225229140

suriya s

In [118]:
```
import  pandas as pd
```

In [119]:
```
import seaborn as sns
```

In [120]:
```
import matplotlib.pyplot as plt
```

In [121]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [122]:
```
#step 1


p=pd.read_csv(r'C:\Users\1mscdsa40\Downloads\piza1.csv')
p
```

Out[122]:

	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 [123]:
```
p.head()
```

Out[123]:

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

In [124]:
```
p.columns
```

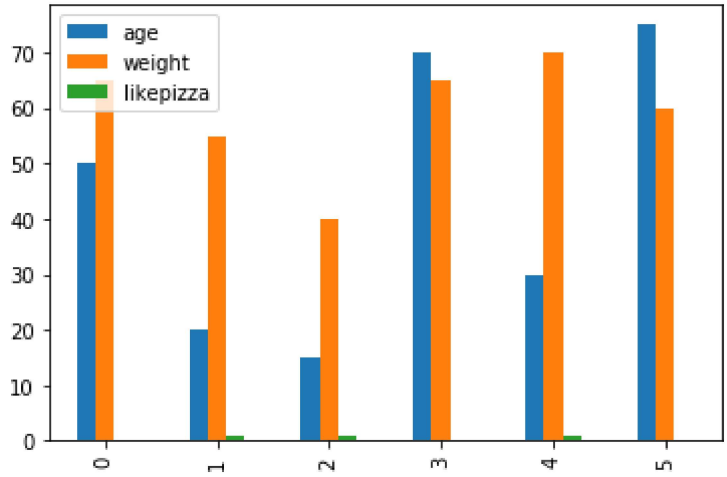
Out[124]:
```
Index(['age', 'weight', 'likepizza'], dtype='object')
```

In [125]:
```
#step2

import matplotlib.pyplot as plt
import seaborn as sns
```

In [126]:
```
p.plot(kind='bar')
```

Out[126]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1dc9b6d7fd0>
```



In [127]:
```
sns.scatterplot(x=pizza['age'], y=pizza['weight'], hue=pizza['likePizza'])
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-127-a3a8b213734b> in <module>()
----> 1 sns.scatterplot(x=pizza['age'], y=pizza['weight'], hue=pizza['likePizz
a'])

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

In [128]:
```
#step 4

Fix = ["age","weight"]
x = p[Fix]
```

In [129]:
```
x
```

Out[129]:

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

In [130]:
```
#step 5

y = p.likepizza
```

In [131]:
```
y
```

Out[131]:
```
0    0
1    1
2    1
3    0
4    1
5    0
Name: likepizza, dtype: int64
```

In [132]:
```
type(a)
```

Out[132]:
```
pandas.core.frame.DataFrame
```

In [133]:
```
x.dtypes
```

Out[133]:
```
age       int64
weight    int64
dtype: object
```

In [134]:
```
type(y)
```

Out[134]:
```
pandas.core.series.Series
```

In [135]:
```
y.dtypes
```

Out[135]:
```
dtype('int64')
```

In [136]:
```
#step 6

piz = KNeighborsClassifier(n_neighbors=2)
```

In [137]:
```
piz.fit(x,y)
```

Out[137]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

In [138]:
```
piz.predict(x)
```

Out[138]:
```
array([0, 1, 1, 0, 1, 0], dtype=int64)
```

In [139]:
```
#step 7

pre = [[25,50]]
print(piz.predict(pre))
```

```
[1]
```

In [140]:
```
pre =[[60,60]]
print(piz.predict(pre))
```

```
[0]
```

In [141]:
```
#step 8

piz = KNeighborsClassifier(n_neighbors=3)
piz.fit(x,y)
```

Out[141]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
```

In [142]:
```
#step 9


pre = [[25,50]]
print(piz.predict(pre))
```

```
[1]
```

In [143]:
```
pre = [[60,60]]
print(piz.predict(pre))
```

```
[0]
```

In [144]:
```
y_pred = piz.predict(x)
y_pred
```

Out[144]:
```
array([0, 1, 1, 0, 1, 0], dtype=int64)
```

In [145]:
```
#step 10

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

In [146]:
```
#step 11



accuracy(y,y_pred)
```

Out[146]:
```
1.0
```

In [ ]:

In [ ]:

In [147]:
```
#step 12

p1 = pd.read_csv(r'C:\Users\1mscdsa40\Downloads\piza2.csv')
p1
```

Out[147]:

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

In [148]:
```
p1.head()
```

Out[148]:

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

In [149]:
```
p1.shape
```

Out[149]:
```
(4, 3)
```

In [150]:
```
p1.columns
```

Out[150]:
```
Index(['age', 'weight', 'likepizza'], dtype='object')
```

In [151]:
```
p1.info
```

Out[151]:
```
<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 [152]:
```
Fix = ["age","weight"]
x = p1[Fix]
```

In [153]:
```
x
```

Out[153]:

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

In [154]:
```
y=p1.likepizza
```

In [155]:
```
y
```

Out[155]:
```
0    1
1    1
2    0
3    0
Name: likepizza, dtype: int64
```

In [156]:
```
pizza = KNeighborsClassifier(n_neighbors=2)
```

In [157]:
```
pizza.fit(x,y)
```

Out[157]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

In [158]:
```
pizza.predict(x)
```

Out[158]:
```
array([0, 0, 0, 0], dtype=int64)
```

In [159]:
```
def accuracy(actual, pred):
    return sum(actual == pred) / (float(actual.shape[0]))
```

In [160]:
```
y_pred = pizza.predict(x)
```

In [161]:
```
accuracy(y,y_pred)
```

Out[161]:
```
0.5
```

In [162]:
```
#stepp 13


scores = []

for k in range(1,4):
    best = KNeighborsClassifier(n_neighbors=k)
    best.fit(x,y)
    y_pred = best.predict(x)
    acc = accuracy(y,y_pred)
    scores.append((k, acc))
```

In [163]:
```
scores
```

Out[163]:
```
[(1, 1.0), (2, 0.5), (3, 0.5)]
```

In [164]:
```
from sklearn.metrics import accuracy_score
```

In [165]:
```
#step 14

accuracy_score(y,y_pred)
```

Out[165]:
```
0.5
```

In [ ]: