Bank Churn Prediction Using Binary Classification Models

Yazeed Alzahrani

January 25, 2024

1 Introduction

In this project, a synthetic dataset of 10,000 bank customers was used to develop machine learning models for predicting customer churn. Each customer record includes 11 variables, one of which is the churn label—a binary variable where 1 indicates that the customer has churned and 0 indicates that the customer has remained with the bank. This label serves as the **target variable** in the classification task. Three supervised machine learning algorithms were used:

- Logistic Regression
- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM)

The objectives of this project are twofold:

1. To build and evaluate binary classification models capable of accurately predicting whether a customer will churn.
2. To identify the features that have the strongest influence on the likelihood of churn.

2 Exploratory Data Analysis

To identify patterns in the data, all variables were examined through scatter plots and histograms. Figure 2 shows that **age** plays a significant role in customer churn. There is a clear difference in the age distributions of churned and non-churned customers: the average age of churned customers is **44.8 years**, compared to **37.4 years** for those who remained with the bank.

Figures 1 and 3 show that churned customers are more likely to have account balances in the **100,000–150,000** range. In contrast, the distributions of **credit score** and **estimated salary** do not show meaningful differences between churned and retained customers, indicating lower predictive value for these features.

The features **tenure** and **credit card ownership** (i.e., whether a customer has a credit card) also showed little to no correlation with the target variable. However, Figure 5 highlights four features that appear to have substantial predictive power:

- **Country**: **48%** of German customers churned, compared to just **19.5%** of non-Germans.
- **Gender**: **33.5%** of female customers churned versus **19.6%** of male customers.
- **Activity status**: Inactive customers had a churn rate of **36.7%**, while active customers had a rate of **16.6%**.
- **Products number**: Customers with more than 2 products churned at a very high rate, although they represented only **3%** of the total. Moreover, customers with only 1 product were **four times more likely to churn** than those with 2 products.

Among these, the **products number** appears to have the strongest predictive potential.

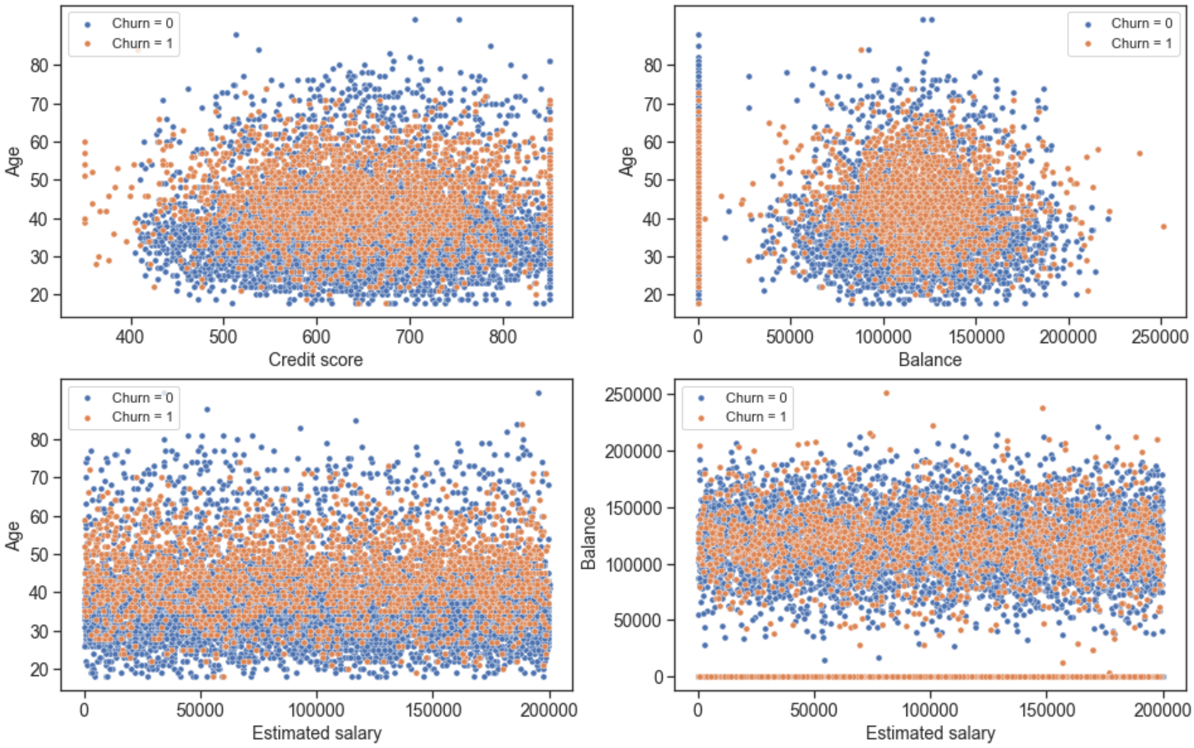

Figure 1: Scatter plots showing the relationship between customer churn and key numerical features: credit score, balance, estimated salary, and age.

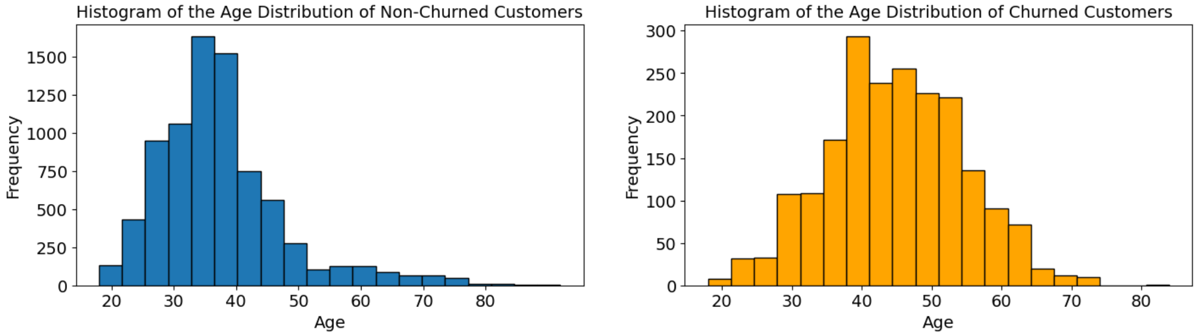

Figure 2: Histograms showing the age distributions of churned and non-churned customers.

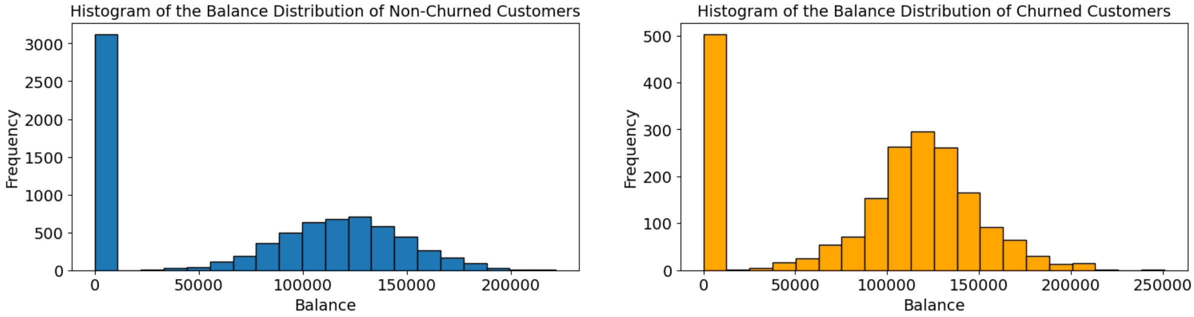

Figure 3: Histograms showing the balance distributions of churned and non-churned customers.

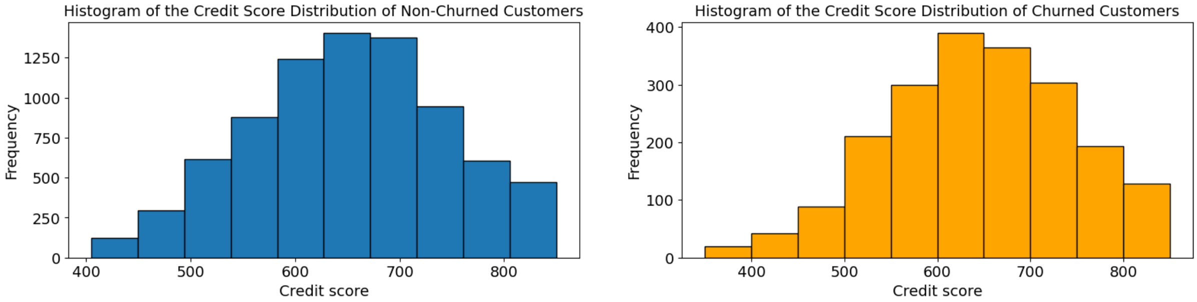

Figure 4: Histograms showing the credit score distributions of churned and non-churned customers.

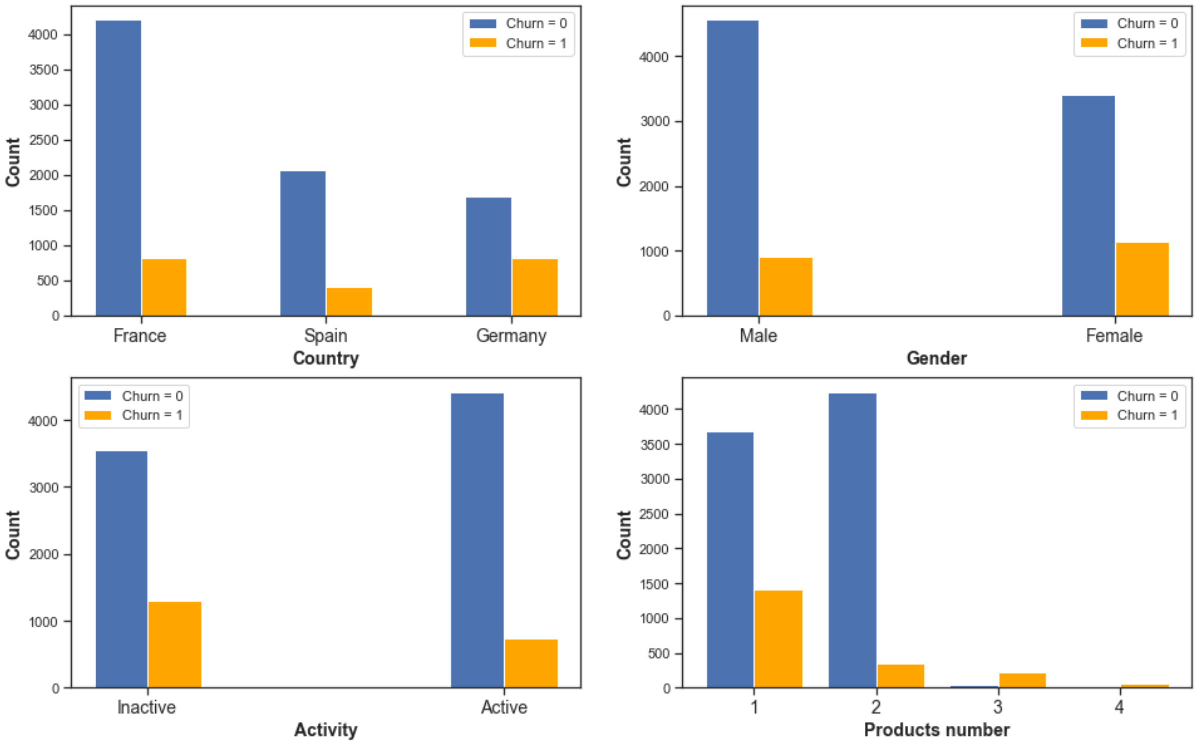

Figure 5: Bar plots showing customer churn counts across categorical features: geography, gender, activity status, and number of products.

3 Methods

This section describes the machine learning algorithms used for binary classification. The Logistic Regression and KNN models were implemented from scratch in Python using the `NumPy` library, while the SVM model was applied using the `scikit-learn` library.

3.1 Logistic Regression

The implementation of logistic regression begins with data standardization, followed by the construction of the data matrix. This matrix includes one column for each feature along with a column of ones as the intercept. To evaluate model performance, we apply 5-fold cross-validation: the dataset is split into five folds, with four used for training and one for testing. The model predicts the class labels of the test set, and the predicted labels are compared against the true labels to compute classification accuracy. This process is repeated across all five folds, with the final model weights and overall accuracy estimated by averaging the results.

Binary logistic regression models the probability that a given input $\mathbf{x}_i$ belongs to class 1 using the logistic (sigmoid) function shown in Equation 1:

$$\sigma(\langle \mathbf{x}_i, \mathbf{w} \rangle) = \frac{1}{1 + e^{-\langle \mathbf{x}_i, \mathbf{w} \rangle}} \tag{1}$$

A predicted label of 1 is assigned if the output exceeds 0.5, and 0 otherwise. To estimate the optimal weights $\mathbf{w}$, we minimize the ridge-regularized logistic regression cost function in Equation 2. Without regularization, this corresponds to maximizing the likelihood of observing the true class labels y_i; the additional penalty term $\frac{\alpha}{2}\|\mathbf{w}\|^2$ controls overfitting by shrinking the weights.

$$L(\mathbf{w}) = \sum_{i=1}^{s} \left[\log\left(1 + e^{\langle \mathbf{x}_i, \mathbf{w} \rangle}\right) - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \right] + \frac{\alpha}{2}\|\mathbf{w}\|^2 \tag{2}$$

The gradient of the cost function, given in Equation 3, is used in the gradient descent algorithm to iteratively update the weights:

$$\nabla L(\mathbf{w}) = \mathbf{X}^T \left(\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}\right) + \alpha \mathbf{w} \tag{3}$$

Here, $\mathbf{X}$ is the data matrix, $\mathbf{y}$ is the vector of true labels, and α is the regularization parameter. A grid search is performed to determine the optimal value of α that maximizes classification accuracy.

3.2 K-Nearest Neighbors

To apply the KNN algorithm, the data is first standardized and structured as a NumPy array, where each row corresponds to a customer and each column represents one of the input features. The dataset is then split into training and testing sets.

For each input in the testing set, the Euclidean distance is computed between that input and every input in the training set. The predicted label is determined based on the majority label among the K closest training inputs (neighbors).

The predicted labels are then compared to the true labels to calculate the classification accuracy. As with logistic regression, K-fold cross-validation is used to estimate the model's performance more reliably by averaging the accuracy across multiple train–test splits. Additionally, a grid search is conducted to determine the optimal value of K that maximizes the classification accuracy.

3.3 Support Vector Machines

SVM classifies data by finding the optimal hyperplane that best separates the data points of different classes. In cases where the data is not linearly separable, the *kernel trick* is applied to map the input data into a higher-dimensional space where a linear separation becomes possible.

A grid search was performed to determine the optimal value of the regularization parameter C. This parameter controls the trade-off between achieving a low training error and maintaining a smooth decision boundary. Higher values of C lead to less regularization and can result in overfitting, whereas lower values increase regularization and may cause underfitting.

4 Results

This section presents the evaluation results for each model.

4.1 Logistic Regression

When all features are included in the model, the classification accuracy reaches **81.04%**, and the optimal regularization parameter is found to be $\alpha = 0$. The resulting weights and their corresponding odds ratios are presented in Table 1.

Feature	**Weight**	**Odds Ratio**
`age`	0.76	2.14
`active_member`	−0.54	0.58
`country_germany`	0.23	0.79
`balance`	0.16	1.18
`gender_female`	0.13	1.14
`gender_male`	−0.13	0.88
`country_france`	−0.12	0.89
`country_spain`	−0.09	0.91
`products_number`	−0.06	0.94
`credit_score`	−0.06	0.94
`tenure`	−0.05	0.95
`estimated_salary`	0.03	1.03
`credit_card`	−0.02	0.98

Table 1: Logistic regression results sorted by absolute weight (feature impact).

The odds ratio for a given feature is calculated as the exponential of its corresponding weight. It indicates how the odds of observing class label 1 change with a one-unit increase in the feature, holding all other features constant.

After removing customers with `products_number` greater than 2 from the dataset, the classification accuracy rises to **84.11%**. The odds ratio for `products_number` also decreases from 0.94 to 0.46. This is due to the fact that nearly all customers who purchased 3 or 4 products ended up churning, even though they represent only 3% of the total dataset. Including them in the model causes their effect to be overrepresented, which ultimately reduces the model's generalization ability.

In another experiment, the model was retrained using only the most influential features, i.e. those with the largest absolute weight values. When `age`, `products_number`, and `active_member` were selected, the classification accuracy dropped slightly to **83.64%**, suggesting that removing less informative features does not necessarily improve performance.

The final model after excluding extreme product numbers and adjusting the selection region for `credit_score` resulted in the weights and odds ratios shown in Table 2.

Feature	**Weight**	**Odds Ratio**
`products_number`	−0.78	0.46
`age`	0.74	2.10
`active_member`	−0.56	0.57
`country_germany`	0.29	1.33
`country_france`	−0.15	0.86
`gender_male`	−0.13	0.88
`country_spain`	−0.11	0.89
`balance`	−0.08	0.93
`tenure`	−0.06	0.95
`credit_score`	−0.04	0.97
`credit_card`	−0.04	0.97
`gender_female`	0.13	1.14
`estimated_salary`	0.01	1.01

Table 2: Logistic regression results after filtering out customers with more than 2 products. Features are sorted by absolute weight.

4.2 K-Nearest Neighbors

This algorithm was executed using four different sets of input features. The results are summarized in Table 3. The selected features were primarily chosen based on their strong predictive potential in the logistic regression model, with the exception of the final case. The feature `estimated_salary` was included in the last experiment to demonstrate its weaker predictive value, as reflected by the lowest classification accuracy.

Selected features	**Classification accuracy (%)**
`age`	83.1
`age`, `products_number`	85.1
`age`, `products_number`, `active_member`	86.7
`estimated_salary`	80.9

Table 3: KNN classification results using different feature subsets.

4.3 Support Vector Machines

SVM classification was performed under two different feature selection scenarios. In the first case, all available input features were used, resulting in a classification accuracy of **85.52%**. In the second case, only the five most influential features—`country_germany`, `gender_male`, `age`, `products_number`, and `active_member`—were selected based on their high absolute weight values in the logistic regression model. This feature reduction led to a slight improvement in performance, with the classification accuracy increasing to **85.88%**.

5 Conclusions

Among the three classification models evaluated, K-Nearest Neighbors achieved the highest accuracy, though not by a substantial margin. This superior performance may be attributed to KNN's ability to capture non-linear relationships in the data more effectively than the other models.

While it was initially expected that SVM classification would yield the best results given its strong track record in handling high-dimensional data, its accuracy fell slightly short of KNN. One likely reason is that only the regularization parameter C was tuned during model selection, whereas the kernel parameter γ was kept fixed. Since γ controls the influence of training examples on the decision boundary (with larger values resulting in more complex boundaries), jointly tuning C and γ via grid search would likely have led to greater performance. More broadly, all models would have benefited from more rigorous feature engineering.

The results of the logistic regression model supported the hypotheses formed during exploratory data analysis. In particular, the features `age` and `products_number` exhibited the largest absolute weights, indicating strong influence on the likelihood of customer churn. In contrast, features such as `estimated_salary` and `credit_card` had minimal weights, suggesting limited predictive power—consistent with earlier observations from the data distribution plots.