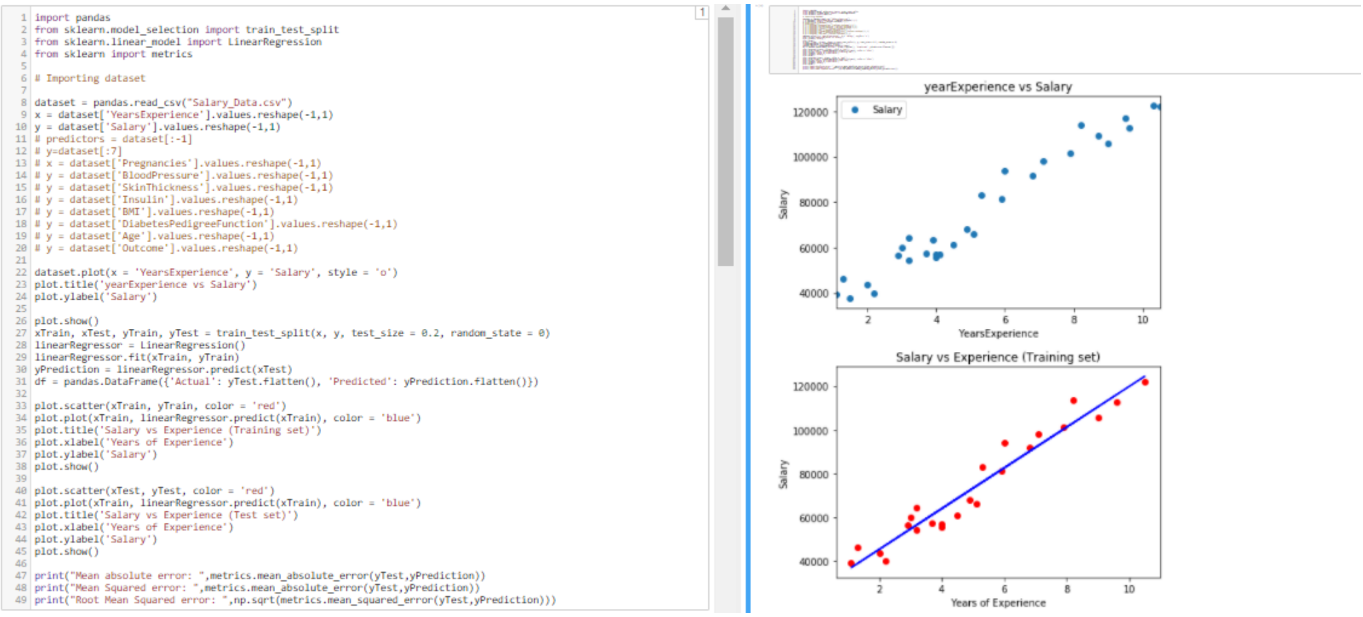
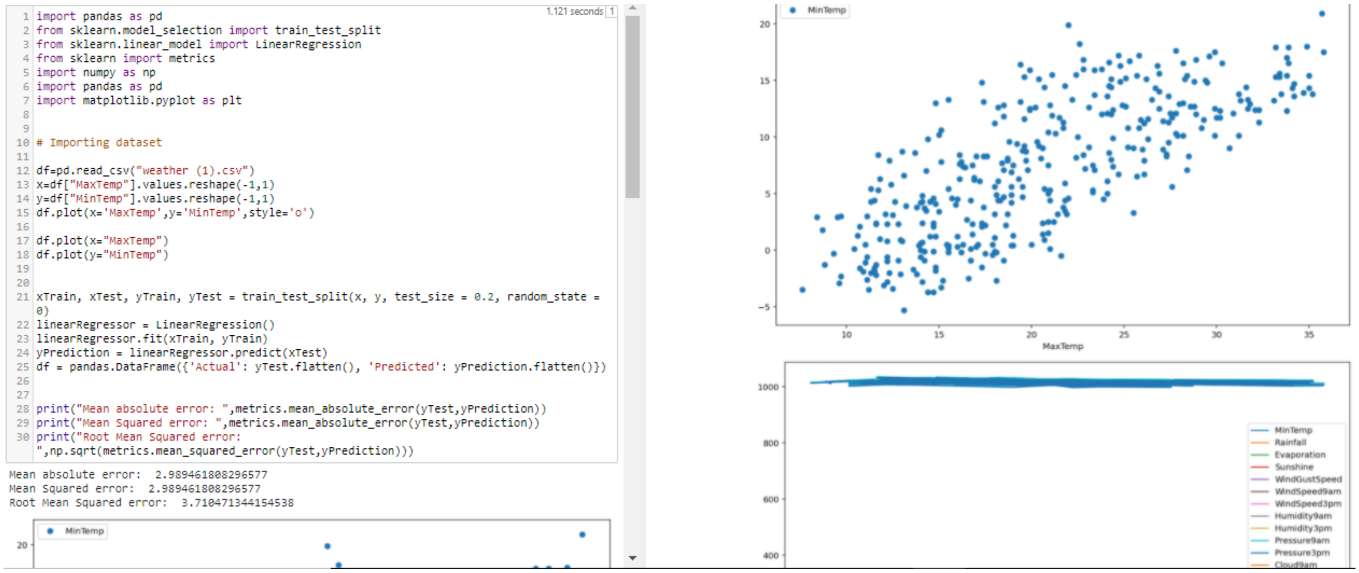
Lab: 07

Task 02



```
import pandas
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

# Importing dataset

dataset = pandas.read_csv("Salary_Data.csv")
x = dataset['YearsExperience'].values.reshape(-1,1)
y = dataset['Salary'].values.reshape(-1,1)
# predictors = dataset[:-1]
# y=dataset[:7]
# x = dataset['Pregnancies'].values.reshape(-1,1)
# y = dataset['BloodPressure'].values.reshape(-1,1)
# y = dataset['SkinThickness'].values.reshape(-1,1)
# y = dataset['Insulin'].values.reshape(-1,1)
# y = dataset['BMI'].values.reshape(-1,1)
# y = dataset['DiabetesPedigreeFunction'].values.reshape(-1,1)
# y = dataset['Age'].values.reshape(-1,1)
# y = dataset['Outcome'].values.reshape(-1,1)

dataset.plot(x = 'YearsExperience', y = 'Salary', style = 'o')
plot.title('yearExperience vs Salary')
plot.ylabel('Salary')

plot.show()
xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size = 0.2, random_state = 0)
linearRegressor = LinearRegression()
linearRegressor.fit(xTrain, yTrain)
yPrediction = linearRegressor.predict(xTest)
df = pandas.DataFrame({'Actual': yTest.flatten(), 'Predicted': yPrediction.flatten()})

plot.scatter(xTrain, yTrain, color = 'red')
plot.plot(xTrain, linearRegressor.predict(xTrain), color = 'blue')
plot.title('Salary vs Experience (Training set)')
plot.xlabel('Years of Experience')
plot.ylabel('Salary')
plot.show()

plot.scatter(xTest, yTest, color = 'red')
plot.plot(xTrain, linearRegressor.predict(xTrain), color = 'blue')
plot.title('Salary vs Experience (Test set)')
plot.xlabel('Years of Experience')
plot.ylabel('Salary')
plot.show()

print("Mean absolute error: ",metrics.mean_absolute_error(yTest,yPrediction))
print("Mean Squared error: ",metrics.mean_absolute_error(yTest,yPrediction))
print("Root Mean Squared error: ",np.sqrt(metrics.mean_squared_error(yTest,yPrediction)))
```

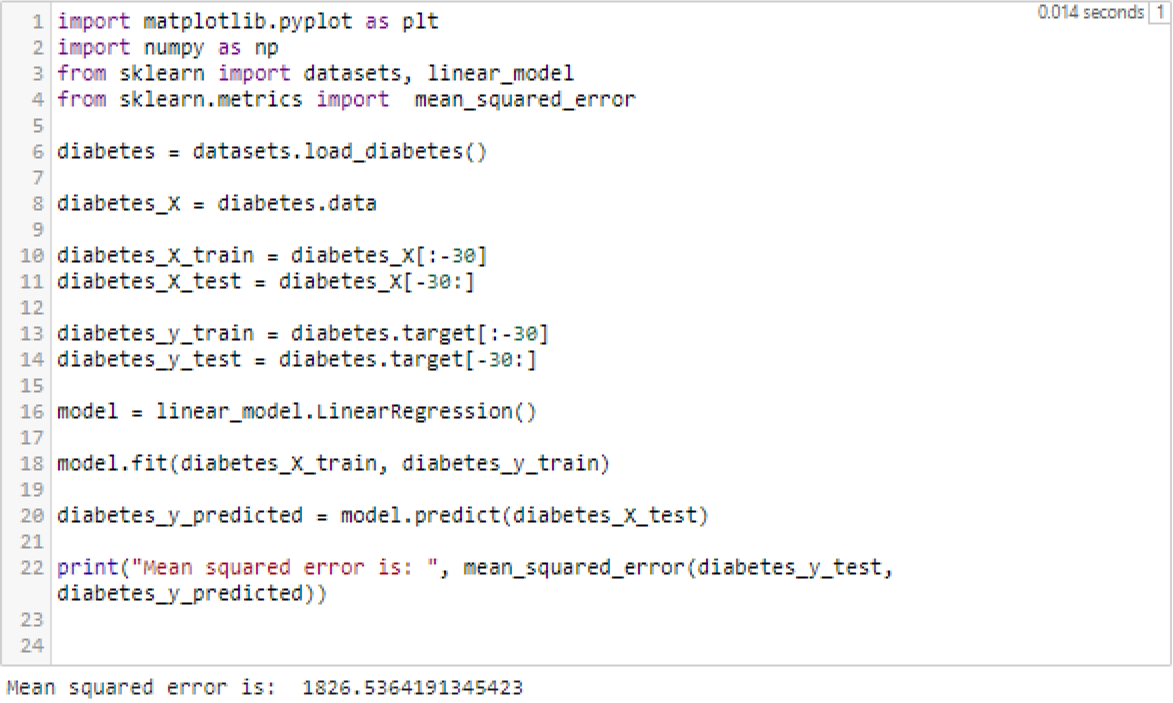
Task 02:



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


# Importing dataset

df=pd.read_csv("weather (1).csv")
x=df["MaxTemp"].values.reshape(-1,1)
y=df["MinTemp"].values.reshape(-1,1)
df.plot(x='MaxTemp',y='MinTemp',style='o')

df.plot(x="MaxTemp")
df.plot(y="MinTemp")


xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size = 0.2, random_state = 0)
linearRegressor = LinearRegression()
linearRegressor.fit(xTrain, yTrain)
yPrediction = linearRegressor.predict(xTest)
df = pandas.DataFrame({'Actual': yTest.flatten(), 'Predicted': yPrediction.flatten()})


print("Mean absolute error: ",metrics.mean_absolute_error(yTest,yPrediction))
print("Mean Squared error: ",metrics.mean_absolute_error(yTest,yPrediction))
print("Root Mean Squared error: ",np.sqrt(metrics.mean_squared_error(yTest,yPrediction)))
```

```
Mean absolute error:  2.989461808296577
Mean Squared error:  2.989461808296577
Root Mean Squared error:  3.710471344154538
```

Task 03:



```python
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import  mean_squared_error

diabetes = datasets.load_diabetes()

diabetes_X = diabetes.data

diabetes_X_train = diabetes_X[:-30]
diabetes_X_test = diabetes_X[-30:]

diabetes_y_train = diabetes.target[:-30]
diabetes_y_test = diabetes.target[-30:]

model = linear_model.LinearRegression()

model.fit(diabetes_X_train, diabetes_y_train)

diabetes_y_predicted = model.predict(diabetes_X_test)

print("Mean squared error is: ", mean_squared_error(diabetes_y_test, diabetes_y_predicted))


```

```
Mean squared error is:  1826.5364191345423
```