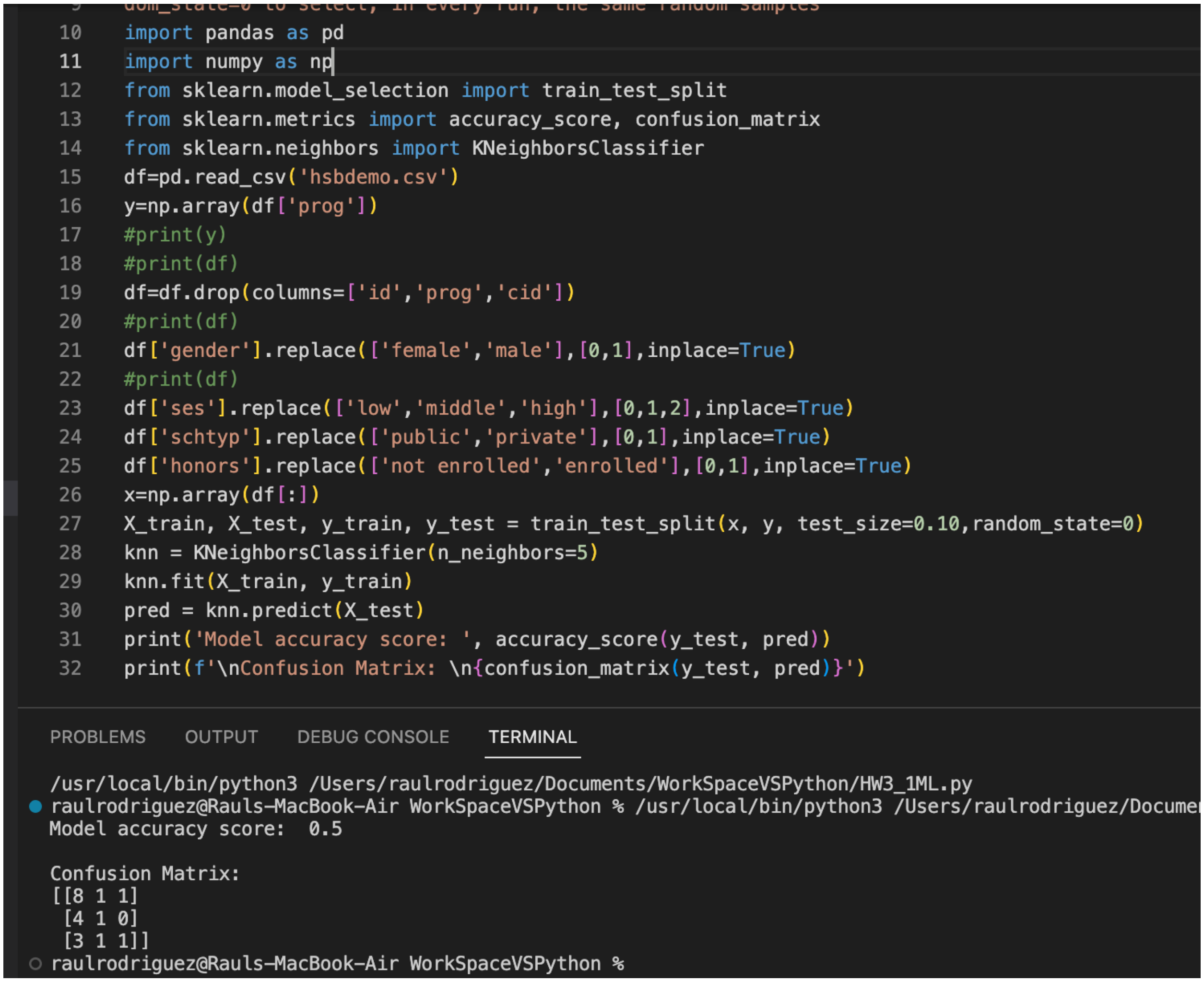
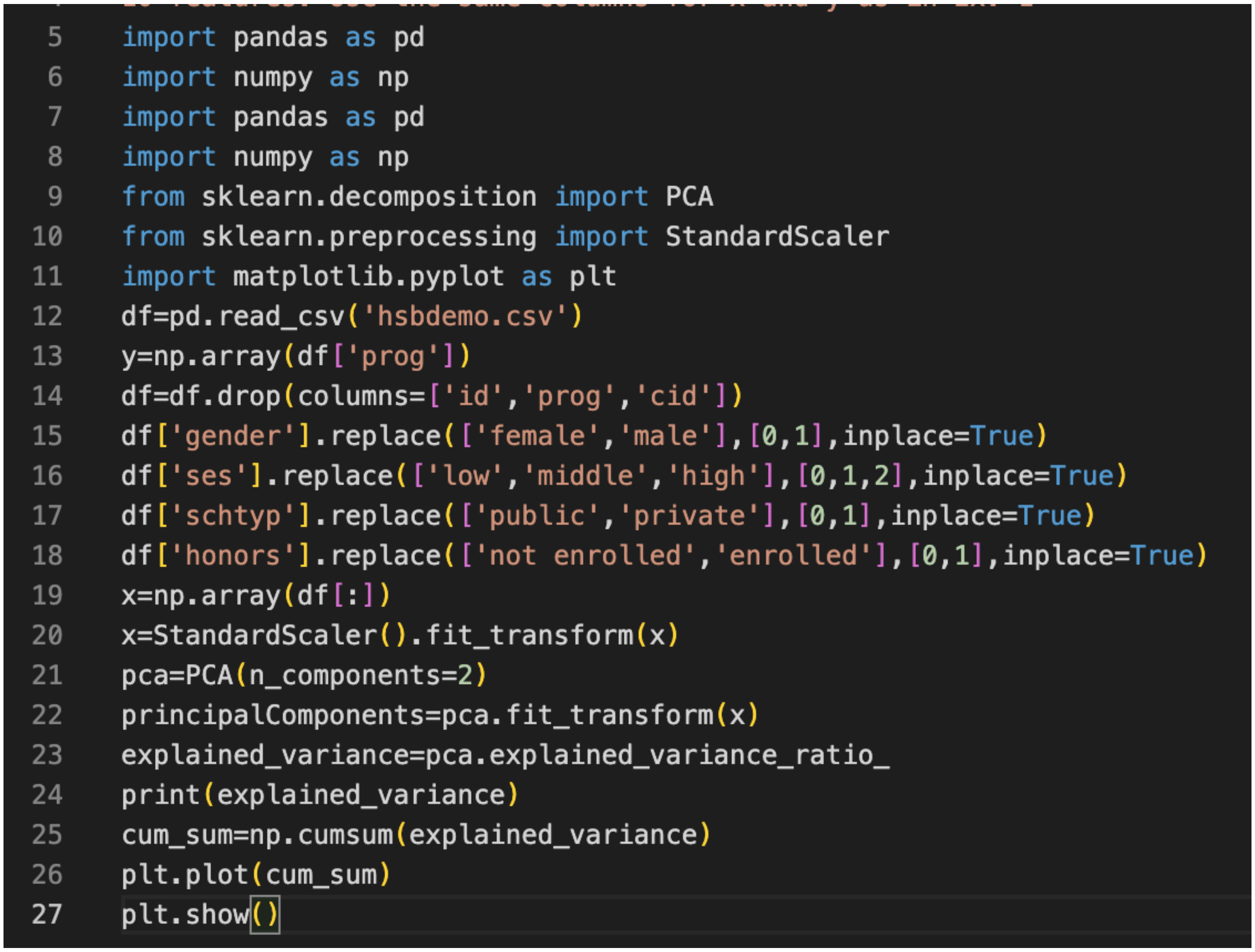
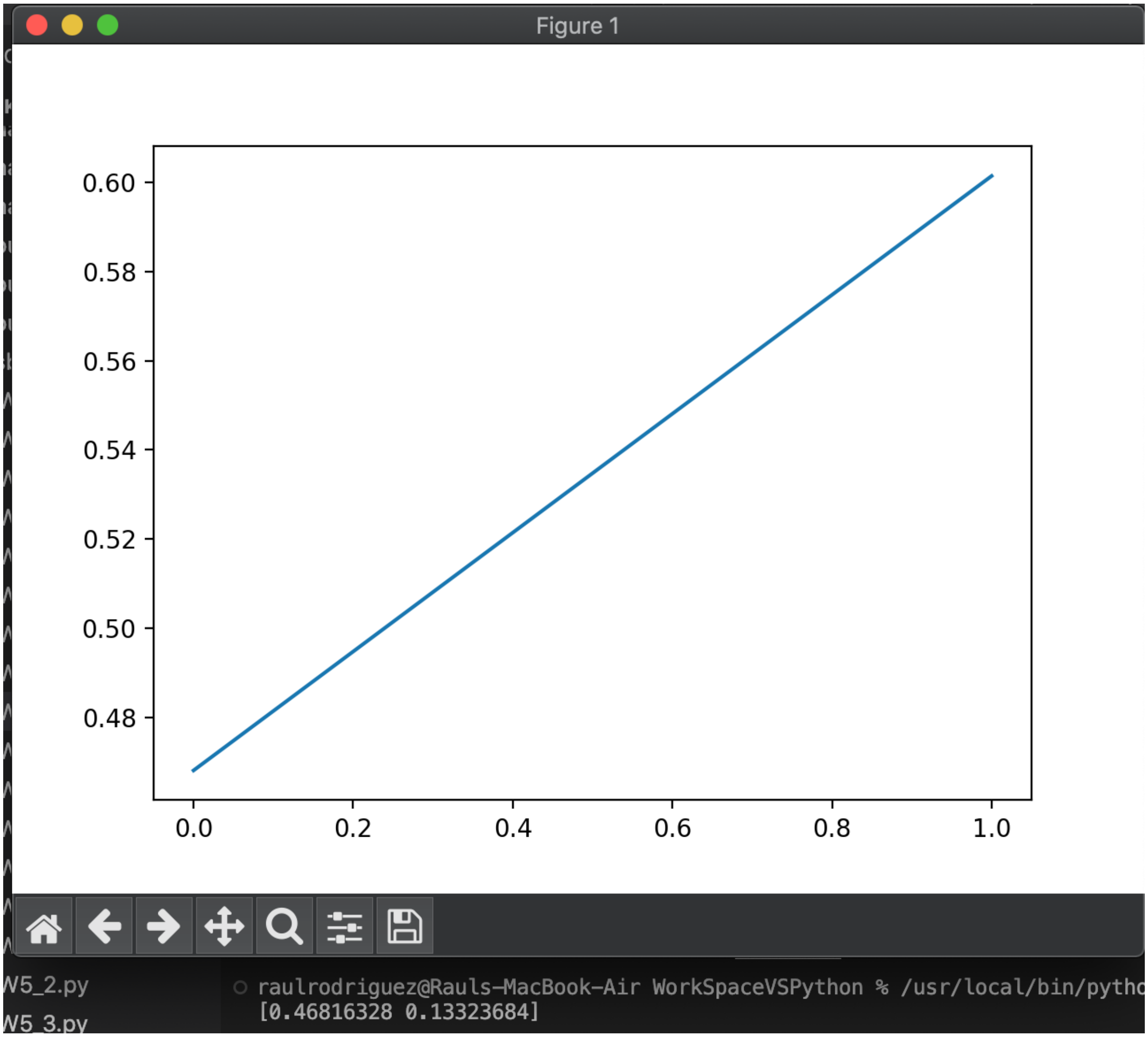


```python
dom_state=0 to select, in every run, the same random samples
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
df=pd.read_csv('hsbdemo.csv')
y=np.array(df['prog'])
#print(y)
#print(df)
df=df.drop(columns=['id','prog','cid'])
#print(df)
df['gender'].replace(['female','male'],[0,1],inplace=True)
#print(df)
df['ses'].replace(['low','middle','high'],[0,1,2],inplace=True)
df['schtyp'].replace(['public','private'],[0,1],inplace=True)
df['honors'].replace(['not enrolled','enrolled'],[0,1],inplace=True)
x=np.array(df[:])
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.10,random_state=0)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
pred = knn.predict(X_test)
print('Model accuracy score: ', accuracy_score(y_test, pred))
print(f'\nConfusion Matrix: \n{confusion_matrix(y_test, pred)}')
```

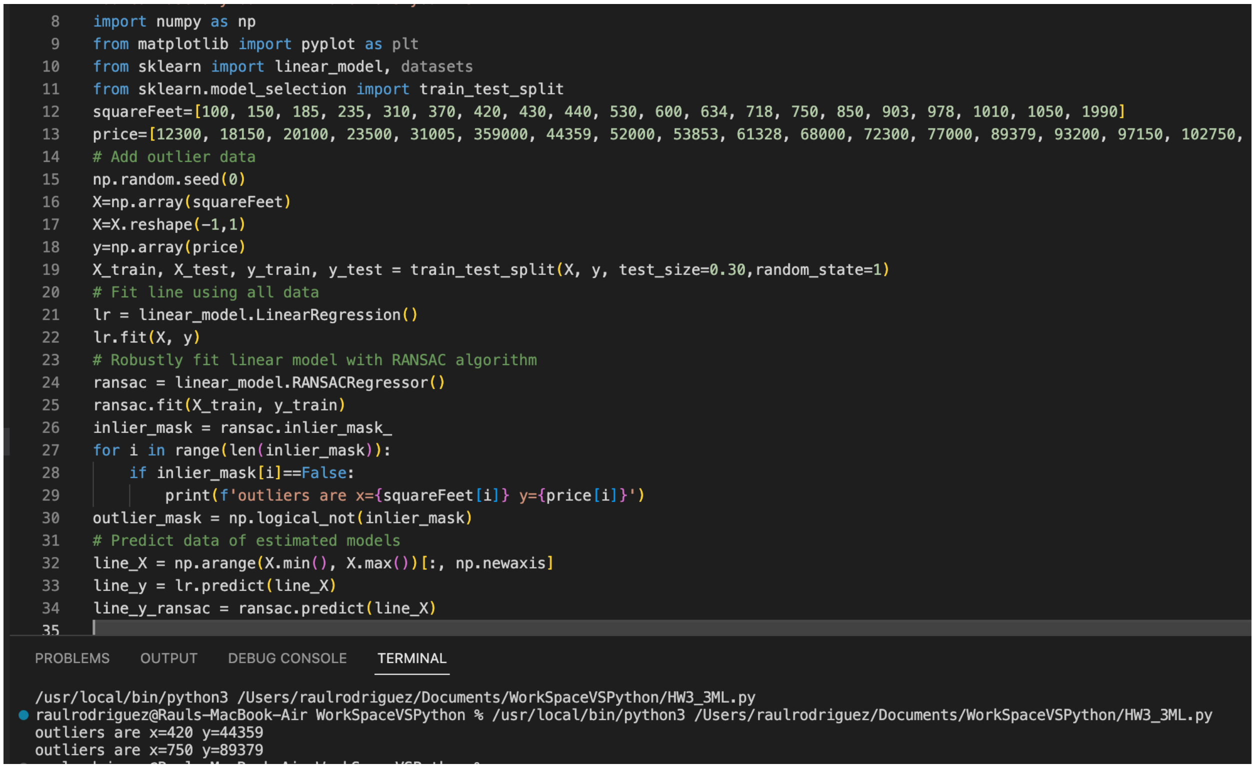
```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW3_1ML.py
● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documen
Model accuracy score:  0.5

Confusion Matrix:
[[8 1 1]
 [4 1 0]
 [3 1 1]]
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```



```python
import pandas as pd
import numpy as np
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
df=pd.read_csv('hsbdemo.csv')
y=np.array(df['prog'])
df=df.drop(columns=['id','prog','cid'])
df['gender'].replace(['female','male'],[0,1],inplace=True)
df['ses'].replace(['low','middle','high'],[0,1,2],inplace=True)
df['schtyp'].replace(['public','private'],[0,1],inplace=True)
df['honors'].replace(['not enrolled','enrolled'],[0,1],inplace=True)
x=np.array(df[:])
x=StandardScaler().fit_transform(x)
pca=PCA(n_components=2)
principalComponents=pca.fit_transform(x)
explained_variance=pca.explained_variance_ratio_
print(explained_variance)
cum_sum=np.cumsum(explained_variance)
plt.plot(cum_sum)
plt.show()
```



```
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/pytho
  [0.46816328 0.13323684]
```



```python
import numpy as np
from matplotlib import pyplot as plt
from sklearn import linear_model, datasets
from sklearn.model_selection import train_test_split
squareFeet=[100, 150, 185, 235, 310, 370, 420, 430, 440, 530, 600, 634, 718, 750, 850, 903, 978, 1010, 1050, 1990]
price=[12300, 18150, 20100, 23500, 31005, 359000, 44359, 52000, 53853, 61328, 68000, 72300, 77000, 89379, 93200, 97150, 102750,
# Add outlier data
np.random.seed(0)
X=np.array(squareFeet)
X=X.reshape(-1,1)
y=np.array(price)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,random_state=1)
# Fit line using all data
lr = linear_model.LinearRegression()
lr.fit(X, y)
# Robustly fit linear model with RANSAC algorithm
ransac = linear_model.RANSACRegressor()
ransac.fit(X_train, y_train)
inlier_mask = ransac.inlier_mask_
for i in range(len(inlier_mask)):
    if inlier_mask[i]==False:
        print(f'outliers are x={squareFeet[i]} y={price[i]}')
outlier_mask = np.logical_not(inlier_mask)
# Predict data of estimated models
line_X = np.arange(X.min(), X.max())[:, np.newaxis]
line_y = lr.predict(line_X)
line_y_ransac = ransac.predict(line_X)
```

```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW3_3ML.py
● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/HW3_3ML.py
outliers are x=420 y=44359
outliers are x=750 y=89379
```