ROHAN NYATI

500075940

R177219148

BATCH – 5 (Ai & Ml)

Experiment -4

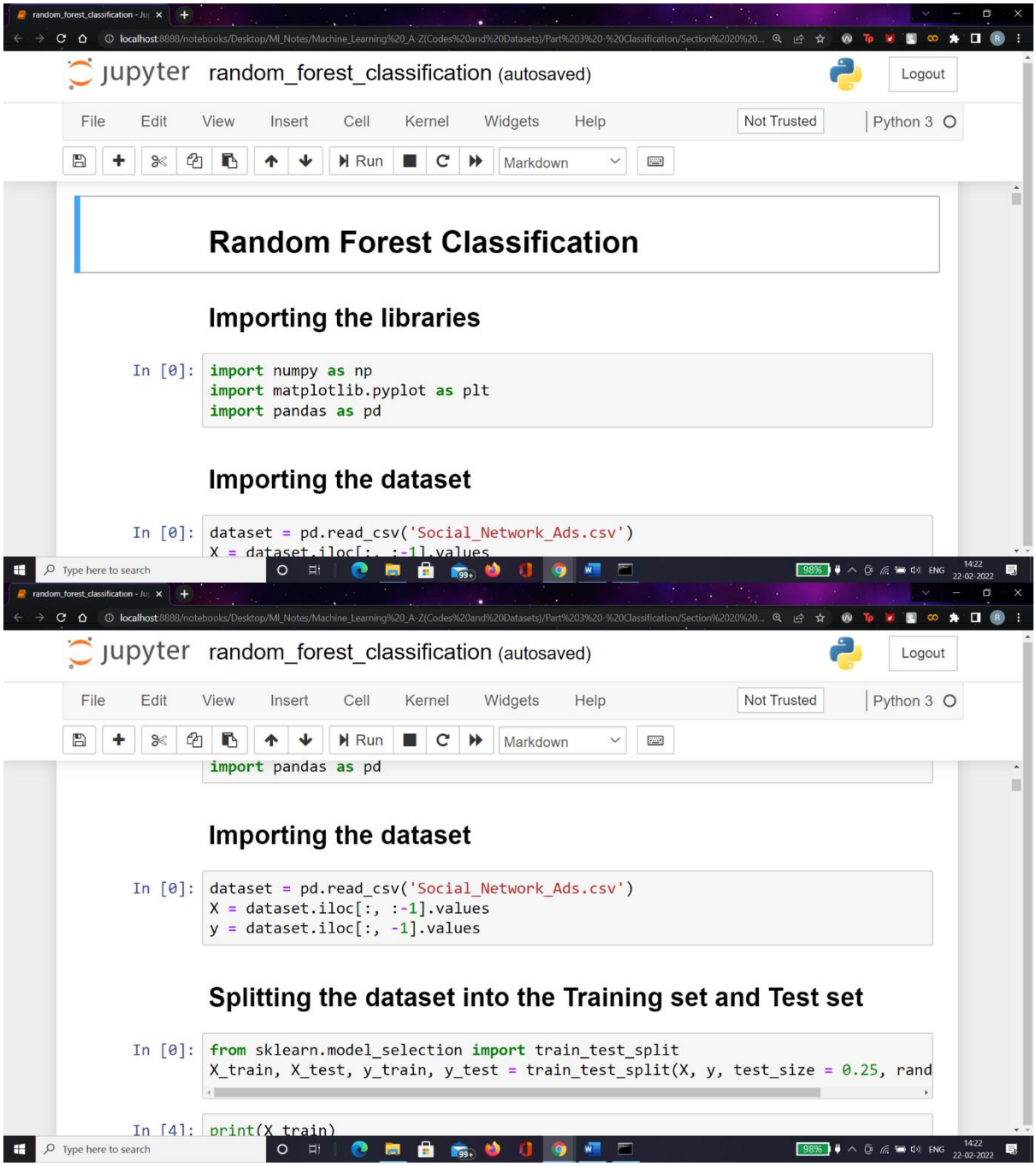
What is Random forest Algorithm? What are the advantages of Random Forest Classification algorithm over decision tree algorithm. Implement Random Forest Classification using suitable dataset. Enumerate its advantages and disadvantages.

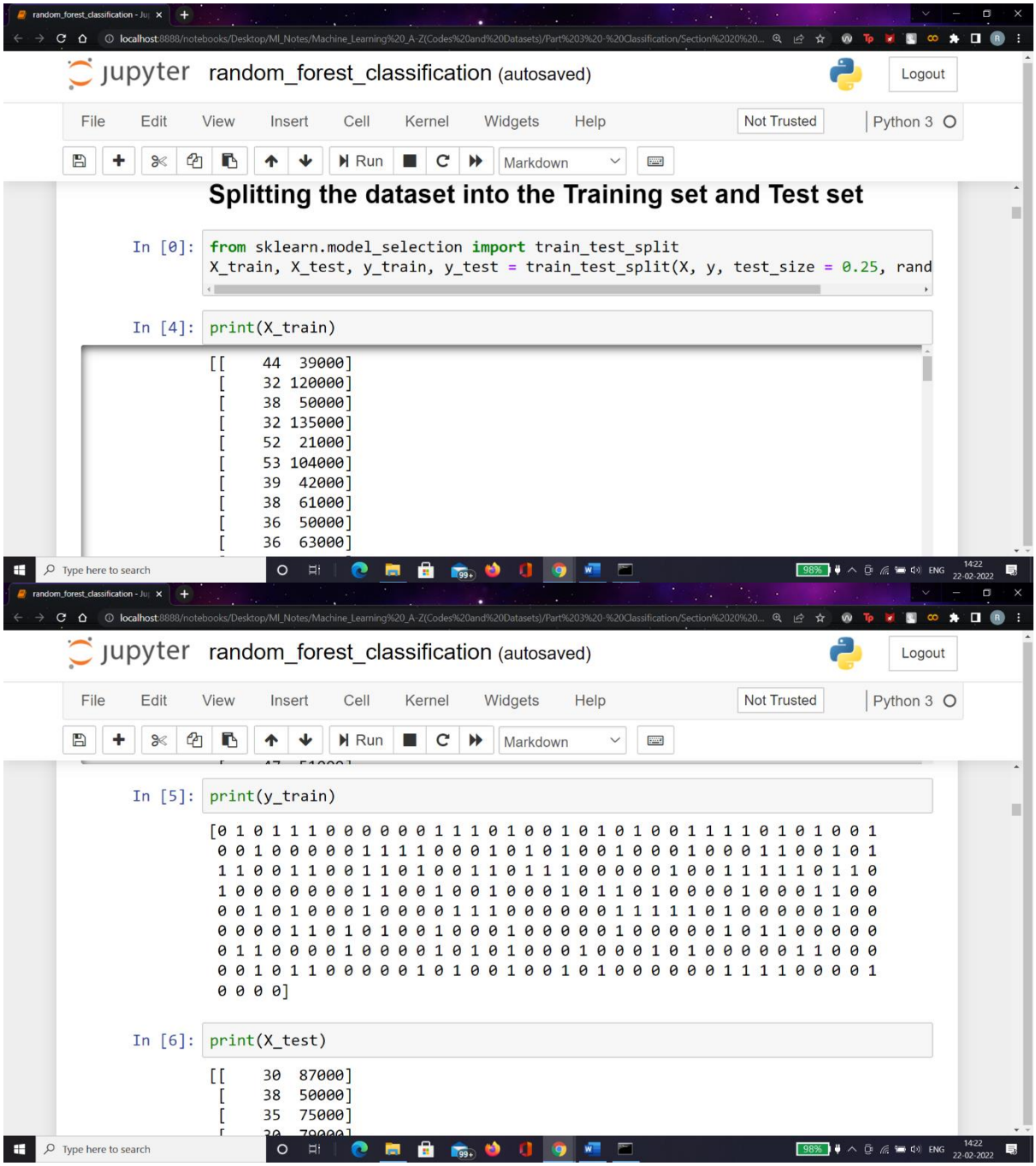
Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

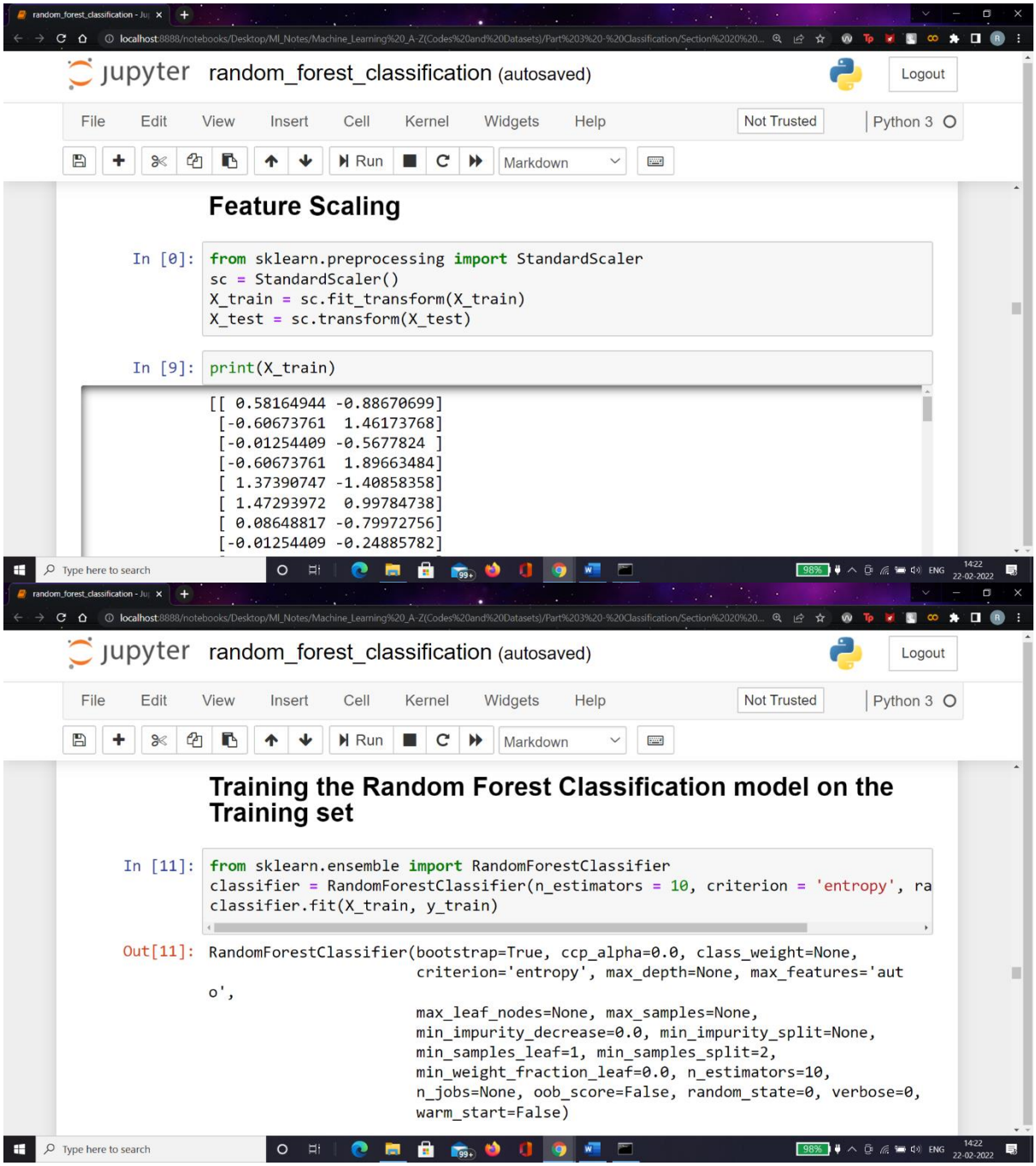
As the name suggests, *"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."* Instead of relying on one decision tree, the random forest takes the prediction from each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

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

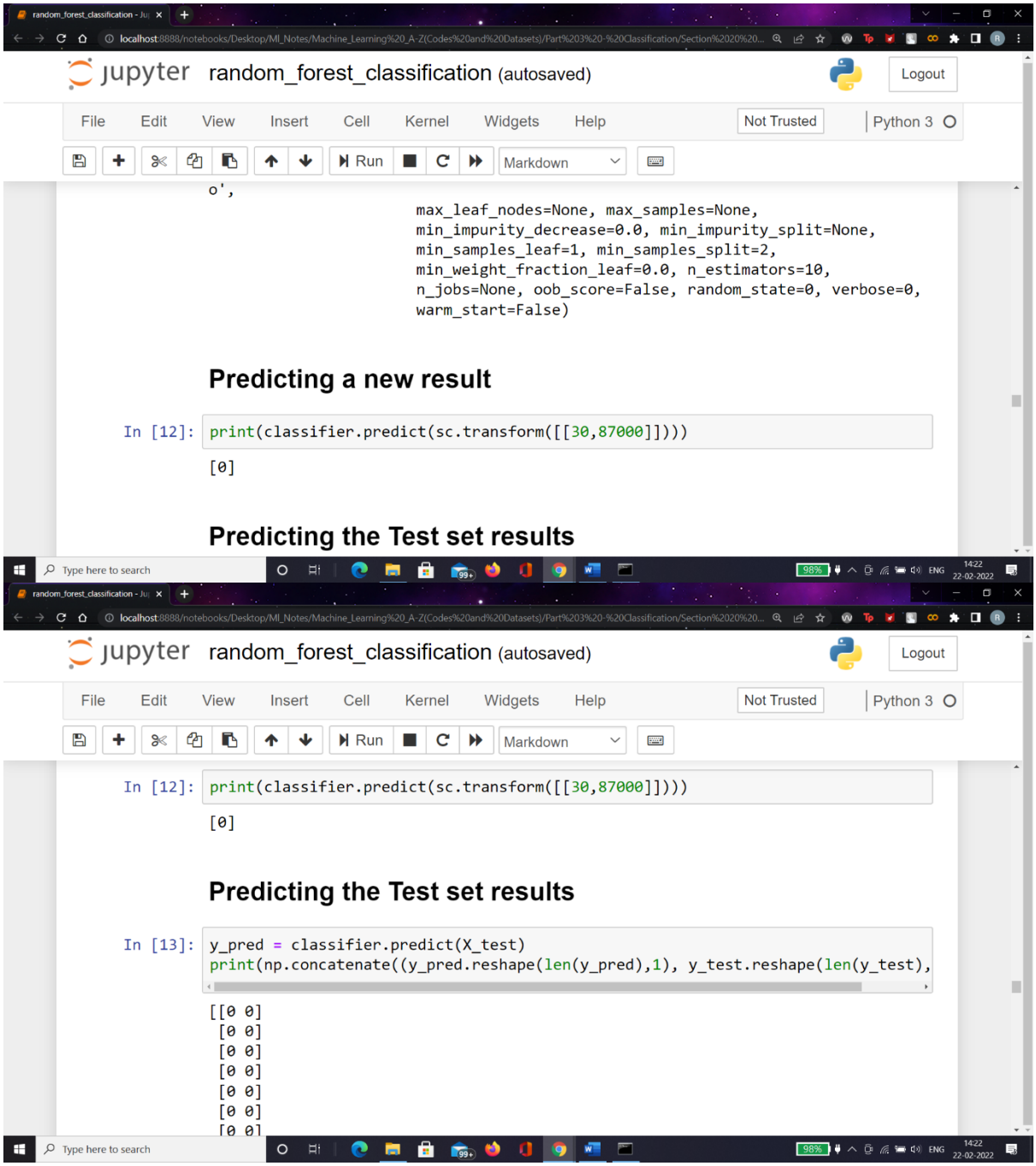
Random Forest is suitable for situations when we have a large dataset, and interpretability is not a major concern. Decision trees are much easier to interpret and understand. Since a random forest combines multiple decision trees, it becomes more difficult to interpret.


random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Random Forest Classification
Importing the libraries
In [0]:
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
Importing the dataset
In [0]:
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, :-1].values
Type here to search
98%
ENG
14:22
22-02-2022
random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
import pandas as pd
Importing the dataset
In [0]:
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
Splitting the dataset into the Training set and Test set
In [0]:
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, rand
In [4]:
print(X_train)
Type here to search
98%
ENG
14:22
22-02-2022

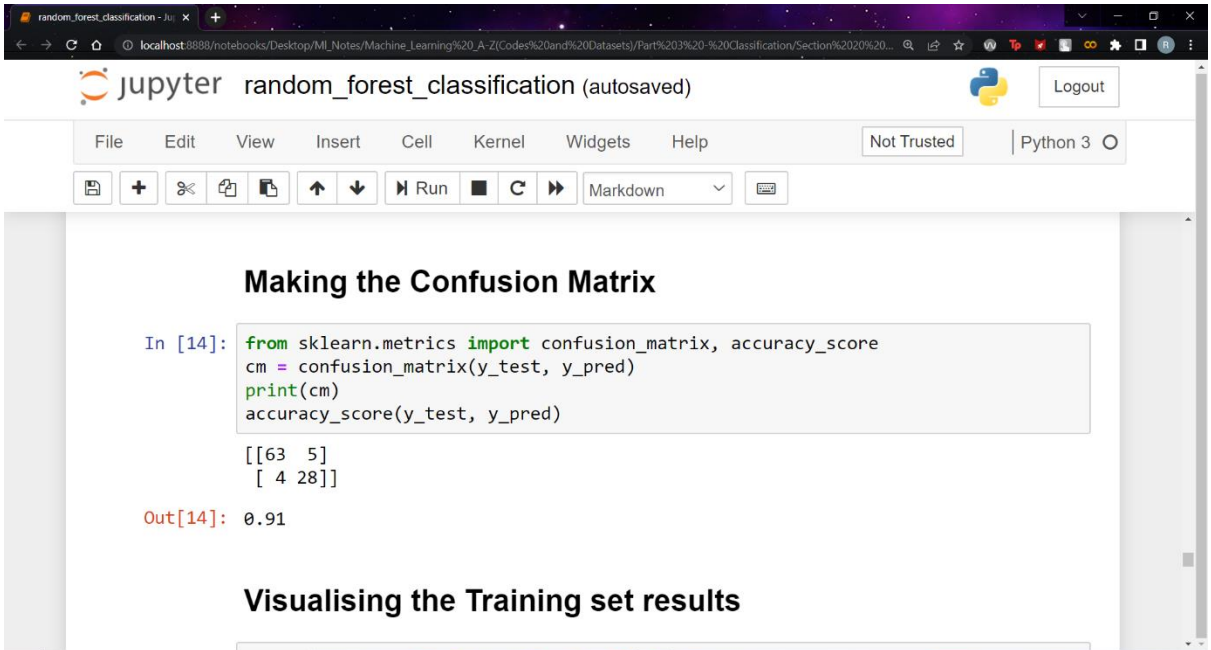

random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Splitting the dataset into the Training set and Test set
In [0]:
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, rand
In [4]:
print(X_train)
[[44 39000]
[32 120000]
[38 50000]
[32 135000]
[52 21000]
[53 104000]
[39 42000]
[38 61000]
[36 50000]
[36 63000]
Type here to search
98%
ENG
14:22
22-02-2022
random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
In [5]:
print(y_train)
[0 1 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 0 0 1
0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 0 1
1 1 0 0 1 1 0 0 1 1 0 1 0 0 1 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 1 0
1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0 0
0 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 1 0 0
0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0
0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 0 0
0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1
0 0 0 0]
In [6]:
print(X_test)
[[30 87000]
[38 50000]
[35 75000]
Type here to search
98%
ENG
14:22
22-02-2022

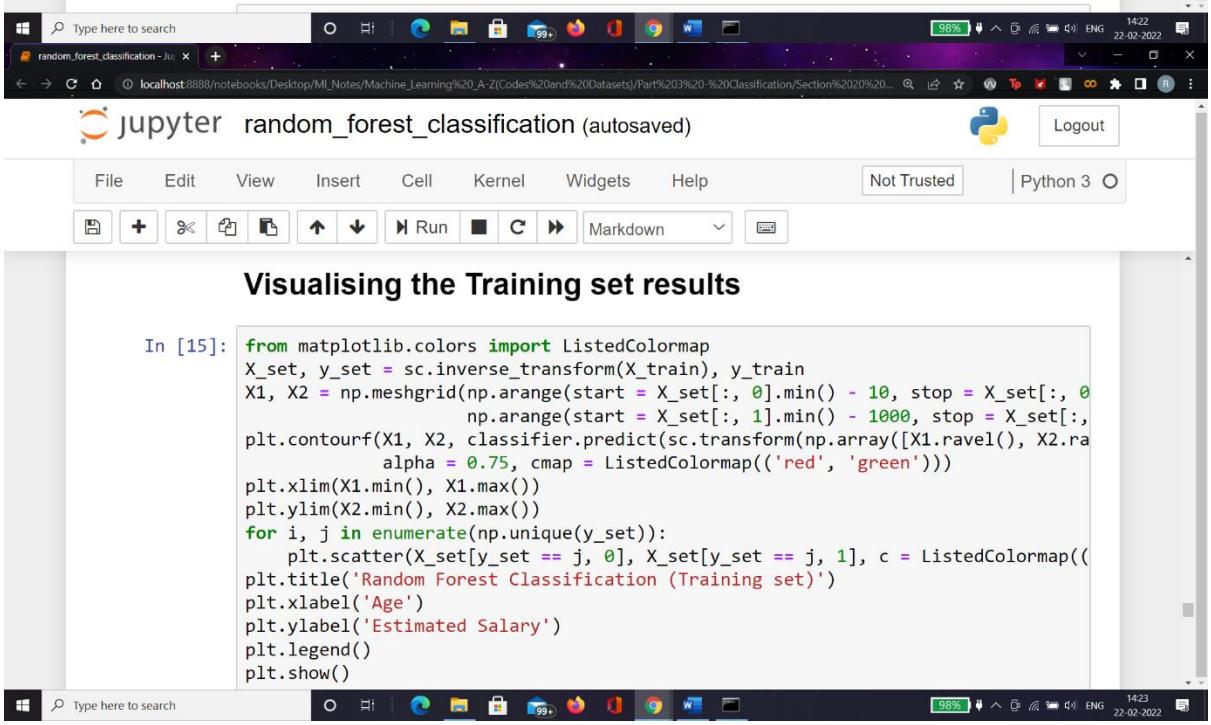

random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Feature Scaling
In [0]:
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
In [9]:
print(X_train)
[[0.58164944 -0.88670699]
[-0.60673761 1.46173768]
[-0.01254409 -0.5677824]
[-0.60673761 1.89663484]
[1.37390747 -1.40858358]
[1.47293972 0.99784738]
[0.08648817 -0.79972756]
[-0.01254409 -0.24885782]
Type here to search
98%
ENG
14:22
22-02-2022

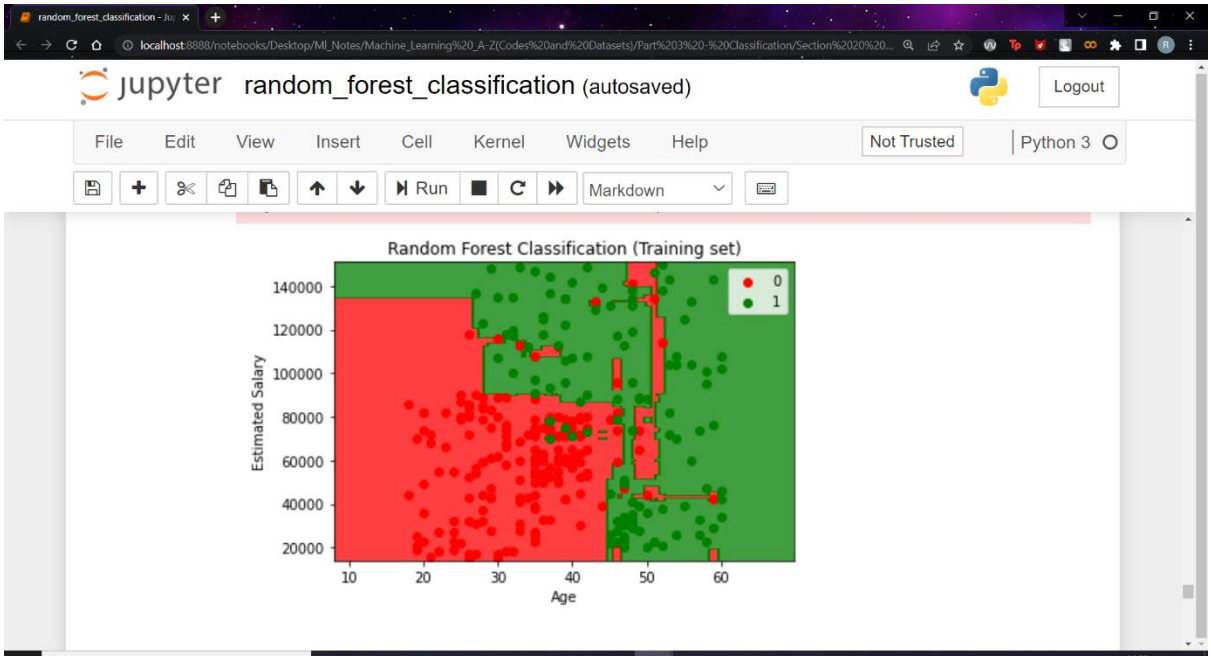
random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Training the Random Forest Classification model on the Training set
In [11]:
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', ra
classifier.fit(X_train, y_train)
Out[11]:
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
criterion='entropy', max_depth=None, max_features='aut
o',
max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=10,
n_jobs=None, oob_score=False, random_state=0, verbose=0,
warm_start=False)
Type here to search
98%
ENG
14:22
22-02-2022

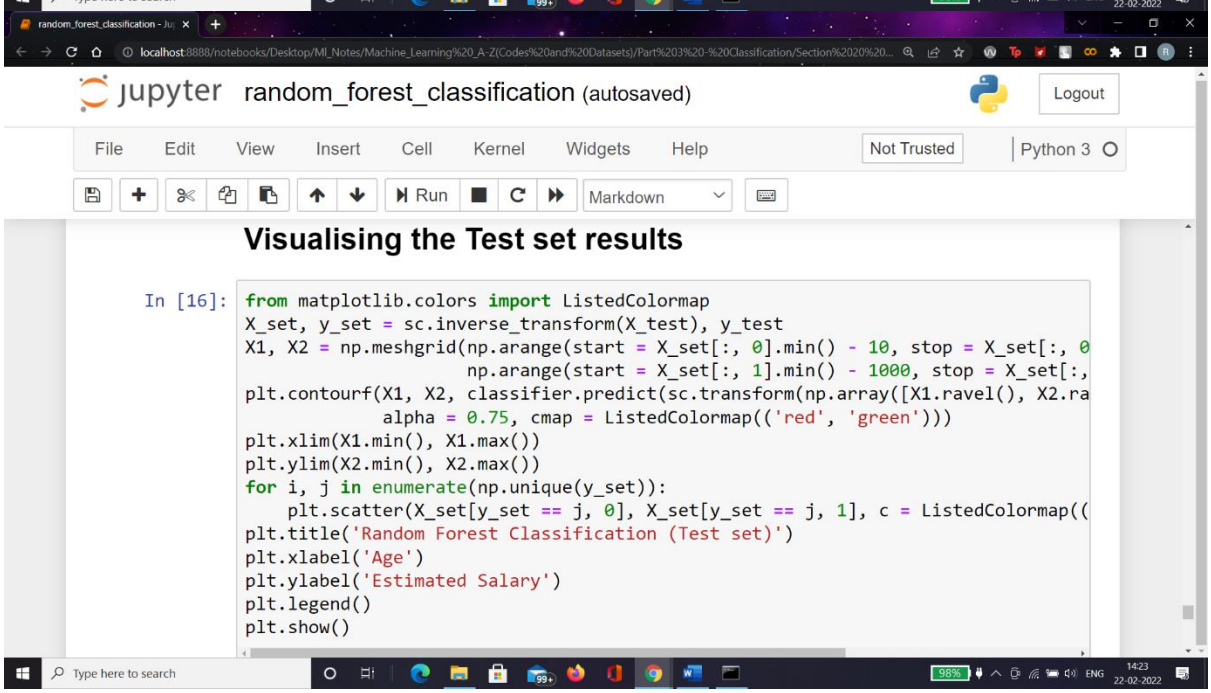

random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter
random_forest_classification (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Markdown
o',
max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=10,
n_jobs=None, oob_score=False, random_state=0, verbose=0,
warm_start=False)
Predicting a new result
In [12]:
print(classifier.predict(sc.transform([[30,87000]])))
[0]
Predicting the Test set results
Type here to search
98%
ENG
14:22
22-02-2022

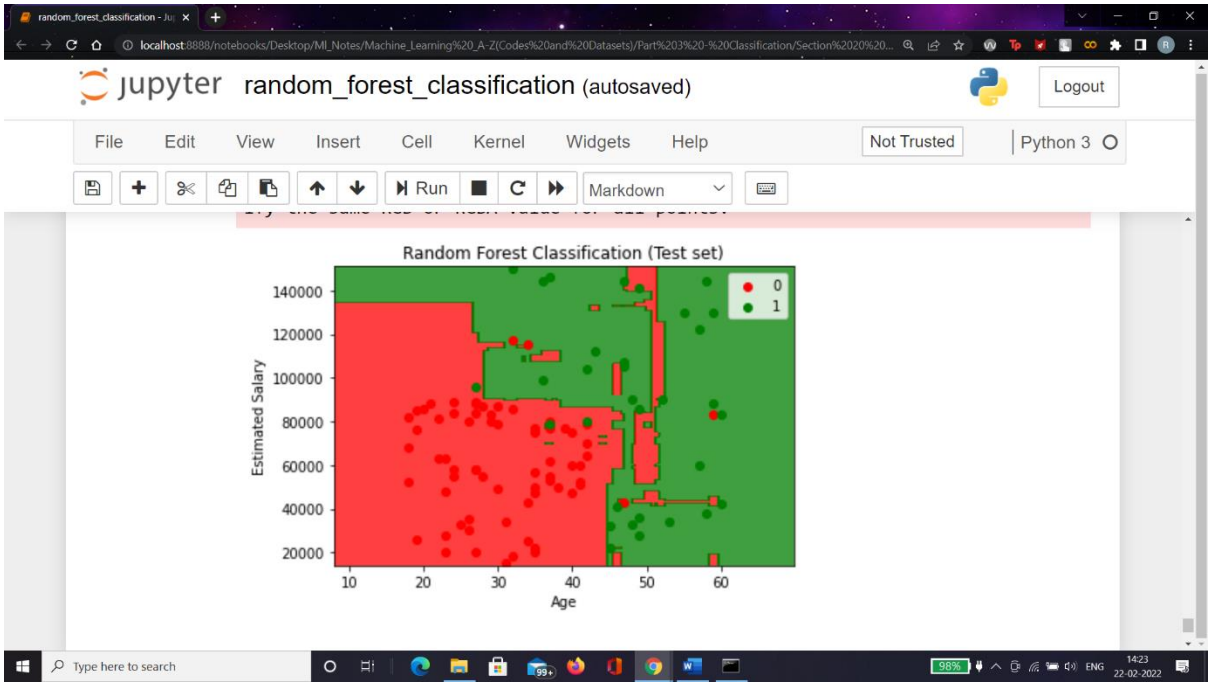
random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter
random_forest_classification (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Markdown
In [12]:
print(classifier.predict(sc.transform([[30,87000]])))
[0]
Predicting the Test set results
In [13]:
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),
[[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
Type here to search
98%
ENG
14:22
22-02-2022


random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Making the Confusion Matrix
In [14]:
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
[[63 5]
[4 28]]
Out[14]: 0.91
Visualising the Training set results


random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Visualising the Training set results
In [15]:
from matplotlib.colors import ListedColormap
X_set, y_set = sc.inverse_transform(X_train), y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 10, stop = X_set[:, 0
np.arange(start = X_set[:, 1].min() - 1000, stop = X_set[:,
plt.contourf(X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ra
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap((
plt.title('Random Forest Classification (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()


random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Random Forest Classification (Training set)
Estimated Salary
Age
0
1
140000
120000
100000
80000
60000
40000
20000
10
20
30
40
50
60
Type here to search
98%
ENG
14:23
22-02-2022


random_forest_classification - Ju
localhost:8888/notebooks/Desktop/ML_Notes/Machine_Learning%20_A-Z(Codes%20and%20Datasets)/Part%203%20-%20Classification/Section%2020%20...
jupyter random_forest_classification (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Visualising the Test set results
In [16]:
from matplotlib.colors import ListedColormap
X_set, y_set = sc.inverse_transform(X_test), y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 10, stop = X_set[:, 0
np.arange(start = X_set[:, 1].min() - 1000, stop = X_set[:,
plt.contourf(X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ra
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap((
plt.title('Random Forest Classification (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
Type here to search
98%
ENG
14:23
22-02-2022



Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.