# Pizza Liking Prediction using KNN

## Name:P.Asha Belcilda

## Roll no:225229104

In [1]:
```
import pandas as pd
```

## Step2.Import dataset

In [2]:
```
pizza_data=pd.read_csv('pizza.csv')
pizza_data.head()
```

Out[2]:

| | age | weight | likePizza |
|---|---|---|---|
| **0** | 50 | 65 | 0 |
| **1** | 20 | 55 | 1 |
| **2** | 15 | 40 | 1 |
| **3** | 70 | 65 | 0 |
| **4** | 30 | 70 | 1 |

In [5]:
```
#shape
pizza_data.shape
```

Out[5]: (6, 3)

In [6]:
```
#columns
pizza_data.shape[1]
```

Out[6]: 3

In [7]:
```
#info
pizza_data.info()
```

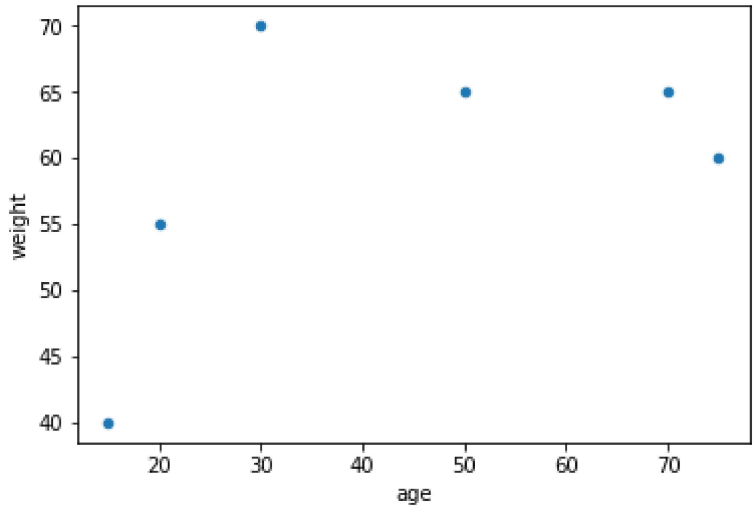
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
age          6 non-null int64
weight       6 non-null int64
likePizza    6 non-null int64
dtypes: int64(3)
memory usage: 224.0 bytes
```

## Step3.Visualize Relationship

In [94]:
```
import matplotlib.pyplot as plt
```

In [106]:
```
df=pd.read_csv("pizza.csv")
df.plot(kind='scatter',x='age',y='weight')
```

Out[106]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1edd4a636d8>
```




## Step-4.Prepare X matrix and y vector

In [84]:
```
x=pd.DataFrame(pizza_data)
cols=[0,1]
x=x[x.columns[cols]]
```

In [85]:
```
y=pizza_data['likePizza'].values
```

## Step-5.Examine x and y

In [86]:
```
x
```

Out[86]:

| | age | weight |
|---|---|---|
| **0** | 50 | 65 |
| **1** | 20 | 55 |
| **2** | 15 | 40 |
| **3** | 70 | 65 |
| **4** | 30 | 70 |
| **5** | 75 | 60 |

In [100]:
```
type(x)
```

Out[100]:
```
pandas.core.frame.DataFrame
```

In [101]:
```
Y
```

Out[101]:
```
array([1, 1, 0, 0])
```

In [102]:
```
type(y)
```

Out[102]:
```
numpy.ndarray
```

## Step-6.Model building

In [123]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=2)
knn.fit(x,y)
```

Out[123]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

## Step-7.Model testing

In [124]:
```
knn.predict(X)
```

Out[124]:
```
array([0, 1, 1, 0], dtype=int64)
```

In [125]:
```
a=[25,50]
knn.predict([a])
```

Out[125]:
```
array([1], dtype=int64)
```

In [126]:
```
b=[60,60]
knn.predict([b])
```

Out[126]:
```
array([0], dtype=int64)
```

## Step-8.Change n_neighbours=3

In [93]:
```
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(x,y)
```

Out[93]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
```

In [33]:
```
c=[25,50]
knn.predict([c])
```

Out[33]:
```
array([1], dtype=int64)
```

In [35]:
```
d=[60,60]
knn.predict([d])
```

Out[35]:
```
array([0], dtype=int64)
```

## Step-9.Predict on entire dataset

In [36]:
```
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x,y)
```

Out[36]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

In [37]:
```
y_pred=knn.predict(x)
y_pred
```

Out[37]:
```
array([0, 1, 1, 0, 1, 0], dtype=int64)
```

## Step-10.Accuracy function

In [39]:
```
def accuracy(actual,pred):
    return sum(actual==pred)/float(actual.shape[0])
```

## Step-11.Find Accuracy

In [40]:
```
accuracy_score=accuracy(y,y_pred)
accuracy_score
```

Out[40]:
```
1.0
```

## Step-12.Prediction on Test set

In [45]:
```
import pandas as pd
df=pd.read_csv("pizza_test.csv")
df.head()
```

Out[45]:

| | age | weight | likepizza |
|---|---|---|---|
| **0** | 48 | 68 | 1 |
| **1** | 35 | 45 | 1 |
| **2** | 15 | 40 | 0 |
| **3** | 55 | 68 | 0 |

In [46]:
```
#shape
df.shape
```

Out[46]:
```
(4, 3)
```

In [48]:
```
#columns
df.shape[1]
```

Out[48]:
```
3
```

In [49]:
```
#info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
age          4 non-null int64
weight       4 non-null int64
likepizza    4 non-null int64
dtypes: int64(3)
memory usage: 176.0 bytes
```

In [50]:
```
x=pd.DataFrame(df)
cols=[0,1]
x=x[x.columns[cols]]
```

In [51]:
```
x
```

Out[51]:

| | age | weight |
|---|---|---|
| **0** | 48 | 68 |
| **1** | 35 | 45 |
| **2** | 15 | 40 |
| **3** | 55 | 68 |

In [52]:
```
Y=df['likepizza'].values
Y
```

Out[52]:
```
array([1, 1, 0, 0], dtype=int64)
```

In [53]:
```
from sklearn.neighbors import KNeighborsClassifier
test=KNeighborsClassifier(n_neighbors=2)
test.fit(x,Y)
```

Out[53]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

In [54]:
```
Y_pred=test.predict(x)
Y_pred
```

Out[54]:
```
array([0, 0, 0, 0], dtype=int64)
```

In [55]:
```
import numpy as np
Y=np.array([1,1,0,0])
Y
```

Out[55]:
```
array([1, 1, 0, 0])
```

In [56]:
```
Y_test=accuracy(Y,Y_pred)
Y_test
```

Out[56]:
```
0.5
```

## Step-13.Find best value for k

In [59]:
```
scores=[]
for k in range(1,4):
    kn=KNeighborsClassifier(n_neighbors=k)
    kn.fit(x,Y)
    kn.predict(x)
    y_test=kn.predict(x)
    a=accuracy(Y,Y_pred)
    scores.append((k,a))
print(scores)
```

```
[(1, 0.5), (2, 0.5), (3, 0.5)]
```

## Step-14.Accuracy_score function

In [60]:
```
from sklearn.metrics import accuracy_score
```

In [61]:
```
accuracy_score(Y,Y_pred)
```

Out[61]:
```
0.5
```