Assignment 2 (A)

```
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
import numpy as np

# --------------------
# Code cell separator
# --------------------

df= pd.read_csv(r'C:\Users\rohit\OneDrive\Desktop\Sem 5\3.End_Sem
ML\temperatures.csv')

# --------------------
# Code cell separator
# --------------------

df.head()

# --------------------
# Code cell separator
# --------------------

df.tail()

# --------------------
# Code cell separator
# --------------------

df.isnull().sum()

# --------------------
# Code cell separator
# --------------------

from sklearn.model_selection import train_test_split

# --------------------
# Code cell separator
# --------------------

x = df[['YEAR']]
y = df[['JAN']]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)

# --------------------
```

```
# Code cell separator
# --------------------

from sklearn.linear_model import LinearRegression

# --------------------
# Code cell separator
# --------------------

xtrain.shape
ytrain.shape

# --------------------
# Code cell separator
# --------------------

model = LinearRegression()
model.fit(xtrain, ytrain)

# --------------------
# Code cell separator
# --------------------

print(f"intercept: {model.intercept_}")

# --------------------
# Code cell separator
# --------------------

print(f"slope: {model.coef_}")

# --------------------
# Code cell separator
# --------------------

prediction=model.predict(xtest)

# --------------------
# Code cell separator
# --------------------

from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(xtest, prediction))
print("Mean Squared Error ", metrics.mean_squared_error(xtest, prediction))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(xtest, prediction)))

# --------------------
# Code cell separator
```

```
# --------------------

import matplotlib.pyplot as mpl
mpl.scatter(x, y, color = 'g')
mpl.plot(x, model.predict(x), color= 'k')
mpl.show()

# --------------------
# Code cell separator
# --------------------

x = df[['YEAR']]
y = df[['MAR']]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)

# --------------------
# Code cell separator
# --------------------

xtrain.shape
ytrain.shape

# --------------------
# Code cell separator
# --------------------

model = LinearRegression()
model.fit(xtrain, ytrain)

# --------------------
# Code cell separator
# --------------------

print(f"intercept: {model.intercept_}")

# --------------------
# Code cell separator
# --------------------

print(f"slope: {model.coef_}")

# --------------------
# Code cell separator
# --------------------

prediction1=model.predict(xtest)

# --------------------
```

```
# Code cell separator
# --------------------

from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(xtest, prediction1))
print("Mean Squared Error ", metrics.mean_squared_error(xtest, prediction1))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(xtest, prediction1)))

# --------------------
# Code cell separator
# --------------------

import matplotlib.pyplot as mpl
mpl.scatter(x, y, color = 'g')
mpl.plot(x, model.predict(x), color= 'k')
mpl.show()

# --------------------
# Code cell separator
# --------------------

x = df[['YEAR']]
y = df[['APR']]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)

# --------------------
# Code cell separator
# --------------------

xtrain.shape
ytrain.shape

# --------------------
# Code cell separator
# --------------------

model = LinearRegression()
model.fit(xtrain, ytrain)

# --------------------
# Code cell separator
# --------------------

print(f"intercept: {model.intercept_}")

# --------------------
# Code cell separator
# --------------------
```

```
print(f"slope: {model.coef_}")

# --------------------
# Code cell separator
# --------------------

prediction2=model.predict(xtest)

# --------------------
# Code cell separator
# --------------------

from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(xtest, prediction2))
print("Mean Squared Error ", metrics.mean_squared_error(xtest, prediction2))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(xtest, prediction2)))

# --------------------
# Code cell separator
# --------------------

import matplotlib.pyplot as mpl
mpl.scatter(x, y, color = 'g')
mpl.plot(x, model.predict(x), color= 'k')
mpl.show()

# --------------------
# Code cell separator
# --------------------

x = df[['YEAR']]
y = df[['JUN']]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)

# --------------------
# Code cell separator
# --------------------

xtrain.shape
ytrain.shape

# --------------------
# Code cell separator
# --------------------

model = LinearRegression()
model.fit(xtrain, ytrain)
```

```
# --------------------
# Code cell separator
# --------------------

print(f"intercept: {model.intercept_}")

# --------------------
# Code cell separator
# --------------------

print(f"slope: {model.coef_}")

# --------------------
# Code cell separator
# --------------------

prediction3=model.predict(xtest)

# --------------------
# Code cell separator
# --------------------

from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(xtest, prediction3))
print("Mean Squared Error ", metrics.mean_squared_error(xtest, prediction3))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(xtest, prediction3)))

# --------------------
# Code cell separator
# --------------------

import matplotlib.pyplot as mpl
mpl.scatter(x, y, color = 'g')
mpl.plot(x, model.predict(x), color= 'k')
mpl.show()

# --------------------
# Code cell separator
# --------------------

x = df[['YEAR']]
y = df[['MAY']]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)

# --------------------
# Code cell separator
# --------------------
```

```
xtrain.shape
ytrain.shape

# --------------------
# Code cell separator
# --------------------

model = LinearRegression()
model.fit(xtrain, ytrain)

# --------------------
# Code cell separator
# --------------------

print(f"intercept: {model.intercept_}")

# --------------------
# Code cell separator
# --------------------

print(f"slope: {model.coef_}")

# --------------------
# Code cell separator
# --------------------

prediction4=model.predict(xtest)

# --------------------
# Code cell separator
# --------------------

from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(xtest, prediction4))
print("Mean Squared Error ", metrics.mean_squared_error(xtest, prediction4))
print("Root Mean Squared Error:", np.sqrt(metrics.mean_squared_error(xtest, prediction4)))

# --------------------
# Code cell separator
# --------------------

import matplotlib.pyplot as mpl
mpl.scatter(x, y, color = 'g')
mpl.plot(x, model.predict(x), color= 'k')
mpl.show()
```