Eunice Olorunshola

10/28/2021

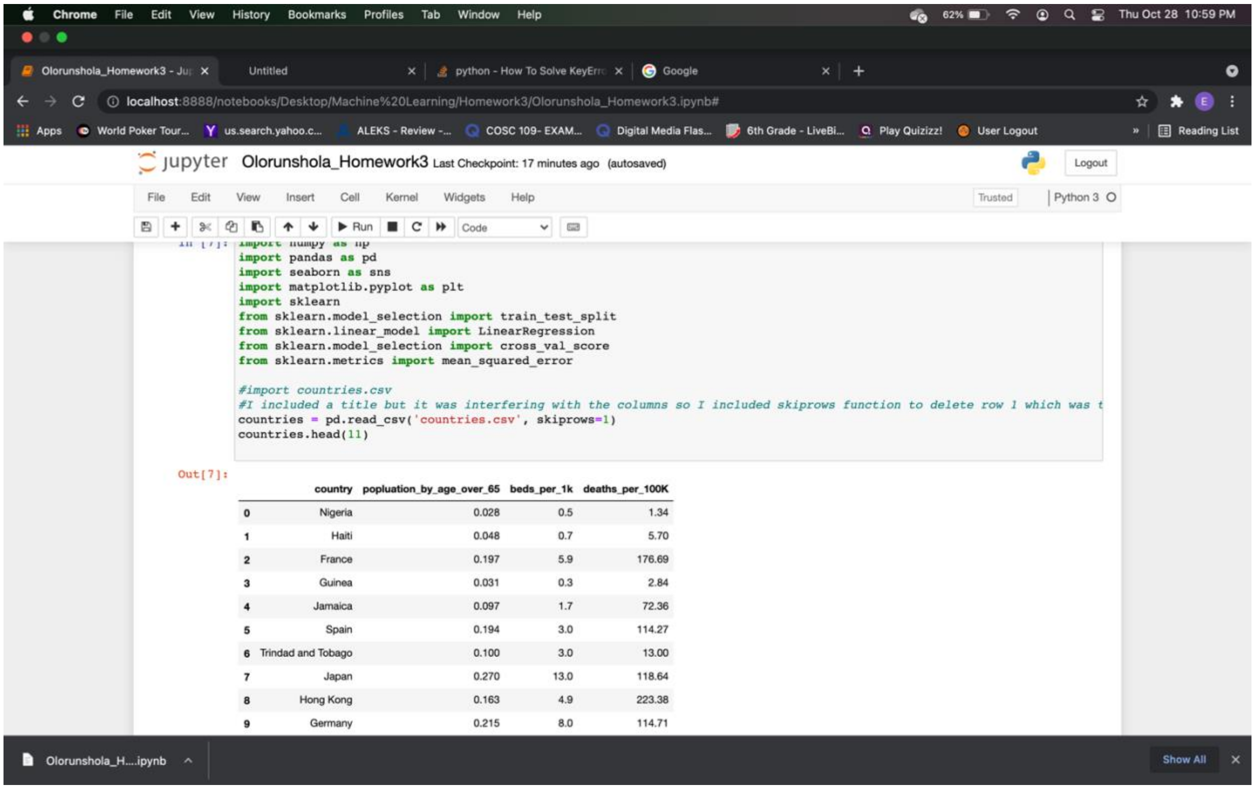
CSC 6850 Machine Learning

Report

Note: I used the link provided for implementing linear regression and also from k fold link on the same website to help with this homework

1) The way that I did this assisgnment is that first step I created my dataset using the 3 links provided and turned it into a csv file than from there I implemented it into my code so it could read the csv file and

Screenshot of the output :


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Oct 28 10:59 PM
Olorunshola_Homework3 - Ju
Untitled
python - How To Solve KeyErr
Google
localhost:8888/notebooks/Desktop/Machine%20Learning/Homework3/Olorunshola_Homework3.ipynb#
Apps World Poker Tour... us.search.yahoo.c... ALEKS - Review -... COSC 109- EXAM... Digital Media Flas... 6th Grade - LiveBi... Play Quizizz! User Logout
Reading List
jupyter Olorunshola_Homework3 Last Checkpoint: 17 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Olorunshola_H....ipynb
Show All

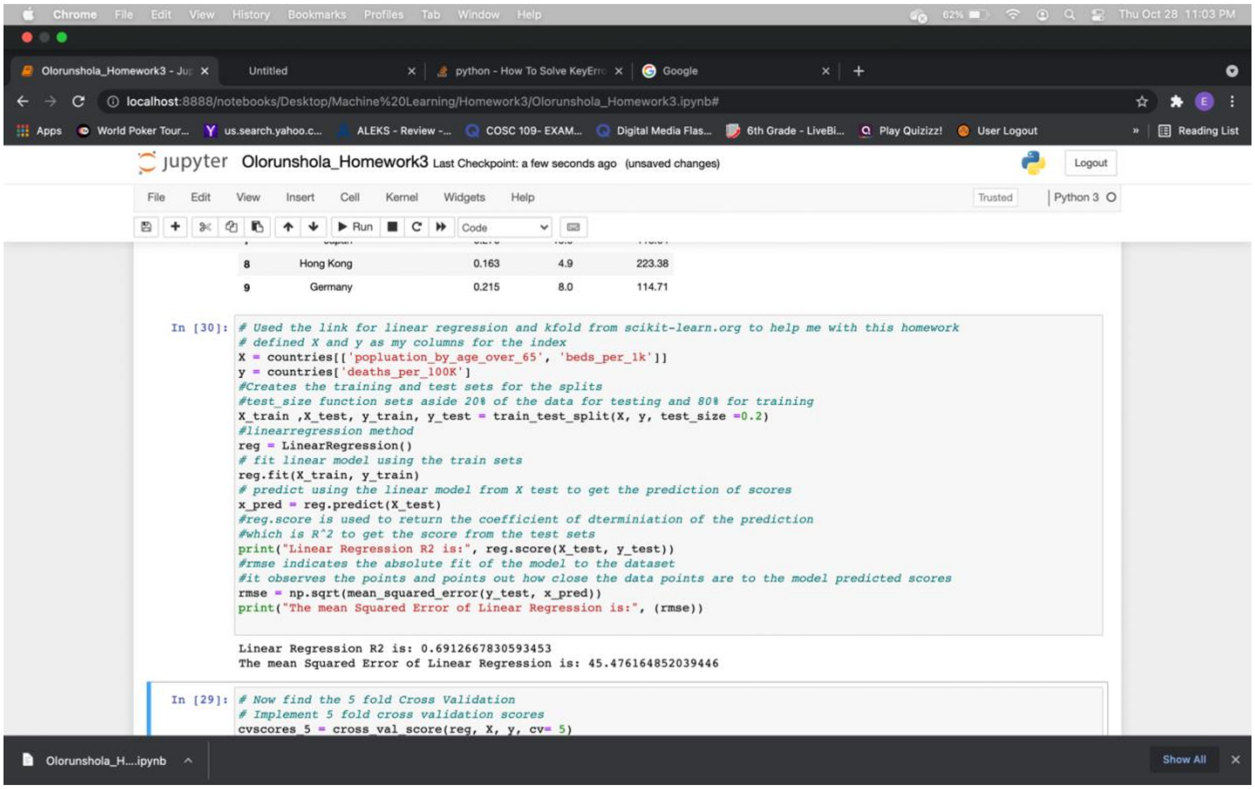
```
In [7]: import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        import sklearn
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from sklearn.model_selection import cross_val_score
        from sklearn.metrics import mean_squared_error

        #import countries.csv
        #I included a title but it was interfering with the columns so I included skiprows function to delete row 1 which was t
        countries = pd.read_csv('countries.csv', skiprows=1)
        countries.head(11)
```

Out[7]:

	country	popluation_by_age_over_65	beds_per_1k	deaths_per_100K
0	Nigeria	0.028	0.5	1.34
1	Haiti	0.048	0.7	5.70
2	France	0.197	5.9	176.69
3	Guinea	0.031	0.3	2.84
4	Jamaica	0.097	1.7	72.36
5	Spain	0.194	3.0	114.27
6	Trindad and Tobago	0.100	3.0	13.00
7	Japan	0.270	13.0	118.64
8	Hong Kong	0.163	4.9	223.38
9	Germany	0.215	8.0	114.71

2) Second step is that I used the link provided to implement the linear regression I also used the k fold link from the same website scikit-learn.org to help implement linear regression below is the output of the results from implementing linear regression

Screenshot of the output:

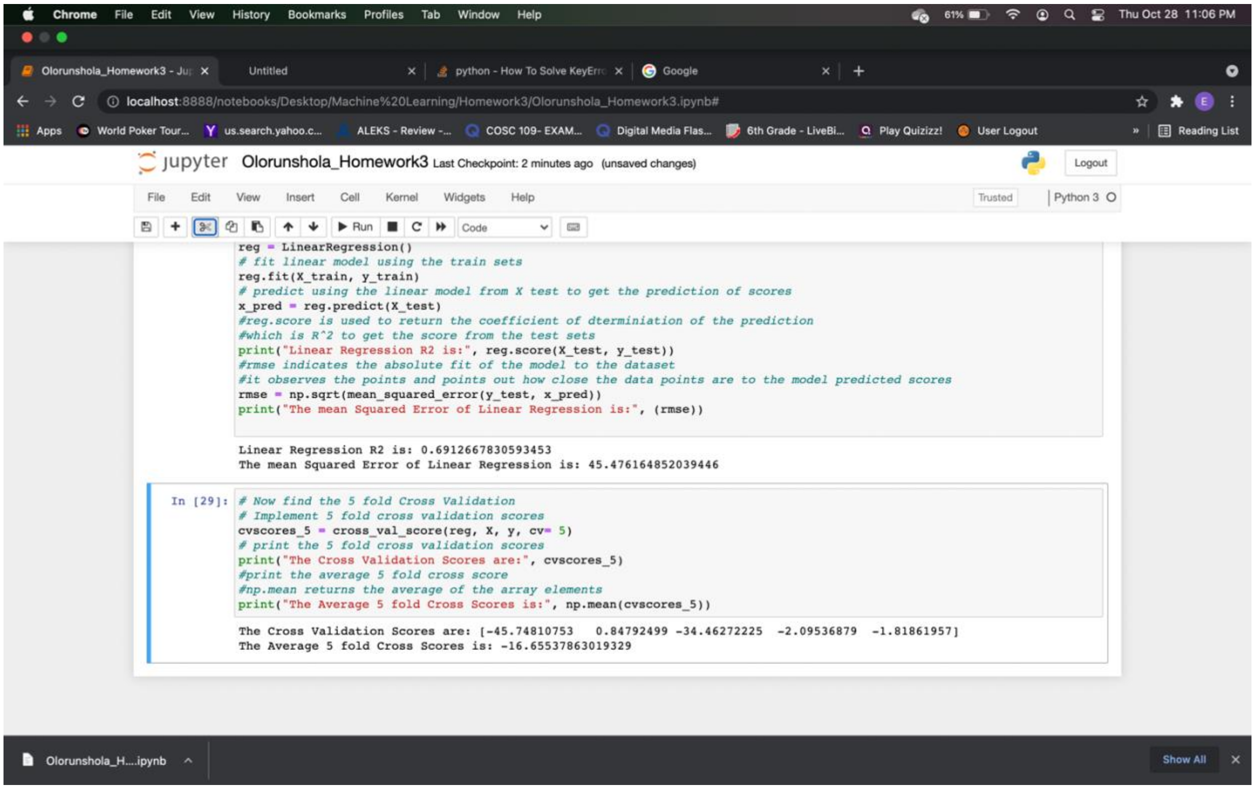

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Oct 28 11:03 PM
Olorunshola_Homework3 - Ju
Untitled
python - How To Solve KeyErr
Google
localhost:8888/notebooks/Desktop/Machine%20Learning/Homework3/Olorunshola_Homework3.ipynb#
Apps
World Poker Tour...
us.search.yahoo.c...
ALEKS - Review -...
COSC 109- EXAM...
Digital Media Flas...
6th Grade - LiveBi...
Play Quizizz!
User Logout
Reading List
jupyter Olorunshola_Homework3 Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Olorunshola_H....ipynb
Show All

8	Hong Kong	0.163	4.9	223.38
9	Germany	0.215	8.0	114.71

```
In [30]: # Used the link for linear regression and kfold from scikit-learn.org to help me with this homework
         # defined X and y as my columns for the index
         X = countries[['popluation_by_age_over_65', 'beds_per_1k']]
         y = countries['deaths_per_100K']
         #Creates the training and test sets for the splits
         #test_size function sets aside 20% of the data for testing and 80% for training
         X_train ,X_test, y_train, y_test = train_test_split(X, y, test_size =0.2)
         #linearregression method
         reg = LinearRegression()
         # fit linear model using the train sets
         reg.fit(X_train, y_train)
         # predict using the linear model from X test to get the prediction of scores
         x_pred = reg.predict(X_test)
         #reg.score is used to return the coefficient of dterminiation of the prediction
         #which is R^2 to get the score from the test sets
         print("Linear Regression R2 is:", reg.score(X_test, y_test))
         #rmse indicates the absolute fit of the model to the dataset
         #it observes the points and points out how close the data points are to the model predicted scores
         rmse = np.sqrt(mean_squared_error(y_test, x_pred))
         print("The mean Squared Error of Linear Regression is:", (rmse))

         Linear Regression R2 is: 0.6912667830593453
         The mean Squared Error of Linear Regression is: 45.476164852039446

In [29]: # Now find the 5 fold Cross Validation
         # Implement 5 fold cross validation scores
         cvscores_5 = cross_val_score(reg, X, y, cv= 5)
```

3) The last step is to implement 5 fold cross validation score from using the module cross_val_score from sk.learn I was able to implement the 5 scores from my dataset model correctly

Screenshot of output:


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Oct 28 11:06 PM
Olorunshola_Homework3 - Jup
Untitled
python - How To Solve KeyErr
Google
localhost:8888/notebooks/Desktop/Machine%20Learning/Homework3/Olorunshola_Homework3.ipynb#
jupyter Olorunshola_Homework3 Last Checkpoint: 2 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Olorunshola_H....ipynb
Show All

```
reg = LinearRegression()
# fit linear model using the train sets
reg.fit(X_train, y_train)
# predict using the linear model from X test to get the prediction of scores
x_pred = reg.predict(X_test)
#reg.score is used to return the coefficient of dterminiation of the prediction
#which is R^2 to get the score from the test sets
print("Linear Regression R2 is:", reg.score(X_test, y_test))
#rmse indicates the absolute fit of the model to the dataset
#it observes the points and points out how close the data points are to the model predicted scores
rmse = np.sqrt(mean_squared_error(y_test, x_pred))
print("The mean Squared Error of Linear Regression is:", (rmse))
```

```
Linear Regression R2 is: 0.6912667830593453
The mean Squared Error of Linear Regression is: 45.476164852039446
```

```
In [29]: # Now find the 5 fold Cross Validation
         # Implement 5 fold cross validation scores
         cvscores_5 = cross_val_score(reg, X, y, cv= 5)
         # print the 5 fold cross validation scores
         print("The Cross Validation Scores are:", cvscores_5)
         #print the average 5 fold cross score
         #np.mean returns the average of the array elements
         print("The Average 5 fold Cross Scores is:", np.mean(cvscores_5))
```

```
The Cross Validation Scores are: [-45.74810753   0.84792499 -34.46272225  -2.09536879  -1.81861957]
The Average 5 fold Cross Scores is: -16.65537863019329
```

Appendix section

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error

#import countries.csv
#I included a title but it was interfering with the columns so I included skiprows function to delete row 1 which was the title
countries = pd.read_csv('countries.csv', skiprows=1)
countries.head(11)


# Used the link for linear regression and kfold from scikit-learn.org to help me with this homework
# defined X and y as my columns for the index
X = countries[['popluation_by_age_over_65', 'beds_per_1k']]
y = countries['deaths_per_100K']
#Creates the training and test sets for the splits
#test_size function sets aside 20% of the data for testing and 80% for training
X_train ,X_test, y_train, y_test = train_test_split(X, y, test_size =0.2)
#linearregression method
reg = LinearRegression()
# fit linear model using the train sets
reg.fit(X_train, y_train)
# predict using the linear model from X test to get the prediction of scores
x_pred = reg.predict(X_test)
#reg.score is used to return the coefficient of dterminiation of the prediction
#which is R^2 to get the score from the test sets
print("Linear Regression R2 is:", reg.score(X_test, y_test))
#rmse indicates the absolute fit of the model to the dataset
#it observes the points and points out how close the data points are to the model predicted scores
rmse = np.sqrt(mean_squared_error(y_test, x_pred))
print("The mean Squared Error of Linear Regression is:", (rmse))
```

```
#Now find the 5 fold Cross Validation
# Implement 5 fold cross validation scores
cvscores_5 = cross_val_score(reg, X, y, cv= 5)
# print the 5 fold cross validation scores
print("The Cross Validation Scores are:", cvscores_5)
#print the average 5 fold cross score
#np.mean returns the average of the array elements
print("The Average 5 fold Cross Scores is:", np.mean(cvscores_5))
```