Step 1:

Homework 3: Linear Regression

Anthony Martinez | amm180005

Date: Feb 06 2021

Step 2: Write code to upload the Boston data. Use the data file in Piazza.

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Boston.csv

- **Boston.csv**(text/csv) - 5348 bytes, last modified: 1/31/2022 - 100% done

```
Saving Boston.csv to Boston.csv
```

```
import io
import pandas as pd

df = pd.read_csv(io.BytesIO(uploaded['Boston.csv']))
```

Step 3: Let X be the room and y be the median home value in 1000s.

```
# use double '[[]]' to keep rm as pandas data frame. Single '[]' gives us pandas serie
X = df['rm']
y = df['medv']
```

Step 4 Write a function to return the mean of a vector (column).

```
def find_mean(column):
  return column.mean()
```

Step 5: Write a function to calculate the coefficients using the formulas in the lecture slides.

```
def calculate_coefficients():
  '''
  This function calculates the intercept and coefficent of the X and y vectors.

  w = Summation from 1-n(xi - xmean)(yi-ymean) / summation from 1 - n(xi-xmean)^2
  b = ymean - (w*xmean)

  returns w and b

  use tuple to retrive both variables when function is called
  '''

  # get mean of x and y vectors
  x_mean = find_mean(X)
  y_mean = find_mean(y)


  numerator = 0
  denominator = 0
  # calculate w
  for i in range(0,len(X)):
    numerator += (X[i] - x_mean) * (y[i] - y_mean)
    denominator += (X[i] - x_mean) * (X[i] - x_mean)


  coefficent = numerator/denominator

  # use w, to calculate b (the intercept)
  intercept = (y_mean) - (coefficent*x_mean)

  return coefficent, intercept # will use a tuple to store both variables when calling
```

Step 6: Output coefficients b and w.

```
# code for step 6

result = calculate_coefficients() # results is a tuple where index 0 is the coefficent
```

```
print('Intercept: ', result[1])
print('Coefficent: ', result[0])
```

```
Intercept:  -34.67062077643857
Coefficent:  9.102108981180303
```

Step 7: Run linear regression in sklearn on all the data. See example in the GitHub.

```
# code for step 7
from sklearn.linear_model import LinearRegression

# use double to keep rm as pandas data frame. Single '[]' gives us pandas series
X = df[['rm']]
y = df['medv']

# run linear regression model
linreg = LinearRegression()
linreg.fit(X, y)

# make predictions
pred = linreg.predict(X)
```

```
## Step 8: Output the coefficients of the model.

print('Intercept ', linreg.intercept_)
print('Coefficient : ', linreg.coef_)
```

```
Intercept  -34.67062077643857
Coefficient :  [9.10210898]
```

Step 9: How similar are the coefficients?

The coefficients are exactly the same.

When calculating from scratch my code calculated an intercept of -34.670 and a coefficient of 9.102.

When using sklearn methods I got an Intercept of -34.670 and a coefficient of 9.102.

Step 10: Comment on any possible reasons for similarity or difference.

The similarity is most likely due to the fact that the calculations are fairly simple. The math boils down to computing the difference between an element and the mean of the elements in the vector and multiplying by some other number.

✓ 0s completed at 4:51 PM