```
import pandas as pd
#import math
#import operator
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
data = pd.read_csv("iris.csv")
data.head()
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
X =  data.drop('Name', axis=1) # features
y = data['Name']  # labels
```

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
# 2. Split into train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 3. Create KNN model with k=3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# 4. Predict
y_pred = knn.predict(X_test)

# 5. Evaluate
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

```
# 2. Split into train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 3. Create KNN model with k=3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# 4. Predict
y_pred = knn.predict(X_test)

# 5. Evaluate
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

```
# 2. Split into train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

# 3. Create KNN model with k=3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# 4. Predict
y_pred = knn.predict(X_test)

# 5. Evaluate
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9833333333333333
```

```
# 2. Split into train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

```
# 3. Create KNN model with k=3
knn = KNeighborsClassifier(n_neighbors=4)
knn.fit(X_train, y_train)

# 4. Predict
y_pred = knn.predict(X_test)

# 5. Evaluate
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9833333333333333
```

```
 #Train KNN with 4 neighbors
knn = KNeighborsClassifier(n_neighbors=4)
knn.fit(X_train, y_train)

# Select a sample test point (e.g., the first one)
sample = X_test.iloc[0].values.reshape(1, -1)

#sample = X_test[0].reshape(1, -1)

# Find its 4 nearest neighbors
distances, indices = knn.kneighbors(sample)

print("Sample test point:", sample)
print("Indices of 4 nearest neighbors in training set:", indices)
print("Distances to 4 nearest neighbors:", distances)
print("Neighbor labels:", y_train.iloc[indices[0]])
```

```
Sample test point: [[6.1 2.8 4.7 1.2]]
Indices of 4 nearest neighbors in training set: [[49 60  9 43]]
Distances to 4 nearest neighbors: [[0.2236068  0.3        0.43588989 0.50990195]]
Neighbor labels: 63    Iris-versicolor
91    Iris-versicolor
97    Iris-versicolor
72    Iris-versicolor
Name: Name, dtype: object
C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but KNeighborsClass
  warnings.warn(
```