LAB 2: Pizza Liking Prediction using kNN

Name : MURALI KUMAR R

Roll no : 225229120

In [1]:

```
import pandas as pd
```

Step 2 : Import dataset:

In [11]:

```
#head:
pizza_data=pd.read_csv('pizza.csv')
pizza_data.head()
```

Out[11]:

	AGE	WEIGHT	LIKEPIZZA
0	50	65	0
1	20	55	1
2	15	40	1
3	70	65	0
4	30	70	1

In [14]:

```
#shape:
pizza_data.shape
```

Out[14]:

```
(6, 3)
```

In [18]:

```
#column shape:
pizza_data.shape[1]
```

Out[18]:

```
3
```

In [19]:

```
#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
```

Step 3 : Visualize Relationships:

In [69]:

```
import matplotlib.pyplot as plt
```

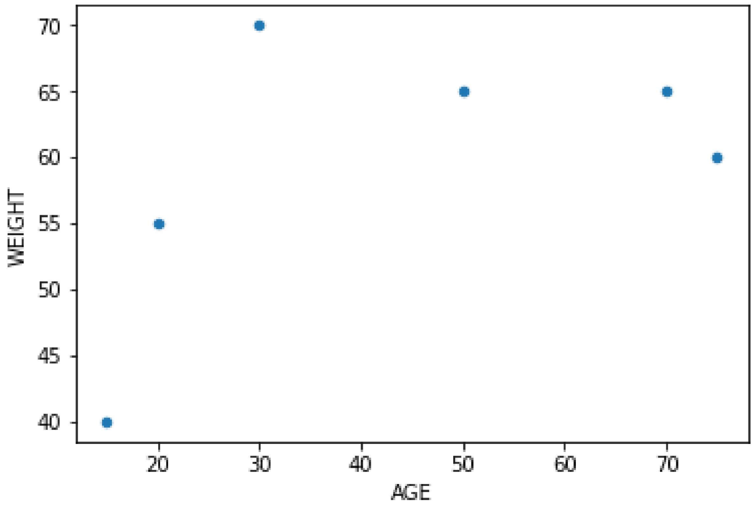
In [70]:

```
#plot:
df  = pd.read_csv("pizza.csv")

df.plot(kind='scatter',x='AGE',y='WEIGHT')
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f2dd513c50>
```



Step 4 : Prepare X matrix and y vectors:

In [54]:

```
#extract age' and 'weight' columns and store into new dataframe X:
X=pd.DataFrame(pizza_data)
cols=[0,1]
X=X[X.columns[cols]]
```

In [57]:

```
#extract 'likepizza' column and store into y:
y=pizza_data['LIKEPIZZA'].values
```

Step 5: Examine X and y:

In [65]:

```
#print X:
X
```

Out[65]:

	AGE	WEIGHT
0	50	65
1	20	55
2	15	40
3	70	65
4	30	70
5	75	60

In [66]:

```
#type of X:
type(X)
```

Out[66]:

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

In [67]:

```
#print y:
y
```

Out[67]:

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

In [68]:

```
#type of y:
type(y)
```

Out[68]:

```
numpy.ndarray
```

Step 6 : Model builing:

In [72]:

```
#create KNeighborsClassifier(n_neighbours=2) from sklearn:
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=2)

knn.fit(X,y)
```

Out[72]:

```
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 [73]:

```
#using your KNN model, predict if a person will like Pizza or not:
knn.predict(X)
```

Out[73]:

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

In [74]:

```
#will a  person who is 25 years with weigth 50kgs like pizza or not:
a=[25,50]
knn.predict([a])
```

Out[74]:

```
array([1], dtype=int64)
```

In [75]:

```
#will a person who is 60 years with weight 60kgs like pizza or not:
b=[60,60]
knn.predict([b])
```

Out[75]:

```
array([0], dtype=int64)
```

Step 8 : Change n_neighbours=3:

In [76]:

```
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X,y)
```

Out[76]:

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

In [77]:

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

Out[77]:

```
array([1], dtype=int64)
```

In [79]:

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

Out[79]:

```
array([0], dtype=int64)
```

Step 9 : Predict on entire dataset:

In [80]:

```
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(X,y)
```

Out[80]:

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

In [81]:

```
y_pred=knn.predict(X)
y_pred
```

Out[81]:

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

Step 10 : Accuracy function:

In [83]:

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

Step 11 : Find accuracy:

In [84]:

```
accuracy_score=accuracy(y,y_pred)
accuracy_score
```

Out[84]:

```
1.0
```

Step 12 : Prediction on Test set:

In [108]:

```
import pandas as pd
df=pd.read_csv("pizza_text.csv")
```

In [90]:

```
df.head()
```

Out[90]:

	age	weight	likepizza
0	48	68	1
1	35	45	1
2	15	40	0
3	55	65	0

In [91]:

```
df.shape
```

Out[91]:

```
(4, 3)
```

In [92]:

```
df.shape[0]
```

Out[92]:

```
4
```

In [93]:

```
df.shape[1]
```

Out[93]:

```
3
```

In [94]:

```
df.size
```

Out[94]:

```
12
```

In [95]:

```
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 [96]:
```
x=pd.DataFrame(df)
cols=[0,1]
x=x[x.columns[cols]]
```

In [97]:
```
x
```

Out[97]:

	age	weight
0	48	68
1	35	45
2	15	40
3	55	65

In [98]:
```
Y=df['likepizza'].values
```

In [99]:
```
Y
```

Out[99]:
```
array([1, 1, 0, 0], dtype=int64)
```

In [100]:
```
from sklearn.neighbors import KNeighborsClassifier
test=KNeighborsClassifier(n_neighbors=2)
test.fit(x,Y)
```

Out[100]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

In [101]:
```
Y_pred=test.predict(x)
Y_pred
```

Out[101]:
```
array([0, 0, 0, 0], dtype=int64)
```

In [102]:
```
import numpy as np
y=np.array([1,1,0,0])
y
```

Out[102]:
```
array([1, 1, 0, 0])
```

In [104]:
```
Y_test=accuracy(Y,Y_pred)
Y_test
```

Out[104]:
```
0.5
```

Step 13 : Find best value for k:

In [105]:

```
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_scroe function:

In [106]:

```
from sklearn.metrics import accuracy_score
```

In [107]:

```
accuracy_score(Y,Y_pred)
```

Out[107]:

```
0.5
```

In [ ]:

```
```