

K-Nearest Neighbors (KNN) - Step by Step Python Demo

1. Objective

Build a KNN model to classify whether a person will buy a product based on age and salary.

2. Install Required Libraries (One Time Only)

```
pip install numpy pandas matplotlib scikit-learn
```

3. Import Libraries

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

4. Create Sample Dataset

```
data = pd.DataFrame({
    "Age": [22,25,47,52,46,56,55,60,62,23],
    "Salary": [20000,25000,50000,60000,52000,58000,57000,65000,70000,22000],
    "Buy": [0,0,1,1,1,1,1,1,1,0]
})

print(data)
```

5. Separate Features and Target

```
X = data[["Age","Salary"]]
y = data["Buy"]
```

6. Split Train and Test Data

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.2, random_state=42  
)
```

7. Create Model

```
model = KNeighborsClassifier(n_neighbors=5)
```

8. Train Model

```
model.fit(X_train, y_train)  
print("Model Trained Successfully")
```

9. Make Predictions

```
y_pred = model.predict(X_test)  
print("Actual:", list(y_test))  
print("Predicted:", list(y_pred))
```

10. Evaluate Model

```
accuracy = accuracy_score(y_test, y_pred)  
print("Accuracy:", accuracy)  
  
print("Confusion Matrix:")  
print(confusion_matrix(y_test, y_pred))  
  
print("Classification Report:")  
print(classification_report(y_test, y_pred))
```

11. Predict New Customer

```
new_customer = [[40, 48000]]  
prediction = model.predict(new_customer)  
  
if prediction[0] == 1:  
    print("Customer WILL BUY")  
else:  
    print("Customer WILL NOT BUY")
```

- KNN is instance-based learning
- No explicit training phase
- Uses distance between points
- Choice of K impacts accuracy

KNN is a supervised learning algorithm that classifies data based on the majority class of nearest neighbors.