```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder
# Load data from CSV file
data = pd.read_csv('data.csv')
# Drop unnecessary columns
columns_to_drop = ['RowNumber', 'CustomerId', 'Surname', 'Complain',
'Satisfaction Sc
data = data.drop(columns=columns_to_drop)
# Convert categorical columns to numerical using Label Encoding
label_encoder = LabelEncoder()
data['Geography'] = label_encoder.fit_transform(data['Geography'])
data['Gender'] = label_encoder.fit_transform(data['Gender'])
# Split the data into features and target
X = data.drop(columns=['Exited'])
y = data['Exited']
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=
# Train Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(X_train, y_train)
rf_predictions = rf_classifier.predict(X_test)
# Train XGBoost Classifier
xgb_classifier = XGBClassifier(random_state=42)
xgb_classifier.fit(X_train, y_train)
xgb_predictions = xgb_classifier.predict(X_test)
# Evaluate and print results
print("Random Forest Classifier:")
print(classification_report(y_test, rf_predictions))
print("XGBoost Classifier:")
print(classification_report(y_test, xgb_predictions))
# Visualize Customer Churn using a line graph
plt.figure(figsize=(10, 6))
sns.lineplot(data=data, x='Age', y='Exited', estimator='mean',
hue='Gender')
plt.title('Customer Churn based on Age')
plt.xlabel('Age')
```

```
plt.ylabel('Churn Probability')
plt.legend(title='Gender', loc='upper right')
plt.show()
```


Customer Churn based on Age