Sardar Patel Institute of Technology,Mumbai
Department of Electronics and Telecommunication Engineering
B.E. Sem-VII (2021-2022)
Data Analytics

Experiment: Exploratory Data Analysis (EDA)

Name: Sneha Ghuge **UID:** 2019110015 **BE ETRX** **DA LAB 2**

Aim: Building Linear Regression model for given dataset.

Problem Statement :

Problem 1.1 - Creating Our First Model

We are interested in how changes in these variables affect future temperatures, as well as how well these variables explain temperature changes so far. To do this, first read the dataset climate_change.csv into Python.
Then, split the data into a training set, consisting of all the observations up to and including 2006, and a testing set consisting of the remaining years (hint: use subset). A training set refers to the data that will be used to build the model, and a testing set refers to the data we will use to test our predictive ability.
Next, build a linear regression model to predict the dependent variable Temp, using MEI, CO2, CH4, N2O, CFC.11, CFC.12, TSI, and Aerosols as independent variables (Year and Month should NOT be used in the model). Use the training set to build the model.
Enter the model R2 (the "Multiple R-squared" value):

Problem 1.2 - Creating Our First Model

Which variables are significant in the model? We will consider a variable significant only if the p-value is below 0.05. (Select all that apply.)
a) MEI b) CO2 c) CH4 d) N2O e) CFC.11 f) CFC.12 g) TSI h) Aerosols

Problem 2.1 - Understanding the Model
Current scientific opinion is that nitrous oxide and CFC-11 are greenhouse gasses:
Gasses that are able to trap heat from the sun and contribute to the heating of the Earth.
However, the regression coefficients of both the N2O and CFC-11 variables are negative, indicating that increasing atmospheric concentrations of either of these two compounds is associated with lower global temperatures.

Exercise 3
Which of the following is the simplest correct explanation for this contradiction?
1. Climate scientists are wrong that N2O and CFC-11 are greenhouse gasses - this regression analysis constitutes part of a disproof.
2. There is not enough data, so the regression coefficients being estimated are not accurate.
3. All of the gas concentration variables reflect human development - N2O and CFC.11 are correlated with other variables in the data set.

CODE & OUTPUT:

```
from sklearn import linear_model
import numpy as np
import pandas as pd
from sklearn.metrics import r2_score
```

```
data = pd.read_csv('climate_change.csv')
```

```
data.head()
```

	Year	Month	MEI	CO2	CH4	N2O	CFC11	CFC12	TSI	Aerosols	Temp
0	1983	5	2.556	345.96	1638.59	303.677	191.324	350.113	1366.1024	0.0863	0.109
1	1983	6	2.167	345.52	1633.71	303.746	192.057	351.848	1366.1208	0.0794	0.118
2	1983	7	1.741	344.15	1633.22	303.795	192.818	353.725	1366.2850	0.0731	0.137
3	1983	8	1.130	342.25	1631.35	303.839	193.602	355.633	1366.4202	0.0673	0.176
4	1983	9	0.428	340.17	1648.40	303.901	194.392	357.465	1366.2335	0.0619	0.149

```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Year	308.0	1995.662338	7.423197	1983.0000	1989.00000	1996.00000	2002.00000	2008.0000
Month	308.0	6.551948	3.447214	1.0000	4.00000	7.00000	10.00000	12.0000
MEI	308.0	0.275555	0.937918	-1.6350	-0.39875	0.23750	0.83050	3.0010
CO2	308.0	363.226753	12.647125	340.1700	353.02000	361.73500	373.45500	388.5000
CH4	308.0	1749.824513	46.051678	1629.8900	1722.18250	1764.04000	1786.88500	1814.1800
N2O	308.0	312.391834	5.225131	303.6770	308.11150	311.50700	316.97900	322.1820
CFC11	308.0	251.973068	20.231783	191.3240	246.29550	258.34400	267.03100	271.4940
CFC12	308.0	497.524782	57.826899	350.1130	472.41075	528.35600	540.52425	543.8130
TSI	308.0	1366.070759	0.399610	1365.4261	1365.71705	1365.98090	1366.36325	1367.3162
Aerosols	308.0	0.016657	0.029050	0.0016	0.00280	0.00575	0.01260	0.1494
Temp	308.0	0.256776	0.179090	-0.2820	0.12175	0.24800	0.40725	0.7390

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 308 entries, 0 to 307
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Year      308 non-null    int64  
 1   Month     308 non-null    int64  
 2   MEI       308 non-null    float64
 3   CO2       308 non-null    float64
 4   CH4       308 non-null    float64
 5   N2O       308 non-null    float64
 6   CFC11     308 non-null    float64
 7   CFC12     308 non-null    float64
 8   TSI       308 non-null    float64
 9   Aerosols  308 non-null    float64
 10  Temp      308 non-null    float64
dtypes: float64(9), int64(2)
memory usage: 26.6 KB
```

```
from sklearn.model_selection import train_test_split
X = data.iloc[:, :-1]
y = data.iloc[:, -1]
# split the dataset
X_train = X[X['Year']<=2006]
X_test = X[X['Year']>2006]
y_train = y[:len(X_train)]
y_test = y[len(X_train):]
print(len(X_train))
print(len(y_train))
print(len(X_test))
print(len(y_test))

# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

```
284
284
24
24
```

```
X_train = X_train.iloc[:,2:]
X_test = X_test.iloc[:,2:]
```

```
X_test["MEI"][284]
```

```
0.974
```

```
print(data.isnull().any())
```

```
Year        False
Month       False
MEI         False
CO2         False
CH4         False
N2O         False
CFC11       False
CFC12       False
TSI         False
Aerosols    False
Temp        False
dtype: bool
```

Now let's plot Linear Regression Models for all attributes.

```
X = X_train[["MEI","CH4","CO2","N2O","CFC11","CFC12","TSI","Aerosols"]]
y = y_train

reg = linear_model.LinearRegression()
reg.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = reg.predict([[X_test["MEI"][284],X_test["CH4"][284],X_test["CO2"][284],X_test["N2O"][284],X_test["CFC11"][284],X_
print("Coefficients for all attributes : ",reg.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [ 6.42053134e-02  1.24041896e-04  6.45735927e-03 -1.65280033e-02
 -6.63048889e-03  3.80810324e-03  9.31410835e-02 -1.53761324e+00]
Predicted temprature :  [0.46860242]
Actual temprature :  0.601
```

R2 score (Coefficient of determination) is used to evaluate the performance of a linear regression model. It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable(s).

It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

```
# testing our model on the test set
f = reg.predict(X_test)
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The feature names should match th
assed during fit. Starting version 1.2, an error will be raised.
Feature names must be in the same order as they were in fit.

  warnings.warn(message, FutureWarning)
```

```
array([9.44120315, 9.43346922, 9.41187035, 9.39730901, 9.37016122,
       9.25378577, 9.19414027, 9.23226039, 9.27636682, 9.32934965,
       9.3260373 , 9.33683534, 9.37565077, 9.31111758, 9.2240278 ,
       9.27103714, 9.3334963 , 9.33048801, 9.26705955, 9.22913469,
       9.30122542, 9.41148048, 9.40805734, 9.4053746 ])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -7210.7701985696585
```

Model : MEI vs Temperature

```
X = X_train[["MEI"]]
y = y_train
regrMEI = linear_model.LinearRegression()
regrMEI.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrMEI.predict([[X_test["MEI"][284]]])
print("Coefficients for all attributes : ",regrMEI.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.03360508]
Predicted temprature :  [0.26904031]
Actual temprature :  0.601
```

```
f = regrMEI.predict(np.array(X_test["MEI"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feat
ression was fitted with feature names
  warnings.warn(
```

```
array([0.26904031, 0.25344755, 0.23879574, 0.23466231, 0.24245869,
       0.22427834, 0.22656349, 0.22152272, 0.19725985, 0.19793196,
       0.19675578, 0.19705822, 0.20233422, 0.18919463, 0.18136465,
       0.20465297, 0.22437916, 0.24061041, 0.23640978, 0.22737001,
       0.21470089, 0.210097  , 0.2154402 , 0.21392798])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -1.662215044104026
```

Model : CH4 vs Temperature

```
X = X_train[["CH4"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["CH4"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.00278925]
Predicted temprature :  [0.39791246]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["CH4"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid featu
ression was fitted with feature names
  warnings.warn(
```

```
array([0.39791246, 0.40745168, 0.40750747, 0.40474611, 0.38672758,
       0.34812439, 0.32045506, 0.34134652, 0.38271106, 0.40549921,
       0.40943205, 0.4144248 , 0.42653013, 0.4084837 , 0.37888979,
       0.37813669, 0.38890319, 0.37598897, 0.35124835, 0.34274115,
       0.3851377 , 0.43841233, 0.43336379, 0.43478631])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -0.2606461625131895
```

Model : CO2 vs Temperature

```
X = X_train[["CO2"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["CO2"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.01248554]
Predicted temprature :  [0.51643483]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["CO2"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have 
ression was fitted with feature names
  warnings.warn(
```

```
array([0.51643483, 0.52742211, 0.53678626, 0.55975964, 0.56200704,
       0.55538971, 0.53591227, 0.50482329, 0.4910892 , 0.49408572,
       0.51006721, 0.52842095, 0.54777353, 0.55139433, 0.55439086,
       0.56924865, 0.58597927, 0.57823824, 0.56000935, 0.53166719,
       0.51843252, 0.51718397, 0.53141748, 0.54927179])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -2.9256224363623864
```

Model : N2O vs Temperature

```
X = X_train[["N2O"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["N2O"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.02963935]
Predicted temprature :  [0.51170135]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["N2O"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid featu
ression was fitted with feature names
  warnings.warn(
```

```
array([0.51170135, 0.51199774, 0.51131604, 0.51042686, 0.50826319,
       0.50491394, 0.50541781, 0.50903381, 0.51339079, 0.52041532,
       0.52655066, 0.53114476, 0.53443473, 0.5349386 , 0.53345663,
       0.53520535, 0.53716155, 0.53796181, 0.53573886, 0.53671696,
       0.54039224, 0.54830594, 0.55473768, 0.55974673])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -2.5408746886236218
```

Model : CFC11 vs Temperature

```
X = X_train[["CFC11"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["CFC11"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.00351877]
Predicted temprature :  [0.23331923]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["CFC11"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not hav
ression was fitted with feature names
  warnings.warn(
```

```
array([0.23331923, 0.2329392 , 0.23199969, 0.23051125, 0.22927968,
       0.22807274, 0.22672153, 0.22605296, 0.22572572, 0.22563775,
       0.22559904, 0.2258911 , 0.22561663, 0.22461378, 0.222967  ,
       0.22175654, 0.22115131, 0.22031385, 0.2194623 , 0.21863891,
       0.21822721, 0.21821666, 0.21872688, 0.21865298])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -1.6945376446857718
```

Model : CFC12 vs Temperature

```
X = X_train[["CFC12"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["CFC12"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.0021092]
Predicted temprature :  [0.3426889]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["CFC12"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid fe
ression was fitted with feature names
  warnings.warn(
```

```
array([0.3426889 , 0.34219746, 0.34185577, 0.3413812 , 0.3404194 ,
       0.33882907, 0.33827435, 0.33829966, 0.33862869, 0.3388375 ,
       0.33870884, 0.33814569, 0.33777447, 0.33694766, 0.33588252,
       0.33518437, 0.33465918, 0.33408759, 0.33387245, 0.33396947,
       0.33391885, 0.33366364, 0.33361934, 0.33382816])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -0.042784892653006557
```

Model : TSI vs Temperature

```
X = X_train[["TSI"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["TSI"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [0.10986083]
Predicted temprature :  [0.20559769]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["TSI"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have v
ression was fitted with feature names
  warnings.warn(
```

```
array([0.20559769, 0.20529008, 0.20967353, 0.20620192, 0.20295004,
       0.21046452, 0.20925605, 0.20991522, 0.20544388, 0.2079597 ,
       0.20018155, 0.20289511, 0.20548783, 0.207718  , 0.20068691,
       0.20530106, 0.20561966, 0.20073085, 0.20062099, 0.19897308,
       0.19981901, 0.20104945, 0.20441119, 0.20288413])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
print('r2 score for this model is : ', r2)
```

```
r2 score for this model is :  -2.25358810624701
```

Model : Aerosols vs Temperature

```
X = X_train[["Aerosols"]]
y = y_train
regrCH4 = linear_model.LinearRegression()
regrCH4.fit(X, y)
```

```
LinearRegression()
```

```
#predict the temperature
predictedTemp = regrCH4.predict([[X_test["Aerosols"][284]]])
print("Coefficients for all attributes : ",regrCH4.coef_)
print("Predicted temprature : ",predictedTemp)
print("Actual temprature : ",y_test[284])
```

```
Coefficients for all attributes :  [-2.32296391]
Predicted temprature :  [0.27642001]
Actual temprature :  0.601
```

```
f = regrCH4.predict(np.array(X_test["Aerosols"]).reshape(-1, 1))
f
```

```
C:\Users\91744\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid fea
ression was fitted with feature names
  warnings.warn(
```

```
array([0.27642001, 0.2771169 , 0.27851068, 0.27851068, 0.27943986,
       0.27967216, 0.27967216, 0.27943986, 0.27920757, 0.27943986,
       0.27920757, 0.27967216, 0.28013675, 0.28060135, 0.28106594,
       0.28129823, 0.28176283, 0.28176283, 0.28129823, 0.28060135,
       0.27897527, 0.27827838, 0.27781379, 0.27827838])
```

```
### Assume y_test is the actual value and f is the predicted values
r2 = r2_score(y_test, f)
```

R2 Scores:

MEI vs Temperature R2 score --1.662215044104026

CH4 vs Temperature R2 score - -0.2606461625131886

CO2 vs Temperature R2 score - -2.92562243636237

N2O vs Temperature R2 score - -2.5408746886236506

CFC11 vs Temperature R2 score - -1.6945376446857718

CFC12 vs Temperature R2 score - -0.042784892653006557

TSI vs Temperature R2 score - -2.253588106246505

Aerosols vs Temperature R2 score - -0.6397171917625124

Here after calculating the individual R2 scores for all the attributes we can see that none of them individually provide a very good estimation of temperature, hence we will now check the correlation of the attributes with temperature to find the ones that contribute to the significant changes in temperature.

Correlation Analysis:

```
from scipy import stats

CATEGORICAL_VARIABLES = ["MEI",
                         "CH4",
                         "CO2",
                         "N2O",
                         "CFC11",
                         "CFC12",
                         "TSI",
                         "Aerosols"]
for c in CATEGORICAL_VARIABLES:
  correlation = stats.pointbiserialr(data[c], data["Temp"])
  print("Correlation of %s to temp is %s" %(c, correlation))
```

```
Correlation of MEI to temp is PointbiserialrResult(correlation=0.13529168433351063, pvalue=0.017518659805993528)
Correlation of CH4 to temp is PointbiserialrResult(correlation=0.6996965803638928, pvalue=1.3362989047670364e-46)
Correlation of CO2 to temp is PointbiserialrResult(correlation=0.7485046457380211, pvalue=1.557880415620173e-56)
Correlation of N2O to temp is PointbiserialrResult(correlation=0.7432418337360966, pvalue=2.3517474412415498e-55)
Correlation of CFC11 to temp is PointbiserialrResult(correlation=0.3801113416532199, pvalue=5.031362179050227e-12)
Correlation of CFC12 to temp is PointbiserialrResult(correlation=0.6889441088656743, pvalue=1.1179206572557341e-44)
Correlation of TSI to temp is PointbiserialrResult(correlation=0.18218560682875687, pvalue=0.0013215404069405788)
Correlation of Aerosols to temp is PointbiserialrResult(correlation=-0.392069446275214, pvalue=9.283071094401667e-13)
```

The correlation analysis of all the attributes with respect to temperature is:

MEI to temperature : correlation=0.13529168433351063, pvalue=0.017518659805993528
CH4 to temperature : correlation=0.6996965803638928, pvalue=1.3362989047670364e-46
CO2 to temperature : correlation=0.7485046457380211, pvalue=1.557880415620173e-56
N2O to temperature: correlation=0.7432418337360966, pvalue=2.3517474412415498e-55
CFC11 to temperature : correlation=0.3801113416532199, pvalue=5.031362179050227e-12
CFC12 to temperature : correlation=0.6889441088656743, pvalue=1.1179206572557341e-44
TSI to temperature : correlation=0.18218560682875687, pvalue=0.0013215404069405788
Aerosols to temperature : correlation=-0.392069446275214, pvalue=9.283071094401667e-13

Conclusion:

After calculating the R2 scores for all the attributes we can see that none of them individually provide a very good estimation of temperature, so we checked the Correlation between the attributes.