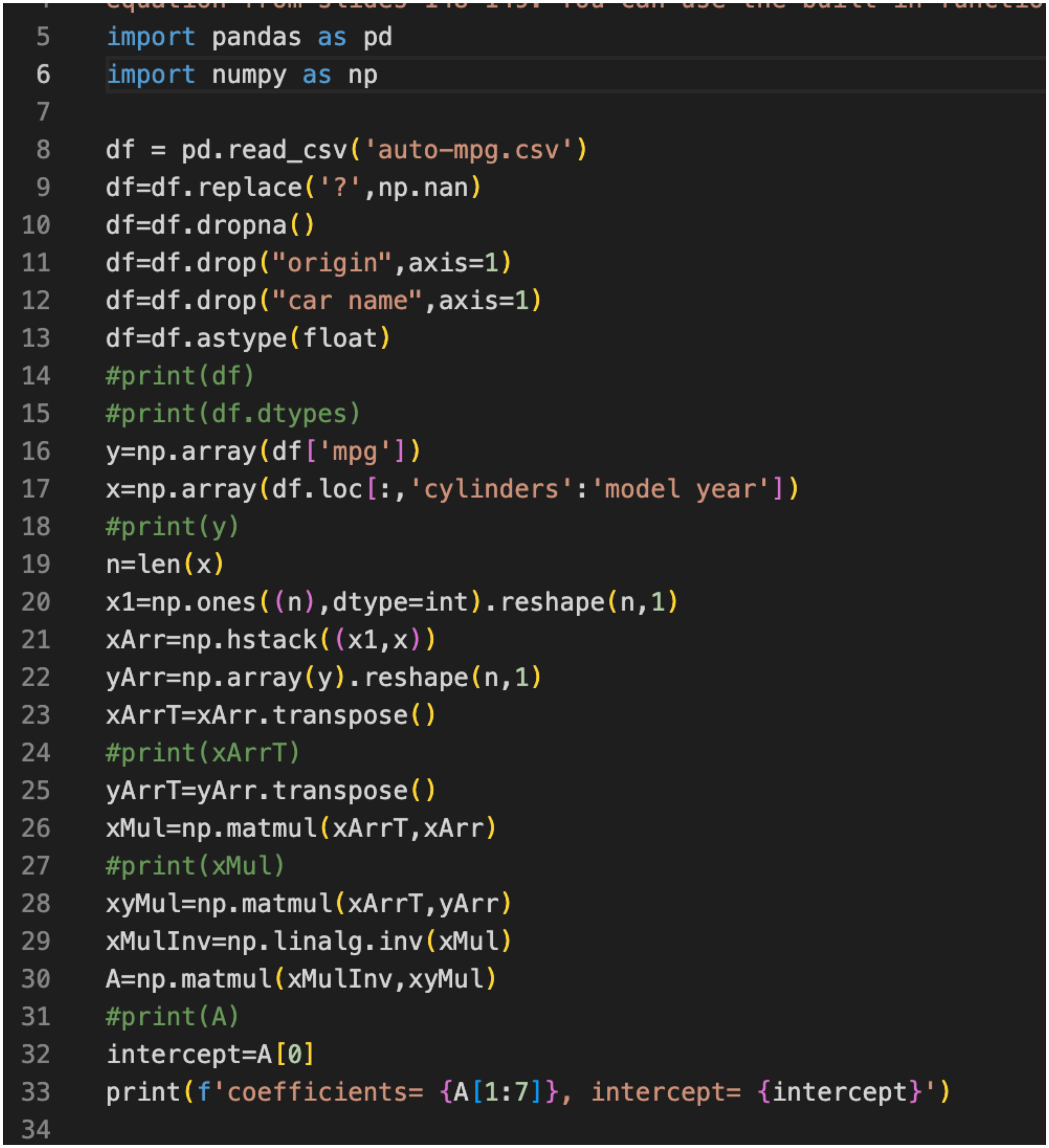
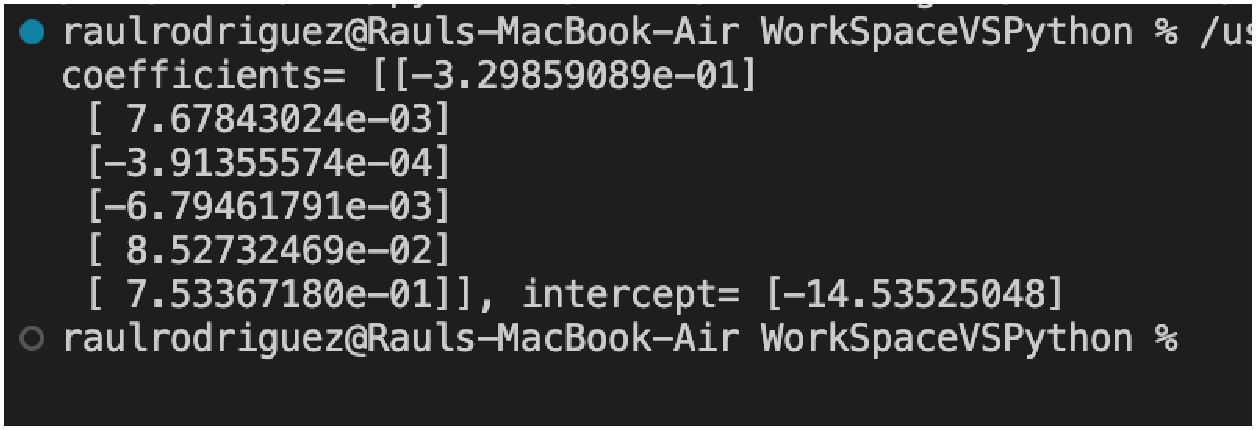
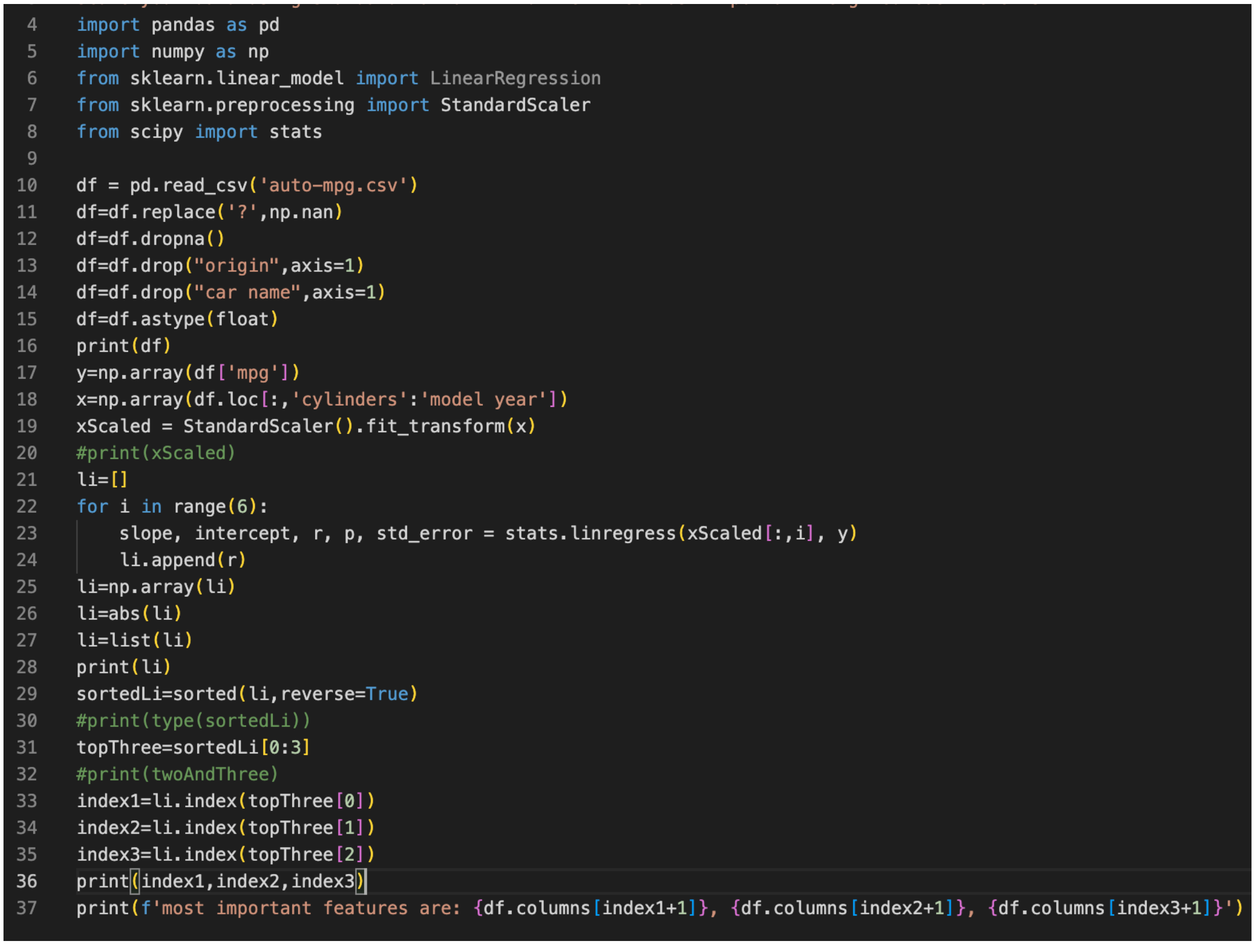


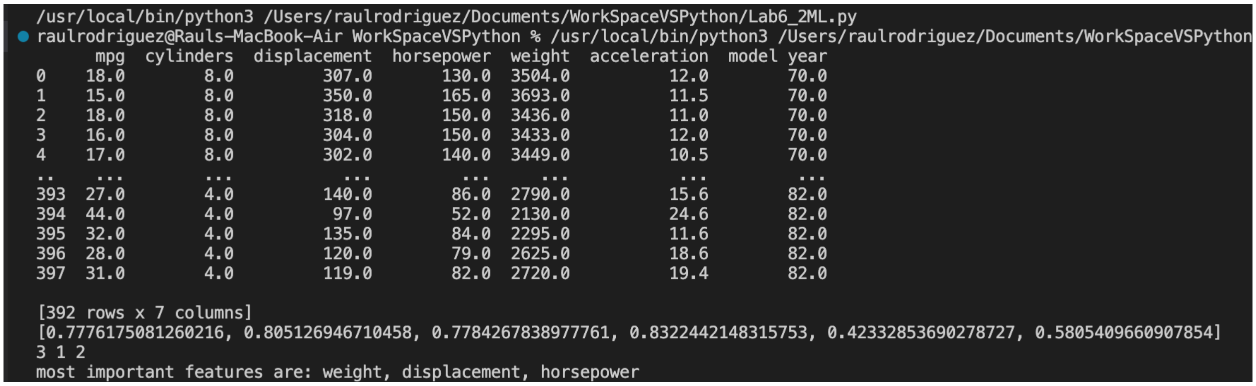
```python
equation from slides 140-145. You can use the built-in functio
import pandas as pd
import numpy as np

df = pd.read_csv('auto-mpg.csv')
df=df.replace('?',np.nan)
df=df.dropna()
df=df.drop("origin",axis=1)
df=df.drop("car name",axis=1)
df=df.astype(float)
#print(df)
#print(df.dtypes)
y=np.array(df['mpg'])
x=np.array(df.loc[:,'cylinders':'model year'])
#print(y)
n=len(x)
x1=np.ones((n),dtype=int).reshape(n,1)
xArr=np.hstack((x1,x))
yArr=np.array(y).reshape(n,1)
xArrT=xArr.transpose()
#print(xArrT)
yArrT=yArr.transpose()
xMul=np.matmul(xArrT,xArr)
#print(xMul)
xyMul=np.matmul(xArrT,yArr)
xMulInv=np.linalg.inv(xMul)
A=np.matmul(xMulInv,xyMul)
#print(A)
intercept=A[0]
print(f'coefficients= {A[1:7]}, intercept= {intercept}')

```



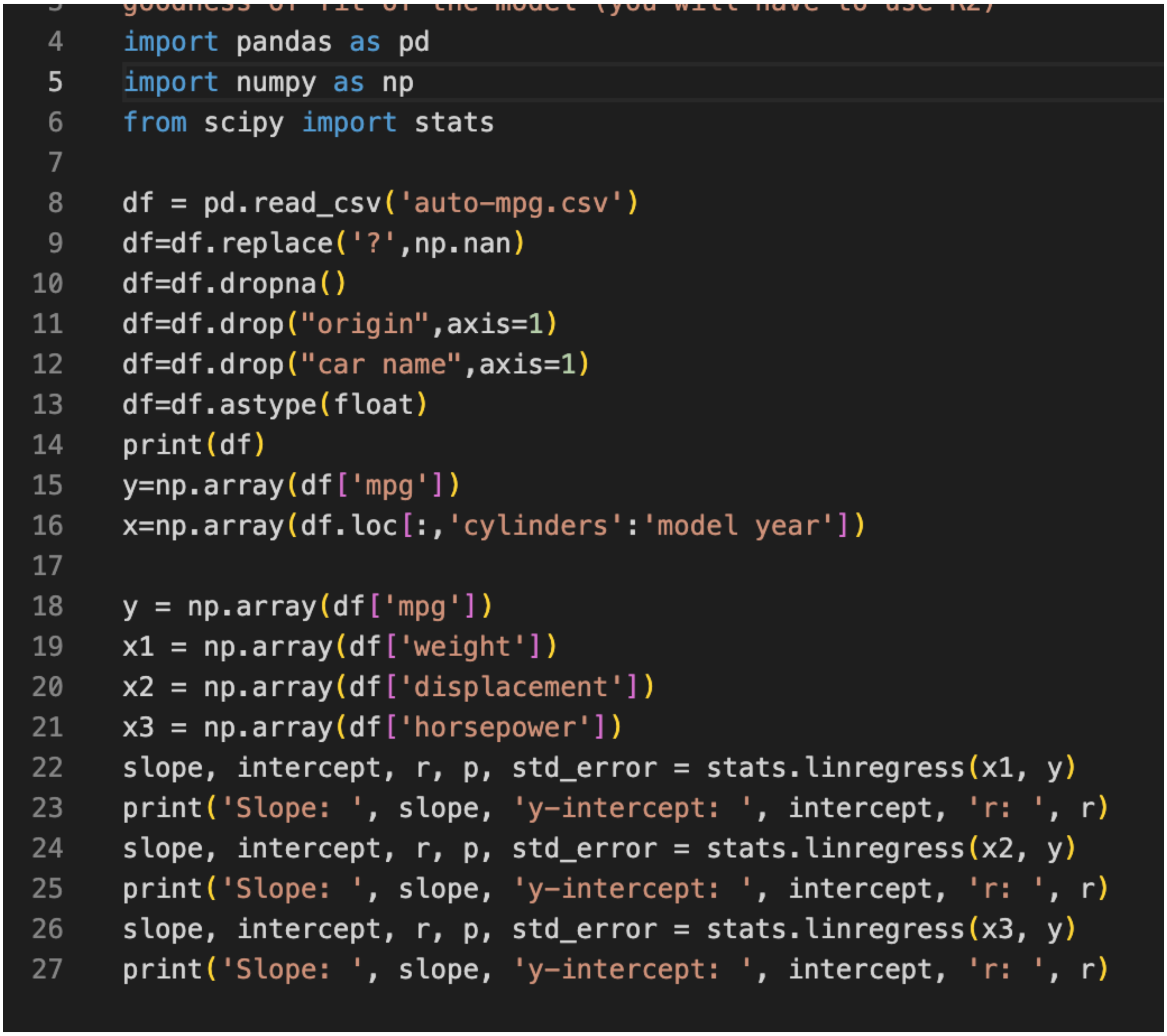
```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /us
coefficients= [[-3.29859089e-01]
 [ 7.67843024e-03]
 [-3.91355574e-04]
 [-6.79461791e-03]
 [ 8.52732469e-02]
 [ 7.53367180e-01]], intercept= [-14.53525048]
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```



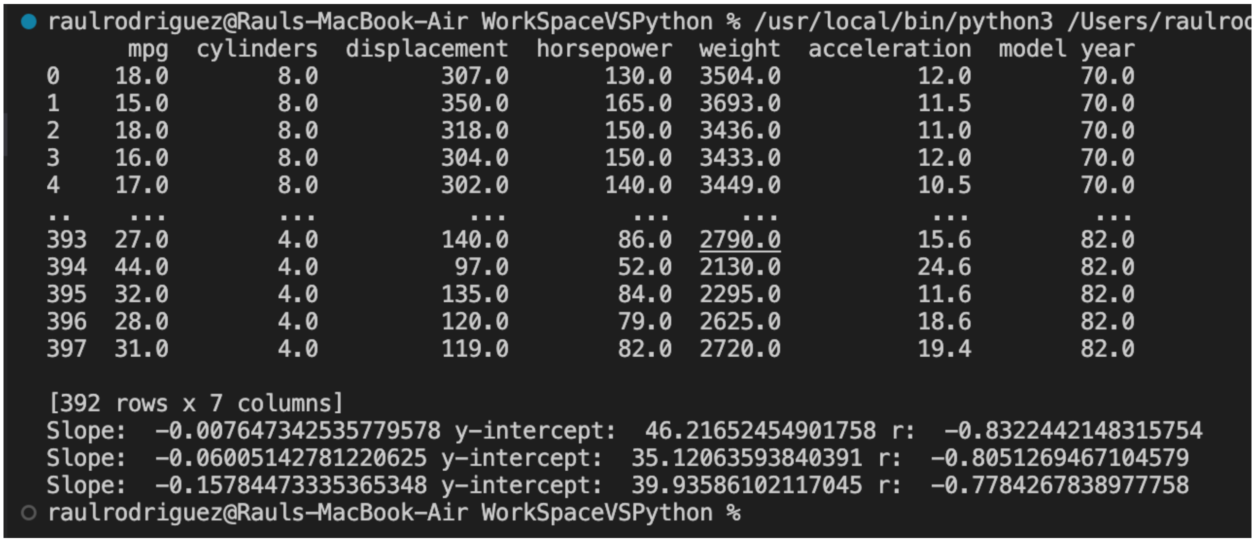
```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from scipy import stats

df = pd.read_csv('auto-mpg.csv')
df=df.replace('?',np.nan)
df=df.dropna()
df=df.drop("origin",axis=1)
df=df.drop("car name",axis=1)
df=df.astype(float)
print(df)
y=np.array(df['mpg'])
x=np.array(df.loc[:,'cylinders':'model year'])
xScaled = StandardScaler().fit_transform(x)
#print(xScaled)
li=[]
for i in range(6):
    slope, intercept, r, p, std_error = stats.linregress(xScaled[:,i], y)
    li.append(r)
li=np.array(li)
li=abs(li)
li=list(li)
print(li)
sortedLi=sorted(li,reverse=True)
#print(type(sortedLi))
topThree=sortedLi[0:3]
#print(twoAndThree)
index1=li.index(topThree[0])
index2=li.index(topThree[1])
index3=li.index(topThree[2])
print(index1,index2,index3)
print(f'most important features are: {df.columns[index1+1]}, {df.columns[index2+1]}, {df.columns[index3+1]}')
```



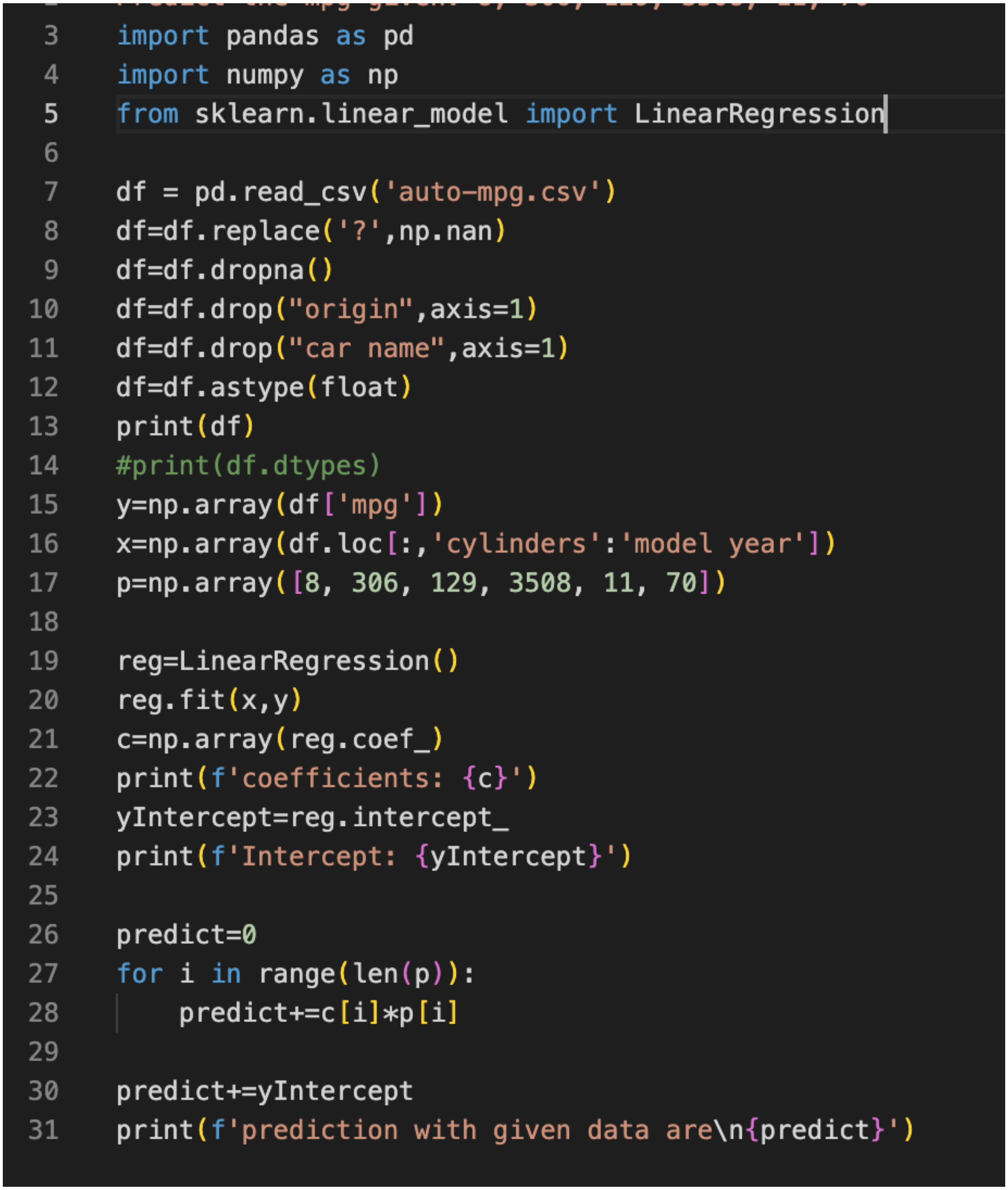
```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab6_2ML.py
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython
      mpg  cylinders  displacement  horsepower  weight  acceleration  model year
0    18.0        8.0         307.0       130.0  3504.0          12.0        70.0
1    15.0        8.0         350.0       165.0  3693.0          11.5        70.0
2    18.0        8.0         318.0       150.0  3436.0          11.0        70.0
3    16.0        8.0         304.0       150.0  3433.0          12.0        70.0
4    17.0        8.0         302.0       140.0  3449.0          10.5        70.0
..    ...        ...           ...         ...     ...           ...         ...
393  27.0        4.0         140.0        86.0  2790.0          15.6        82.0
394  44.0        4.0          97.0        52.0  2130.0          24.6        82.0
395  32.0        4.0         135.0        84.0  2295.0          11.6        82.0
396  28.0        4.0         120.0        79.0  2625.0          18.6        82.0
397  31.0        4.0         119.0        82.0  2720.0          19.4        82.0

[392 rows x 7 columns]
[0.7776175081260216, 0.805126946710458, 0.7784267838977761, 0.8322442148315753, 0.42332853690278727, 0.5805409660907854]
3 1 2
most important features are: weight, displacement, horsepower
```



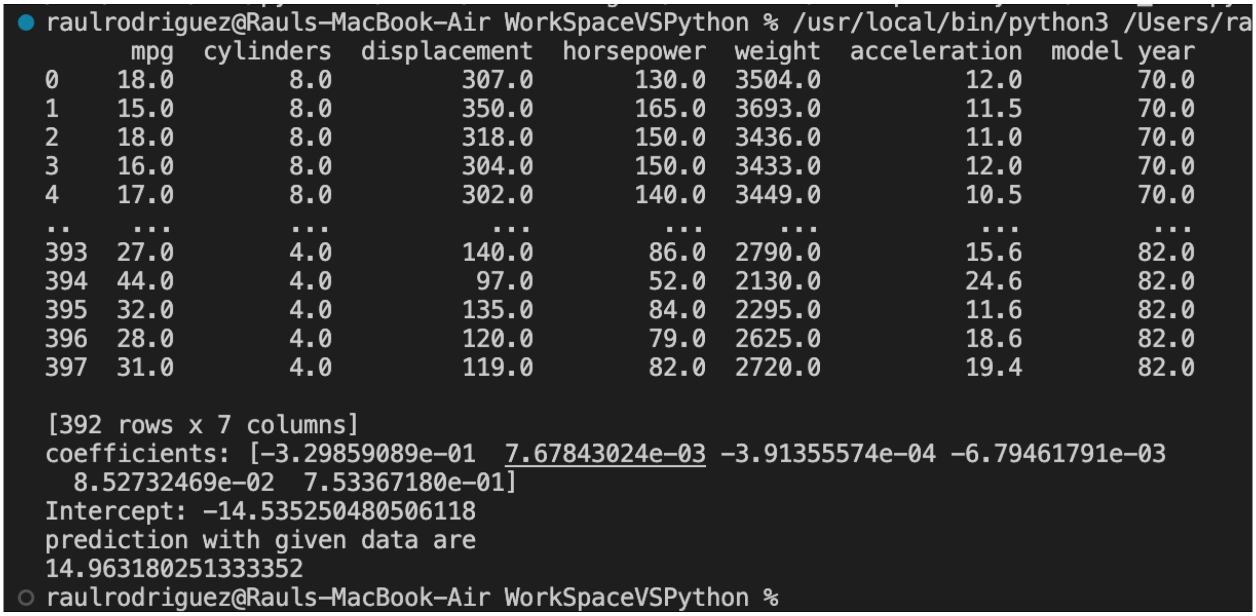
```python
goodness of fit of the model (you will have to use R2)
import pandas as pd
import numpy as np
from scipy import stats

df = pd.read_csv('auto-mpg.csv')
df=df.replace('?',np.nan)
df=df.dropna()
df=df.drop("origin",axis=1)
df=df.drop("car name",axis=1)
df=df.astype(float)
print(df)
y=np.array(df['mpg'])
x=np.array(df.loc[:,'cylinders':'model year'])

y = np.array(df['mpg'])
x1 = np.array(df['weight'])
x2 = np.array(df['displacement'])
x3 = np.array(df['horsepower'])
slope, intercept, r, p, std_error = stats.linregress(x1, y)
print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)
slope, intercept, r, p, std_error = stats.linregress(x2, y)
print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)
slope, intercept, r, p, std_error = stats.linregress(x3, y)
print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)
```



```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulroc
      mpg  cylinders  displacement  horsepower  weight  acceleration  model year
0    18.0        8.0         307.0       130.0  3504.0          12.0        70.0
1    15.0        8.0         350.0       165.0  3693.0          11.5        70.0
2    18.0        8.0         318.0       150.0  3436.0          11.0        70.0
3    16.0        8.0         304.0       150.0  3433.0          12.0        70.0
4    17.0        8.0         302.0       140.0  3449.0          10.5        70.0
..    ...        ...           ...         ...     ...           ...         ...
393  27.0        4.0         140.0        86.0  2790.0          15.6        82.0
394  44.0        4.0          97.0        52.0  2130.0          24.6        82.0
395  32.0        4.0         135.0        84.0  2295.0          11.6        82.0
396  28.0        4.0         120.0        79.0  2625.0          18.6        82.0
397  31.0        4.0         119.0        82.0  2720.0          19.4        82.0

[392 rows x 7 columns]
Slope:  -0.007647342535779578 y-intercept:  46.21652454901758 r:  -0.8322442148315754
Slope:  -0.06005142781220625 y-intercept:  35.12063593840391 r:  -0.8051269467104579
Slope:  -0.15784473335365348 y-intercept:  39.93586102117045 r:  -0.7784267838977758
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```



```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

df = pd.read_csv('auto-mpg.csv')
df=df.replace('?',np.nan)
df=df.dropna()
df=df.drop("origin",axis=1)
df=df.drop("car name",axis=1)
df=df.astype(float)
print(df)
#print(df.dtypes)
y=np.array(df['mpg'])
x=np.array(df.loc[:,'cylinders':'model year'])
p=np.array([8, 306, 129, 3508, 11, 70])

reg=LinearRegression()
reg.fit(x,y)
c=np.array(reg.coef_)
print(f'coefficients: {c}')
yIntercept=reg.intercept_
print(f'Intercept: {yIntercept}')

predict=0
for i in range(len(p)):
    predict+=c[i]*p[i]

predict+=yIntercept
print(f'prediction with given data are\n{predict}')
```



```
● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/ra
       mpg  cylinders  displacement  horsepower  weight  acceleration  model year
0     18.0        8.0         307.0       130.0  3504.0          12.0        70.0
1     15.0        8.0         350.0       165.0  3693.0          11.5        70.0
2     18.0        8.0         318.0       150.0  3436.0          11.0        70.0
3     16.0        8.0         304.0       150.0  3433.0          12.0        70.0
4     17.0        8.0         302.0       140.0  3449.0          10.5        70.0
..     ...        ...           ...         ...     ...           ...         ...
393   27.0        4.0         140.0        86.0  2790.0          15.6        82.0
394   44.0        4.0          97.0        52.0  2130.0          24.6        82.0
395   32.0        4.0         135.0        84.0  2295.0          11.6        82.0
396   28.0        4.0         120.0        79.0  2625.0          18.6        82.0
397   31.0        4.0         119.0        82.0  2720.0          19.4        82.0

[392 rows x 7 columns]
coefficients: [-3.29859089e-01  7.67843024e-03 -3.91355574e-04 -6.79461791e-03
  8.52732469e-02  7.53367180e-01]
Intercept: -14.535250480506118
prediction with given data are
14.963180251333352
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```