PARSHVANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

Department of Computer Science and Engineering
Data Science

Academic Year: 2024-2025
Semester: VI
Class / Branch: CSE Data Science
Subject: ML lab
Name of Instructor:Prof. Ujwala Pagare

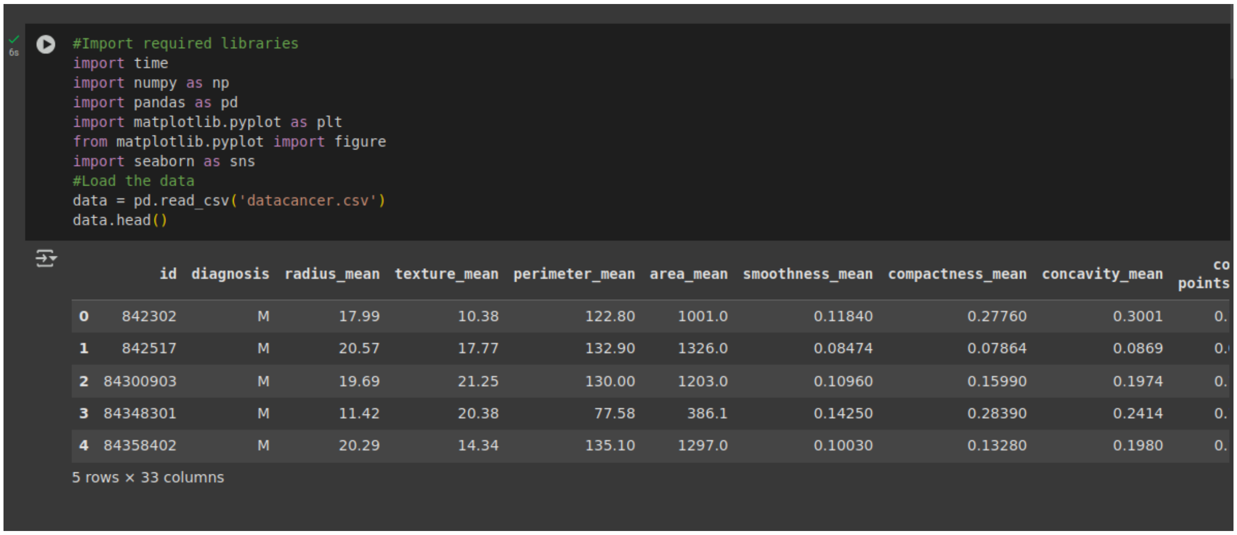
Name of Student:Annsh Yadav
Student ID:22107012
Date Of Performance: 03/04/2025
Date Of Submission:03/04/2025

Experiment No. 9

Aim:- To implement the Feature Selection technique for dimensionality reduction using python.

Program:-

```
#Import required libraries

import time

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib.pyplot import figure

import seaborn as sns

#Load the data

data = pd.read_csv('datacancer.csv')

data.head()
```



```
#Import required libraries
import time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
#Load the data
data = pd.read_csv('datacancer.csv')
data.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.

5 rows × 33 columns

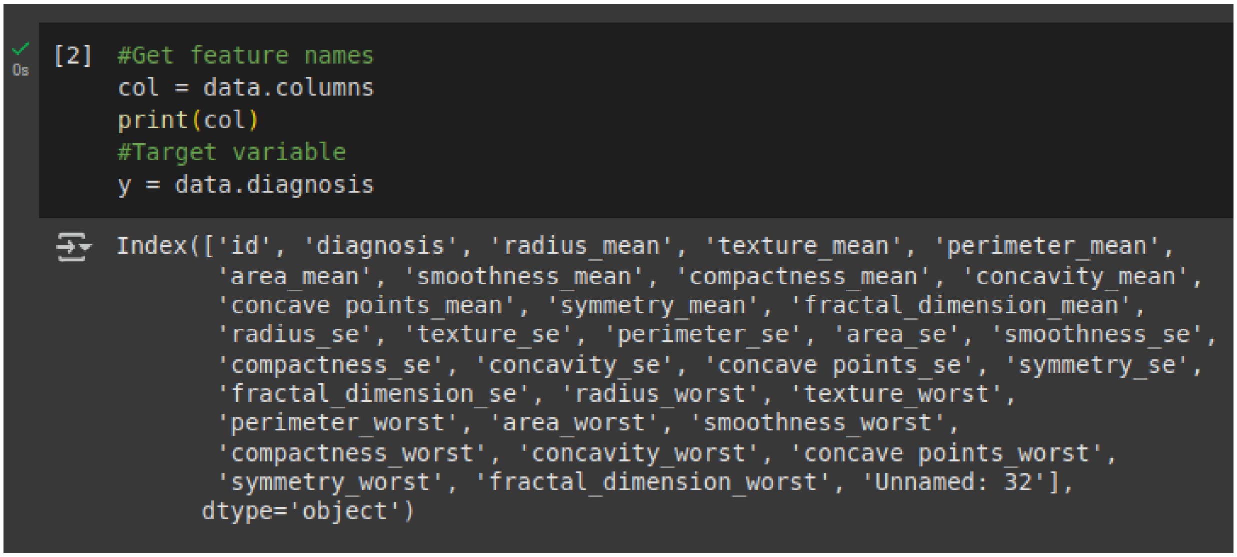
#Get feature names

col = data.columns

print(col)

#Target variable

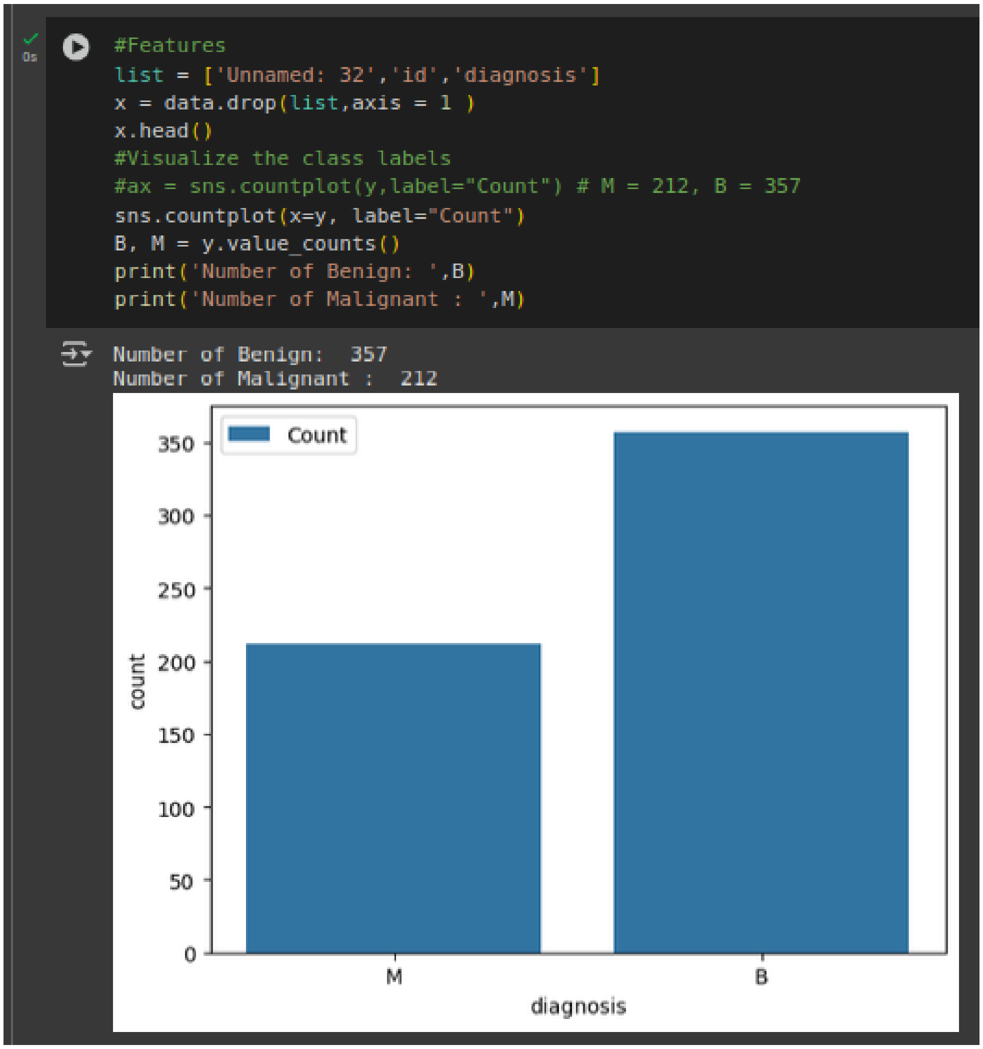
y = data.diagnosis



```
[2] #Get feature names
    col = data.columns
    print(col)
    #Target variable
    y = data.diagnosis
```

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

```
list = ['Unnamed: 32','id','diagnosis']

x = data.drop(list,axis = 1 )

x.head()

#Visualize the class labels

#ax = sns.countplot(y,label="Count") # M = 212, B = 357

sns.countplot(x=y, label="Count")

B, M = y.value_counts()

print('Number of Benign: ',B)

print('Number of Malignant : ',M)
```



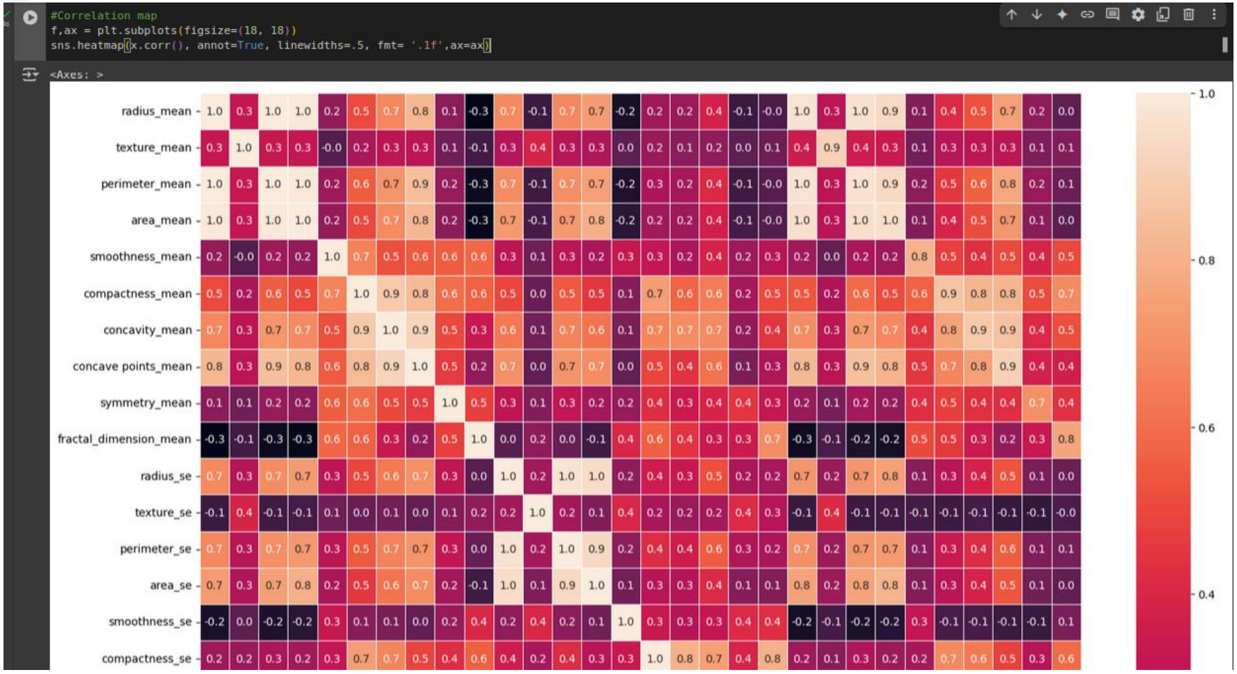
```
#Features
list = ['Unnamed: 32','id','diagnosis']
x = data.drop(list,axis = 1 )
x.head()
#Visualize the class labels
#ax = sns.countplot(y,label="Count") # M = 212, B = 357
sns.countplot(x=y, label="Count")
B, M = y.value_counts()
print('Number of Benign: ',B)
print('Number of Malignant : ',M)
```

```
Number of Benign:  357
Number of Malignant :  212
```

#Correlation map

f,ax = plt.subplots(figsize=(18, 18))

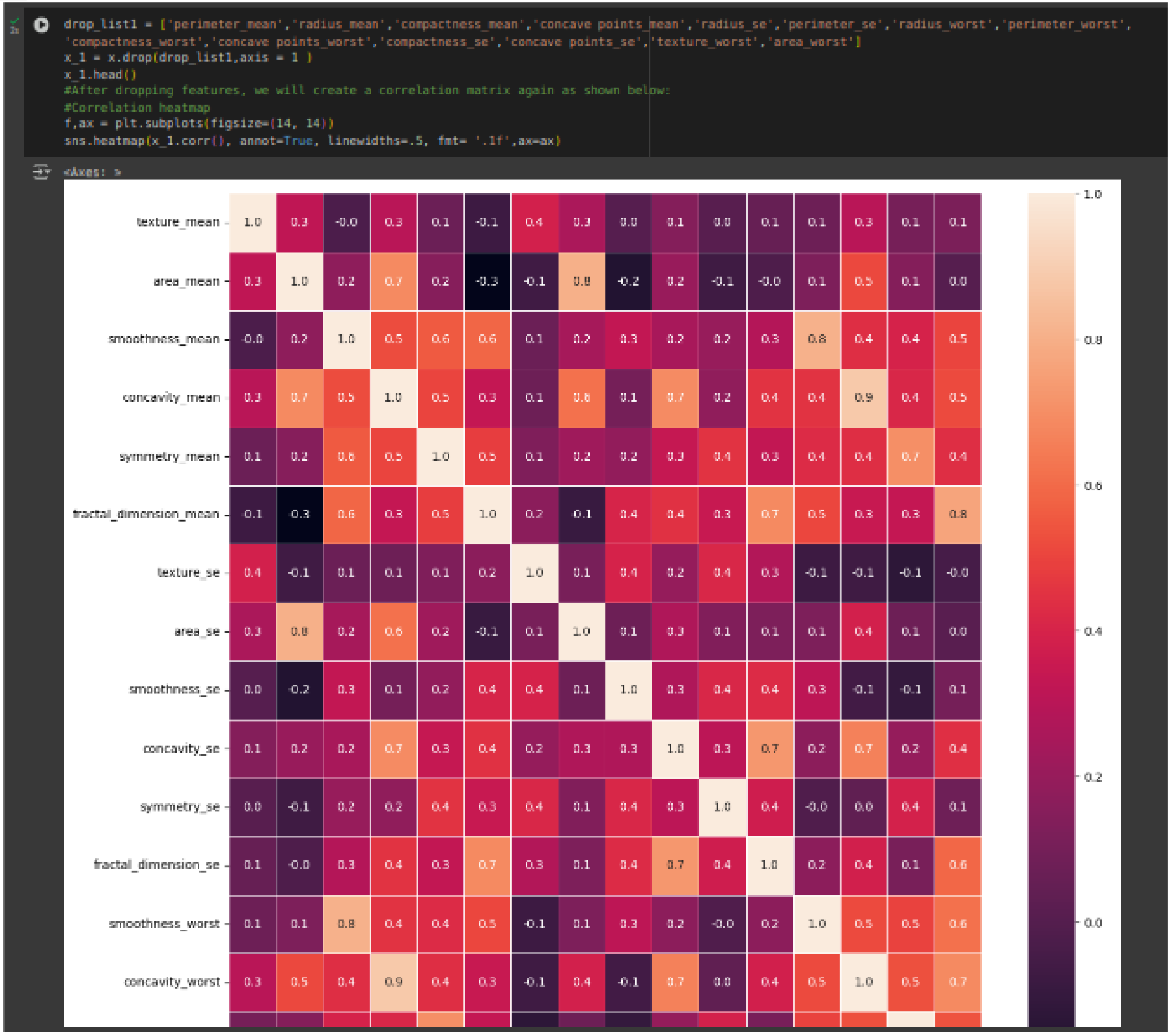
sns.heatmap(x.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)



```
#Correlation map
f,ax = plt.subplots(figsize=(18, 18))
sns.heatmap(x.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```

```
<Axes: >
```

```
drop_list1 = ['perimeter_mean','radius_mean','compactness_mean','concave
points_mean','radius_se','perimeter_se','radius_worst','perimeter_worst',
'compactness_worst','concave points_worst','compactness_se','concave
points_se','texture_worst','area_worst']
x_1 = x.drop(drop_list1,axis = 1 )
x_1.head()
#After dropping features, we will create a correlation matrix again as shown
below:
#Correlation heatmap
f,ax = plt.subplots(figsize=(14, 14))
sns.heatmap(x_1.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```

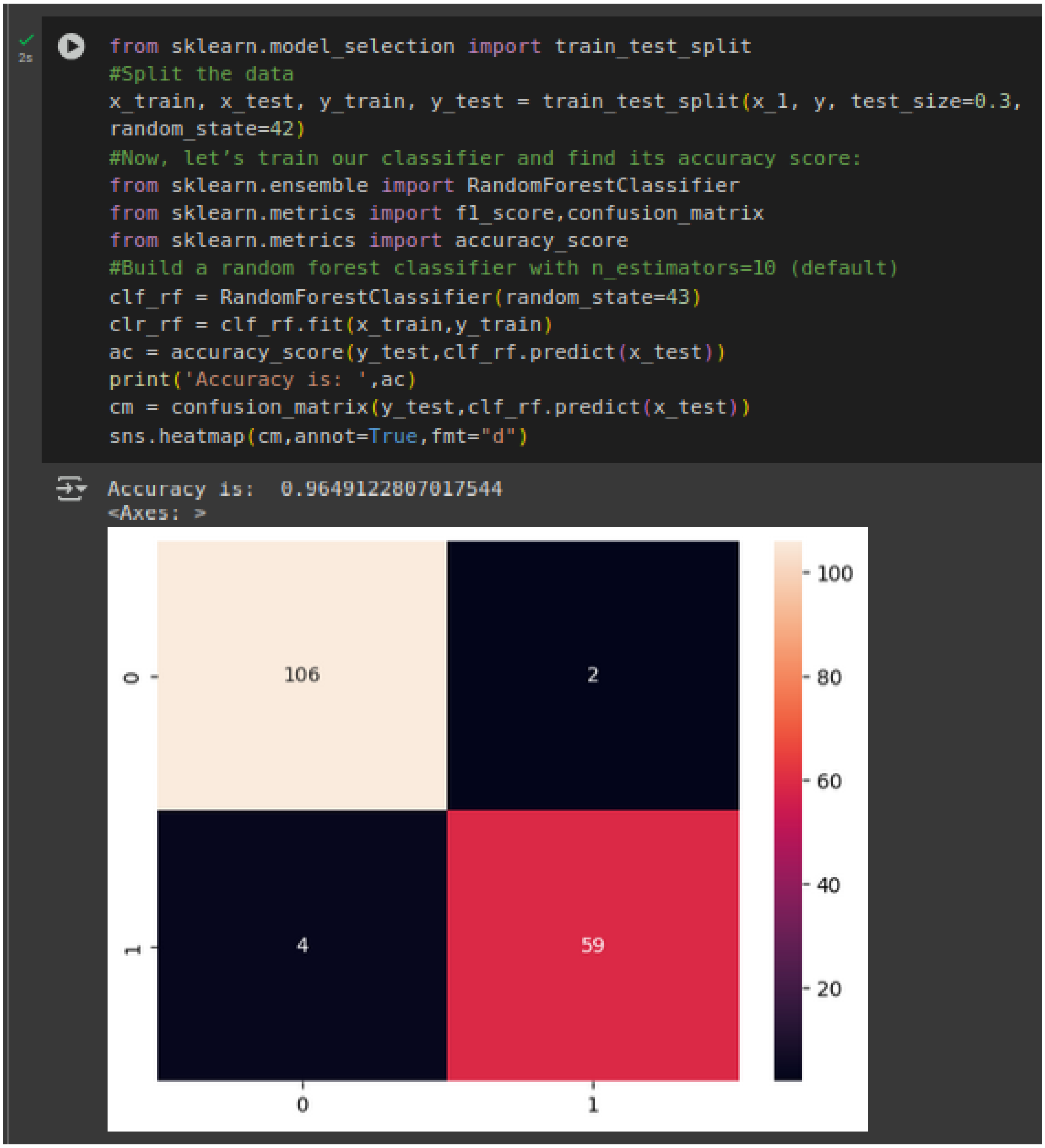


```
drop_list1 = ['perimeter_mean','radius_mean','compactness_mean','concave points_mean','radius_se','perimeter_se','radius_worst','perimeter_worst',
'compactness_worst','concave points_worst','compactness_se','concave points_se','texture_worst','area_worst']
x_1 = x.drop(drop_list1,axis = 1 )
x_1.head()
#After dropping features, we will create a correlation matrix again as shown below:
#Correlation heatmap
f,ax = plt.subplots(figsize=(14, 14))
sns.heatmap(x_1.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```

```
<Axes: >
```

from sklearn.model_selection import train_test_split

```
#Split the data

x_train, x_test, y_train, y_test = train_test_split(x_1, y, test_size=0.3,

random_state=42)

#Now, let’s train our classifier and find its accuracy score:

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import f1_score,confusion_matrix

from sklearn.metrics import accuracy_score

#Build a random forest classifier with n_estimators=10 (default)

clf_rf = RandomForestClassifier(random_state=43)

clr_rf = clf_rf.fit(x_train,y_train)

ac = accuracy_score(y_test,clf_rf.predict(x_test))

print('Accuracy is: ',ac)

cm = confusion_matrix(y_test,clf_rf.predict(x_test))

sns.heatmap(cm,annot=True,fmt="d")
```



```
from sklearn.model_selection import train_test_split
#Split the data
x_train, x_test, y_train, y_test = train_test_split(x_l, y, test_size=0.3,
random_state=42)
#Now, let’s train our classifier and find its accuracy score:
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score,confusion_matrix
from sklearn.metrics import accuracy_score
#Build a random forest classifier with n_estimators=10 (default)
clf_rf = RandomForestClassifier(random_state=43)
clr_rf = clf_rf.fit(x_train,y_train)
ac = accuracy_score(y_test,clf_rf.predict(x_test))
print('Accuracy is: ',ac)
cm = confusion_matrix(y_test,clf_rf.predict(x_test))
sns.heatmap(cm,annot=True,fmt="d")
```

```
Accuracy is:  0.9649122807017544
<Axes: >
```

Conclusion: Thus, we have implemented the Feature Selection technique for dimensionality reduction using Python.