

A retail company wants to predict customer purchasing behavior based on their age, salary and past purchase history. The company aims to use K-Nearest Neighbors (KNN) algorithm to classify customers into potential buying groups to personalize marketing strategies. This predictive model will help the company understand and target specific customer segments more effectively, thereby increasing sales and customer satisfaction.

Suggested code may be subject to a license | towardsdatascience.com/feature-engineering-on-time-series-data-transforming-signal-data-of-a-smartphone-accelerometer-for-72cbe34b8a60

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

data = np.array([[25,50000,2], [30,80000,1], [35,60000,3], [20, 30000,2], [40, 90000,1],[45,75000,2]])
labels = np.array ([1,2,1,0,2,1]) # 0:Low, 1:medium, 2:high
```

Suggested code may be subject to a license | [Gabozm09/Music-Genre-Classification](https://github.com/Gabozm09/Music-Genre-Classification)

```
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
```

+ Code

+ Text

Suggested code may be subject to a license | [ktzlx/CameraPricePrediction](https://github.com/ktzlx/CameraPricePrediction)

```
scaler=StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
```



```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
accuracy=knn.score(X_test,y_test)
print(f"Accuracy:{accuracy}")
```



Accuracy:0.5

```
user_input = np.array([[30, 60000, 1]])
user_input_scaled = scaler.transform(user_input)
knn.predict(user_input_scaled)
```



array([0])

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.