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 [1]:

```
# Let's import the packages that we need
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
%matplotlib inline
```

In [2]:

```
# we have to load data (already we have split) so we have test and train
dtest = pd.read_csv("test_set.csv")
dtrain = pd.read_csv("Train_set.csv")
```

In [3]:

```
# we sholud understanding what the data is ?
# import first 5 (haed of table ) data in table
```

In [4]:

```
dtrain.head()
```

Out[4]:

	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

In [ ]:

```
# same as previous
```

In [5]:

```
dtrain.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2998 entries, 0 to 2997
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Height  2998 non-null   float64
 1   Weight  2998 non-null   float64
 2   Sex     2998 non-null   object 
dtypes: float64(2), object(1)
memory usage: 70.4+ KB
```

In [ ]:

```
# to get the type
```

In [6]:

```
type(dtrain)
```

Out[6]:

```
pandas.core.frame.DataFrame
```

In [7]:

```
dtrain.keys()
```

Out[7]:

```
Index(['Height', 'Weight', 'Sex'], dtype='object')
```

In [ ]:

```
# here er can also know the shape of data (row,columns)
```

In [8]:

```
dtrain.shape
```

Out[8]:

```
(2998, 3)
```

In [ ]:

```
# i decided that i have to count how many fmale and male we have
```

In [9]:

```
dtrain.Sex.value_counts()
```

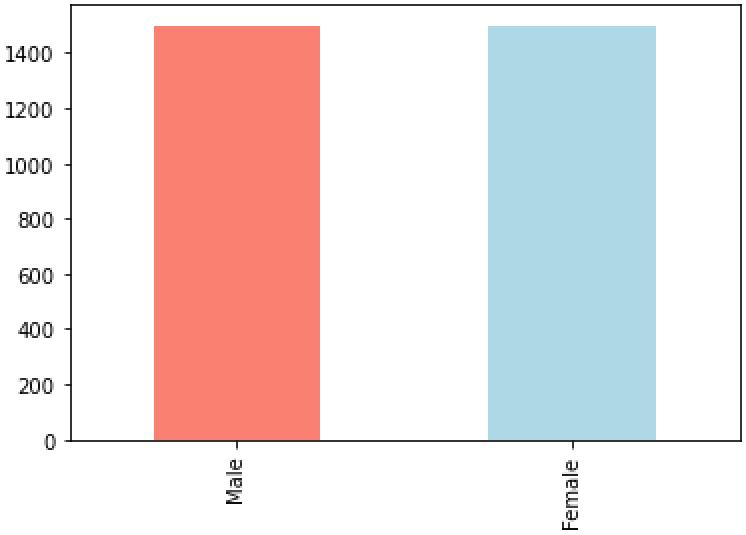
Out[9]:

```
Male      1500
Female    1498
Name: Sex, dtype: int64
```

In [ ]:

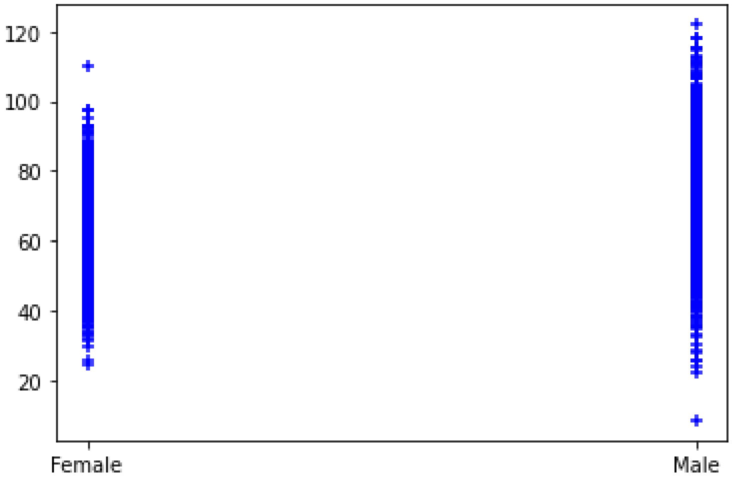
```
# represent part
```

In [10]:
```
dtrain.Sex.value_counts().plot(kind="bar", color=["salmon", "lightblue"]);
```



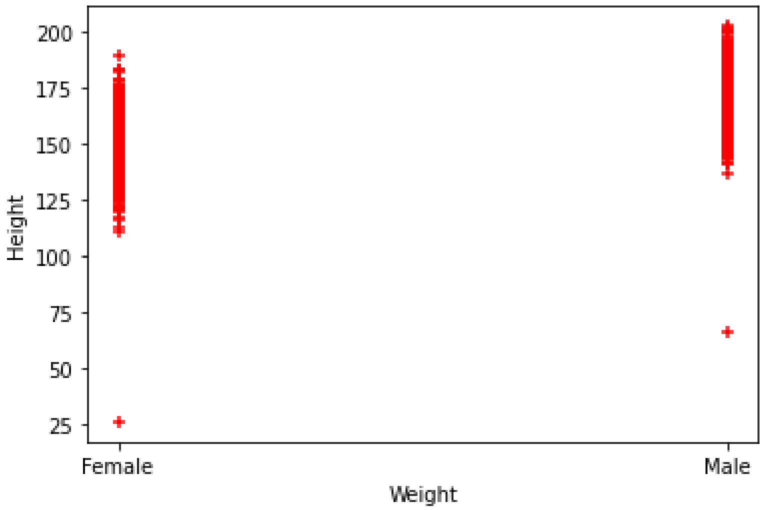
In [11]:
```
plt.scatter(dtrain.Sex,dtrain.Weight,marker='+',color='blue')
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x20b265cd9a0>
```



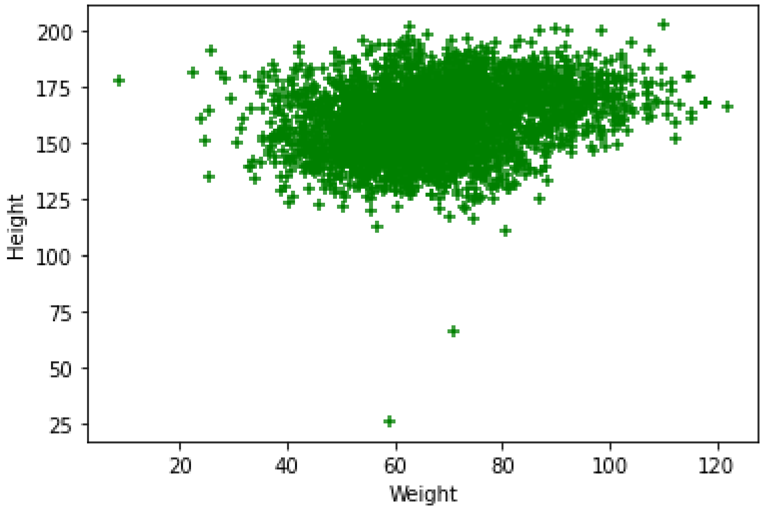
```
In [12]: plt.scatter(dtrain.Sex,dtrain.Height,marker='+',color='red')
         plt.ylabel("Height")
         plt.xlabel("Weight")
```

```
Out[12]: Text(0.5, 0, 'Weight')
```



```
In [13]: plt.scatter(dtrain.Weight,dtrain.Height,marker='+',color='green')
         plt.ylabel("Height")
         plt.xlabel("Weight")
```

```
Out[13]: Text(0.5, 0, 'Weight')
```



In [14]:

```
# and we have to drop the target from table ,
#Split the dataset into features and target
X_train=dtrain.drop('Sex',axis='columns')
X_train
```

Out[14]:

	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
...	...	...
2993	150.83	49.66
2994	157.09	64.34
2995	162.99	45.58
2996	154.76	48.92
2997	185.08	82.74

2998 rows × 2 columns

In [15]:

```
y_train=dtrain['Sex']
y_train
```

Out[15]:

```
0       Female
1       Female
2         Male
3         Male
4         Male
         ...
2993    Female
2994    Female
2995    Female
2996    Female
2997      Male
Name: Sex, Length: 2998, dtype: object
```

In [16]:

```
# and we have to drop the target from table ,
#Split the dataset into features and target

X_test=dtest.drop('Sex',axis='columns')
X_test
```

Out[16]:

	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 [17]:

```
y_test=dtest['Sex']
y_test
```

Out[17]:

```
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 [18]:

```
#Import package LogisticRegression

from sklearn.linear_model import LogisticRegression
lgrgmodel=LogisticRegression()
```

In [ ]:

```
# fti the model !
```

In [19]:

```
lgrgmodel.fit(X_train,y_train)
```

Out[19]:

```
LogisticRegression()
```

In [20]:

```
#optimization
lgrg_pred=lgrgmodel.predict(X_test)
lgrg_pred
```

Out[20]:

```
array(['Female', 'Male', 'Male', 'Male', 'Female', 'Female', 'Male',
       'Female', 'Male', 'Male', 'Male', 'Male', 'Female', 'Female',
       'Female', 'Female', 'Male', 'Male', 'Male', 'Male', 'Female',
       'Male', 'Male', 'Female', 'Female', 'Female', 'Female', 'Male',
       'Male', 'Female', 'Female', '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', 'Male',
       'Male', 'Male', 'Female', 'Female', 'Female', 'Female', 'Male',
       'Male', 'Male', 'Female', 'Male', 'Male', 'Male', 'Male', 'Female',
       'Female', 'Female', 'Male', 'Female', 'Female', 'Male', 'Male',
       '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', 'Male',
       'Male', 'Female', 'Male', 'Male', 'Male', 'Female', 'Female',
       'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Male',
       'Male', 'Female', 'Female', 'Female', 'Female', 'Male', 'Female',
       'Male', 'Male', 'Male', 'Female', 'Female', 'Male', 'Female',
       'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Male',
       'Male', 'Female', 'Female', 'Male', 'Male', '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', 'Male', 'Female',
       'Female', 'Female', 'Female', 'Female', 'Male', 'Male', 'Female'],
      dtype=object)
```

In [21]:

```
# get the accuracy and compare it with Knn
# our accuracy is 87
lgrgmodel.score(X_test,y_test)
```

Out[21]:

```
0.8780487804878049
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: