

NAME: Nivetha M

ROLL NO.: 215229126

LAB 2 PIZZA LIKING PREDICTION USING KNN

STEP1: PREPARE YOUR DATASET

STEP 2: IMPORT DATASET

```
In [140]: 1 import pandas as pd
```

```
In [141]: 1 df=pd.read_csv("pizza.csv")  
2 df
```

```
Out[141]:
```

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

```
In [142]: 1 df.head()
```

```
Out[142]:
```

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

```
In [143]: 1 df.shape
```

```
Out[143]: (6, 3)
```

```
In [144]: 1 df.columns
```

```
Out[144]: Index(['age', 'weight', 'likePizza'], dtype='object')
```

In [145]:

```
1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   age         6 non-null      int64  
1   weight      6 non-null      int64  
2   likePizza   6 non-null      int64  
dtypes: int64(3)
memory usage: 272.0 bytes
```

STEP 3: VISUALIZE RELATIONSHIPS

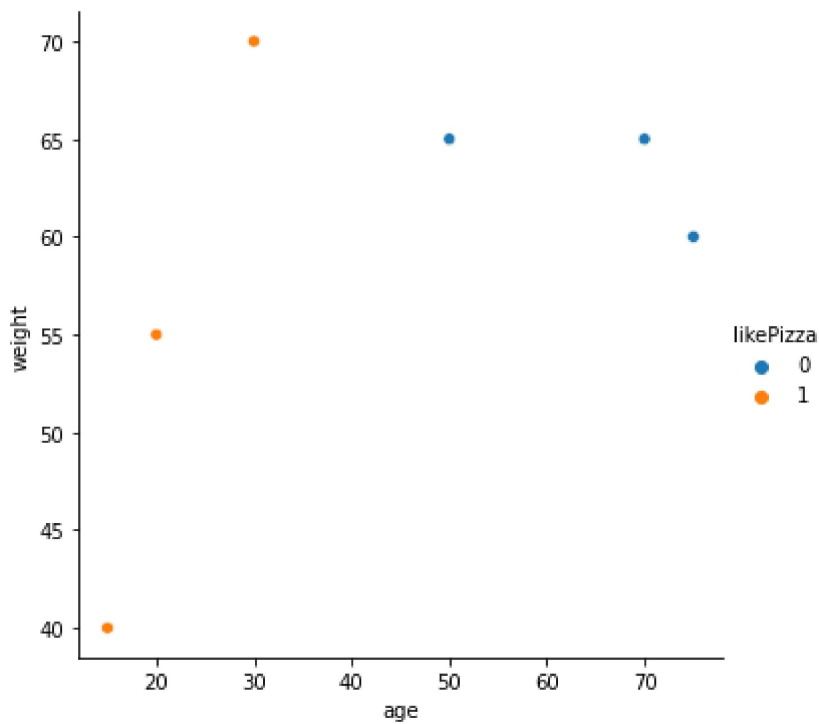
In [146]:

```
1 import seaborn as sns
```

In [147]:

```
1 sns.relplot(x='age',y='weight',hue="likePizza",data=df)
```

Out[147]: <seaborn.axisgrid.FacetGrid at 0x25972190070>



Step4:Prepare X matrix and y vector

```
In [148]: 1 X = df[["age", "weight"]]
          2 X
```

```
Out[148]:
```

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

```
In [149]: 1 y = df[["likePizza"]]
          2 y
```

```
Out[149]:
```

	likePizza
0	0
1	1
2	1
3	0
4	1
5	0

Step5:Examine X and Y

```
In [150]: 1 type(X)
```

```
Out[150]: pandas.core.frame.DataFrame
```

```
In [151]: 1 type(y)
```

```
Out[151]: pandas.core.frame.DataFrame
```

Step6:Model building

```
In [152]: 1 from sklearn.neighbors import KNeighborsClassifier
          2 neigh = KNeighborsClassifier(n_neighbors=2)
```

```
In [153]: 1 neigh.fit(X,y)
```

```
C:\Users\Harini\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:179: Dat
aConversionWarning: A column-vector y was passed when a 1d array was expected. Please cha
nge the shape of y to (n_samples,), for example using ravel().
    return self._fit(X, y)
```

```
Out[153]: KNeighborsClassifier(n_neighbors=2)
```

```
In [154]: 1 neigh.predict(X)
```

```
Out[154]: array([0, 1, 1, 0, 1, 0], dtype=int64)
```

STEP 7: MODEL TESTING

```
In [155]: 1 data1=[25,50]
          2 neigh.predict([data1])
```

```
Out[155]: array([1], dtype=int64)
```

```
In [156]: 1 data1=[60,60]
          2 neigh.predict([data1])
```

```
Out[156]: array([0], dtype=int64)
```

STEP 8: CHANGE n NEIGHBORS=3

```
In [157]: 1 from sklearn.neighbors import KNeighborsClassifier
          2 neigh = KNeighborsClassifier(n_neighbors=3)
```

```
In [158]: 1 neigh.fit(X,y)
```

```
C:\Users\Harini\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:179: Dat
aConversionWarning: A column-vector y was passed when a 1d array was expected. Please cha
nge the shape of y to (n_samples,), for example using ravel().
    return self._fit(X, y)
```

```
Out[158]: KNeighborsClassifier(n_neighbors=3)
```

```
In [159]: 1 data1=[25,50]
          2 neigh.predict([data1])
```

```
Out[159]: array([1], dtype=int64)
```

```
In [160]: 1 data1=[25,50]
          2 neigh.predict([data1])
```

```
Out[160]: array([1], dtype=int64)
```

STEP 10: ACCURACY FUNCTION

```
In [161]: 1 def accuracy(actual,predict):
          2     return sum(actual == predict)/ float(actual.shape[0])
```

```
In [164]: 1 Y_pred=df.likePizza
          2
```

```
In [165]: 1 y_predict = neigh.predict(X)
          2 y_predict
```

```
Out[165]: array([0, 1, 1, 0, 1, 0], dtype=int64)
```

```
In [169]: 1 accuracy(Y_pred,y_predict)
```

```
Out[169]: 1.0
```

STEP 12: PREDICTION ON TEST SET

```
In [179]: 1 import pandas as pd
```

```
In [180]: 1 df2=pd.read_csv("pizza_test.csv")
          2 df2
```

```
Out[180]:
```

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

```
In [181]: 1 df2.head()
```

```
Out[181]:
```

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

```
In [182]: 1 df2.shape
```

```
Out[182]: (4, 3)
```

```
In [183]: 1 df2.columns
```

```
Out[183]: Index(['age', 'weight', 'likePizza'], dtype='object')
```

```
In [184]: 1 df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         4 non-null      int64
1   weight      4 non-null      int64
2   likePizza   4 non-null      int64
dtypes: int64(3)
memory usage: 224.0 bytes
```

```
In [185]: 1 X = df2[["age","weight"]]
          2 X
```

```
Out[185]:
```

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

```
In [186]: 1 y = df2[["likePizza"]]
          2 y
```

```
Out[186]:
```

	likePizza
0	1
1	1
2	0
3	0

```
In [187]: 1 from sklearn.neighbors import KNeighborsClassifier
          2
```

```
In [188]: 1 testing = KNeighborsClassifier(n_neighbors =2)
          2 testing.fit(X,y)
```

C:\Users\Harini\anaconda3\lib\site-packages\sklearn\neighbors_classification.py:179: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
return self._fit(X, y)
```

```
Out[188]: KNeighborsClassifier(n_neighbors=2)
```

```
In [189]: 1 testing.predict(X)
```

```
Out[189]: array([0, 0, 0, 0], dtype=int64)
```

```
In [194]: 1 a=df2.likePizza
```

```
In [195]: 1 y_testingpred = testing.predict(X)
          2 y_testingpred
```

```
Out[195]: array([0, 0, 0, 0], dtype=int64)
```

```
In [196]: 1 accuracy(a,y_testingpred)
```

```
Out[196]: 0.5
```

STEP 13: FIND THE BEST VALUE FOR k

```
In [65]: 1 scores = []
          2 for k in range(1,4):
          3     knn = KNeighborsClassifier(n_neighbors = k)
          4     knn.fit(X,y)
          5     knn.predict(X)
          6     y_pr = knn.predict(X)
          7     a = accuracy(y,y_pr)
          8     scores.append((k,a))
```

```
In [66]: 1 scores
```

```
Out[66]: [(1, 1.0), (2, 1.0), (3, 1.0)]
```

STEP 14: ACCURACY_SCORE FUNCTION

```
In [67]: 1 from sklearn.metrics import accuracy_score
```

```
In [68]: 1 accuracy_score(y,y_pred)
```

```
Out[68]: 1.0
```