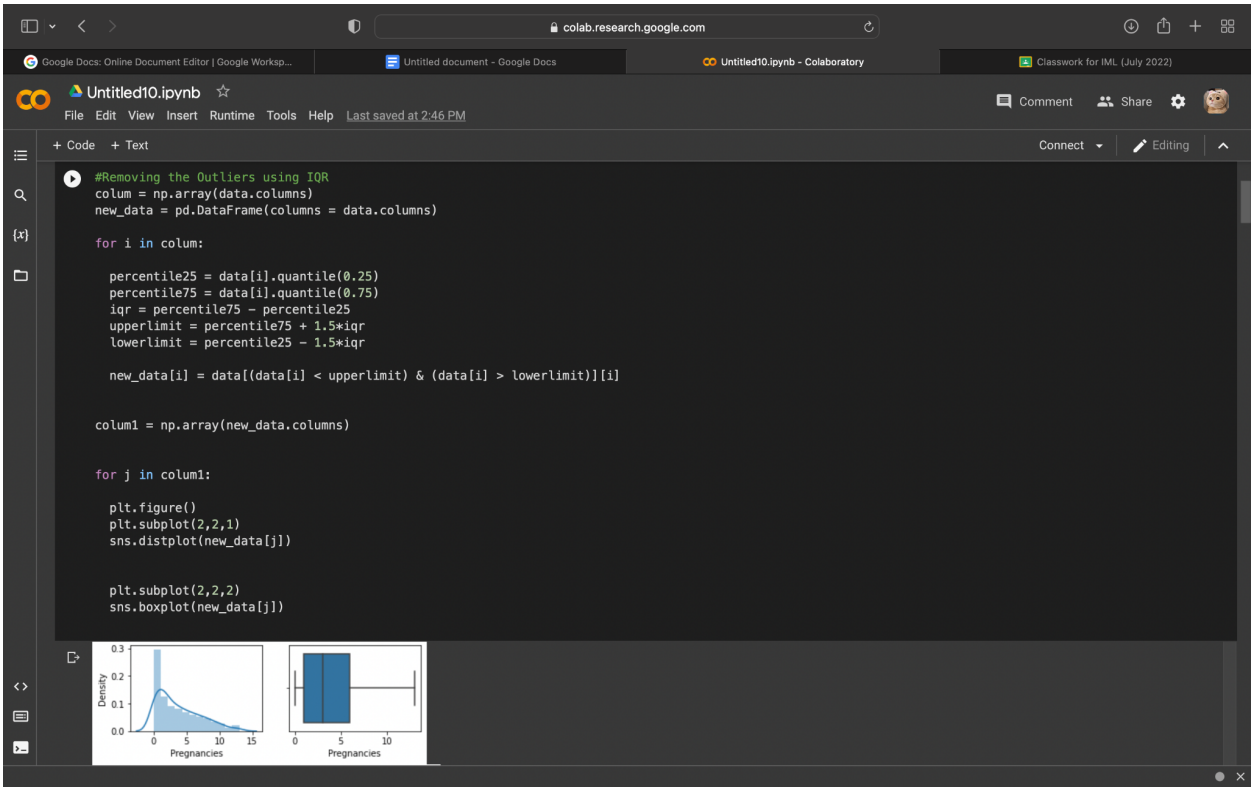
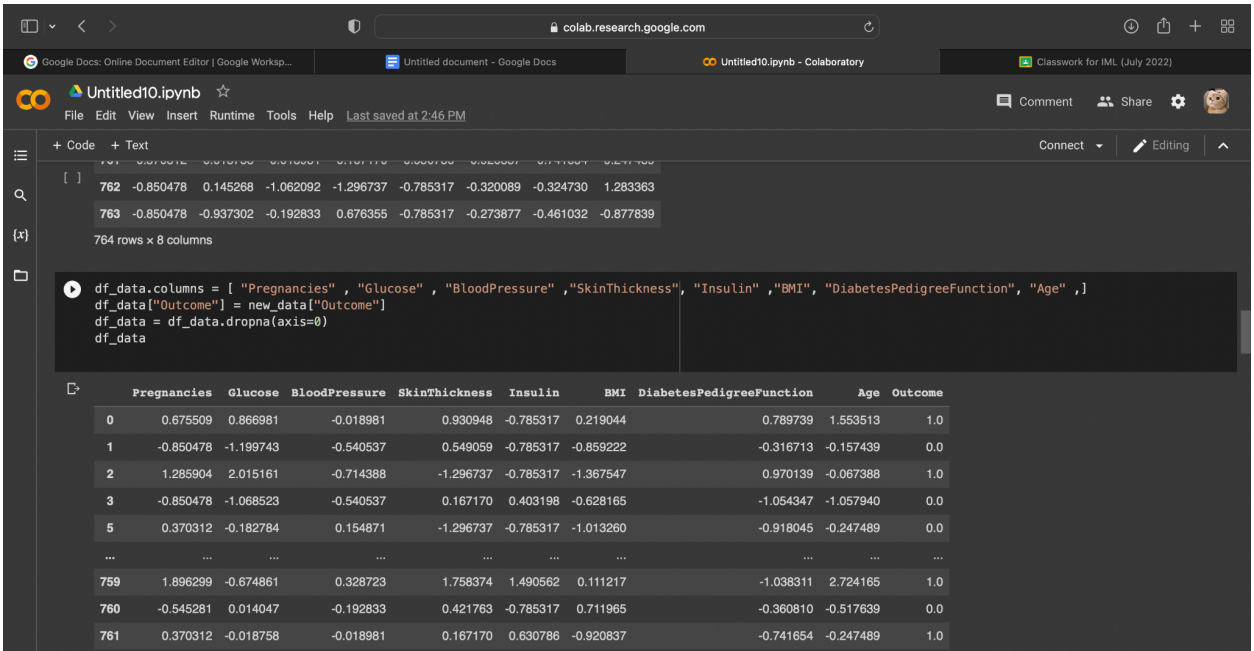
SMRUTI DAWALE
B21BB007
LAB 4

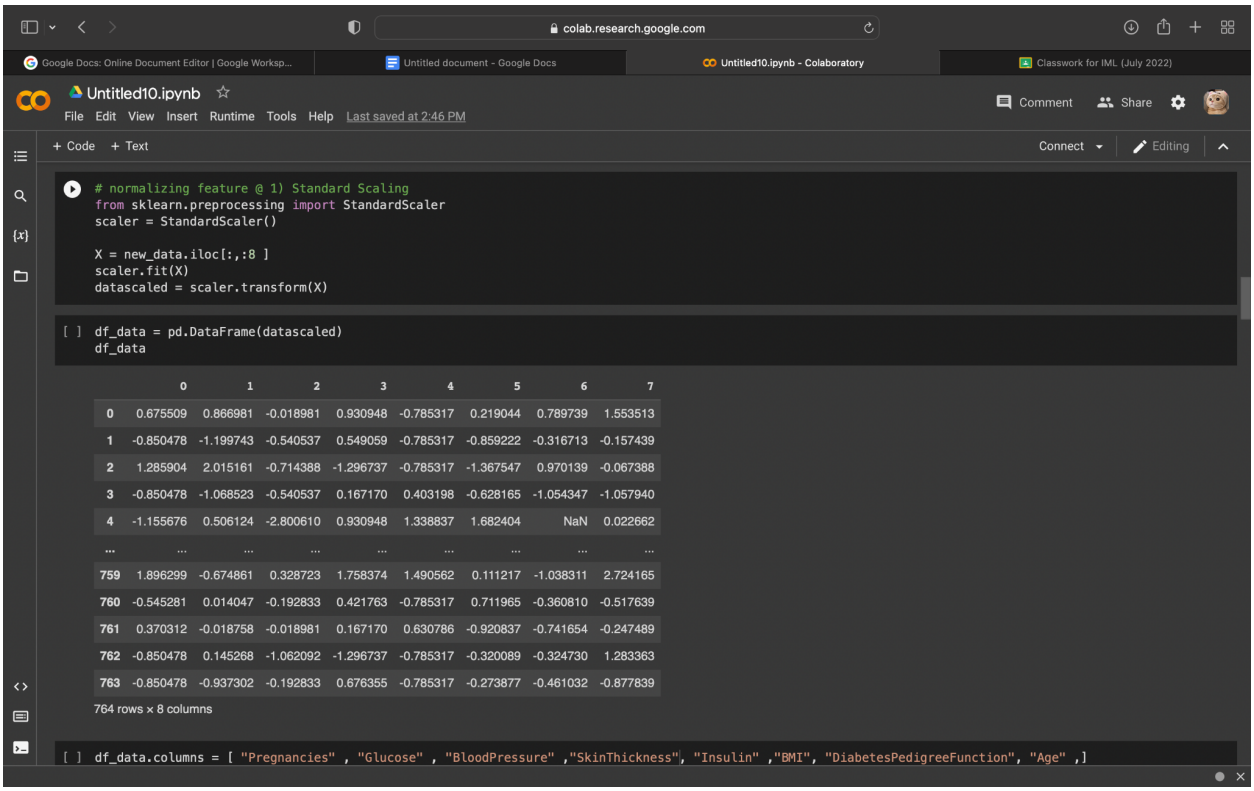
Que 1)

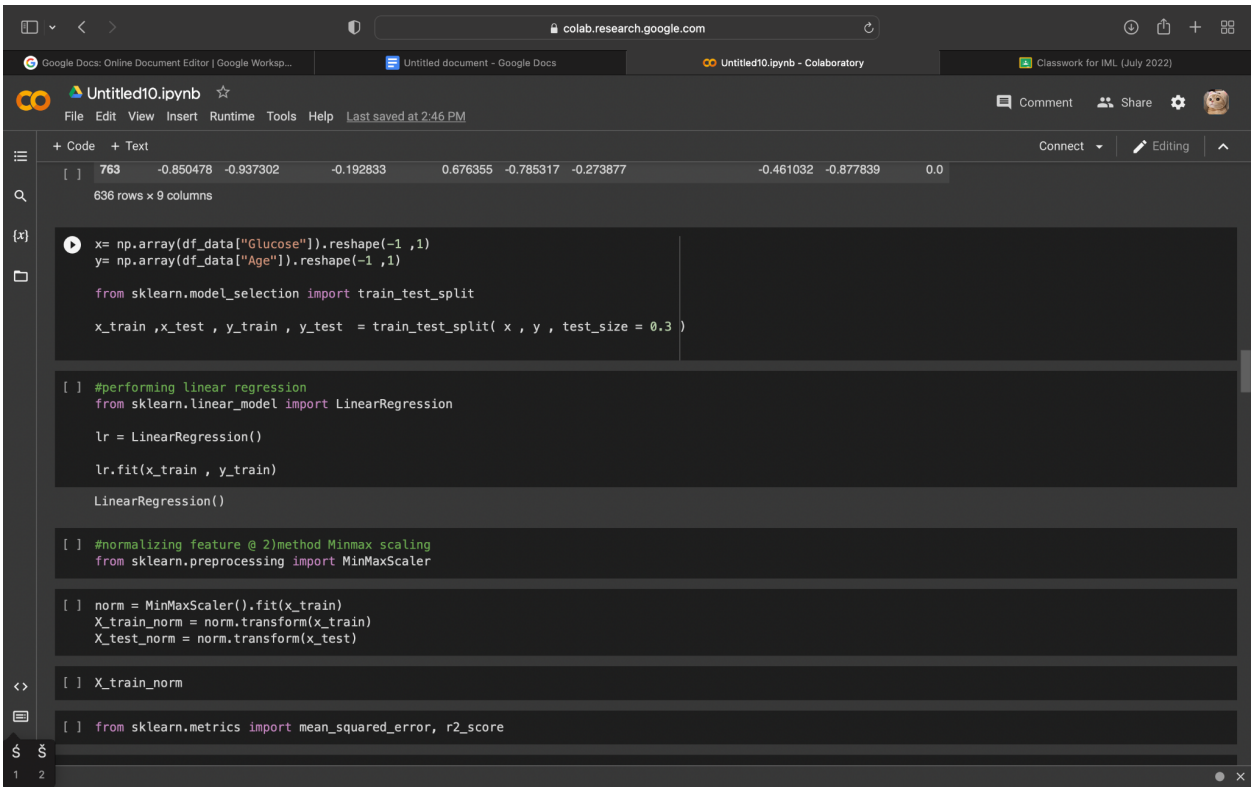
A) I have done the steps of preprocessing i.e removing outliers , standard scaling and checking and removing null values.
Then i trained the model using linear regression.





	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0.675509	0.866981	-0.018981	0.930948	-0.785317	0.219044	0.789739	1.553513	1.0
1	-0.850478	-1.199743	-0.540537	0.549059	-0.785317	-0.859222	-0.316713	-0.157439	0.0
2	1.285904	2.015161	-0.714388	-1.296737	-0.785317	-1.367547	0.970139	-0.067388	1.0
3	-0.850478	-1.068523	-0.540537	0.167170	0.403198	-0.628165	-1.054347	-1.057940	0.0
5	0.370312	-0.182784	0.154871	-1.296737	-0.785317	-1.013260	-0.918045	-0.247489	0.0
...	...	...	...	...	...	...	...	...	...
759	1.896299	-0.674861	0.328723	1.758374	1.490562	0.111217	-1.038311	2.724165	1.0
760	-0.545281	0.014047	-0.192833	0.421763	-0.785317	0.711965	-0.360810	-0.517639	0.0
761	0.370312	-0.018758	-0.018981	0.167170	0.630786	-0.920837	-0.741654	-0.247489	1.0

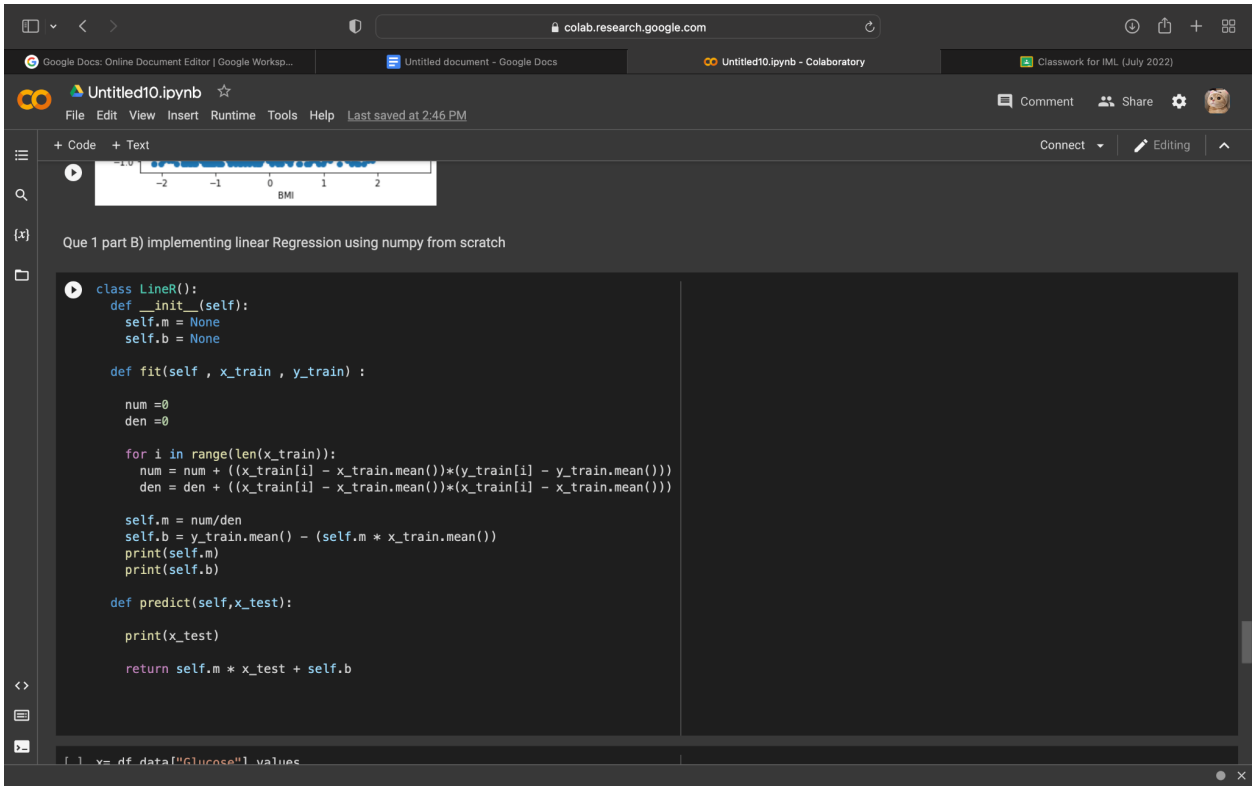

colab.research.google.com
Google Docs: Online Document Editor | Google Worksp...
Untitled document - Google Docs
Untitled10.ipynb - Colaboratory
Classwork for IML (July 2022)
Untitled10.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:46 PM
Comment
Share
+ Code + Text
Connect
Editing
# normalizing feature @ 1) Standard Scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = new_data.iloc[:,:8]
scaler.fit(X)
datascaled = scaler.transform(X)
[ ] df_data = pd.DataFrame(datascaled)
df_data
0 1 2 3 4 5 6 7
0 0.675509 0.866981 -0.018981 0.930948 -0.785317 0.219044 0.789739 1.553513
1 -0.850478 -1.199743 -0.540537 0.549059 -0.785317 -0.859222 -0.316713 -0.157439
2 1.285904 2.015161 -0.714388 -1.296737 -0.785317 -1.367547 0.970139 -0.067388
3 -0.850478 -1.068523 -0.540537 0.167170 0.403198 -0.628165 -1.054347 -1.057940
4 -1.155676 0.506124 -2.800610 0.930948 1.338837 1.682404 NaN 0.022662
... ...
759 1.896299 -0.674861 0.328723 1.758374 1.490562 0.111217 -1.038311 2.724165
760 -0.545281 0.014047 -0.192833 0.421763 -0.785317 0.711965 -0.360810 -0.517639
761 0.370312 -0.018758 -0.018981 0.167170 0.630786 -0.920837 -0.741654 -0.247489
762 -0.850478 0.145268 -1.062092 -1.296737 -0.785317 -0.320089 -0.324730 1.283363
763 -0.850478 -0.937302 -0.192833 0.676355 -0.785317 -0.273877 -0.461032 -0.877839
764 rows × 8 columns
[ ] df_data.columns = ["Pregnancies" , "Glucose" , "BloodPressure" ,"SkinThickness", "Insulin" ,"BMI", "DiabetesPedigreeFunction", "Age" ,]

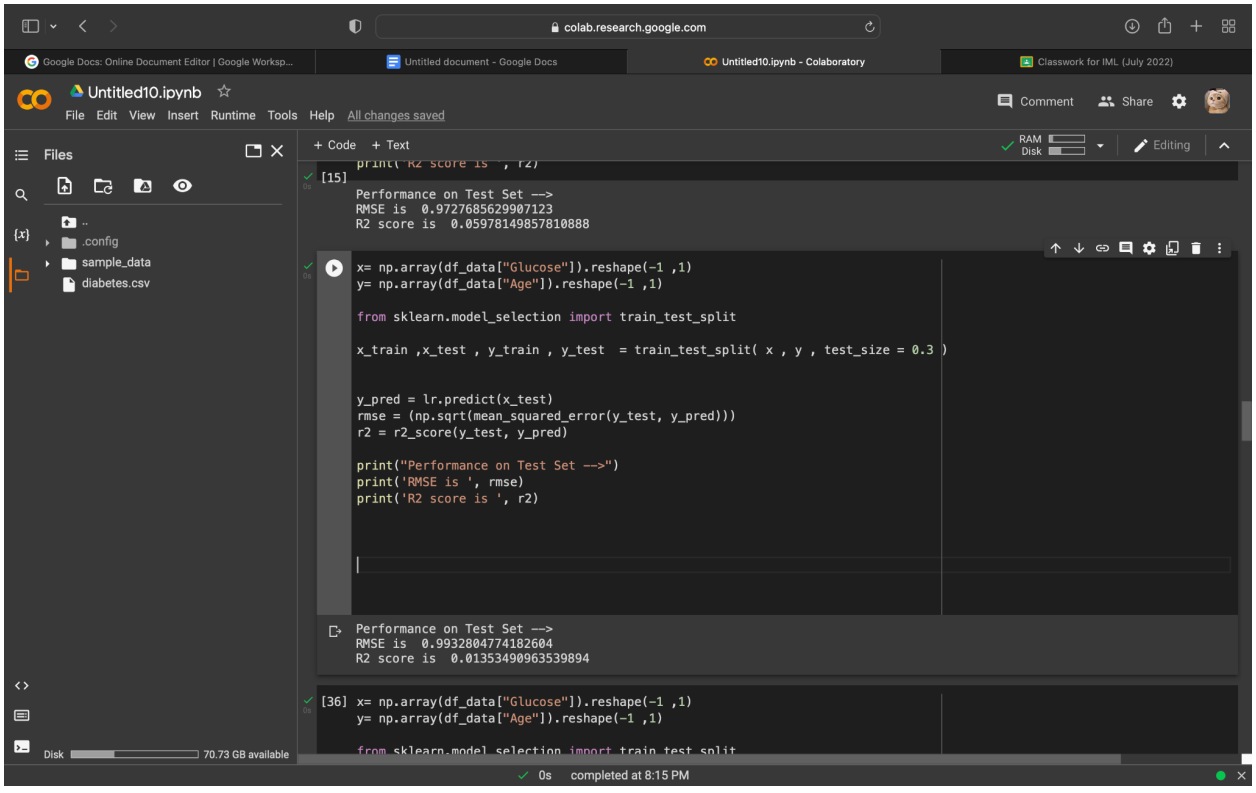

colab.research.google.com
Google Docs: Online Document Editor | Google Worksp...
Untitled document - Google Docs
Untitled10.ipynb - Colaboratory
Classwork for IML (July 2022)
Untitled10.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:46 PM
Comment
Share
+ Code + Text
Connect
Editing
[ ] 763 -0.850478 -0.937302 -0.192833 0.676355 -0.785317 -0.273877 -0.461032 -0.877839 0.0
636 rows × 9 columns
x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train ,x_test , y_train , y_test = train_test_split(x , y , test_size = 0.3)
[ ] #performing linear regression
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train , y_train)
LinearRegression()
[ ] #normalizing feature @ 2)method Minmax scaling
from sklearn.preprocessing import MinMaxScaler
[ ] norm = MinMaxScaler().fit(x_train)
X_train_norm = norm.transform(x_train)
X_test_norm = norm.transform(x_test)
[ ] X_train_norm
[ ] from sklearn.metrics import mean_squared_error, r2_score

B) I have made a class for this question named LineR() , where I have made functions for training and predicting the x_train .

Rms error using code :`RMSE is  0.9500411120764171`

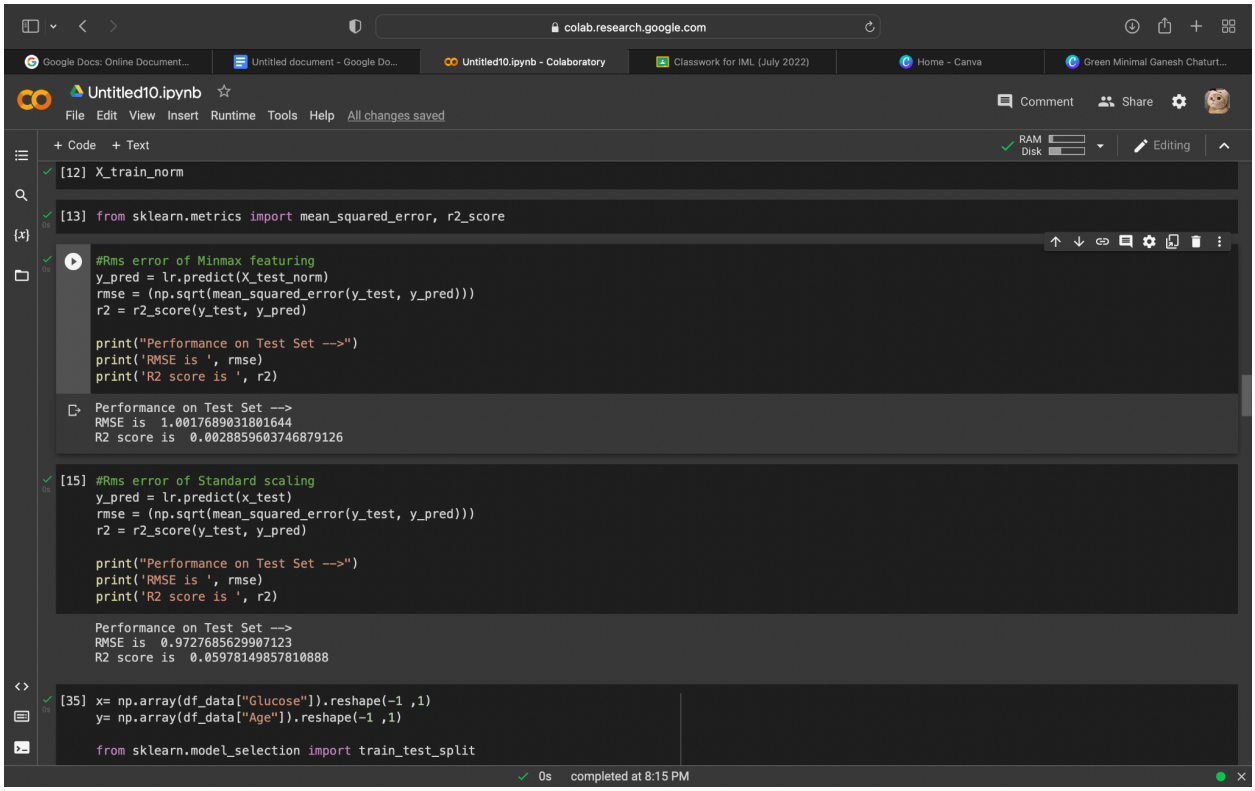
Rms error using sklearn: `RMSE is  0.9932804774182604`


colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:46 PM
Comment Share
+ Code + Text
Connect Editing
BMI
Que 1 part B) implementing linear Regression using numpy from scratch
class LineR():
def __init__(self):
self.m = None
self.b = None
def fit(self , x_train , y_train) :
num =0
den =0
for i in range(len(x_train)):
num = num + ((x_train[i] - x_train.mean())*(y_train[i] - y_train.mean()))
den = den + ((x_train[i] - x_train.mean())*(x_train[i] - x_train.mean()))
self.m = num/den
self.b = y_train.mean() - (self.m * x_train.mean())
print(self.m)
print(self.b)
def predict(self,x_test):
print(x_test)
return self.m * x_test + self.b

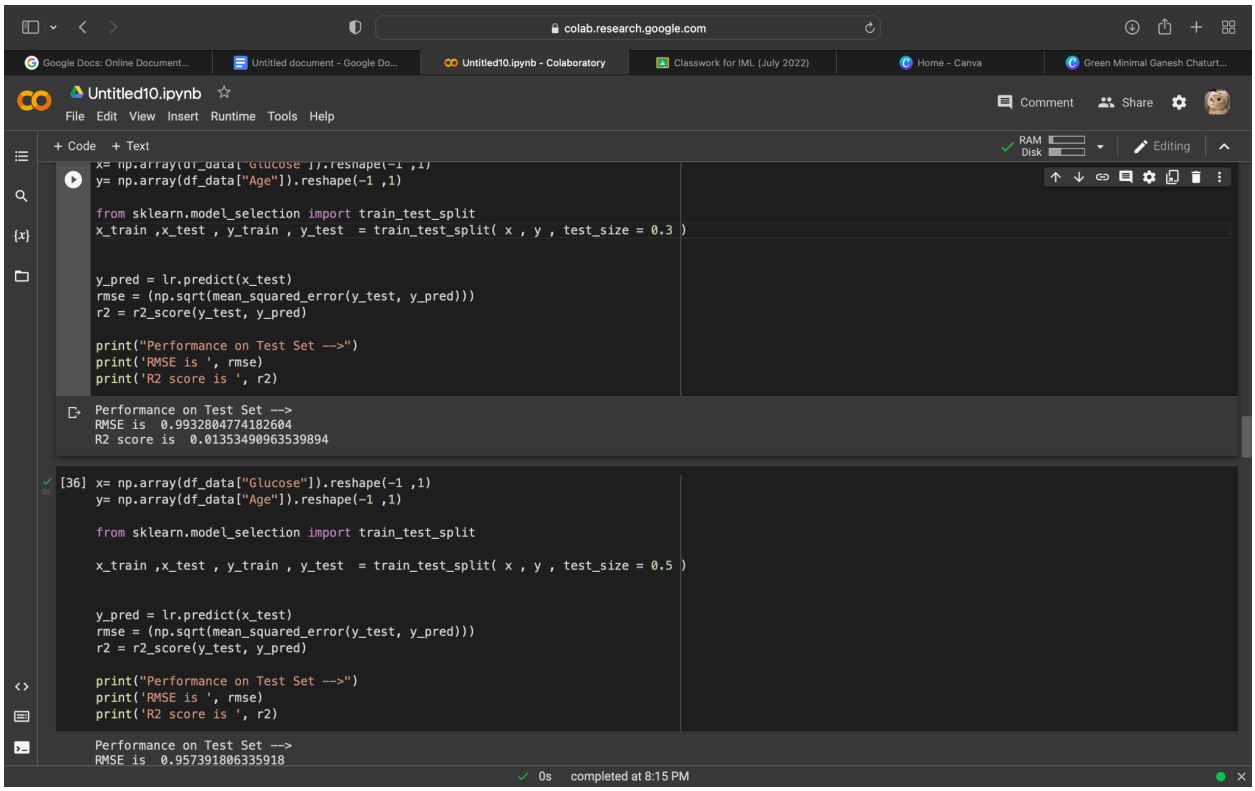

colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
RAM Disk
Editing
.config
sample_data
diabetes.csv
Disk 70.73 GB available
[15]
Performance on Test Set -->
RMSE is 0.9727685629907123
R2 score is 0.05978149857810888
x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train ,x_test , y_train , y_test = train_test_split(x , y , test_size = 0.3)
y_pred = lr.predict(x_test)
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
r2 = r2_score(y_test, y_pred)
print("Performance on Test Set -->")
print('RMSE is ', rmse)
print('R2 score is ', r2)
Performance on Test Set -->
RMSE is 0.9932804774182604
R2 score is 0.01353490963539894
[36] x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
0s completed at 8:15 PM

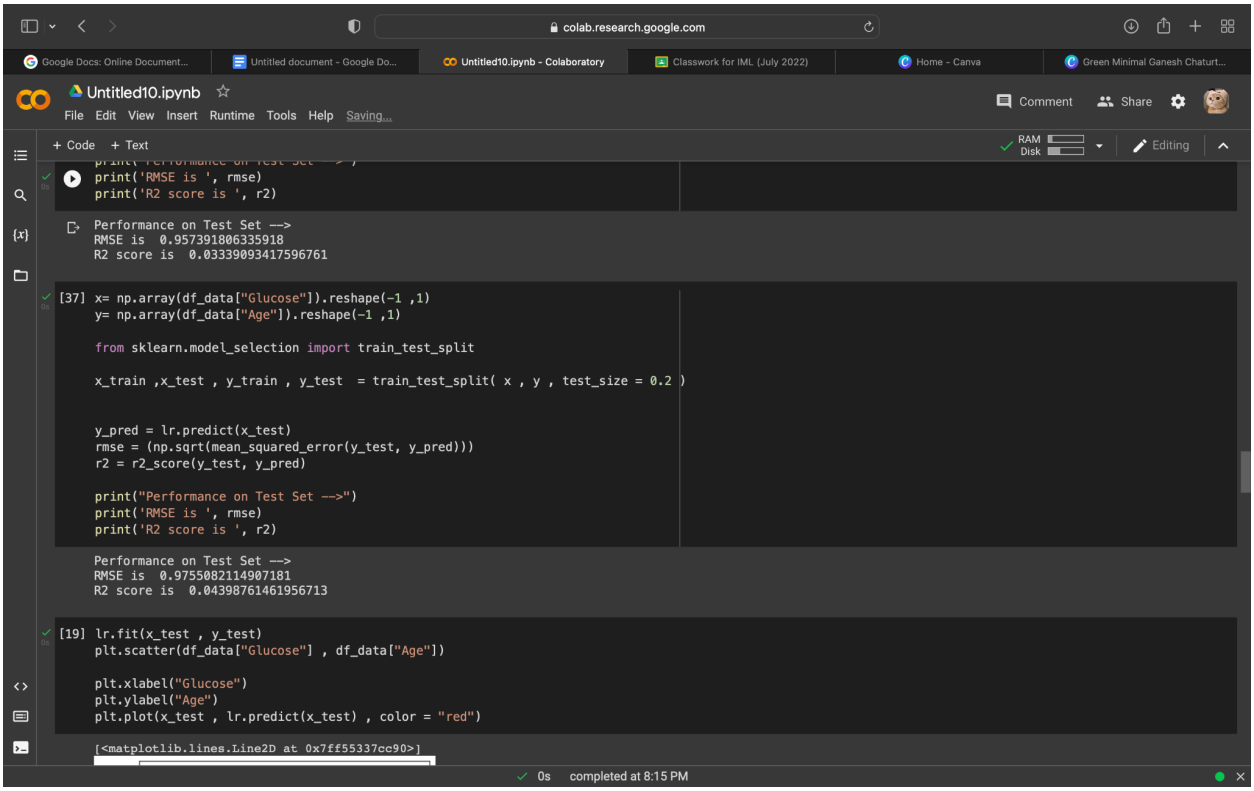
Que 2)

A) I have used Standard scaling and Minmax Scaling
The rms values using
Standard scaling-`RMSE is  0.9727685629907123`
Minmax Scaling -`RMSE is  1.0017689031801644`

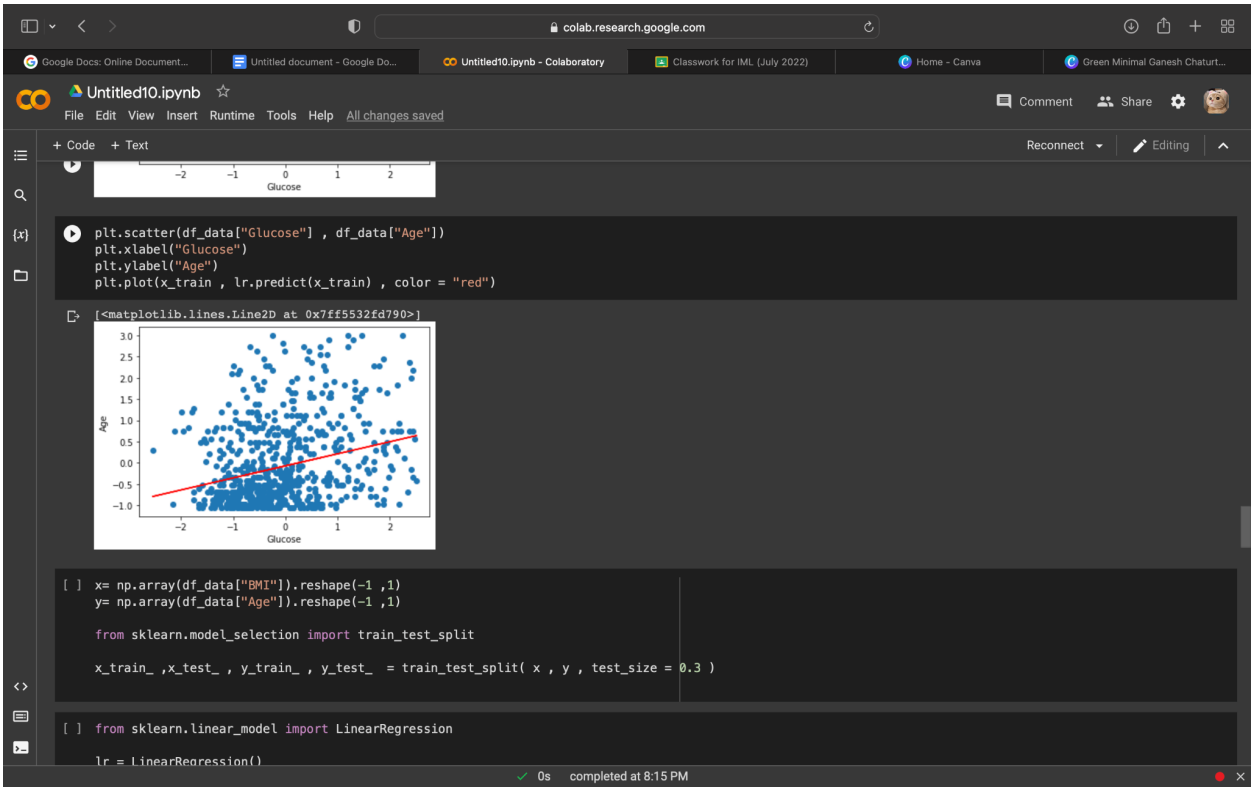


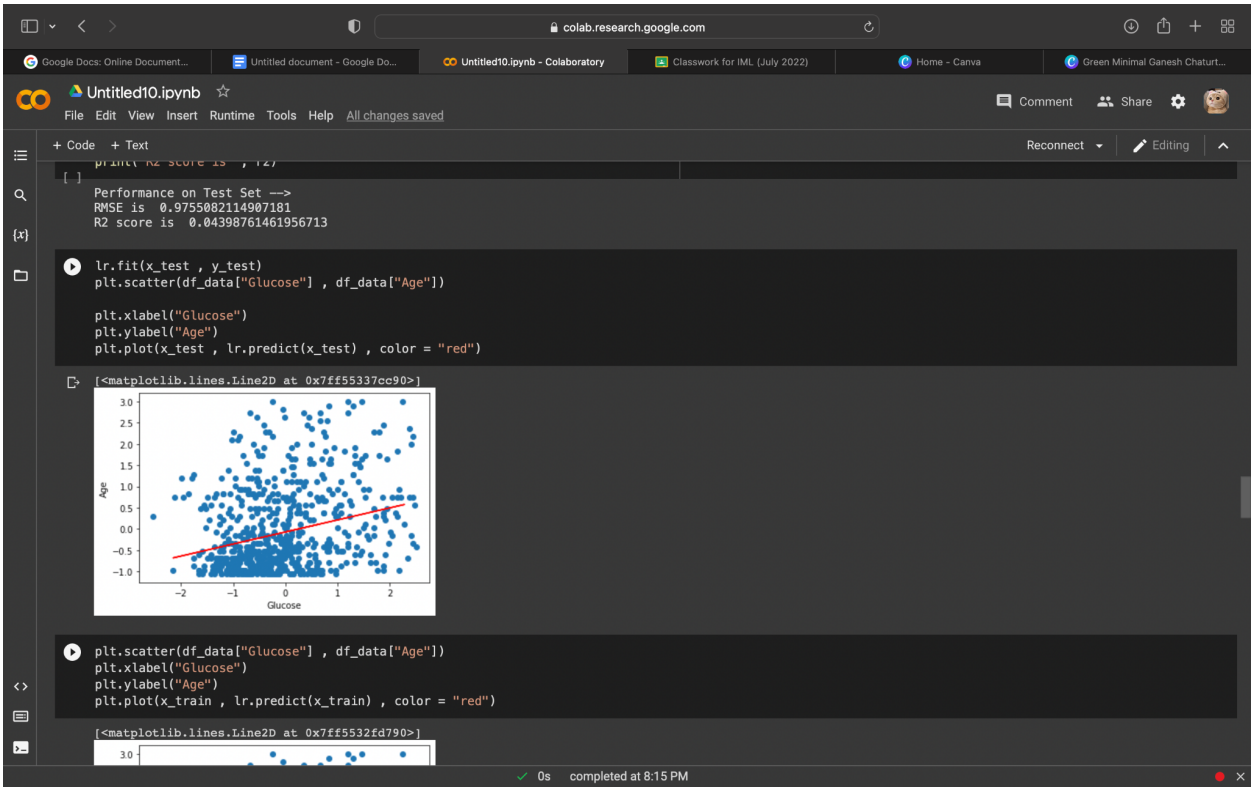
B) Ratio
test = 0.3 , train = 0.7 `RMSE is  0.9932804774182604`
test = 0.2 , train = 0.8 `RMSE is  0.9755082114907181`
Test = 0.5 , tain = 0.5 `RMSE is  0.957391806335918`


colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train ,x_test , y_train , y_test = train_test_split(x , y , test_size = 0.3)
y_pred = lr.predict(x_test)
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
r2 = r2_score(y_test, y_pred)
print("Performance on Test Set -->")
print('RMSE is ', rmse)
print('R2 score is ', r2)
Performance on Test Set -->
RMSE is 0.9932804774182604
R2 score is 0.01353490963539894
[36] x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train ,x_test , y_train , y_test = train_test_split(x , y , test_size = 0.5)
y_pred = lr.predict(x_test)
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
r2 = r2_score(y_test, y_pred)
print("Performance on Test Set -->")
print('RMSE is ', rmse)
print('R2 score is ', r2)
Performance on Test Set -->
RMSE is 0.957391806335918
0s completed at 8:15 PM


colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment
Share
+ Code + Text
RAM
Disk
Editing
print('RMSE is ', rmse)
print('R2 score is ', r2)
Performance on Test Set -->
RMSE is 0.957391806335918
R2 score is 0.03339093417596761
[37] x= np.array(df_data["Glucose"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train ,x_test , y_train , y_test = train_test_split(x , y , test_size = 0.2)
y_pred = lr.predict(x_test)
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
r2 = r2_score(y_test, y_pred)
print("Performance on Test Set -->")
print('RMSE is ', rmse)
print('R2 score is ', r2)
Performance on Test Set -->
RMSE is 0.9755082114907181
R2 score is 0.04398761461956713
[19] lr.fit(x_test , y_test)
plt.scatter(df_data["Glucose"] , df_data["Age"])
plt.xlabel("Glucose")
plt.ylabel("Age")
plt.plot(x_test , lr.predict(x_test) , color = "red")
[<matplotlib.lines.Line2D at 0x7ff55337cc90>]
0s completed at 8:15 PM

C) Graphs


colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Reconnect Editing
Glucose
plt.scatter(df_data["Glucose"] , df_data["Age"])
plt.xlabel("Glucose")
plt.ylabel("Age")
plt.plot(x_train , lr.predict(x_train) , color = "red")
[<matplotlib.lines.Line2D at 0x7ff5532fd790>]
Age
Glucose
x= np.array(df_data["BMI"]).reshape(-1 ,1)
y= np.array(df_data["Age"]).reshape(-1 ,1)
from sklearn.model_selection import train_test_split
x_train_ ,x_test_ , y_train_ , y_test_ = train_test_split(x , y , test_size = 0.3)
from sklearn.linear_model import LinearRegression
0s completed at 8:15 PM


colab.research.google.com
Untitled10.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Reconnect Editing
Performance on Test Set -->
RMSE is 0.9755082114907181
R2 score is 0.04398761461956713
lr.fit(x_test , y_test)
plt.scatter(df_data["Glucose"] , df_data["Age"])
plt.xlabel("Glucose")
plt.ylabel("Age")
plt.plot(x_test , lr.predict(x_test) , color = "red")
[<matplotlib.lines.Line2D at 0x7ff55337cc90>]
Age
Glucose
plt.scatter(df_data["Glucose"] , df_data["Age"])
plt.xlabel("Glucose")
plt.ylabel("Age")
plt.plot(x_train , lr.predict(x_train) , color = "red")
[<matplotlib.lines.Line2D at 0x7ff5532fd790>]
0s completed at 8:15 PM

colab.research.google.com

Google Docs: Online Document... | Untitled document - Google Do... | Untitled10.ipynb - Colaboratory | Classwork for IML (July 2022) | Home - Canva | Green Minimal Ganesh Chaturt...

Untitled10.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

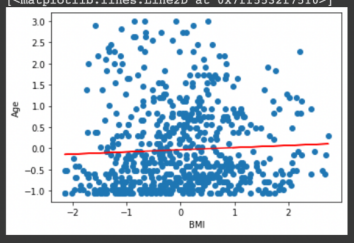
Reconnect Editing

```
[ ] lr.fit(x_train_ , y_train_)

    LinearRegression()
```

```
plt.scatter(df_data["BMI"] , df_data["Age"])
plt.xlabel("BMI")
plt.ylabel("Age")
plt.plot(x_train_ , lr.predict(x_train_) , color = "red")
```

```
[<matplotlib.lines.Line2D at 0x7ff5532f7510>]
```




```
[ ] lr.fit(x_test_ , y_test_)
    plt.scatter(df_data["BMI"] , df_data["Age"])
    plt.xlabel("BMI")
    plt.ylabel("Age")
    plt.plot(x_test_ , lr.predict(x_test_) , color = "red")
```

```
[<matplotlib.lines.Line2D at 0x7ff5532e3450>]
```

0s completed at 8:15 PM

colab.research.google.com

Google Docs: Online Document... | Untitled document - Google Do... | Untitled10.ipynb - Colaboratory | Classwork for IML (July 2022) | Home - Canva | Green Minimal Ganesh Chaturt...

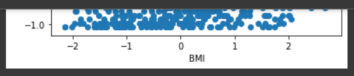
Untitled10.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

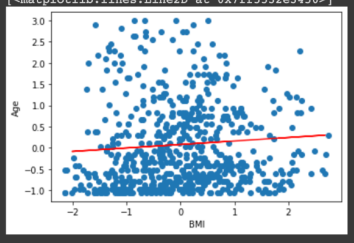
+ Code + Text

Reconnect Editing




```
lr.fit(x_test_ , y_test_)
plt.scatter(df_data["BMI"] , df_data["Age"])
plt.xlabel("BMI")
plt.ylabel("Age")
plt.plot(x_test_ , lr.predict(x_test_) , color = "red")
```

```
[<matplotlib.lines.Line2D at 0x7ff5532e3450>]
```



Que 1 part B) implementing linear Regression using numpy from scratch

```
class LineR():
    def __init__(self):
        self.m = None
        self.b = None

    def fit(self , x_train , y_train) :
```

0s completed at 8:15 PM