Assignment-4
Statistical Machine Learning Approaches
To Liver Disease Prediction

Student Name	Kamali R
Student Roll no	621319104019
Maximum Marks	2 Marks

Customer Segmentation Analysis:

+ Code + Text Connect

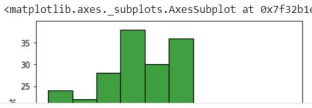
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
data = pd.read_csv('/content/Kamali R_Mall_Customers.csv')
data.head()
```

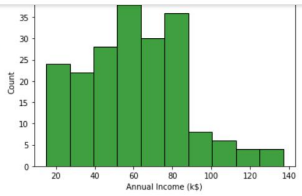
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
[ ] sns.histplot(data['Annual Income (k$)'], color="green")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f32b1ea1750>
```

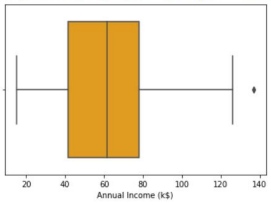


+ Code + Text Connect Editin




```
[ ] sns.boxplot(data['Annual Income (k$)'], color="orange")
```

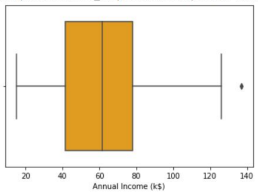
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329beba750>
```



+ Code + Text Connect Editing

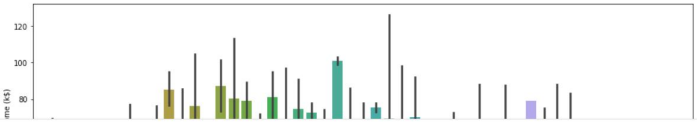
```
[ ] sns.boxplot(data['Annual Income (k$)'], color="orange")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329b98a590>
```



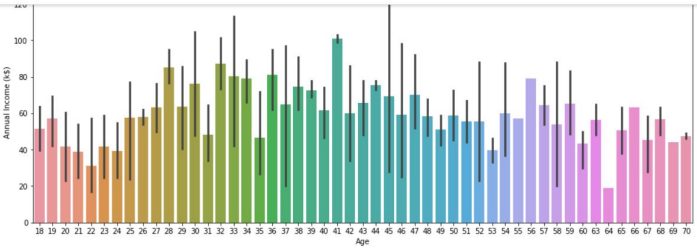

```
[ ] plt.figure(figsize=(16,6))
    sns.barplot(data['Age'],data['Annual Income (k$)'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329b992f90>
```



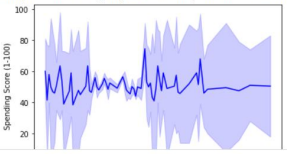
+ Code + Text Connect Editing

```
[ ]
```



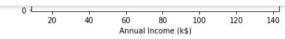

```
sns.lineplot(data['Annual Income (k$)'], data['Spending Score (1-100)'], color="blue")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329b6bb6d0>
```



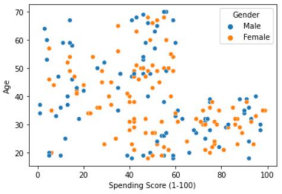
+ Code + Text Connect Editing

```
[ ]
```

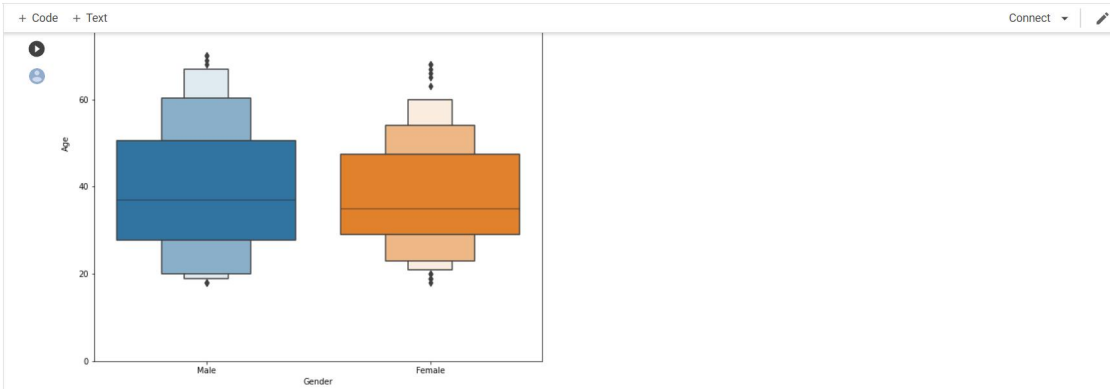



```
sns.scatterplot(data['Spending Score (1-100)'], data['Age'], hue = data['Gender'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329b63d210>
```

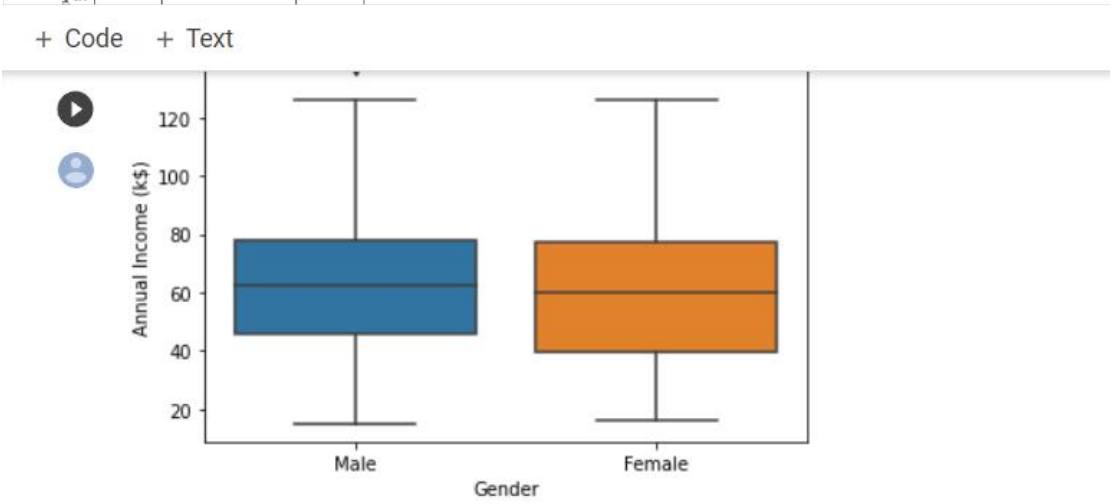



```
[ ] temp = pd.concat([data['Age'], data['Gender']], axis=1)
    f, ax = plt.subplots(figsize=(10,10))
    fig = sns.boxenplot(x='Gender', y="Age", data=data)
    fig.axis(ymin=0, ymax=100);
```




```
[ ] sns.boxplot(x=data['Gender'],y=data['Annual Income (k$)'])
```

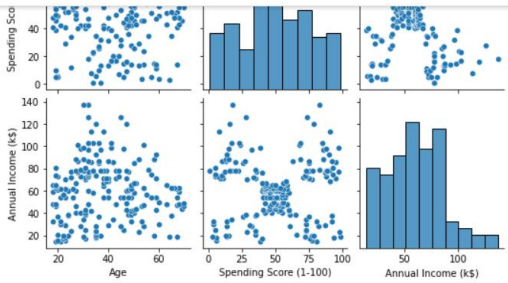
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329b532690>
```




```
[ ] sns.pairplot(data=data[["Age", "Gender","Spending Score (1-100)","Annua
```

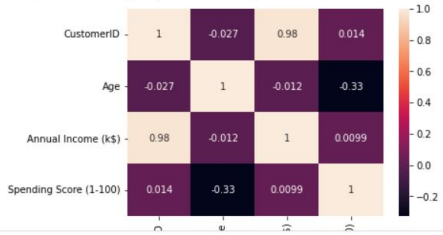
```
<seaborn.axisgrid.PairGrid at 0x7f3298ca7490>
```

+ Code + Text




```
[ ] sns.heatmap(data.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f329880b590>
```



+ Code + Text Connect E

```
data.describe()
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
[ ] data.info
```

```
<bound method DataFrame.info of      CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male   19                  15                      39
1             2    Male   21                  15                      81
2             3  Female   20                  16                       6
3             4  Female   23                  16                      77
4             5  Female   31                  17                      40
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83
```

+ Code + Text Connect Editing

```
[200 rows x 5 columns]>
```

```
[ ] data.shape
```

```
(200, 5)
```

```
[ ] data.isnull().any() #Inference: The dataset has no null values
```

```
CustomerID              False
Gender                  False
Age                     False
Annual Income (k$)      False
Spending Score (1-100)  False
dtype: bool
```

```
data.drop('CustomerID',axis=1,inplace=True)
data.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

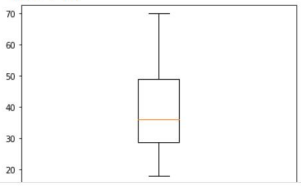
```
[ ] for i in data:
```

+ Code + Text Connect

```
for i in data:
    if data[i].dtype=='int64':
        q1=data[i].quantile(0.25)
        q3=data[i].quantile(0.75)
        iqr=q3-q1
        upper=q3+1.5*iqr
        lower=q1-1.5*iqr
        data[i]=np.where(data[i] >upper, upper, data[i])
        data[i]=np.where(data[i] <lower, lower, data[i])
```

```
[ ] plt.boxplot(data['Age'])
```

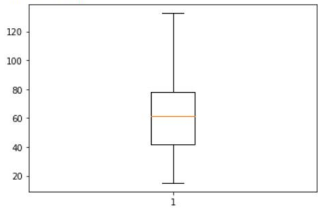
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f3296f28f50>,
  <matplotlib.lines.Line2D at 0x7f3296f2c4d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f3296f2ca10>,
  <matplotlib.lines.Line2D at 0x7f3296f2cf50>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f3296f289d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f3296f33510>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f3296f33a50>],
 'means': []}
```



+ Code + Text Conne

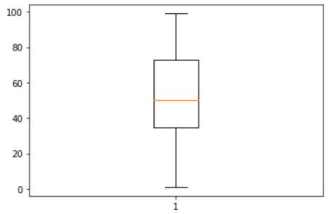
```
plt.boxplot(data['Annual Income (k$)'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f3296e94b50>,
  <matplotlib.lines.Line2D at 0x7f3296e990d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f3296e99610>,
  <matplotlib.lines.Line2D at 0x7f3296e99b50>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f3296e945d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f3296ea1110>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f3296ea1650>],
 'means': []}
```




```
[ ] plt.boxplot(data['Spending Score (1-100)'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f3296e07490>,
  <matplotlib.lines.Line2D at 0x7f3296e079d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f3296e07f10>,
  <matplotlib.lines.Line2D at 0x7f3296e0d490>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f3296e7ee90>],
 'medians': [<matplotlib.lines.Line2D at 0x7f3296e0da10>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f3296e0df50>],
 'means': []}
```




```
[ ] from sklearn.preprocessing import LabelEncoder
    l_en = LabelEncoder()
```

```
data['Gender'] = l_en.fit_transform(data['Gender'])
data.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19.0	15.0	39.0
1	1	21.0	15.0	81.0
2	0	20.0	16.0	6.0
3	0	23.0	16.0	77.0

+ Code + Text

Connect | Editing

4	0	31.0	17.0	40.0

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
data_scaled[0:5]
```

```
array([[1.        , 0.01923077, 0.        , 0.3877551 ],
       [1.        , 0.05769231, 0.        , 0.81632653],
       [0.        , 0.03846154, 0.00849257, 0.05102041],
       [0.        , 0.09615385, 0.00849257, 0.7755102 ],
       [0.        , 0.25      , 0.01698514, 0.39795918]])
```

```
from sklearn.cluster import KMeans
km = KMeans()
res = km.fit_predict(data_scaled)
res
```

```
array([1, 1, 6, 6, 6, 3, 6, 5, 6, 5, 6, 3, 6, 7, 1, 6, 1, 5, 6, 1, 1,
       3, 1, 3, 1, 3, 1, 3, 6, 5, 6, 5, 1, 3, 6, 3, 6, 3, 6, 3, 1, 5, 6,
       3, 6, 3, 6, 6, 6, 3, 1, 6, 5, 3, 5, 3, 5, 6, 5, 5, 1, 3, 3, 5, 1,
       3, 3, 1, 6, 5, 3, 3, 3, 5, 1, 3, 1, 6, 3, 5, 1, 5, 3, 6, 5, 3, 6,
       6, 3, 3, 1, 5, 3, 6, 1, 3, 6, 5, 1, 6, 3, 5, 1, 5, 6, 3, 5, 5, 5,
       5, 6, 3, 1, 6, 6, 3, 3, 3, 3, 1, 2, 0, 4, 6, 0, 7, 4, 5, 4, 7, 4,
       6, 0, 7, 0, 2, 4, 7, 0, 2, 4, 6, 0, 7, 4, 5, 0, 2, 4, 7, 4, 2, 0,
       2, 0, 7, 0, 7, 0, 2, 0, 7, 0, 7, 0, 7, 0, 2, 4, 7, 4, 7, 4, 2, 0,
       5, 4, 5, 4, 2, 0, 7, 0, 2, 4, 2, 4, 2, 0, 2, 0, 7, 0, 2, 0, 2, 4,
       7, 4], dtype=int32)
```

```
data1 = pd.DataFrame(data_scaled, columns = data.columns)
```

+ Code + Text

```
data1 = pd.DataFrame(data_scaled, columns = data.columns)
data1.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1.0	0.019231	0.000000	0.387755
1	1.0	0.057692	0.000000	0.816327
2	0.0	0.038462	0.008493	0.051020
3	0.0	0.096154	0.008493	0.775510
4	0.0	0.250000	0.016985	0.397959

```
data1['kclus']  = pd.Series(res)
data1.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	kclus
0	1.0	0.019231	0.000000	0.387755	1
1	1.0	0.057692	0.000000	0.816327	1
2	0.0	0.038462	0.008493	0.051020	6
3	0.0	0.096154	0.008493	0.775510	6
4	0.0	0.250000	0.016985	0.397959	6

```
data1['kclus'].unique()
```

```
array([1, 6, 3, 5, 7, 2, 0, 4], dtype=int32)
```

(no subject) - kamaliragupathi@ | Kamali_Assignment_4.ipynb - Co | IBM | github login - Yahoo India Search | IBM-EPBL/IBM-Project-2677-165

colab.research.google.com/drive/1u7RaTp12RwqGJKxfJUyUFWe6MlyTygjy

Kamali_Assignment_4.ipynb

File Edit View Insert Runtime Tools Help Last saved at 10:00 PM

Comment | Share

+ Code + Text

Connect | Editing

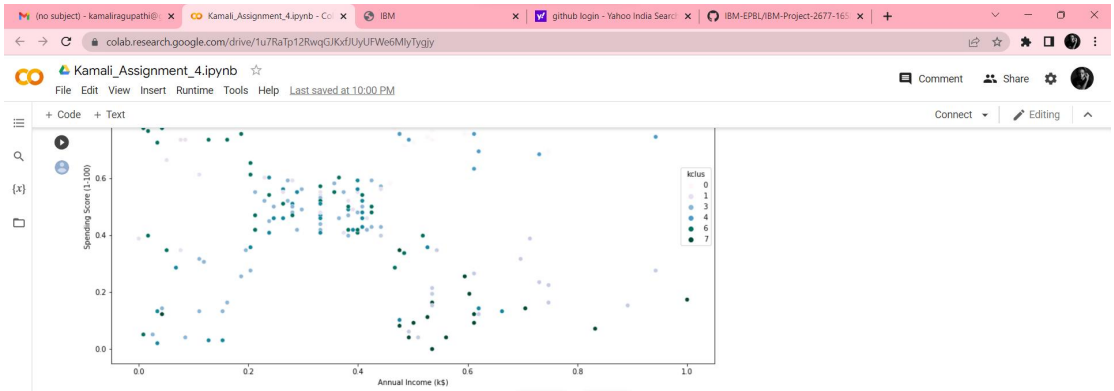
```
data1['kclus'].value_counts()
```

```
6    37
3    37
5    29
1    24
0    22
4    18
7    17
2    16
Name: kclus, dtype: int64
```

```
import matplotlib.pyplot as plt

fig,ax = plt.subplots(figsize=(15,8))
sns.scatterplot(x=data1['Annual Income (k$)'],
                y=data1['Spending Score (1-100)'],
                hue=data1['kclus'],
                palette='PuBuGn')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3295b7c350>
```




```
[ ] ind = data1.iloc[:,0:4]
    ind.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1.0	0.019231	0.000000	0.387755
1	1.0	0.057692	0.000000	0.816327
2	0.0	0.038462	0.008493	0.051020
3	0.0	0.096154	0.008493	0.775510
4	0.0	0.250000	0.016985	0.397959

```
[ ] dep = data1.iloc[:,4:]
    dep.head()
```

	kclus
0	1
1	1
2	6
3	6
4	6

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(ind,dep,test_size=0.3,random_state=1)
x_train.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
116	0.0	0.865385	0.424628	0.428571
67	0.0	0.961538	0.280255	0.479592
78	0.0	0.096154	0.331210	0.520408
42	1.0	0.576923	0.203822	0.357143
17	1.0	0.038462	0.050955	0.663265

```
[ ] x_test.head()
```

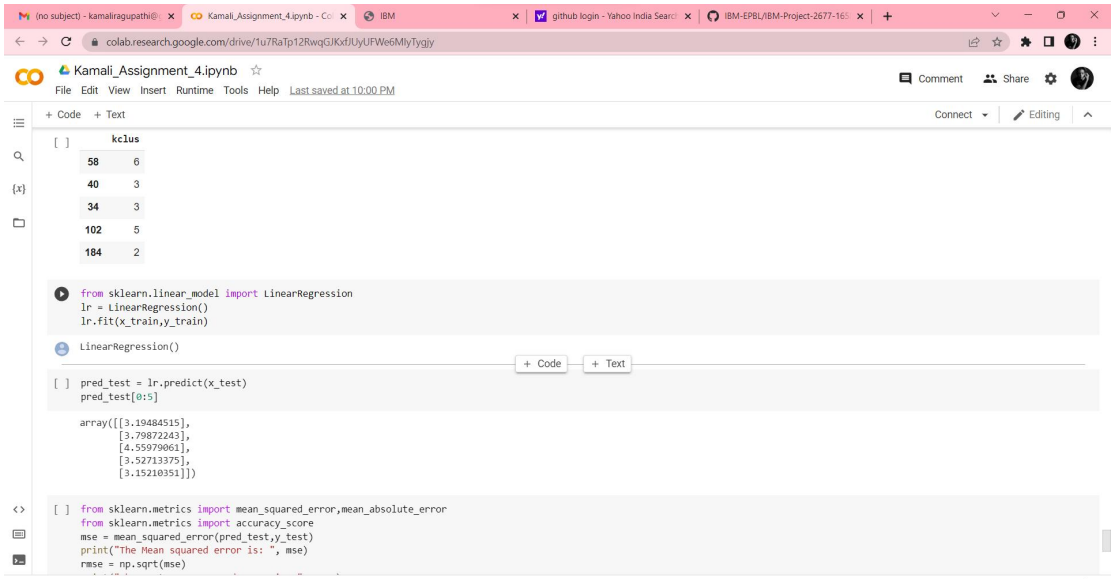
	Gender	Age	Annual Income (k$)	Spending Score (1-100)
58	0.0	0.173077	0.263270	0.510204
40	0.0	0.903846	0.195329	0.346939
34	0.0	0.596154	0.152866	0.132653
102	1.0	0.942308	0.399151	0.591837
184	0.0	0.442308	0.713376	0.387755

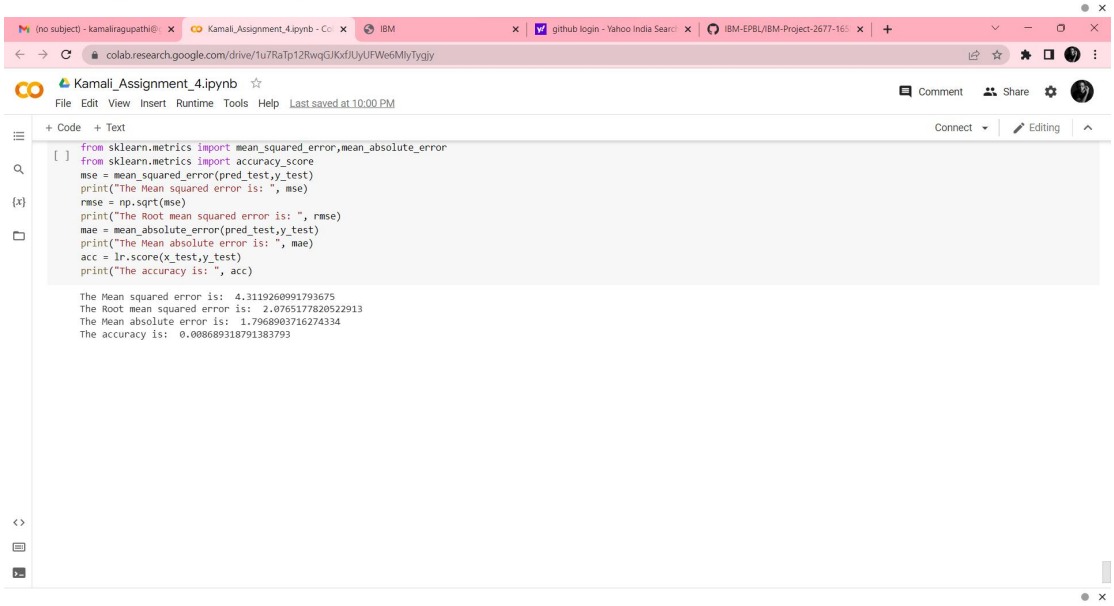
```
y_train.head()
```

	kclus
116	3
67	3
78	6
42	5
17	1

```
[ ] y_test.head()
```

	kclus


Kamali_Assignment_4.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:00 PM
Comment
Share
+ Code + Text
Connect
Editing
kclus
58 6
40 3
34 3
102 5
184 2
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
LinearRegression()
+ Code + Text
pred_test = lr.predict(x_test)
pred_test[0:5]
array([[3.19484515],
[3.79872243],
[4.55979061],
[3.52713375],
[3.15210351]])
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
mse = mean_squared_error(pred_test,y_test)
print("The Mean squared error is: ", mse)
rmse = np.sqrt(mse)


Kamali_Assignment_4.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:00 PM
Comment
Share
+ Code + Text
Connect
Editing
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
mse = mean_squared_error(pred_test,y_test)
print("The Mean squared error is: ", mse)
rmse = np.sqrt(mse)
print("The Root mean squared error is: ", rmse)
mae = mean_absolute_error(pred_test,y_test)
print("The Mean absolute error is: ", mae)
acc = lr.score(x_test,y_test)
print("The accuracy is: ", acc)
The Mean squared error is: 4.3119260991793675
The Root mean squared error is: 2.0765177820522913
The Mean absolute error is: 1.7968903716274334
The accuracy is: 0.008689318791383793