So we have to compare two machine learning algorithm to see which one is better based on our dataset.

Knn vs Logistc reggresion

I will inform you my answer here when i done the compare i will upload a picture of knn score and LG score !

Logistc reggresion score is 0.8780487804878049

knn score is 0.8926829268292683 k = 5 same as previous lab (lab 1)

thanks dr.saeed <3

```
In [287]:   # Let's import the packages that we need

            import numpy as np
            from collections import Counter
            import matplotlib.pyplot as plt
            import pandas as pd
```

```
In [265]:   # we have to load data (already we have split) so we have test and train

            datatrain=pd.read_csv('Train_set.csv')
            datatrain.head(10)
```

Out[265]:

	Height	Weight	Sex
0	165.65	35.41	Female
1	148.53	74.45	Female
2	167.04	81.22	Male
3	161.54	71.47	Male
4	174.31	78.18	Male
5	143.05	79.27	Female
6	149.16	63.71	Female
7	155.80	86.01	Female
8	145.37	88.33	Female
9	162.56	87.36	Male

In [266]:
```python
# we sholud understanding what the data is ?
# import first 10 (haed of table ) data in table
datatest=pd.read_csv('test_set.csv')
datatest.head(10)
```

Out[266]:

	Height	Weight	Sex
0	146.323241	59.861065	Female
1	175.695412	77.863687	Male
2	183.216164	72.131992	Male
3	184.245269	77.546000	Male
4	132.302261	55.188496	Female
5	149.863914	66.055655	Female
6	191.173088	83.631339	Male
7	135.407174	47.014457	Female
8	184.995763	71.935828	Male
9	179.990887	65.920795	Male

In [267]:

```python
df=datatrain
columns=['Sex']
df=df.drop(columns, axis=1)
df.head(15)
```

Out[267]:

	Height	Weight
0	165.65	35.41
1	148.53	74.45
2	167.04	81.22
3	161.54	71.47
4	174.31	78.18
5	143.05	79.27
6	149.16	63.71
7	155.80	86.01
8	145.37	88.33
9	162.56	87.36
10	163.84	78.58
11	172.73	88.08
12	142.03	56.30
13	150.17	53.46
14	188.62	60.95

In [268]:

```python
# to get the type
```

In [269]:

```python
type(df)
```

Out[269]: pandas.core.frame.DataFrame

In [270]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2998 entries, 0 to 2997
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Height  2998 non-null   float64
 1   Weight  2998 non-null   float64
dtypes: float64(2)
memory usage: 47.0 KB
```
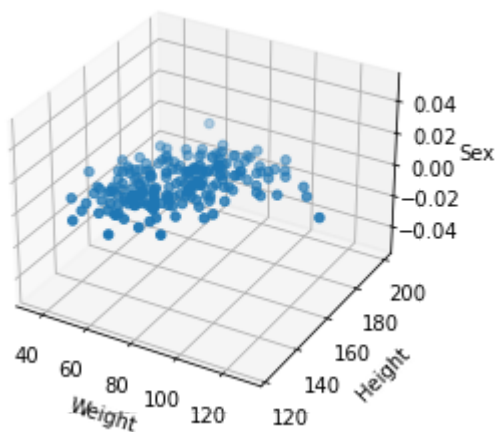
In [271]: `df.keys()`

Out[271]: `Index(['Height', 'Weight'], dtype='object')`

In [272]: `df.shape`

Out[272]: `(2998, 2)`

In [273]:
```python
# Visualization 3D
from mpl_toolkits.mplot3d import Axes3D
fig=plt.figure()
ax=fig.add_subplot(111, projection='3d')
ax.scatter(X_train['Weight'],X_train['Height'])
ax.set_xlabel('Weight')
ax.set_ylabel('Height')
ax.set_zlabel('Sex')
plt.show
```

Out[273]: `<function matplotlib.pyplot.show(close=None, block=None)>`

In [274]:
```python
#represent
x = datatrain['Height']
y=datatrain['Weight']
male=x[0:25]
female=x[5001:5026]
wm=y[0:25]
wf=y[5001:5026]
plt.scatter(male,wm, color='g')
plt.scatter(female,wf,color='r')
plt.xlabel('Male')
plt.ylabel('Female')
plt.show()
```
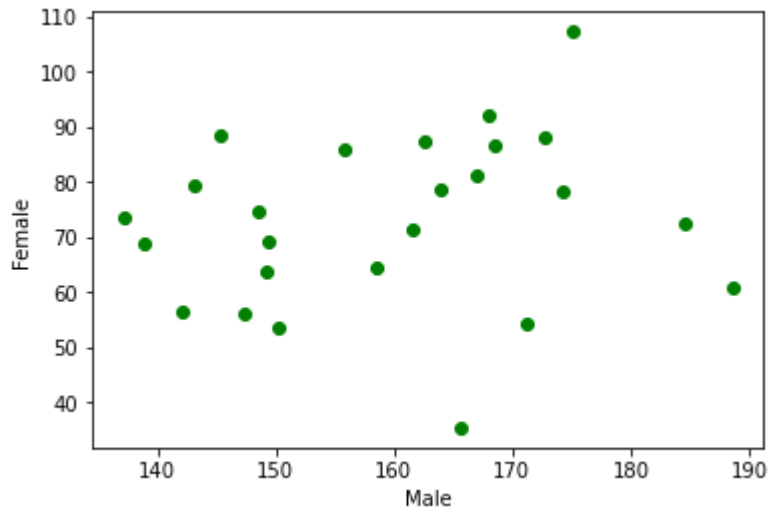

In [275]:
```python
testx=df[400:410]
testy=df[9000:9010]
frames=[testx,testy]
X=pd.concat(frames)
X=np.array(X)
print(X)
X.shape
```

```
[[177.22  63.66]
 [153.93  72.61]
 [160.53  63.31]
 [148.88  81.01]
 [164.37  71.42]
 [175.62  83.99]
 [178.62  80.94]
 [174.33  91.15]
 [180.87  75.98]
 [152.29  49.65]]
```

Out[275]: (10, 2)

In [276]:
```python
# and we have to drop the target from table ,
#Split the dataset into features and target
X_train=datatest.drop('Sex',axis='columns')
X_train
```

Out[276]:

	Height	Weight
0	146.323241	59.861065
1	175.695412	77.863687
2	183.216164	72.131992
3	184.245269	77.546000
4	132.302261	55.188496
...	...	...
200	155.090314	77.248911
201	149.175907	93.231692
202	168.030874	63.640623
203	172.608090	55.189983
204	145.082128	45.583285

205 rows × 2 columns

In [277]:
```python
y_train=datatest['Sex']
y_train
```

Out[277]:
```
0       Female
1         Male
2         Male
3         Male
4       Female
         ...
200     Female
201       Male
202     Female
203       Male
204     Female
Name: Sex, Length: 205, dtype: object
```

In [278]:
```python
# and we have to drop the target from table ,
#Split the dataset into features and target
X_test=datatest.drop('Sex',axis='columns')
X_test
```

Out[278]:

	Height	Weight
0	146.323241	59.861065
1	175.695412	77.863687
2	183.216164	72.131992
3	184.245269	77.546000
4	132.302261	55.188496
...	...	...
200	155.090314	77.248911
201	149.175907	93.231692
202	168.030874	63.640623
203	172.608090	55.189983
204	145.082128	45.583285

205 rows × 2 columns

In [279]:
```python
y_test=datatest['Sex']
y_test
```

Out[279]:
```
0       Female
1         Male
2         Male
3         Male
4       Female
         ...
200     Female
201       Male
202     Female
203       Male
204     Female
Name: Sex, Length: 205, dtype: object
```

In [280]:
```python
#Import package KNN
from sklearn.neighbors import KNeighborsClassifier
#Create KNN Classifier
knn = KNeighborsClassifier(n_neighbors=5 )
```

In [281]:
```python
#fit the model by data
```

In [282]:
```python
knn.fit(X_train,y_train)
```

Out[282]: KNeighborsClassifier()

In [283]:
```python
knn.predict(X_test)
```

Out[283]: array(['Female', 'Male', 'Male', 'Male', 'Female', 'Female', 'Male',
               'Female', 'Male', 'Male', 'Male', 'Male', 'Female', 'Female',
               'Male', 'Female', 'Male', 'Male', 'Male', 'Male', 'Female', 'Male',
               'Male', 'Female', 'Female', 'Female', 'Female', 'Male', 'Male',
               'Female', 'Male', 'Female', 'Female', 'Male', 'Male', 'Male',
               'Female', 'Male', 'Female', 'Male', 'Female', 'Female', 'Male',
               'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female',
               'Male', 'Male', 'Female', 'Male', 'Male', 'Male', 'Female', 'Male',
               'Male', 'Female', 'Female', 'Female', 'Female', 'Male', 'Male',
               'Male', 'Female', 'Male', 'Male', 'Male', 'Male', 'Female',
               'Female', 'Female', 'Male', 'Female', 'Female', 'Male', 'Female',
               'Female', 'Male', 'Female', 'Female', 'Female', 'Female', 'Female',
               'Male', 'Male', 'Male', 'Female', 'Female', 'Female', 'Male',
               'Female', 'Male', 'Male', 'Male', 'Female', 'Female', 'Male',
               'Male', 'Female', 'Female', 'Male', 'Male', 'Female', 'Female',
               'Male', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female',
               'Female', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female',
               'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Male',
               'Male', 'Female', 'Female', 'Female', 'Female', 'Female', 'Female',
               'Male', 'Male', 'Female', 'Female', 'Female', 'Male', 'Female',
               'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Male',
               'Male', 'Female', 'Female', 'Male', 'Female', 'Male', 'Male',
               'Female', 'Female', 'Male', 'Female', 'Male', 'Male', 'Male',
               'Male', 'Male', 'Female', 'Female', 'Male', 'Male', 'Male',
               'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Male',
               'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Male',
               'Male', 'Female', 'Female', 'Male', 'Male', 'Male', 'Female',
               'Male', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female',
               'Female', 'Female', 'Female', 'Male', 'Male', 'Male', 'Female'],
              dtype=object)

In [ ]:

In [284]:
```python
#accuracy

knn.score(X_test, y_test)
```

Out[284]: 0.8926829268292683

In [285]:
```python
# prediction system !

sex_prediction = knn.predict([[140, 38]])
lookup_sex_name=[sex_prediction[0]]
lookup_sex_name
```

C:\Users\PC\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
  warnings.warn(

Out[285]: ['Female']

In [286]:
```python
sex_prediction = knn.predict([[190, 120]])
lookup2_sex_name=[sex_prediction[0]]
lookup2_sex_name
```

C:\Users\PC\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
  warnings.warn(

Out[286]: ['Male']

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: