In [0]:

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

In [2]:

```
data=pd.read_csv('https://raw.githubusercontent.com/shawon100/Web-Log-Dataset/master/weight-height
.csv')
data.head(10)
```

Out[2]:

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801
5	Male	67.253016	152.212156
6	Male	68.785081	183.927889
7	Male	68.348516	167.971110
8	Male	67.018950	175.929440
9	Male	63.456494	156.399676

In [3]:

```
df=data
columns=['Gender']
df = df.drop(columns, axis=1)
df.head(10)
```

Out[3]:

	Height	Weight
0	73.847017	241.893563
1	68.781904	162.310473
2	74.110105	212.740856
3	71.730978	220.042470
4	69.881796	206.349801
5	67.253016	152.212156
6	68.785081	183.927889
7	68.348516	167.971110
8	67.018950	175.929440
9	63.456494	156.399676

In [ ]:

```
Xm=df[0:25]
Xf=df[5001:5026]
frame=[Xm,Xf]
X_train=pd.concat(frame)
```

```
X_train=pd.concat(frames)
X_train=np.array(X_train)
print(X_train)
X_train.shape
```

In [ ]:

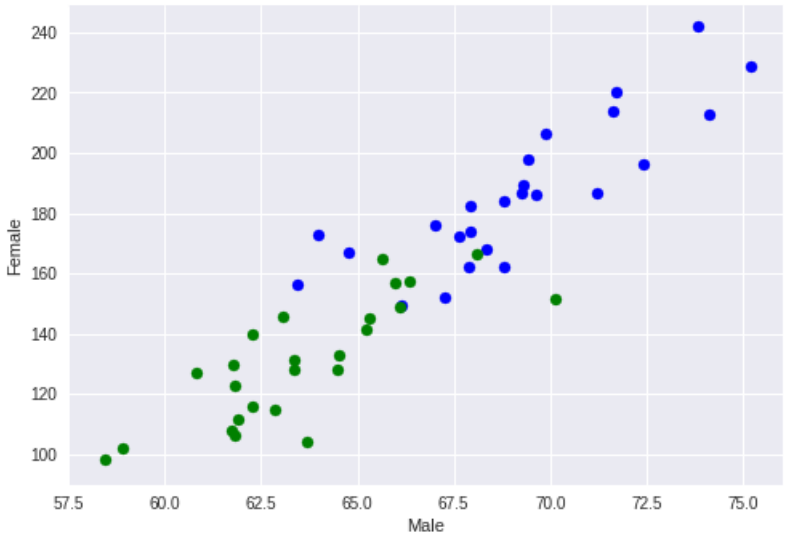
```
lbl=data['Gender']
ym=lbl[0:25]
yf=lbl[5001:5026]
frames=[ym,yf]
Y_train=pd.concat(frames)
Y_train=np.array(Y_train)
print(Y_train)
Y_train.shape
```

In [ ]:

```
label=data['Gender']
lb=label[400:410]
lc=label[9000:9010]
frames=[lb,lc]
Y=pd.concat(frames)
Y=np.array(Y)
print(Y)
```

In [10]:

```
x = data['Height']
y=data['Weight']
male=x[0:25]
female=x[5001:5026]
wm=y[0:25]
wf=y[5001:5026]
plt.scatter(male,wm, color='b')
plt.scatter(female,wf,color='g')
plt.xlabel('Male')
plt.ylabel('Female')
plt.show()
```



In [ ]:

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

In [0]:

In [0]:

```
def predict(input_feature_set, k):
    distances = []
    z=0
    for training_feature_set in X_train:
        group=Y_train[z]
        #print("Group=",group)
        #print("Training Feature=",training_feature_set)
        euclidean_distance = np.linalg.norm(np.array(input_feature_set) -
np.array(training_feature_set))
        #print("Distance=",euclidean_distance)
        distances.append([euclidean_distance, group])
        z=z+1
        #print(z)

    nearest = sorted(distances)[:k]
    #print("Sorted=",nearest)
    votes=[]
    #votes = [d[1] for d in nearest]
    for d in nearest:
        votes.append(d[1])
    #print(votes)
    #prediction = Counter(votes).most_common(1)[0][0]
    item={}
    for i in votes:
        if i in item:
            item[i]=item[i]+1
        else:
            item[i]=1

    #finding most common class
    m=0
    for k in item:
        if item[k]>m:
            m=item[k]

    for k in item:
        if item[k]==m:
            index=k

    prediction=index


    return prediction
```

In [ ]:

```
#Training
toutput=[]
for j in X_train:
    predicted=predict(j,3)
    toutput.append(predicted)
print("Output=",toutput)
print(len(toutput))
```

In [24]:

```
c=0
for p in range(0,50):
        if toutput[p]==Y_train[p]:
            c=c+1
result=(c/50.0)*100
print("Training Accuracy=",result,"%")
```

```
Training Accuracy= 94.0 %
```

In [ ]:

```
output=[]
for j in X:
    predicted=predict(j,3)
    output.append(predicted)
print("Output=",output)
```

In [26]:

```
c=0
for p in range(0,20):
        if output[p]==Y[p]:
            c=c+1
result=(c/20.0)*100
print("Testing Accuracy=",result,"%")
```

```
Testing Accuracy= 80.0 %
```

In [0]:

```

```