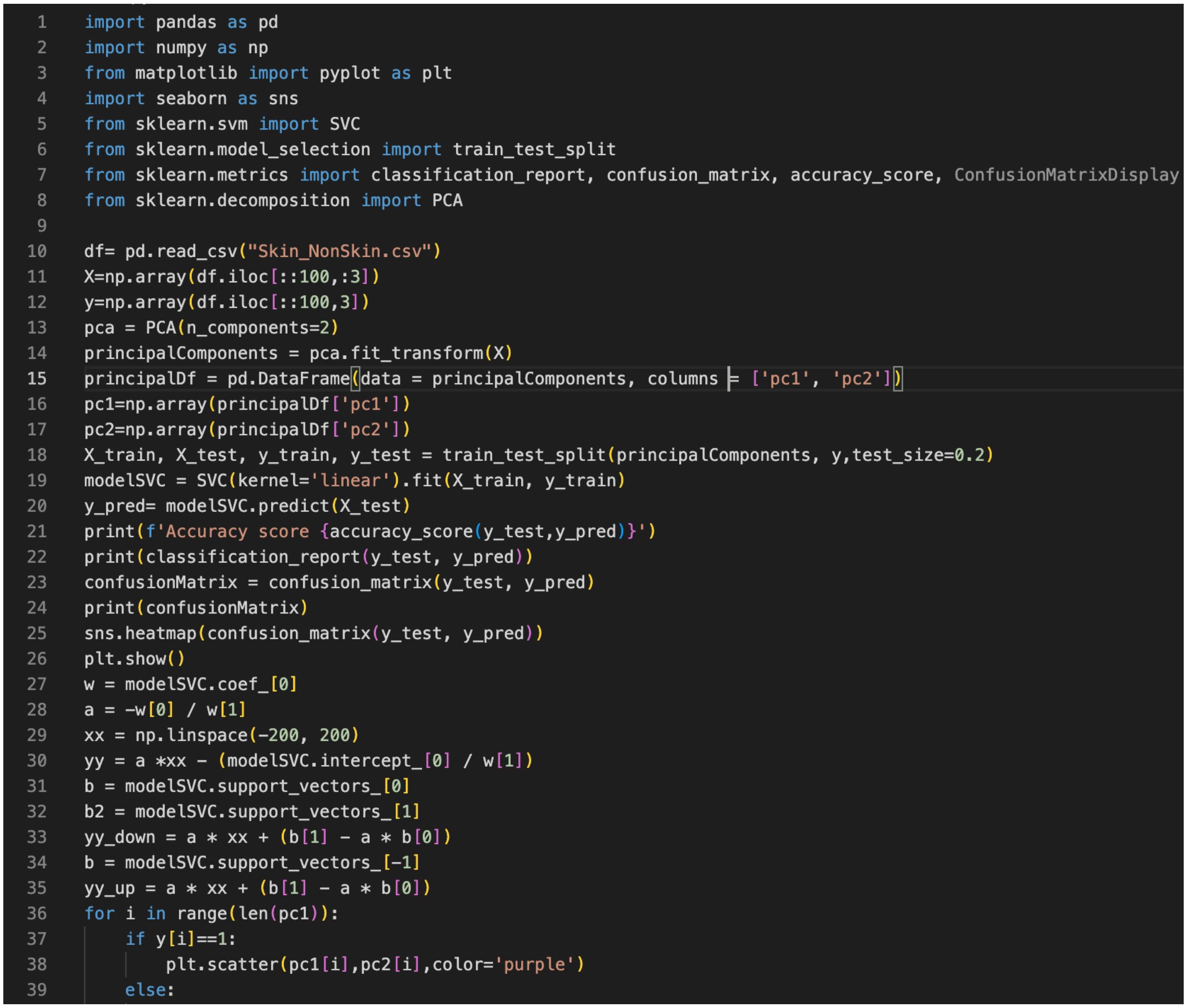
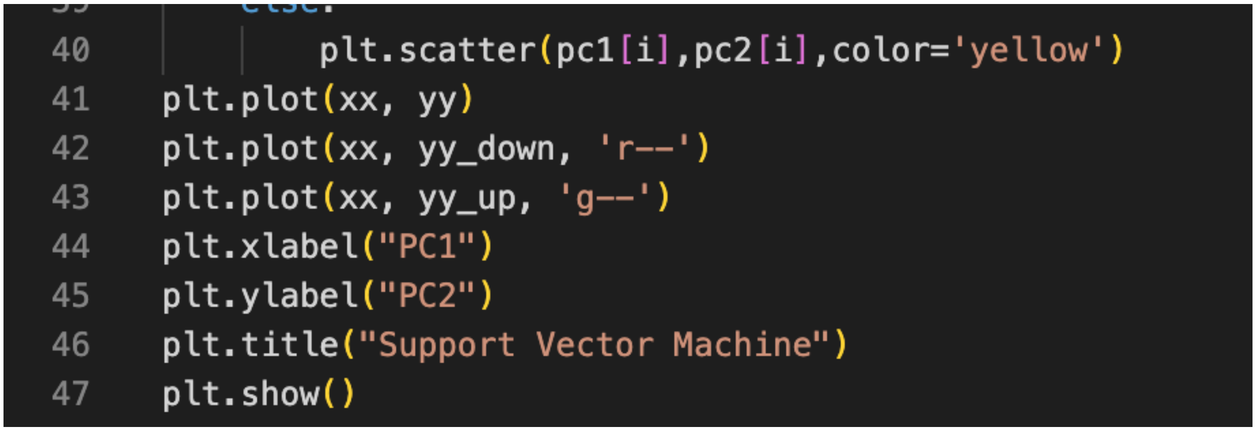
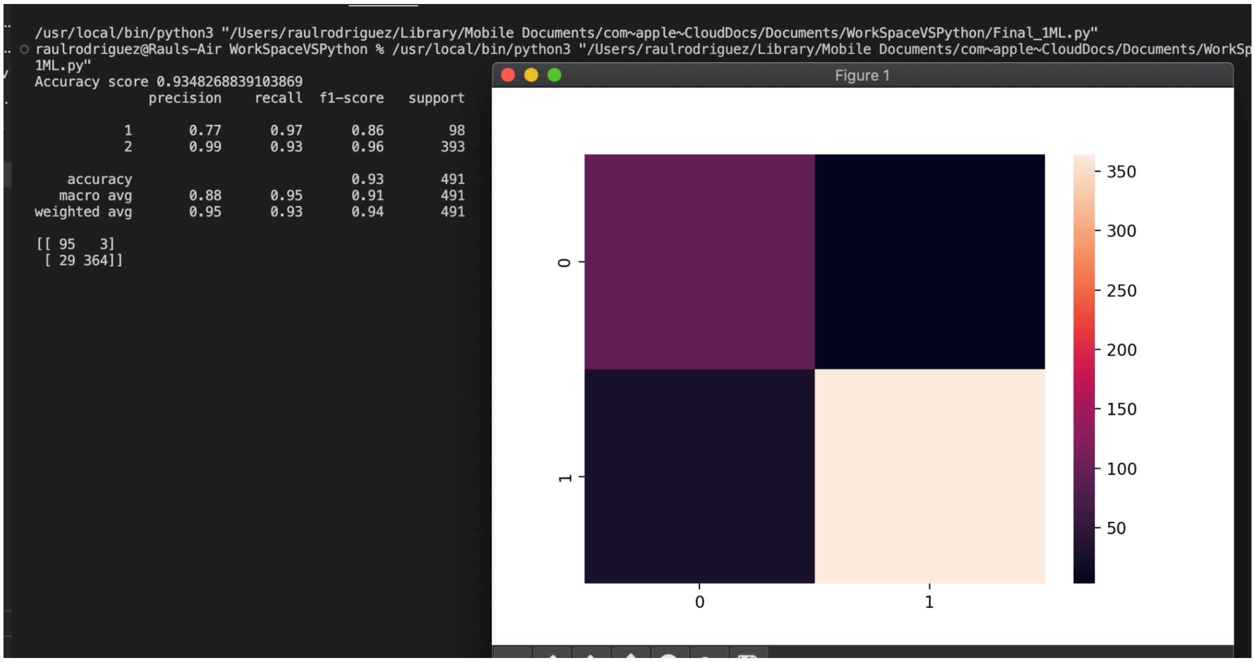


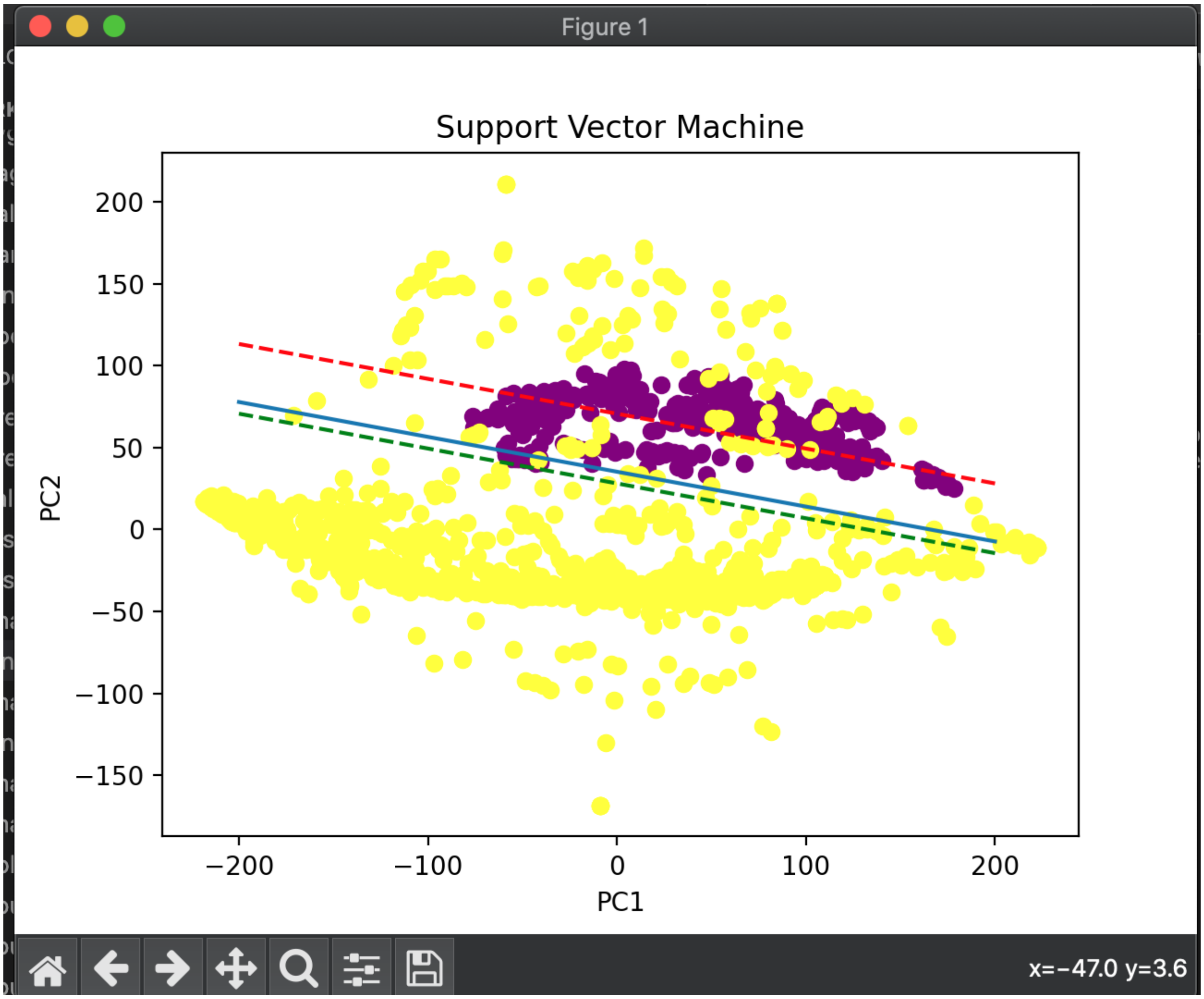
```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, ConfusionMatrixDisplay
from sklearn.decomposition import PCA

df= pd.read_csv("Skin_NonSkin.csv")
X=np.array(df.iloc[::100,:3])
y=np.array(df.iloc[::100,3])
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(X)
principalDf = pd.DataFrame(data = principalComponents, columns = ['pc1', 'pc2'])
pc1=np.array(principalDf['pc1'])
pc2=np.array(principalDf['pc2'])
X_train, X_test, y_train, y_test = train_test_split(principalComponents, y,test_size=0.2)
modelSVC = SVC(kernel='linear').fit(X_train, y_train)
y_pred= modelSVC.predict(X_test)
print(f'Accuracy score {accuracy_score(y_test,y_pred)}')
print(classification_report(y_test, y_pred))
confusionMatrix = confusion_matrix(y_test, y_pred)
print(confusionMatrix)
sns.heatmap(confusion_matrix(y_test, y_pred))
plt.show()
w = modelSVC.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(-200, 200)
yy = a *xx - (modelSVC.intercept_[0] / w[1])
b = modelSVC.support_vectors_[0]
b2 = modelSVC.support_vectors_[1]
yy_down = a * xx + (b[1] - a * b[0])
b = modelSVC.support_vectors_[-1]
yy_up = a * xx + (b[1] - a * b[0])
for i in range(len(pc1)):
    if y[i]==1:
        plt.scatter(pc1[i],pc2[i],color='purple')
    else:
```

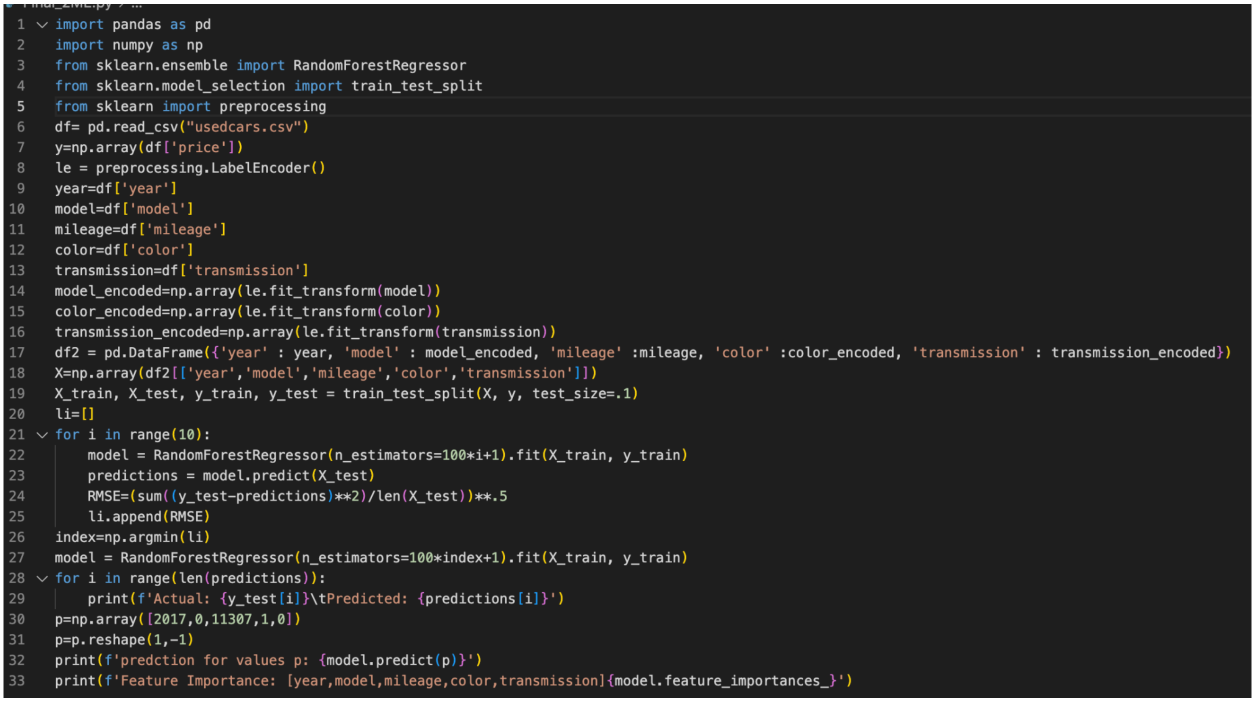


```python
        plt.scatter(pc1[i],pc2[i],color='yellow')
plt.plot(xx, yy)
plt.plot(xx, yy_down, 'r--')
plt.plot(xx, yy_up, 'g--')
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.title("Support Vector Machine")
plt.show()
```

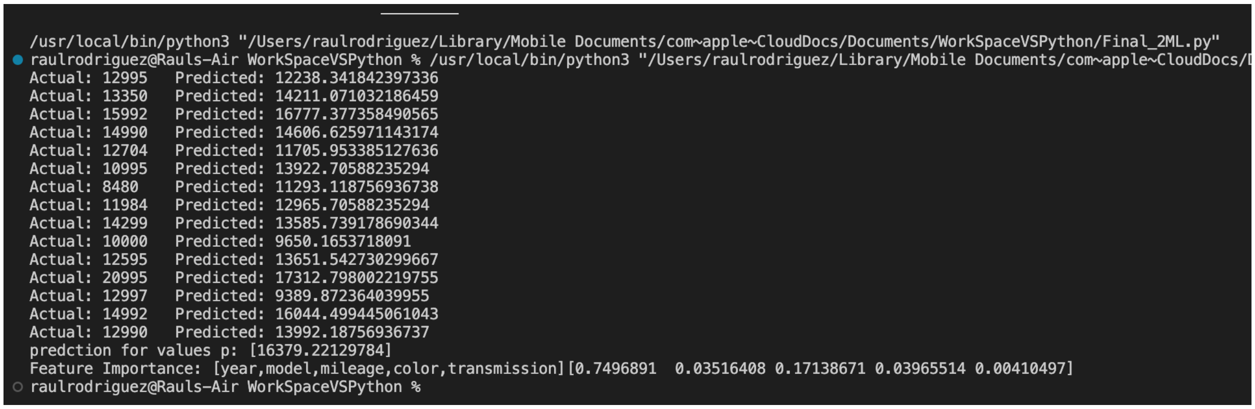


```
/usr/local/bin/python3 "/Users/raulrodriguez/Library/Mobile Documents/com~apple~CloudDocs/Documents/WorkSpaceVSPython/Final_1ML.py"
○ raulrodriguez@Rauls-Air WorkSpaceVSPython % /usr/local/bin/python3 "/Users/raulrodriguez/Library/Mobile Documents/com~apple~CloudDocs/Documents/WorkSp
1ML.py"
Accuracy score 0.9348268839103869
              precision    recall  f1-score   support

           1       0.77      0.97      0.86        98
           2       0.99      0.93      0.96       393

    accuracy                           0.93       491
   macro avg       0.88      0.95      0.91       491
weighted avg       0.95      0.93      0.94       491

[[ 95   3]
 [ 29 364]]
```





```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
df= pd.read_csv("usedcars.csv")
y=np.array(df['price'])
le = preprocessing.LabelEncoder()
year=df['year']
model=df['model']
mileage=df['mileage']
color=df['color']
transmission=df['transmission']
model_encoded=np.array(le.fit_transform(model))
color_encoded=np.array(le.fit_transform(color))
transmission_encoded=np.array(le.fit_transform(transmission))
df2 = pd.DataFrame({'year' : year, 'model' : model_encoded, 'mileage' :mileage, 'color' :color_encoded, 'transmission' : transmission_encoded})
X=np.array(df2[['year','model','mileage','color','transmission']])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.1)
li=[]
for i in range(10):
    model = RandomForestRegressor(n_estimators=100*i+1).fit(X_train, y_train)
    predictions = model.predict(X_test)
    RMSE=(sum((y_test-predictions)**2)/len(X_test))**.5
    li.append(RMSE)
index=np.argmin(li)
model = RandomForestRegressor(n_estimators=100*index+1).fit(X_train, y_train)
for i in range(len(predictions)):
    print(f'Actual: {y_test[i]}\tPredicted: {predictions[i]}')
p=np.array([2017,0,11307,1,0])
p=p.reshape(1,-1)
print(f'predction for values p: {model.predict(p)}')
print(f'Feature Importance: [year,model,mileage,color,transmission]{model.feature_importances_}')
```



```
  /usr/local/bin/python3 "/Users/raulrodriguez/Library/Mobile Documents/com~apple~CloudDocs/Documents/WorkSpaceVSPython/Final_2ML.py"
● raulrodriguez@Rauls-Air WorkSpaceVSPython % /usr/local/bin/python3 "/Users/raulrodriguez/Library/Mobile Documents/com~apple~CloudDocs/D
  Actual: 12995   Predicted: 12238.341842397336
  Actual: 13350   Predicted: 14211.071032186459
  Actual: 15992   Predicted: 16777.377358490565
  Actual: 14990   Predicted: 14606.625971143174
  Actual: 12704   Predicted: 11705.953385127636
  Actual: 10995   Predicted: 13922.70588235294
  Actual: 8480    Predicted: 11293.118756936738
  Actual: 11984   Predicted: 12965.70588235294
  Actual: 14299   Predicted: 13585.739178690344
  Actual: 10000   Predicted: 9650.1653718091
  Actual: 12595   Predicted: 13651.542730299667
  Actual: 20995   Predicted: 17312.798002219755
  Actual: 12997   Predicted: 9389.872364039955
  Actual: 14992   Predicted: 16044.499445061043
  Actual: 12990   Predicted: 13992.18756936737
  predction for values p: [16379.22129784]
  Feature Importance: [year,model,mileage,color,transmission][0.7496891  0.03516408 0.17138671 0.03965514 0.00410497]
○ raulrodriguez@Rauls-Air WorkSpaceVSPython %
```