Saurav Barua
Assistant Professor,
Daffodil International University

Program for Route Choice Model using Support Vector Machine (SVM)

written by Saurav Barua, Assistant Professor, Daffodil International University

Problem statement: A and B are two alternative routes, free flow travel times are-- 10 and 20 minutes respectively. Four decision variables are--(1) Travel time, (2) Travel time fluctuation, (3) Trip purpose and

(4) Familiarity. Travel times and travel time fluctuation are in minutes Trip purpose '1' for work-based trip and '0' for non-work based trip. Familiarity is '1' for familiar and '0' for unfamiliar with the route.

Codes:

```
#Program for Route Choice Model using Support Vector Machine (SVM)
# written by Saurav Barua, Assistant Professor, Daffodil International University
#Problem statement: A and B are two alternative routes, free flow travel
# times are-- 10 and 20 minutes respectively. Four decision variables are--
#(1) Travel time, (2)Travel time fluctuation, (3) Trip purpose and
#(4)Familiarity. Travel times and travel time fluctuation are in minutes
# Trip purpose '1' for workbased trip and '0' for non-work based trip
# Familiarity is '1' for familar and '0' for unfamiliar with the route.

# import numpy, pandas, matplotlib and seaborn library in python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
# Retrive data from excel csv format
data1=pd.read_csv(r'C:\Users\Administrator\Desktop/route choice.csv')
data1.head()
# pair plot
sns.pairplot(data=data1, hue='Route', palette='Set2')
from sklearn.model_selection import train_test_split
```

Saurav Barua
Assistant Professor,
Daffodil International University

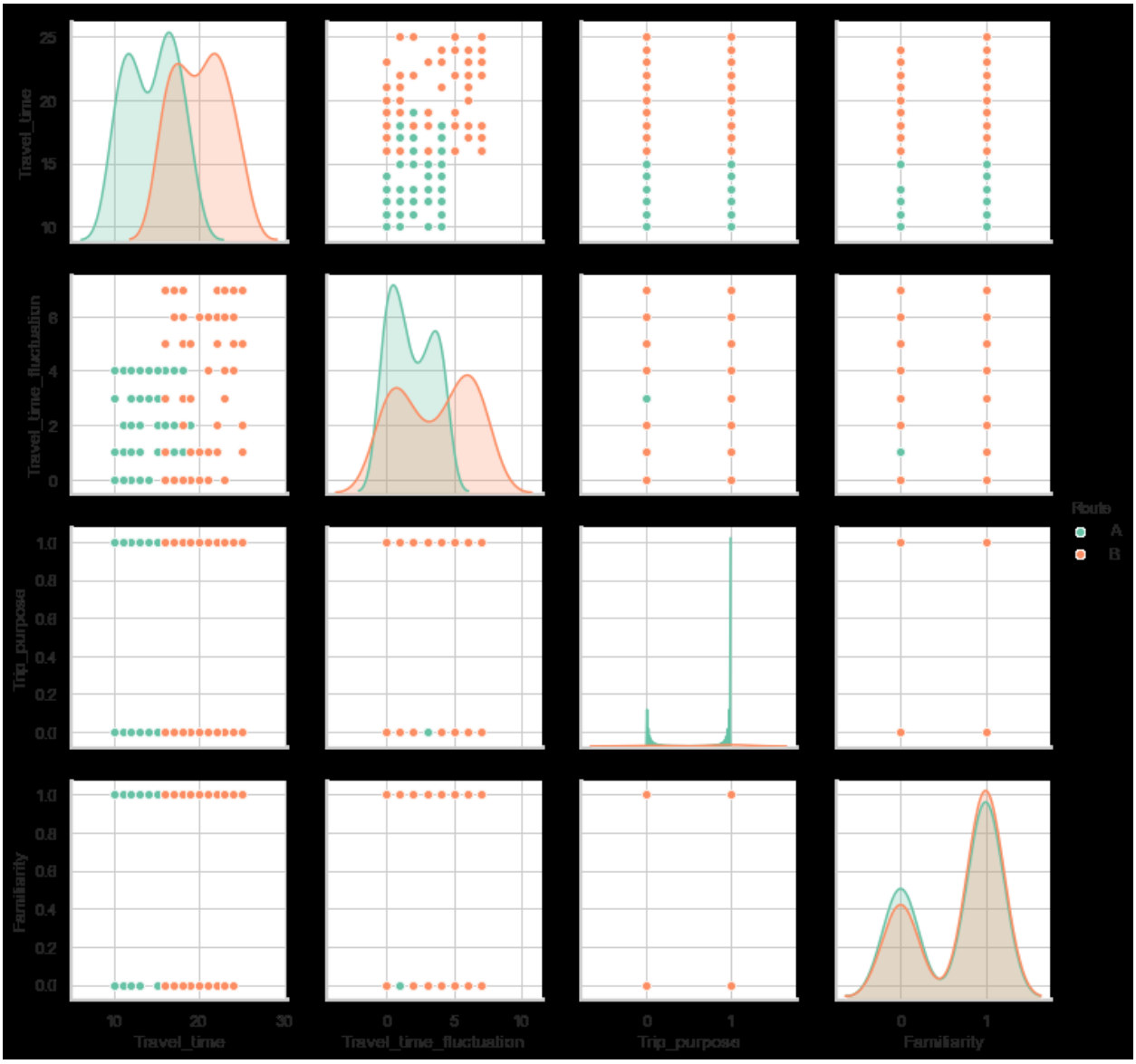
```
x=data1.iloc[:,:-1]

y=data1.iloc[:,4]

x_train,x_test, y_train, y_test=train_test_split(x,y,test_size=0.30)

from sklearn.svm import SVC

# Use linear kernel trick in Support vector machine

model=SVC(kernel='linear')

model.fit(x_train, y_train)

pred=model.predict(x_test)

from sklearn.metrics import classification_report, confusion_matrix

# generate confusion matrix

print(confusion_matrix(y_test,pred))

print(classification_report(y_test, pred))

# coefficient of decision variables which can be used for ranking attributes

a = model.coef_

print(a)

# interception

b = model.intercept_

print(b)
```

Results:

```
[[17  3]
 [ 2 20]]
              precision    recall  f1-score   support

           A       0.89      0.85      0.87        20
           B       0.87      0.91      0.89        22

   micro avg       0.88      0.88      0.88        42
   macro avg       0.88      0.88      0.88        42
weighted avg       0.88      0.88      0.88        42

[[ 0.69562186  0.36971044 -1.06499018 -0.69601219]]
[-11.84743498]
```

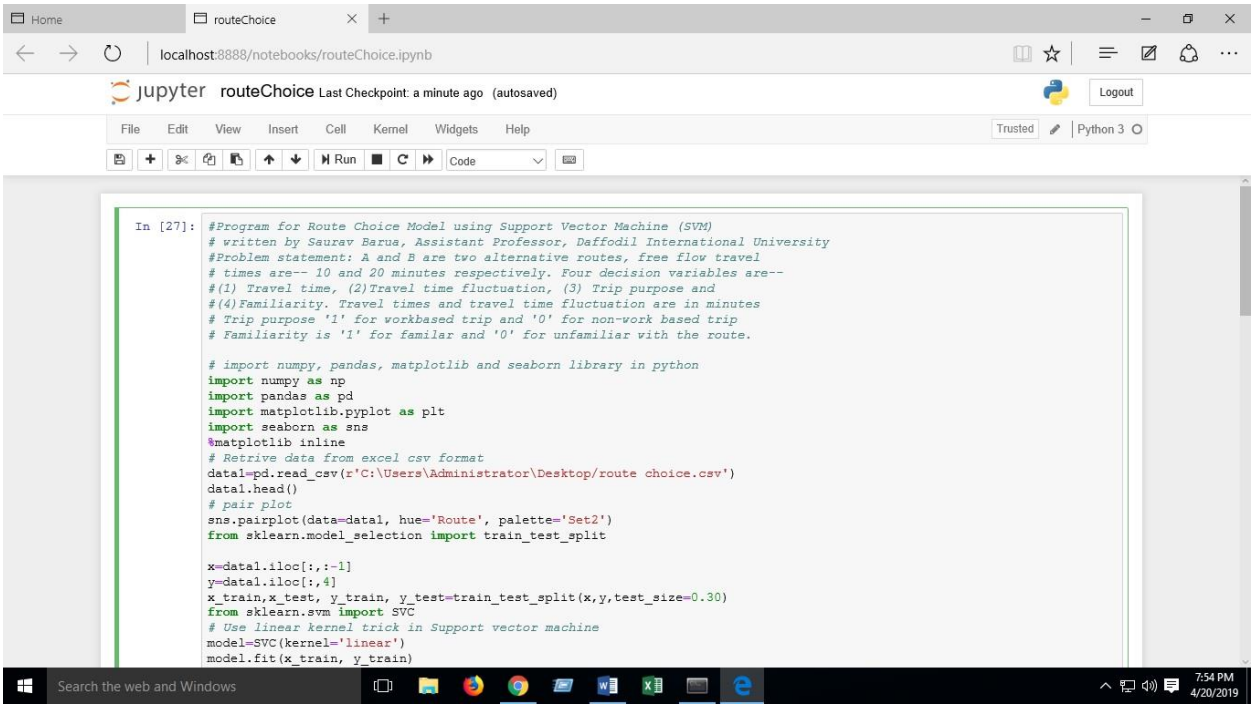
Output plots:

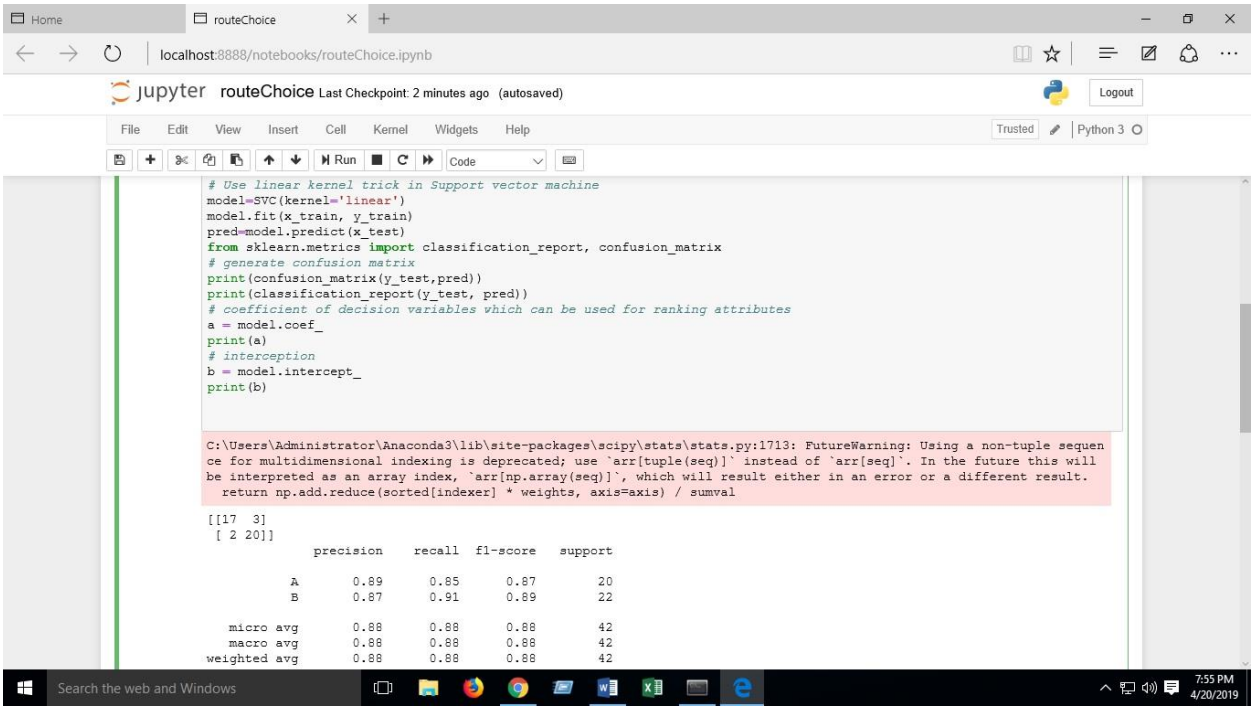
Saurav Barua
Assistant Professor,
Daffodil International University



Snapshots:

Saurav Barua
Assistant Professor,
Daffodil International University





Saurav Barua
Assistant Professor,
Daffodil International University