Prediction of Customer Churn

İrem Şahar
Osmancan Çağlayan

Problem

The project addresses the issue of Customer Churn Prediction, which is a critical problem faced in bank industry.

- Customer churn happens when customers stop using a company's products or services.
  This leads to losing money and spending more to find new customers.

Experimental Scheme

Data Source

- **We used the Kaggle Churn Modelling Dataset, which contains 10,000 records.**

The data was provided in a CSV file format, which we then loaded and converted into a DataFrame for further analysis.

Experimental Scheme

Data Source

Features of Dataset

- RowNumber
- CustomerId
- Surname
- CreditScore
- Age
- Tenure
- Balance
- NumOfProducts
- HasCrCard
- IsActiveMember
- EstimatedSalary
- Exited
- Gender
- Geography

Experimental Scheme

Variables

❖ <u>Independent Variables</u>

- ➢ Age
- ➢ NumOfProducts
- ➢ IsActiveMember
- ➢ Gender
- ➢ Geography

❖ <u>Dependent Variable</u>

- ➢ Exited

Experimental Scheme

Variables

- **Control Variables**
  - Standardization and Min-Max Scaling
  - One-Hot Encoding
  - Outlier Removal
  - Feature Selection

Experimental Scheme

Libraries

```
import numpy as np
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler, OneHotEncoder
import pandas as pd
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
```

- **Data Analysis and Manipulation**

We used **numpy** and **pandas** for handling the arrays and dataframe for structured data analysis and **scipy** for Z-score calculations for outlier detection.

- **Visualization**

We used **matplotlib** and **seaborn** libraries to create various plots to visualize data distributions and models performance.

- **Machine Learning Utilities**

Sklearn library is used to implement the machine learning algorithms and utilities.

Algorithm & Motivation

Algorithm

1. Load and Explore Data
2. Data Cleaning
3. Data Analysis and Visualization
4. Handle Outliers
5. Encode Categorical Features
6. Scale Features
7. Correlation Analysis
8. Model Selection and Training
9. Compare Models
10. Visualize Performance Metrics

Algorithm & Motivation

Motivation

The entire process is designed to create a reliable and effective solution for predicting customer churn, based on these ideas:

- ➢ **Data Reliability**
- ➢ **Model Robustness**
- ➢ **Business Clarity**

Algorithm & Motivation

Machine Learning Models

- Logistic Regression
- K- Nearest Neighbors
- Support Vector Machine

Algorithm & Motivation

Logistic Regression

Sigmoid Function

- It gives probabilities which values between 0 and 1 using the formula:

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}}$$

Gradient Descent

- It updates the weights and bias iteratively to minimize the cost function. For every iteration, it computes the **predictions:**

$$\text{predictions} = \text{sigmoid}(X \cdot \text{weights} + \text{bias})$$

- After that, it computes the **gradients:**

Weight gradient:

$$dw = \frac{1}{m} \cdot X^T \cdot (\text{predictions} - y)$$

Bias gradient:

$$db = \frac{1}{m} \cdot \sum(\text{predictions} - y)$$

- Lastly, it updates **weights** and **bias:**

$$\text{weights} - = \text{learning rate} \cdot dw$$

$$\text{bias} - = \text{learning rate} \cdot db$$

Predict Function

- Based on the threshold 0.5, it predict labels as 0 or 1.

Algorithm & Motivation

K-Nearest Neighbors (KNN)

It assigns number of neighbors as 5 and divides the data into training (80%) and test (20%) subsets.

- ### Euclidean Distance
  - Takes two data points as input and computes their scalar **distance** value:

$$\text{distance} = \sqrt{\sum_{i=1}^{n} (x_1[i] - x_2[i])^2}$$

- ### KNN Classifier
  - For each test point:
    - Computes the distances to all training points using the Euclidean distance formula.
    - Sorts the distances and selects the k-nearest training points.
    - Extracts the labels of the k-nearest points.
    - Counts the frequency of each point.
    - Assigns the most frequent label as the prediction.

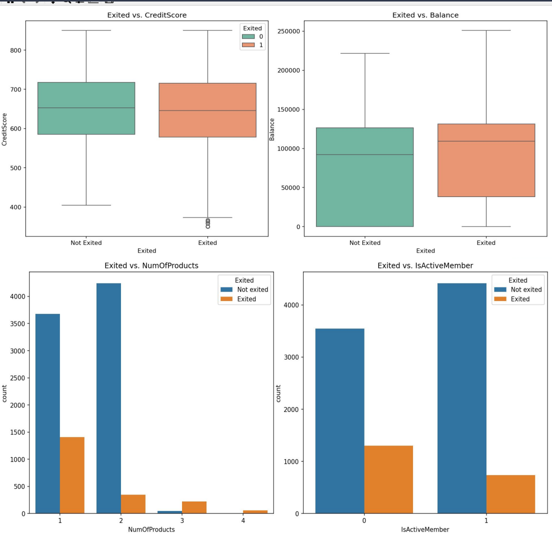
Algorithm & Motivation

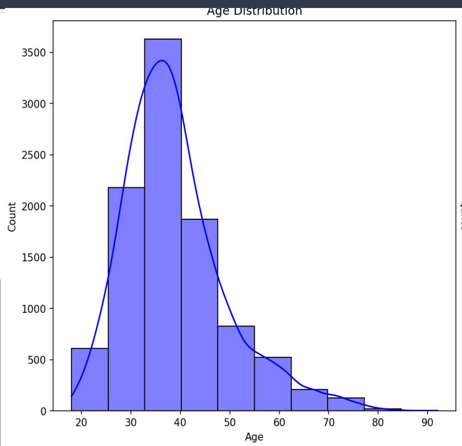
Support Vector Machine (SVM)

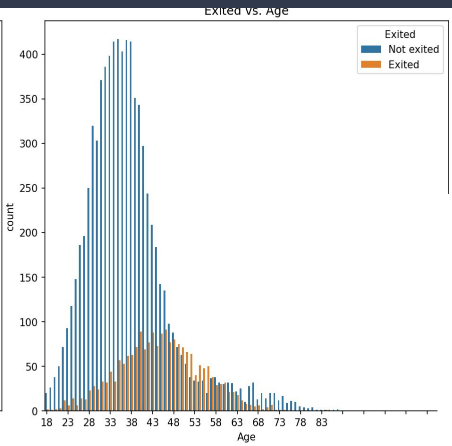
Firstly, it divides the data into training (80%) and testing (20%) subsets using a random state of 50 to ensure reproducibility.

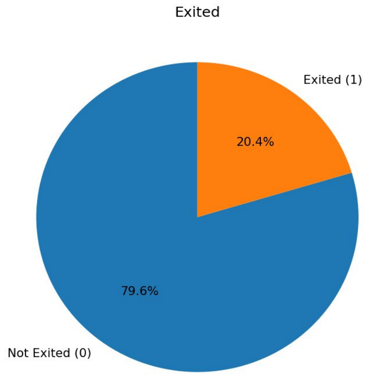
- ### SVM Classifier
  - It creates an instance of the Support Vector Classifier (SVC) with the RBF kernel to ensure the ability to handle non-linear data distributions and consistent behavior.
  - The RBF kernel maps data into a higher-dimensional space to make it easier to classify.
  - **random_state=50 in train_test_split:**
    This ensures that splits are the same each time the code is run. Otherwise, the data could be split randomly, which could lead to different results on each run.
    random_state=50 in SVC:
    This ensures that that the model is trained the same way every time, resulting in consistent model training and results.
- ### Training the Model
  - Using the fit() method to train the SVM model on the training dataset.

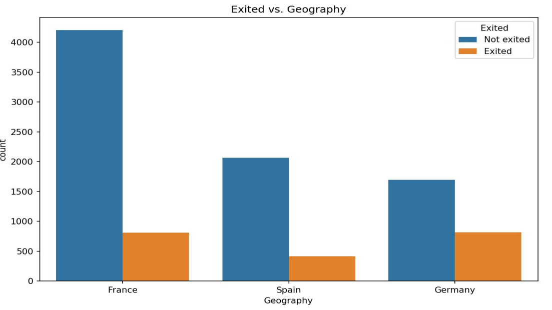
Results

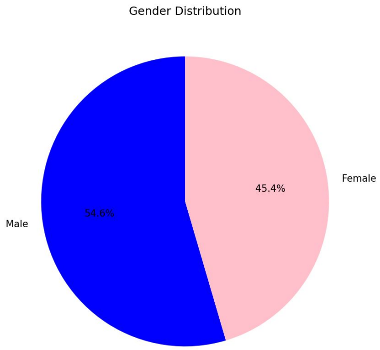

Exited vs. CreditScore
Exited vs. Balance
Exited vs. NumOfProducts
Exited vs. IsActiveMember
Not Exited
Exited
Not exited
CreditScore
Balance
count
NumOfProducts
IsActiveMember

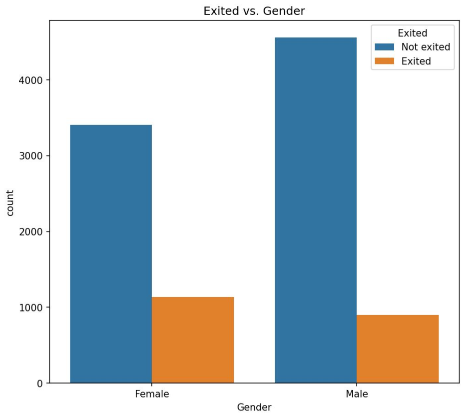

Age Distribution
Count
Age

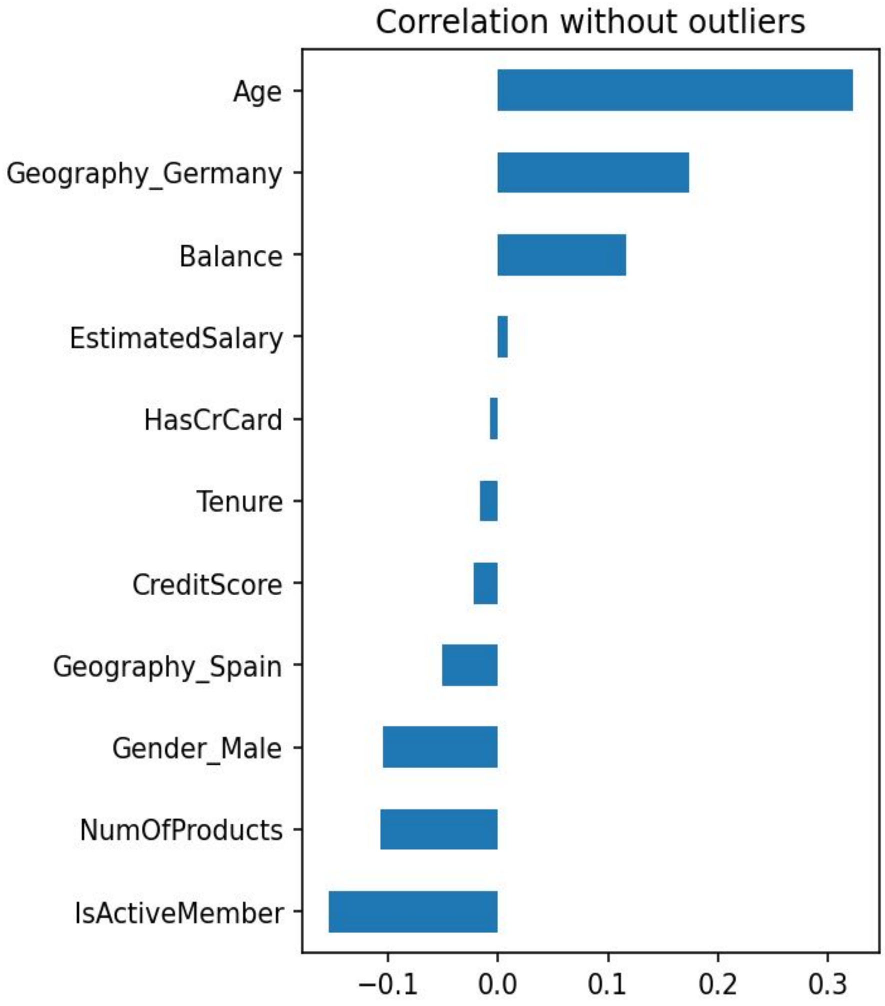

Exited vs. Age
Exited
Not exited
Exited
count
Age

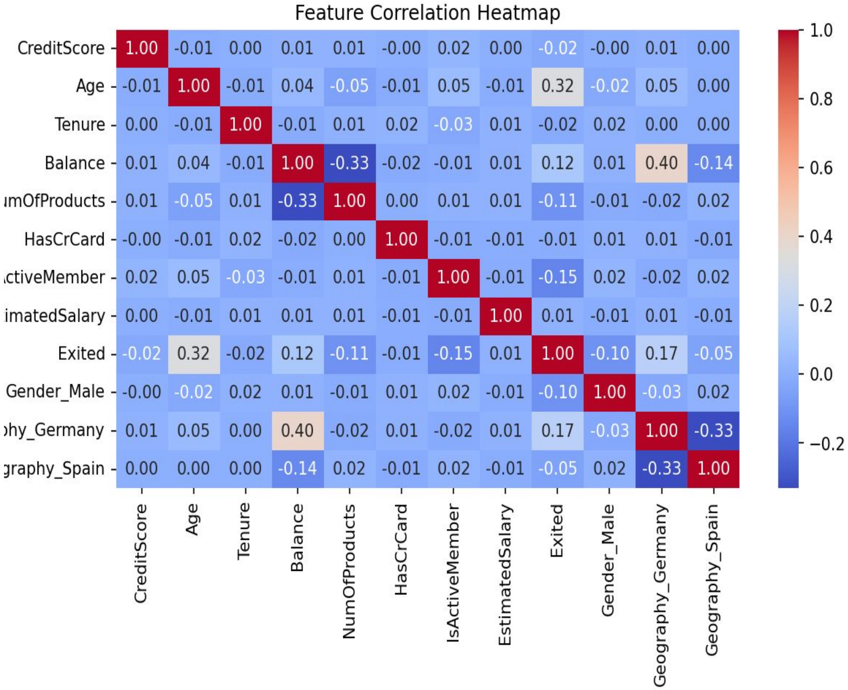

Exited
Exited (1)
20.4%
79.6%
Not Exited (0)


Exited vs. Geography
Exited
Not exited
Exited
count
France
Spain
Germany
Geography


Gender Distribution
Male
54.6%
45.4%
Female


Exited vs. Gender
Exited
Not exited
Exited
count
Female
Male
Gender


Correlation without outliers
Age
Geography_Germany
Balance
EstimatedSalary
HasCrCard
Tenure
CreditScore
Geography_Spain
Gender_Male
NumOfProducts
IsActiveMember
−0.1
0.0
0.1
0.2
0.3


Feature Correlation Heatmap
CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Exited
Gender_Male
Geography_Germany
Geography_Spain
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

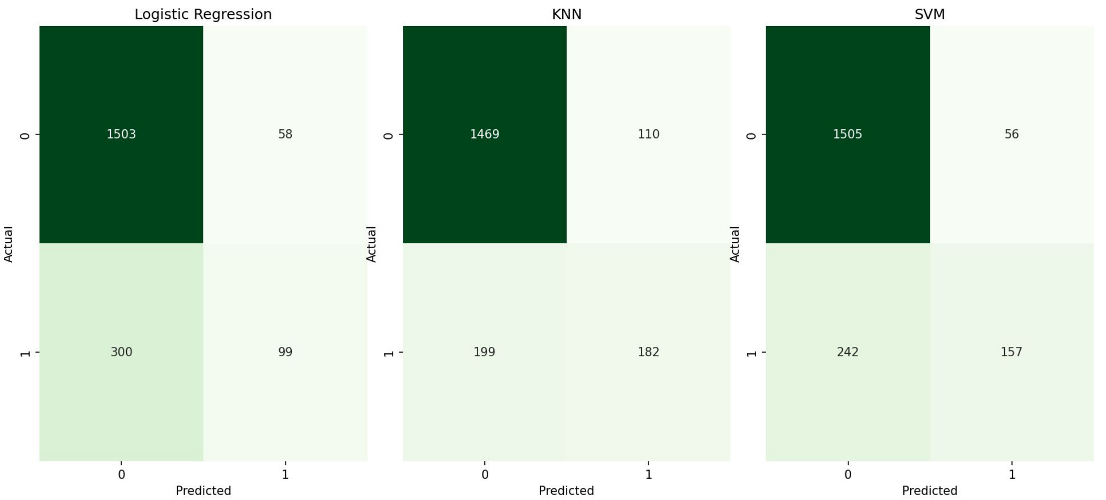
Results

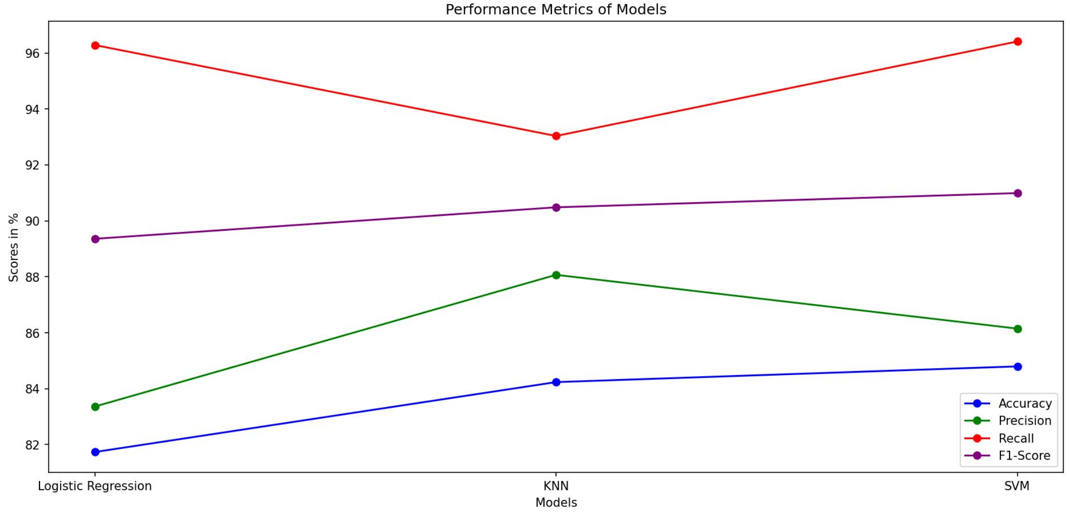
```
========================================
Logistic Regression
========================================
Accuracy: 0.82
Confusion Matrix:
[[1503   58]
 [ 300   99]]
Performance:
              precision    recall  f1-score   support

           0       0.83      0.96      0.89      1561
           1       0.63      0.25      0.36       399

    accuracy                           0.82      1960
   macro avg       0.73      0.61      0.62      1960
weighted avg       0.79      0.82      0.78      1960
```

```
========================================
K-Nearest Neighbors
========================================
KNN Accuracy: 0.84
Confusion Matrix:
[[1469  110]
 [ 199  182]]
Performance:
              precision    recall  f1-score   support

           0       0.88      0.93      0.90      1579
           1       0.62      0.48      0.54       381

    accuracy                           0.84      1960
   macro avg       0.75      0.70      0.72      1960
weighted avg       0.83      0.84      0.83      1960
```

```
========================================
Support Vector Machine
========================================
SVM Accuracy: 0.85
Confusion Matrix:
[[1505   56]
 [ 242  157]]
Performance:
              precision    recall  f1-score   support

           0       0.86      0.96      0.91      1561
           1       0.74      0.39      0.51       399

    accuracy                           0.85      1960
   macro avg       0.80      0.68      0.71      1960
weighted avg       0.84      0.85      0.83      1960
```

MODEL NAME	Execution Time
Logical Regression	0.43 s
SVM	0.96 s
KNN	120.24 s





Improvements

- Explore additional features
- Handle class imbalances using techniques like oversampling (e.g., SMOTE) or undersampling
- Collect more data or use data augmentation techniques to artificially increase dataset size.
- Create better fine-tuning the model settings