```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("teleCust.csv")
df.head()
```

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	13	44	1	9	64.0	4	5	0.0	0	2	1
1	3	11	33	1	7	136.0	5	5	0.0	0	6	4
2	3	68	52	1	24	116.0	1	29	0.0	1	2	3
3	2	33	33	0	12	33.0	2	0	0.0	1	1	1
4	2	23	30	1	9	30.0	1	2	0.0	0	4	3

```
X = df.drop("custcat", axis=1).values
y = df["custcat"].values
```

```
from sklearn.model_selection import train_test_split

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

```
from sklearn.preprocessing import StandardScaler

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

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Accuracy of KNN Model (k=5): {accuracy:.4f}")
```

```
Accuracy of KNN Model (k=5): 0.3350
```

```
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier

accuracies = []
for k in range(1, 21):
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(X_train, y_train)
    y_pred_k = model.predict(X_test)
    accuracies.append(accuracy_score(y_test, y_pred_k))

best_k = np.argmax(accuracies) + 1
print(f"Best k value: {best_k}, Accuracy: {accuracies[best_k-1]:.4f}")
```

```
Best k value: 18, Accuracy: 0.3800
```