

```
In [109]: # Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score

# Import dataset
df = pd.read_csv("C:/Users/hkeim/OneDrive/Documents/School/D209/churn_clean.csv")

In [110]: # Drop extraneous variables
df=df.drop(['Customer_id', 'Interaction', 'UID', 'City', 'County', 'Job'], axis=1)

In [111]: # Change object type to category codes
df['State'] = df['State'].astype('category')
df['State'] = df['State'].cat.codes

In [112]: df['Area'] = df['Area'].astype('category')
df['Area'] = df['Area'].cat.codes

In [113]: df['TimeZone'] = df['TimeZone'].astype('category')
df['TimeZone'] = df['TimeZone'].cat.codes

In [114]: df['Marital'] = df['Marital'].astype('category')
df['Marital'] = df['Marital'].cat.codes

In [115]: df['Gender'] = df['Gender'].astype('category')
df['Gender'] = df['Gender'].cat.codes

In [116]: df['Contract'] = df['Contract'].astype('category')
df['Contract'] = df['Contract'].cat.codes

In [117]: df['PaymentMethod'] = df['PaymentMethod'].astype('category')
df['PaymentMethod'] = df['PaymentMethod'].cat.codes

In [118]: df['InternetService'] = df['InternetService'].astype('category')
df['InternetService'] = df['InternetService'].cat.codes

In [119]: # Create dummies for bianry objects
df=pd.get_dummies(df, columns=['Churn', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
                              'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSup'])

# Drop 'No' dummies
df=df.drop(['Churn_No', 'Techie_No', 'Port_modem_No', 'Tablet_No', 'Phone_No', 'Multiple_No',
            'StreamingTV_No', 'StreamingMovies_No', 'PaperlessBilling_No'], axis=1)
```

```
In [120]: # Split data into training and testing sets
train, test = train_test_split(df, test_size=.2, random_state=1111)
```

```
In [121]: # Create train features array
X_train=train.drop('Churn_Yes', axis=1).values
```

```
In [122]: # Create train target array
y_train=train['Churn_Yes'].values
```

```
In [123]: # Create test features array
X_test=test.drop('Churn_Yes', axis=1).values
```

```
In [124]: # Create test target array
y_test=test['Churn_Yes'].values
```

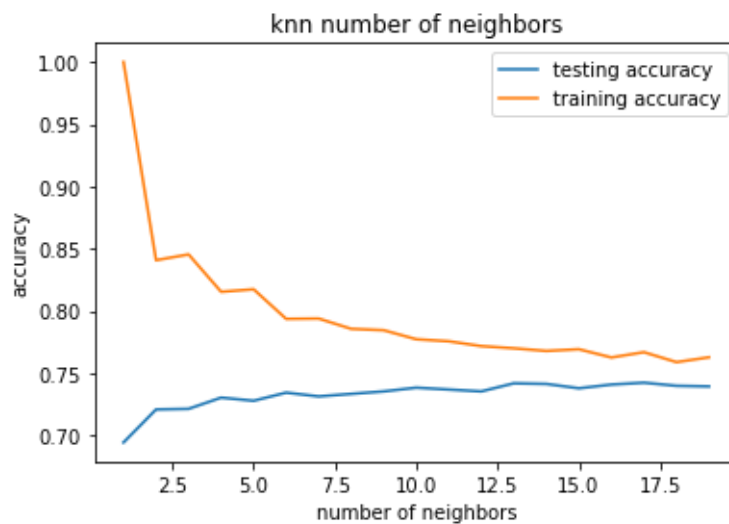
```
In [125]: # Scale features data
scaler=StandardScaler()
scaler.fit(X_train)
scaler.fit(X_test)
```

```
Out[125]: StandardScaler()
```

```
In [126]: # Determine best number of neighbors
neighbors=np.arange(1, 20)
train_accuracy=np.empty(len(neighbors))
test_accuracy=np.empty(len(neighbors))

for i, k in enumerate(neighbors):
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    train_accuracy[i]=knn.score(X_train, y_train)
    test_accuracy[i]=knn.score(X_test, y_test)

plt.title('knn number of neighbors')
plt.plot(neighbors, test_accuracy, label='testing accuracy')
plt.plot(neighbors, train_accuracy, label='training accuracy')
plt.legend()
plt.xlabel('number of neighbors')
plt.ylabel('accuracy')
plt.show()
```



In [127]:

```
# Build model
knn=KNeighborsClassifier(n_neighbors=17)
knn.fit(X_train, y_train)
pred=knn.predict(X_test)
```

In [128]:

```
# Evaluate model
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
print(roc_auc_score(y_test, pred))
```

```
[[1348 125]
 [ 390 137]]
      precision    recall  f1-score   support

     0       0.78        0.92        0.84       1473
     1       0.52        0.26        0.35        527

 accuracy          0.74       2000
 macro avg         0.65        0.59        0.59       2000
 weighted avg         0.71        0.74        0.71       2000

0.5875506105470899
```

In [129]:

```
# Save CSV file of prepared data
df.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D209/Keim D209 Task One Clean Data.csv",
```

In [130]:

```
# Save CSV files of training and testing data
train.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D209/Keim D209 Task One Training Data.
test.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D209/Keim D209 Task One Testing Data.cs
```