yield (Quintal/ Hectare) '],
           dtype='object')
```

DATA PRE PREPROCESSING

Data pre-processing is a process of creating suitable data by cleaning, integrating, transforming, and reducing raw data and extracting required data from it which is suitable for a machine learning model. Data pre-processing plays a major role while creating a machine learning model. In general, The raw data that's been collected from various sources consist of a lot of potentially incorrect data and instrument faulty errors by Data pre-processing those errors will be removed and the model can be developed well.

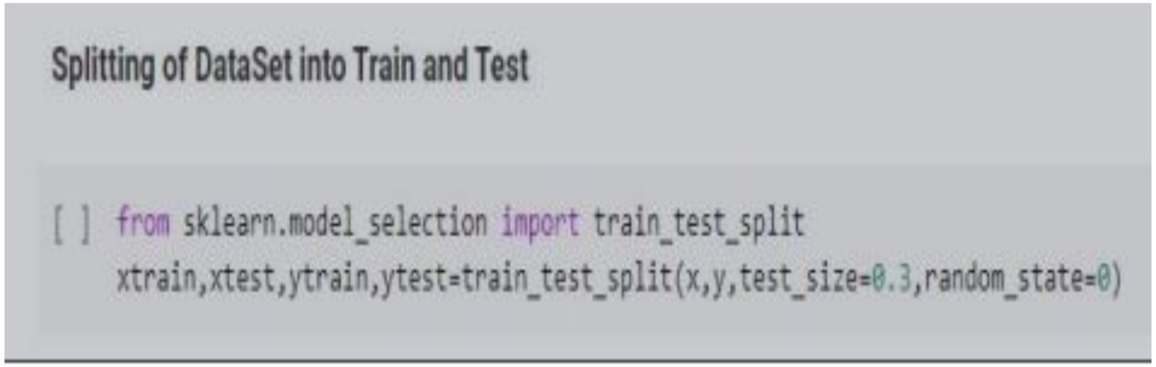


```
[ ]  data.isnull().any()

     Crop                                      False
     State                                     False
     Cost of Cultivation (`/Hectare) A2+FL     False
     Cost of Cultivation (`/Hectare) C2        False
     Cost of Production (`/Quintal) C2         False
     Yield (Quintal/ Hectare)                  False
     dtype: bool
```

DATA SPLITTING

Data is divided into two parts Training and Testing dataset because if we test our training set with a whole different testing set, we can't predict the model and correlations properly. So, we divide the same data set into training and testing data.

- Training data: a subset of data set to train a model.
- Testing data: a subset of data set to test the model.

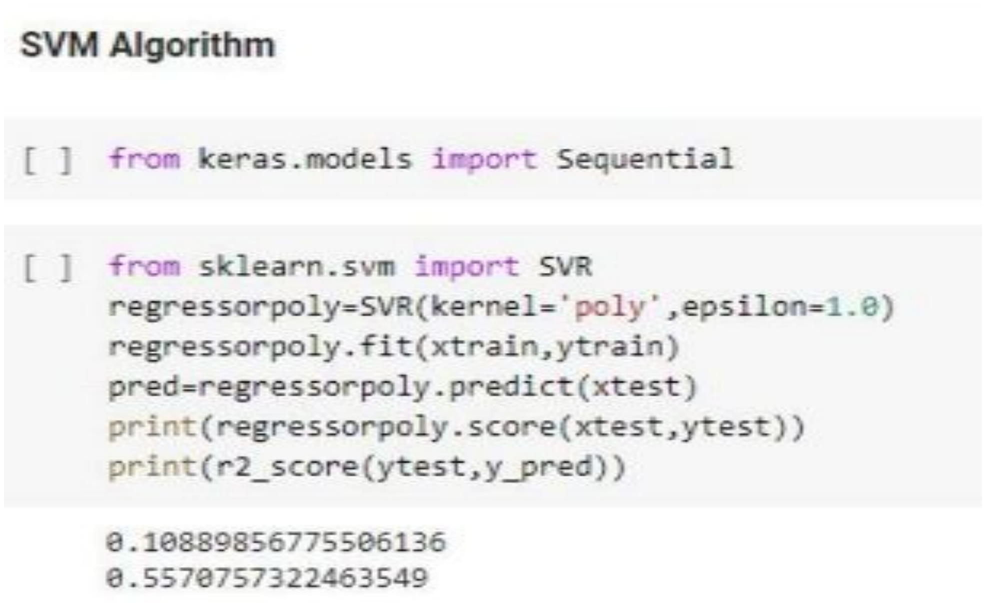


Splitting of DataSet into Train and Test

```
[ ] from sklearn.model_selection import train_test_split
    xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=0)
```

SUPPORT VECTOR MACHINE ALGORITHM

Support Vector Machines (SVMs) are Machine Learning models used for both classification and regression problems. An SVM model represents the training data as points in space so that examples falling in different categories are divided by a hyperplane that is as far as possible from the nearest data point.



SVM Algorithm

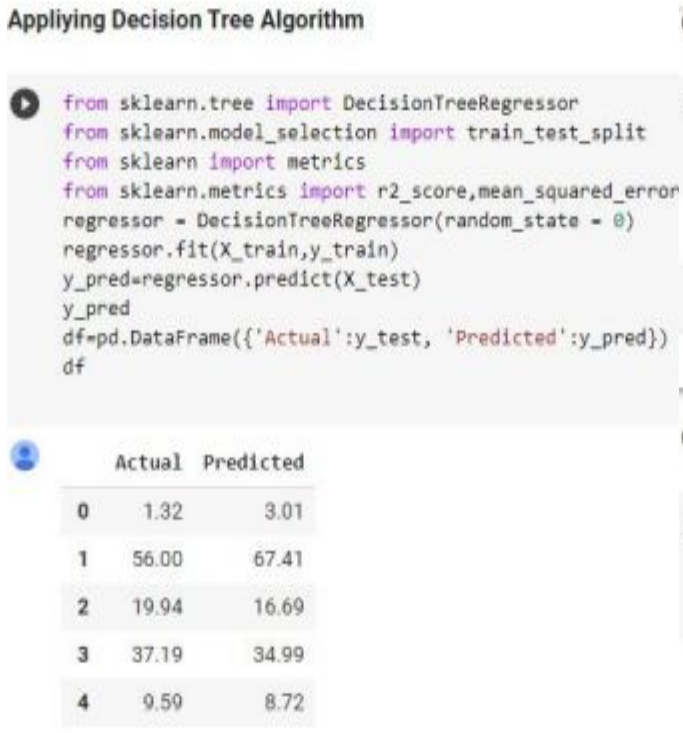
```
[ ]  from keras.models import Sequential
```

```
[ ]  from sklearn.svm import SVR
     regressorpoly=SVR(kernel='poly',epsilon=1.0)
     regressorpoly.fit(xtrain,ytrain)
     pred=regressorpoly.predict(xtest)
     print(regressorpoly.score(xtest,ytest))
     print(r2_score(ytest,y_pred))
```

```
0.10889856775506136
0.5570757322463549
```

DECISION TREE ALGORITHM

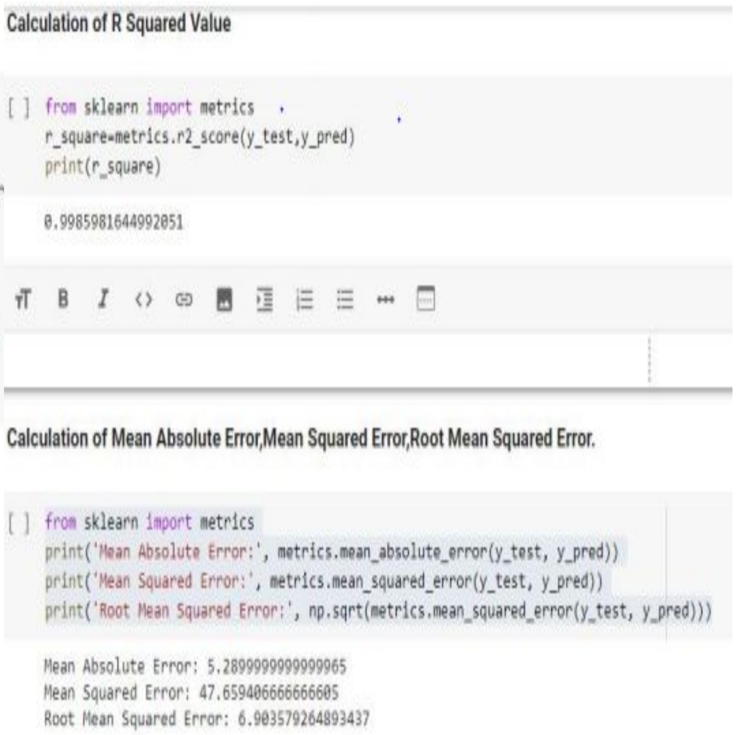
Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with **decision nodes** and **leaf nodes**. A decision node each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target.The topmost decision node in a tree which irresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.



Appliying Decision Tree Algorithm

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import r2_score,mean_squared_error
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train,y_train)
y_pred=regressor.predict(X_test)
y_pred
df=pd.DataFrame({'Actual':y_test, 'Predicted':y_pred})
df
```

	Actual	Predicted
0	1.32	3.01
1	56.00	67.41
2	19.94	16.69
3	37.19	34.99
4	9.59	8.72



Calculation of R Squared Value

```
from sklearn import metrics
r_square=metrics.r2_score(y_test,y_pred)
print(r_square)
```

```
0.9985981644992051
```

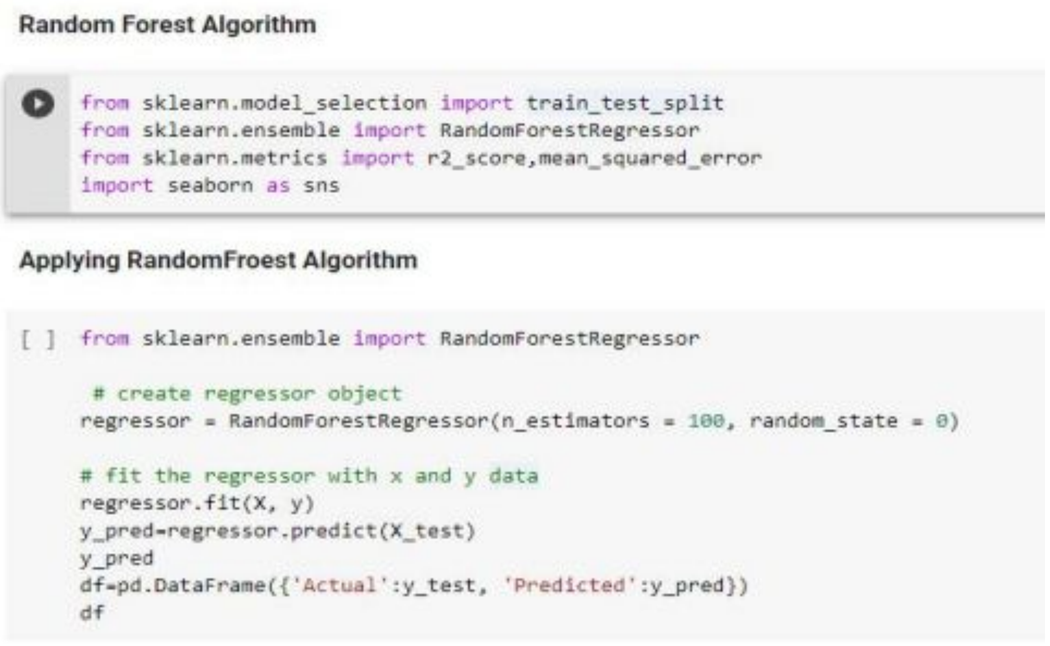
Calculation of Mean Absolute Error,Mean Squared Error,Root Mean Squared Error.

```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 5.2899999999999965
Mean Squared Error: 47.659406666666605
Root Mean Squared Error: 6.903579264893437
```

RANDOM FOREST ALGORITHM

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms



Random Forest Algorithm

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score,mean_squared_error
import seaborn as sns
```

Applying RandomFroest Algorithm

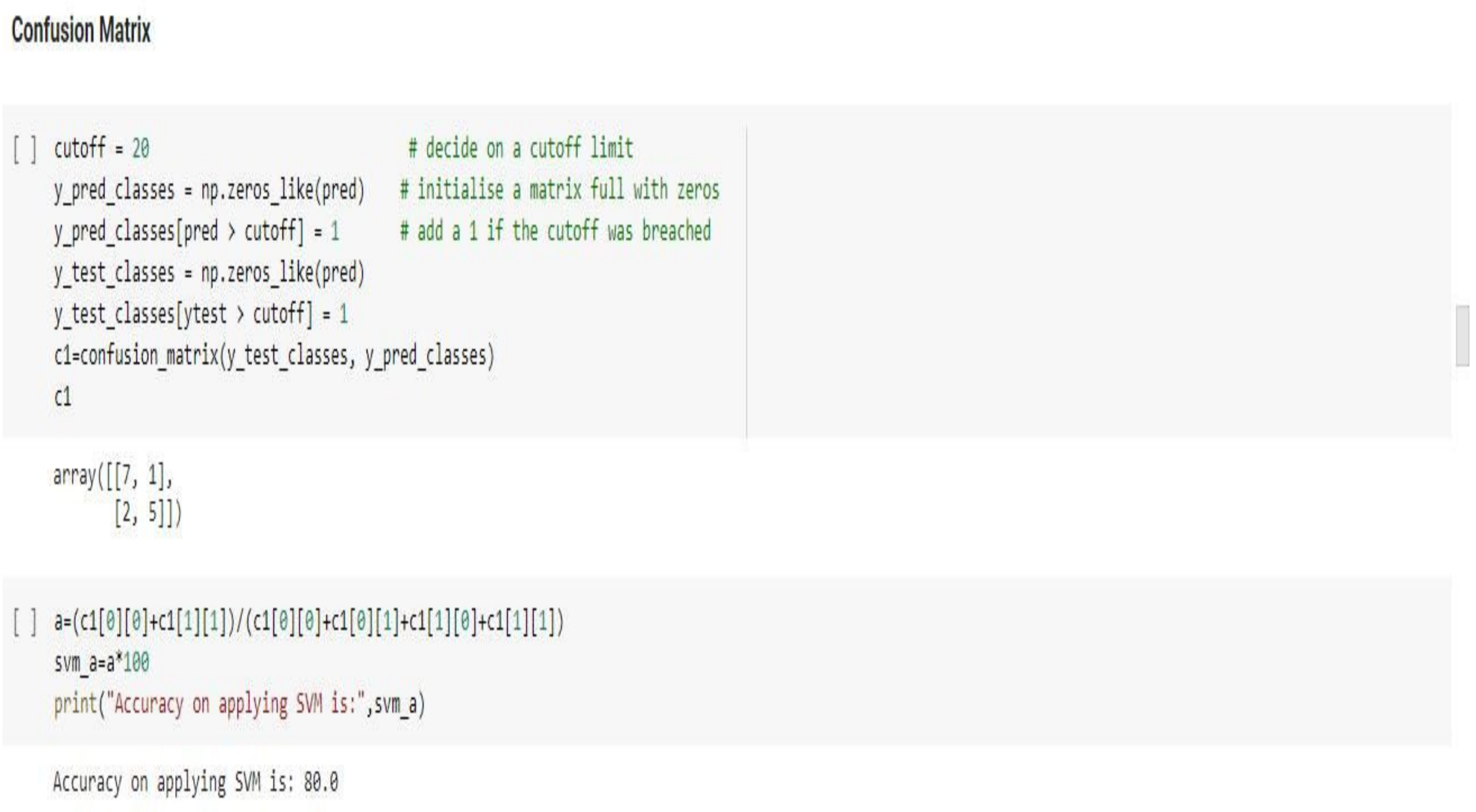
```
[ ] from sklearn.ensemble import RandomForestRegressor

     # create regressor object
    regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)

    # fit the regressor with x and y data
    regressor.fit(X, y)
    y_pred=regressor.predict(X_test)
    y_pred
    df=pd.DataFrame({'Actual':y_test, 'Predicted':y_pred})
    df
```

Result

The model is built successfully using SVM algorithm, Decision Tree algorithm and Random Forest algorithm.

- Applying Confusion Matrix on SVM Algorithm
- Applying Confusion Matrix on Decision Tree Algorithm
- Applying Confusion Matrix on Random Forest Algorithm
- Comparing Accuracy of three different algorithm

APPLYING CONFUSION MATRIX ON SVM



Confusion Matrix

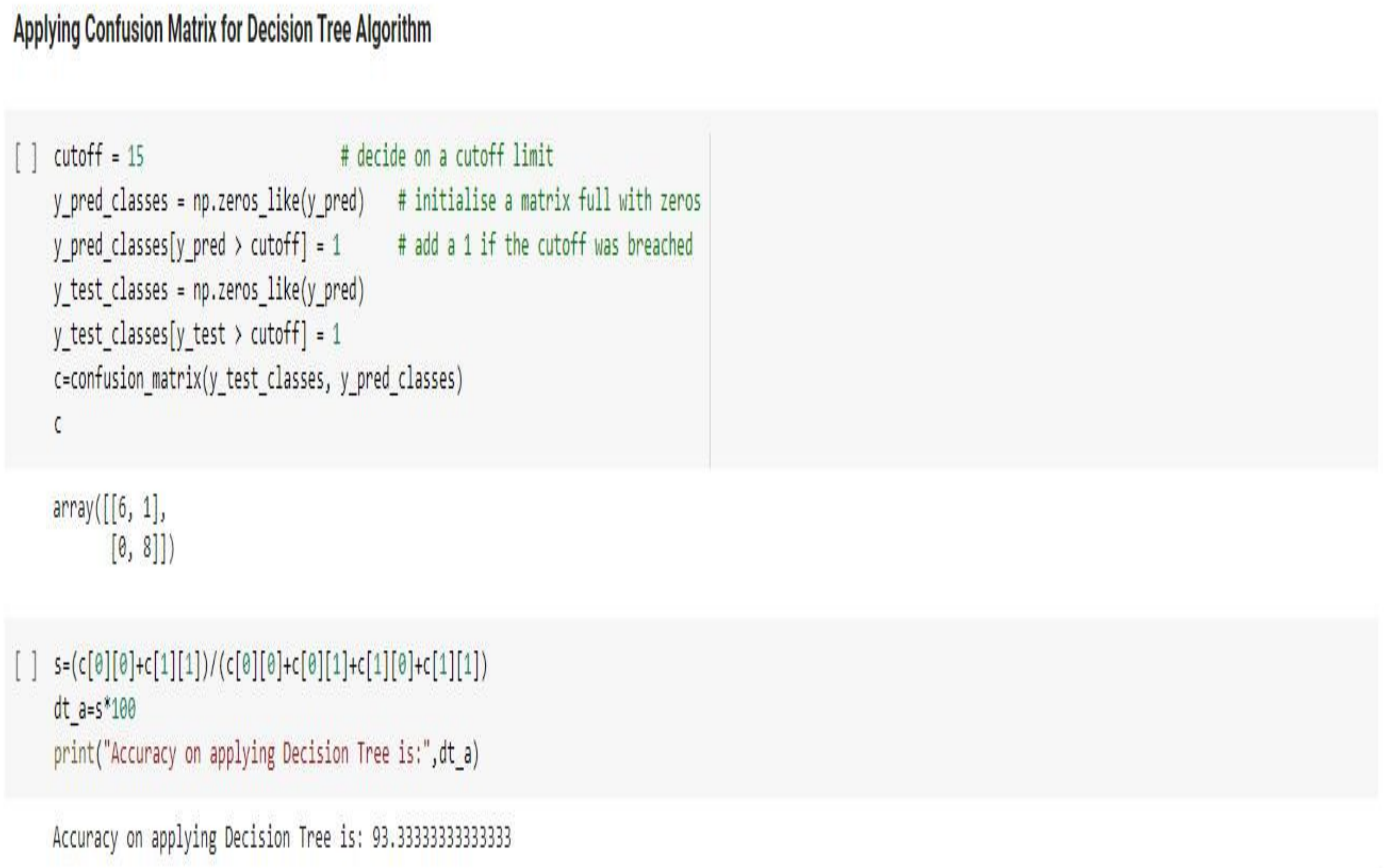
```
[ ] cutoff = 20                           # decide on a cutoff limit
    y_pred_classes = np.zeros_like(pred)  # initialise a matrix full with zeros
    y_pred_classes[pred > cutoff] = 1     # add a 1 if the cutoff was breached
    y_test_classes = np.zeros_like(pred)
    y_test_classes[ytest > cutoff] = 1
    c1=confusion_matrix(y_test_classes, y_pred_classes)
    c1
```

```
array([[7, 1],
       [2, 5]])
```

```
[ ] a=(c1[0][0]+c1[1][1])/(c1[0][0]+c1[0][1]+c1[1][0]+c1[1][1])
    svm_a=a*100
    print("Accuracy on applying SVM is:",svm_a)
```

```
Accuracy on applying SVM is: 80.0
```

APPLYING CONFUSION MATRIX ON DECISION TREE



Applying Confusion Matrix for Decision Tree Algorithm

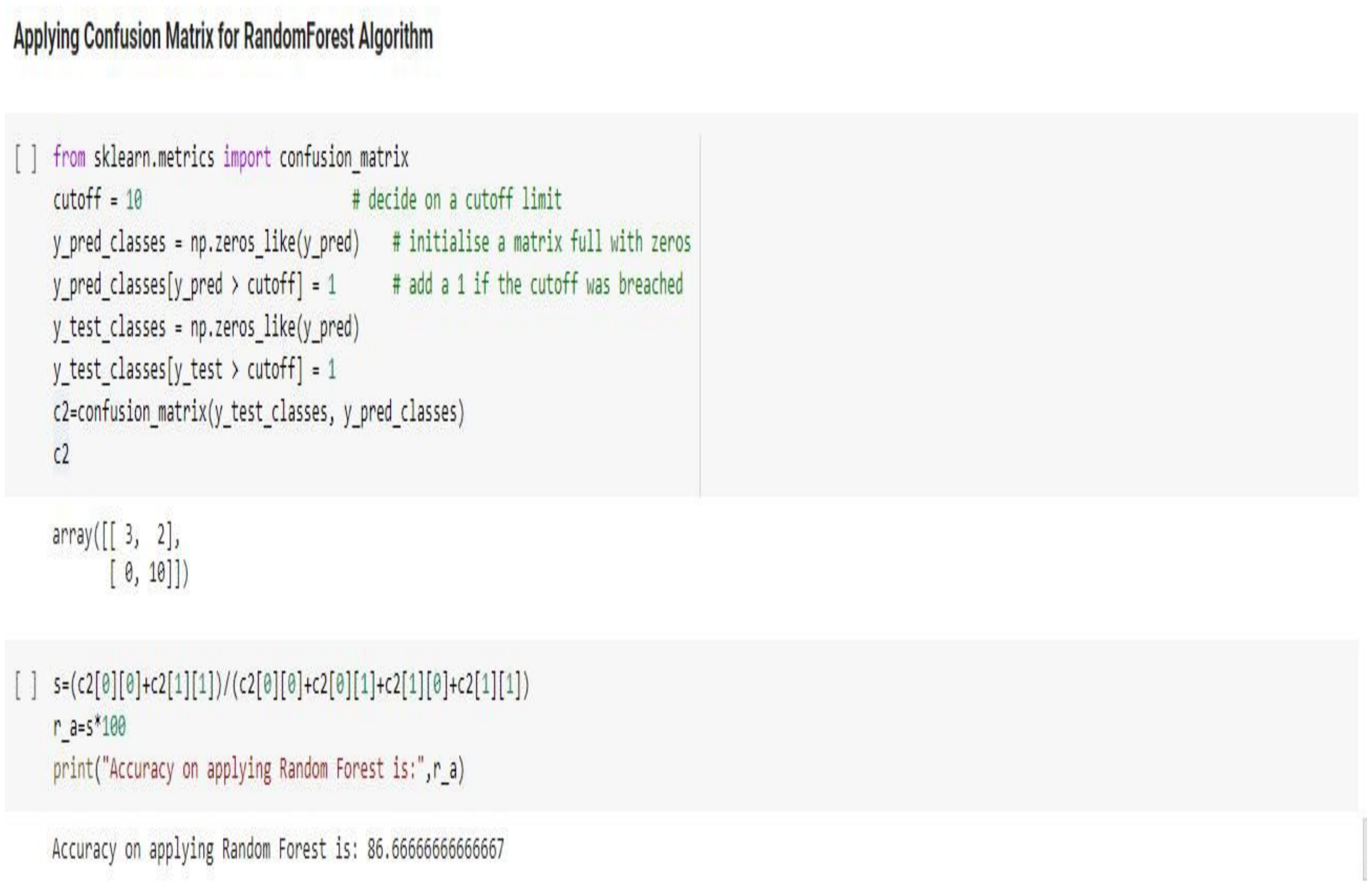
```
[ ] cutoff = 15                          # decide on a cutoff limit
    y_pred_classes = np.zeros_like(y_pred)    # initialise a matrix full with zeros
    y_pred_classes[y_pred > cutoff] = 1       # add a 1 if the cutoff was breached
    y_test_classes = np.zeros_like(y_pred)
    y_test_classes[y_test > cutoff] = 1
    c=confusion_matrix(y_test_classes, y_pred_classes)
    c
```

```
array([[6, 1],
       [0, 8]])
```

```
[ ] s=(c[0][0]+c[1][1])/(c[0][0]+c[0][1]+c[1][0]+c[1][1])
    dt_a=s*100
    print("Accuracy on applying Decision Tree is:",dt_a)
```

```
Accuracy on applying Decision Tree is: 93.33333333333333
```

APPLYING CONFUSION MATRIX ON RANDOM FOREST



Applying Confusion Matrix for RandomForest Algorithm

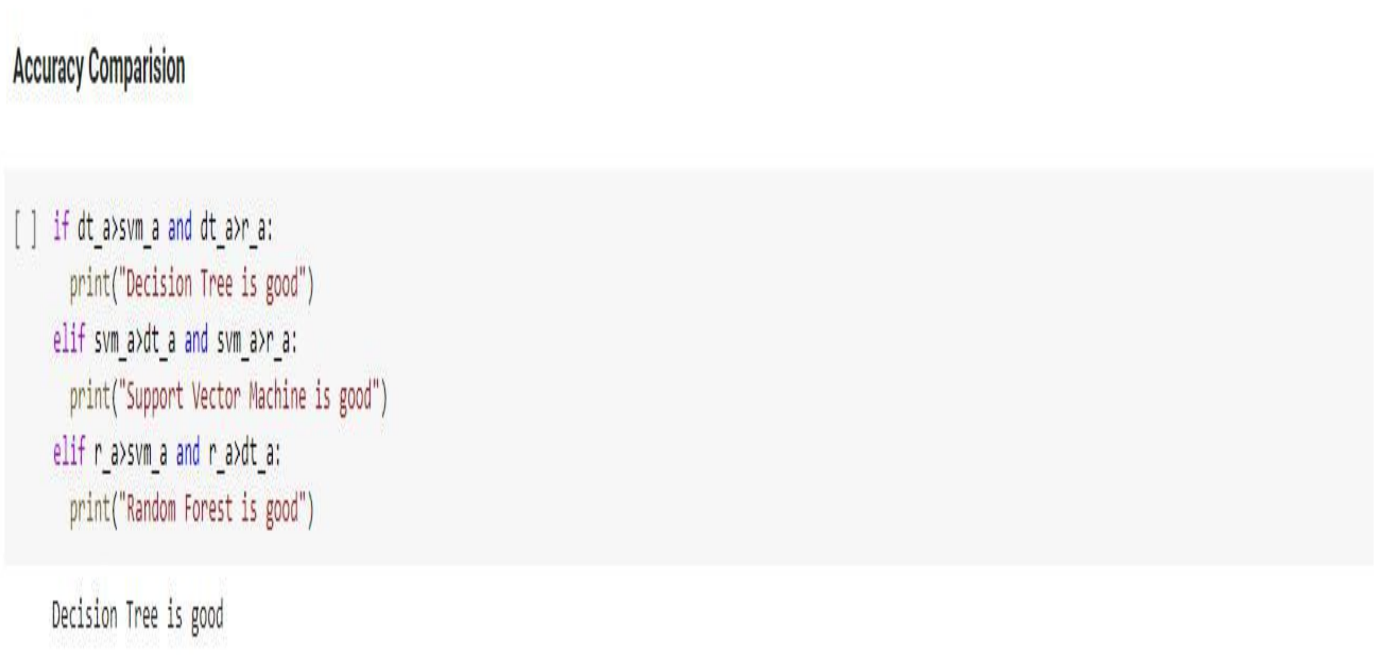
```
[ ] from sklearn.metrics import confusion_matrix
    cutoff = 10                          # decide on a cutoff limit
    y_pred_classes = np.zeros_like(y_pred)    # initialise a matrix full with zeros
    y_pred_classes[y_pred > cutoff] = 1       # add a 1 if the cutoff was breached
    y_test_classes = np.zeros_like(y_pred)
    y_test_classes[y_test > cutoff] = 1
    c2=confusion_matrix(y_test_classes, y_pred_classes)
    c2
```

```
    array([[ 3,  2],
           [ 0, 10]])
```

```
[ ] s=(c2[0][0]+c2[1][1])/(c2[0][0]+c2[0][1]+c2[1][0]+c2[1][1])
    r_a=s*100
    print("Accuracy on applying Random Forest is:",r_a)
```

```
    Accuracy on applying Random Forest is: 86.66666666666667
```

COMPARING ACCURACY



Accuracy Comparision

```
[ ] if dt_a>svm_a and dt_a>r_a:
      print("Decision Tree is good")
    elif svm_a>dt_a and svm_a>r_a:
      print("Support Vector Machine is good")
    elif r_a>svm_a and r_a>dt_a:
      print("Random Forest is good")
```

```
Decision Tree is good
```

CONCLUSION

Our project focuses on the prediction of crop and calculation of its yield with the help of machine learning techniques. Several machine learning methodologies used for the calculation of accuracy. K-Means clustering was used for the crop prediction which is the existing methodology. The proposed technique helps farmers in decision making of which crop to cultivate in the field which included Support Vector Machine (SVM) and Decision Tree algorithms. Implemented a system to crop prediction from the collection of past data. This work is employed to search out the gain knowledge about the crop that can be deployed to make an efficient and useful harvesting.The accurate prediction of different specified crops will help farmers. This improves our Indian economy by maximizing the yield rate of crop production.

Thank you