

Name:Viviyen Richards W

Roll no:205229133

step 2

```
In [9]: import pandas as pd
```

```
In [10]: df = pd.read_csv("pizza.csv")
```

```
In [11]: df.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 [12]: df.shape
```

```
Out[12]: (6, 3)
```

```
In [13]: for col in df.columns:  
          print(col)
```

```
age  
weight  
likepizza
```

```
In [14]: df.columns
```

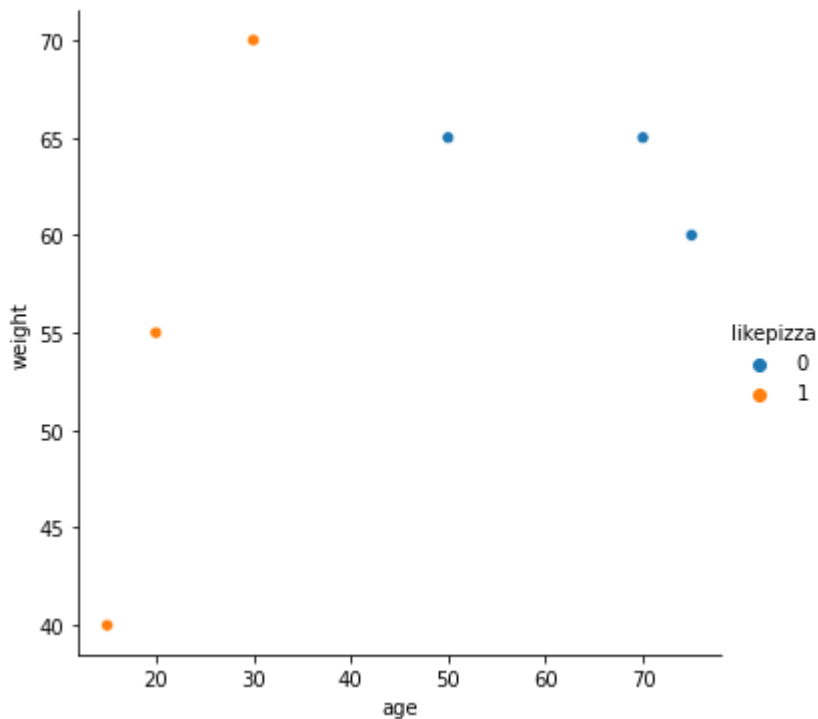
```
Out[14]: Index(['age', 'weight', 'likepizza'], dtype='object')
```

step:3

```
In [15]: import seaborn as sns
```

```
In [16]: sns.relplot(x="age", y="weight", data=df, hue='likepizza')
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x2b5101f7040>
```



step:4

```
In [74]: y=df.likepizza  
deat = ['age', 'weight']  
x=df[deat]
```

```
In [75]: x
```

```
Out[75]:
```

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

```
In [76]: y
```

```
Out[76]: 0    0
          1    1
          2    1
          3    0
          4    1
          5    0
          Name: likepizza, dtype: int64
```

```
In [77]: x.dtypes
```

```
Out[77]: age      int64
          weight  int64
          dtype: object
```

```
In [78]: y.dtypes
```

```
Out[78]: dtype('int64')
```

step 6

```
In [80]: from sklearn.neighbors import KNeighborsClassifier
          model = KNeighborsClassifier(n_neighbors=2)
          model.fit(x,y)
```

```
Out[80]: KNeighborsClassifier(n_neighbors=2)
```

step 7

```
In [81]: data = [[20,50]]
```

```
In [83]: model.predict(data)
```

```
Out[83]: array([1], dtype=int64)
```

```
In [84]: data = [[60,60]]
```

```
In [85]: model.predict(data)
```

```
Out[85]: array([0], dtype=int64)
```

step 8

```
In [86]: model2 =KNeighborsClassifier(n_neighbors=3)
model.fit(x,y)
data = [[32,35]]
```

```
In [29]: model.predict(data)
```

```
Out[29]: array([1], dtype=int64)
```

```
In [30]: print(model.predict(data))

[1]
```

```
In [31]: print(model.predict(x))

[0 1 1 0 1 0]
```

step 9

```
In [32]: y_pred = model.predict(x)
def accuracy(actual, pred):
    return sum(actual==pred) / float(actual.shape[0])
print('Accuracy:',accuracy(y,y_pred))
```

```
Accuracy: 1.0
```

step 10

```
In [34]: from sklearn.metrics import accuracy_score
```

```
In [35]: print(accuracy_score(y,y_pred))

1.0
```

step 12

```
In [54]: p1=pd.read_csv("pizza_test.csv")
```

```
In [55]: p1.head()
```

```
Out[55]:
```

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

```
In [56]: p1.shape
```

```
Out[56]: (4, 3)
```

```
In [57]: p1.columns
```

```
Out[57]: Index(['age', 'weight', 'likepizza'], dtype='object')
```

```
In [58]: p1.info
```

```
Out[58]: <bound method DataFrame.info of      age  weight  likepizza
0    48      68         1
1    35      45         1
2    15      40         0
3    55      65         0>
```

```
In [59]: age_weight=['age', 'weight']
X=p1[age_weight]
X
```

```
Out[59]:
```

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

```
In [60]: y=p1.likepizza
y
```

```
Out[60]: 0    1
1    1
2    0
3    0
Name: likepizza, dtype: int64
```

```
In [61]: knn=KNeighborsClassifier(n_neighbors=2)
knn.fit(X,y)
```

```
Out[61]: KNeighborsClassifier(n_neighbors=2)
```

```
In [62]: knn.predict(X)
```

```
Out[62]: array([0, 0, 0, 0], dtype=int64)
```

```
In [63]: def accuracy(actual,pred):
          return sum(actual == pred) / float(actual.shape[0])
          return accuracy
```

```
In [64]: y_pred=knn.predict(X)
```

```
In [65]: accuracy(y,y_pred)
print("accuracy score: ", accuracy(y,y_pred) )
```

accuracy score: 0.5

step 13

```
In [66]: scores = []
for k in range(1,4):
    best = KNeighborsClassifier(n_neighbors=k)
    best.fit(X, y)
    y_predt = best.predict(X)
    acc=accuracy(y,y_predt)
    scores.append((k,acc))
```

```
In [67]: scores
```

```
Out[67]: [(1, 1.0), (2, 0.5), (3, 0.5)]
```

step 14

```
In [68]: from sklearn.metrics import accuracy_score
```

```
In [69]: accuracy(y,y_pred)
print("accuracy_score: ", accuracy_score(y,y_pred) )
```

accuracy_score: 0.5