labsheet 2 Pizza liking prediction using kNN

Step 1: Creating two datasets

Step 2: Import dataset

In [36]:

```
import pandas as pd
dr = pd.read_csv(r'C:\Users\1mscdsa51\Documents\pizza.csv')
dr.head()
```

Out[36]:

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

In [37]:

```
dr.shape
```

Out[37]:

```
(6, 3)
```

In [38]:

```
dr.shape[1]
```

Out[38]:

```
3
```

In [73]:

```
dr.columns
```

Out[73]:

```
Index(['age ', 'weight', 'likePizza'], dtype='object')
```

In [39]:

```
dr.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 Relationship

In [111]:

```
import seaborn as sns
sns.relplot(x='age',y='weight',hue='likepizza',data=dr)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-111-e7d4eb4dc1a9> in <module>()
      1 import seaborn as sns
----> 2 sns.relplot(x='age',y='weight',hue='likepizza',data=dr)

AttributeError: module 'seaborn' has no attribute 'relplot'
```

Step 4: Prepare X matrix and y vector

In [50]:

```
X=pd.DataFrame(dr)
cols=[0,1]
X=X[X.columns[cols]]
X
```

Out[50]:

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

In [51]:

```
y=dr['likePizza'].values
y
```

Out[51]:

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

Step 5: Examine x and y

In [47]:

```
print(y)
```

```
[0 1 1 0 1 0]
```

In [46]:

```
print(X)
```

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

In [48]:

```
type(X)
```

Out[48]:

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

In [49]:

```
type(y)
```

Out[49]:

```
numpy.ndarray
```

Step 6: Model building

In [75]:

```
from sklearn.neighbors import KNeighborsClassifier as ts
knn=ts(n_neighbors=2)
knn.fit(X,y)
```

Out[75]:

```
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 [53]:

```
knn.predict(X)
```

Out[53]:

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

In [60]:

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

Out[60]:

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

In [61]:

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

Out[61]:

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

Step 8: Change n_neighbors=3

In [76]:
```
knn=ts(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]:
```
s=[25,50]
knn.predict([s])
```

Out[77]:
```
array([1], dtype=int64)
```

In [78]:
```
e=[60,60]
knn.predict([e])
```

Out[78]:
```
array([0], dtype=int64)
```

Step 9: Predict on entire dataset

In [79]:
```
knn=ts(n_neighbors=5)
knn.fit(X,y)
```

Out[79]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
           weights='uniform')
```

In [80]:
```
y_pred=knn.predict(X)
y_pred
```

Out[80]:
```
array([0, 1, 1, 0, 1, 0], dtype=int64)
```

Step 10: Accuracy function

In [72]:
```
def accuracy(actual,pred):
    return sum(actual==pred)/float(actual.shape[0])
```

Step 11 Find Accurancy

In [82]:
```
accuracy_score=accuracy(y,y_pred)
accuracy_score
```

Out[82]:
```
1.0
```

Step 12: Prediction on test set

In [83]:
```
ds = pd.read_csv(r'C:\Users\1mscdsa51\Documents\pizza_test.csv')
ds.head()
```

Out[83]:

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

In [85]:
```
ds.shape
```

Out[85]:
```
(4, 3)
```

In [86]:
```
ds.columns
```

Out[86]:
```
Index(['age', 'weight', 'likePizza'], dtype='object')
```

In [87]:
```
ds.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 [93]:
```
S=pd.DataFrame(ds)
cols=[0,1]
S=S[S.columns[cols]]
S
```

Out[93]:

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

In [94]:
```
r=ds['likePizza'].values
r
```

Out[94]:
```
array([1, 1, 0, 0], dtype=int64)
```

```
In [99]: print(S)
         type(S)
```

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

```
Out[99]: pandas.core.frame.DataFrame
```

```
In [100]: print(r)
          type(r)
```

```
[1 1 0 0]
```

```
Out[100]: numpy.ndarray
```

```
In [97]: from sklearn.neighbors import KNeighborsClassifier as rs
         ts=rs(n_neighbors=2)
         ts.fit(S,r)
```

```
Out[97]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=1, n_neighbors=2, p=2,
                    weights='uniform')
```

```
In [106]: r_pred=ts.predict(S)
          r_pred
```

```
Out[106]: array([0, 0, 0, 0], dtype=int64)
```

```
In [107]: import numpy as np
          r=np.array([1,1,0,0])
          r
```

```
Out[107]: array([1, 1, 0, 0])
```

```
In [108]: r_test=accuracy(r,r_pred)
          r_test
```

```
Out[108]: 0.5
```

Step 13 Find the best value for K

In [110]:

```
scores=[]
for k in range(1,4):
    kn=KNeighborsClassifier(n_neighbors=k)
    kn.fit(S,r)
    kn.predict(S)
    y_test=kn.predict(S)
    a=accuracy(r,r_pred)
    scores.append((k,a))

print(scores)
```

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

Step 14: Accuracy_score function

In [112]:

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

In [114]:

```
accuracy_score(r,r_pred)
```

Out[114]:

```
0.5
```

In [ ]: