Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 9
AIDML 2025

Customer Churn Prediction System Based on Flask

Jiasong Ye
Engineering Department, Shenzhen MSU-BIT University, Shenzhen, China
1120220629@smbu.edu.cn

Abstract. The article designs and implements a customer churn prediction system based on Flask, aimed at helping businesses identify potential customer churn risks through machine learning techniques. The system first loads and processes the input customer data, then utilizes various classifiers (such as Random Forest, Support Vector Machine, and XGBoost) to predict churn probabilities and provide corresponding customer retention suggestions. Methodologically, this study employs grid search for hyperparameter optimization to help the models achieve optimal performance. Additionally, a stacking approach is used to integrate the top-performing models from earlier stages, further enhancing predictive accuracy. Experimental results show that the ensemble method yields the best overall performance, achieving a prediction accuracy of 87%. In the future, the system can be expanded to include multi-source data fusion, real-time predictions, and the integration of feedback mechanisms, thereby realizing intelligent and personalized customer relationship management, providing new insights and technical support for customer retention in businesses.

Keywords: Customer Churn Prediction; Machine Learning; Flask; Stacking Ensemble Model.

1. Introduction

In today's highly competitive and rapidly changing business environment, customer retention has become a major challenge for companies. To enhance customer loyalty, businesses urgently need to leverage customer churn prediction models to identify potential churners in advance and develop corresponding retention strategies. According to research, it frequently costs five to ten times as much to acquire new clients as it does to keep existing ones [1]. Therefore, by analyzing customer behavior and purchase history to prevent churn, companies can not only significantly reduce the economic losses associated with churn but also utilize advanced data-driven methods and artificial intelligence technologies to build accurate predictive models, thereby optimizing marketing strategies and enhancing the customer experience [2, 3].

Research in the field of churn prediction has recently shown a trend of evolving from traditional statistical methods to complex machine learning models. Early studies primarily relied on linear models such as logistic regression, emphasizing interpretability and simplicity, which were suitable for small-scale, linearly separable data [4]. As data complexity has increased, ensemble learning methods like Random Forest have become mainstream due to their ability to handle nonlinear relationships and high-dimensional data [5], while also demonstrating strong resistance to overfitting.

In recent years, the rise of deep learning has further pushed the boundaries of churn prediction, particularly with time series models like LSTM, which can capture the dynamic changes in user behavior and show significant advantages on large-scale, high-dimensional data [6]. Current research trends often focus on integrating industry knowledge with multimodal data fusion, as well as improving model interpretability, in order to enhance predictive performance while meeting practical business needs [7, 8].

In order to create a thorough prediction system, this study applies machine learning techniques to address the problem of customer churn prediction. The research employs the full process at the methodological level, from feature engineering and data preparation to model training and optimization. In particular, it uses grid search for hyperparameter tweaking to incorporate nine popular classifiers, such as Random Forest, Support Vector Machines, and XGBoost [9, 10].

In terms of technological innovation, the system not only realizes probability-based risk classification (low/medium/high risk) but also innovatively incorporates the DeepSeek API, which can automatically generate personalized retention suggestions based on customer characteristics and risk levels.

Regarding engineering implementation, the system utilizes the Flask framework to develop a web service interface that supports real-time predictions and visual output, enabling companies to quickly identify high-risk customers and take intervention measures [11]. This research, through a deep integration of machine learning and business scenarios, provides data-driven decision support tools for enterprise customer relationship management.

Chapter 2 of this paper presents the dataset processing and models, including the customer churn dataset obtained from Kaggle and its preprocessing steps, as well as the hyperparameter tuning of various classifiers and the construction of stacking ensemble models. Chapter 3 compares the performance metrics of different models through experiments and analyzes feature importance. Chapter 4 introduces the Flask-based web prediction system, which implements a user-friendly interface and front-end and back-end interaction features, visually displaying churn risk prediction results and retention suggestions.

2. Dataset and Methods

2.1. Dataset

This paper extracted information on 10,000 customers from the Customer Churn Dataset on Kaggle, creating a customer churn dataset. Among this information, the study references 14 initial features, as detailed in Table 1.

Table 1. Feature Names and Explanations

Feature Name	Description
RowNumber	Row Number
CustomerId	Customer's Unique Identifier
Surname	Customer's Surname
CreditScore	Customer's Credit Score (the higher the value, the better the credit)
Geography	Customer's Country
Gender	Customer's Gender
Age	Customer's Age
Tenure	Tenure as a Bank User (the number of years the customer has been a bank user)
Balance	Customer's Account Balance
NumOfProducts	Number of Bank Products Used by the Customer
HasCrCard	Whether the Customer Has a Credit Card (1 = Yes, 0 = No)
IsActiveMember	Whether the Customer is an Active Member (1 = Yes, 0 = No)
EstimatedSalary	Customer's Estimated Annual Income
Exited	Whether the Customer has Churned (1 = Churned, 0 = Not Churned)

To improve the predictive performance and interpretability of the model, this paper conducted data preprocessing on the dataset. The data processing methods included data cleaning, feature selection, feature engineering, and data preprocessing. First, irrelevant columns (such as RowNumber, CustomerId, and Surname) were removed. Next, the features were categorized into numerical and categorical types using a ColumnTransformer, with missing values handled separately using SimpleImputer. Numerical features were standardized using StandardScaler, while categorical features were transformed using OneHotEncoder. Subsequently, the dataset was split into training and testing sets using train_test_split. This processing ensured that the model could be trained on clean and standardized data, enhancing the effectiveness of the subsequent model training.

2.2. Model

Using a variety of machine learning methods, the model seeks to forecast customer attrition. To improve prediction performance, it uses an ensemble learning approach. Random Forest, Support Vector Machine, Logistic Regression, Gradient Boosting, XGBoost, K-Nearest Neighbors, Naive Bayes, Decision Tree, AdaBoost, and CatBoost are among the several classifiers that are included in it. Every classifier has its own benefits and guiding principles. For example, Support Vector Machine achieves efficient classification by identifying the optimal hyperplane that maximizes the margin between classes; Random Forest increases prediction accuracy and robustness by building multiple decision trees and voting; and Logistic Regression employs a linear model for probability prediction, making it appropriate for binary classification problems.

To optimize model performance, GridSearchCV was used to fine-tune the hyperparameters of each classifier, with particular emphasis on the F1 score as the evaluation metric to balance precision and recall. Five-fold cross-validation was employed to ensure the model's stability and generalization ability, reducing the risk of overfitting. After training, evaluation metrics such as accuracy, recall, precision, and F1 score were calculated and printed for each model to provide a comprehensive assessment of their performance.

Additionally, a Stacking Classifier was constructed, combining the best-performing base models from earlier stages to further enhance prediction performance through a meta-model (Logistic Regression). The stacking model uses the predictions from multiple base learners as input for a new learner to make the final prediction, thereby fully leveraging the strengths of each model and improving overall predictive accuracy. This model design maximizes the advantages of ensemble learning by combining the powerful capabilities of various algorithms to achieve higher prediction accuracy and better generalization capability.

3. Experiment

3.1. Model Evaluation Comparison

Table 2. Comparison of Classifier Performance

Model	Precision (0)	Recall (0)	F1 Score (0)	Precision (1)	Recall (1)	F1 Score (1)	Accuracy
Random Forest	0.91	0.88	0.9	0.59	0.66	0.62	0.84
SVM	0.92	0.8	0.86	0.49	0.74	0.59	0.79
Logistic Regression	0.91	0.73	0.81	0.4	0.72	0.52	0.73
Gradient Boosting	0.88	0.96	0.92	0.74	0.47	0.58	0.86
XGBoost	0.88	0.96	0.92	0.75	0.48	0.58	0.86
KNN	0.86	0.95	0.9	0.66	0.39	0.49	0.84
Naive Bayes	0.86	0.92	0.89	0.56	0.41	0.47	0.81
Decision Tree	0.86	0.85	0.86	0.45	0.47	0.46	0.77
AdaBoost	0.88	0.95	0.91	0.72	0.48	0.57	0.86
CatBoost	0.88	0.96	0.92	0.78	0.48	0.6	0.87
Stacking Model	0.88	0.96	0.92	0.78	0.5	0.61	0.87
Random Forest	0.91	0.88	0.9	0.59	0.66	0.62	0.84
SVM	0.92	0.8	0.86	0.49	0.74	0.59	0.79
Logistic Regression	0.91	0.73	0.81	0.4	0.72	0.52	0.73

The performance metrics of each model are presented in Table 2. The Random Forest model performed best in identifying non-churn customers, achieving a precision of 0.91 and a recall of 0.88; however, its capability to identify churn customers was relatively poor, with a precision of only 0.59. The Support Vector Machine had the highest precision for non-churn customers at 0.92, but its recall for churn customers was low at 0.49, indicating that the model struggled to effectively capture instances of the minority class. The overall performance of Logistic Regression was relatively weak, particularly in identifying churn customers, with a precision of only 0.40. Both Gradient Boosting and XGBoost excelled in recall for non-churn customers (both at 0.96), but their precision and recall for churn customers were moderate (0.74 and 0.75, respectively). Meanwhile, both CatBoost and the stacking model demonstrated similar overall performance, each with an accuracy of 0.87, while maintaining good performance in identifying non-churn customers (with precision and recall of 0.88 and 0.96, respectively).

3.2. Feature Importance Analysis

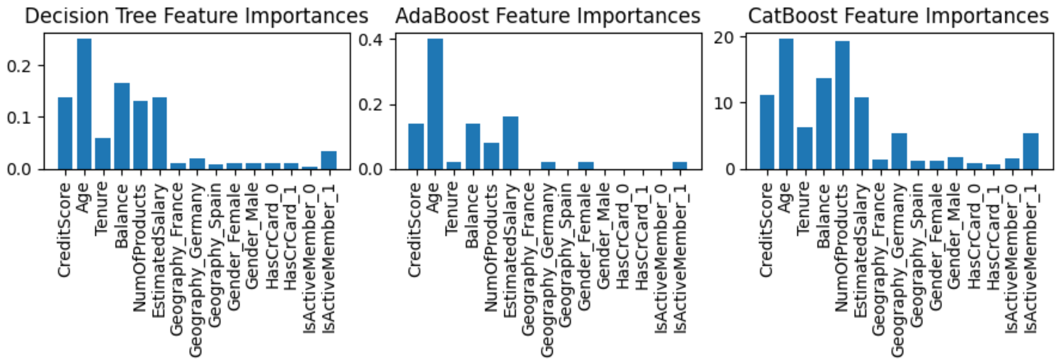

Fig. 1 Feature Importance. (Picture credit: Original)

As shown in Figure 1, the feature importance of different models is presented, including the assessments from Decision Tree, AdaBoost, and CatBoost. In the Decision Tree model, the features "CreditScore" and "Age" exhibit significantly higher importance compared to other features. In contrast, AdaBoost shows a more uniform distribution of feature importance, with "CreditScore" and "Balance" being particularly important in the model. The feature importance in CatBoost also concentrates in a higher range, indicating that "CreditScore" remains a critical feature, while others such as "Age" and "NumOfProducts" also demonstrate considerable importance.

4. Customer Churn Prediction System

4.1. Composition of the Predictive System

The Web Prediction Interface is built on the Flask framework, designed to provide users with a friendly customer churn prediction system. The interface is primarily implemented using HTML, CSS, and JavaScript.

First, in the HTML section, a `<form>` tag is used to create an input form that includes all features related to the dataset, complete with corresponding `<label>` elements to ensure readability and user-friendliness.

CSS styles are employed to enhance the visual appeal of the interface. By setting background colors, rounding borders, and adding shadows, the form becomes more attractive. In the JavaScript portion, event listeners are used to capture the form submission event, utilizing the `fetch` API to send user input data to the Flask backend at the `/predict` route without reloading the page. Upon receiving a response, JavaScript dynamically updates the page to display the churn risk prediction results and recommendations.

On the backend, Flask processes the submitted data, applies the trained machine learning model to make predictions, and returns the results to the front end. The system clearly communicates the churn risk level and retention strategies to the user. This design of frontend-backend interaction enhances the user experience, making the model's output more intuitive and efficient.

4.2. Steps for Using the Predictive System

Users first access the webpage of the customer churn prediction system and fill out a form containing customer information, including credit score, region, gender, age, length of customer service, account balance, number of products, whether they have a credit card, whether they are an active member, and estimated salary. Once the form is completed, the user clicks the "Start Prediction" button, and the system sends this data to the backend for processing.

In the backend, the system analyzes the input data using a pre-trained machine learning model to calculate the probability of customer churn and assess the customer's churn risk level (low risk, medium risk, or high risk) based on that probability. Once the risk assessment is obtained, the system calls the DeepSeek API to send customer information and the risk evaluation results, requesting tailored retention recommendations.

DeepSeek generates specialized customer retention strategies based on the provided information and returns the suggestions to the system. Finally, the system presents the prediction results in a user-friendly format, including the churn probability, risk level, and specific retention suggestions generated by DeepSeek, ensuring that users can clearly understand the customer's churn risk and corresponding measures to take. The second image is the interface for users to input customer information, and the third image is a summary of the prediction results and recommendations.

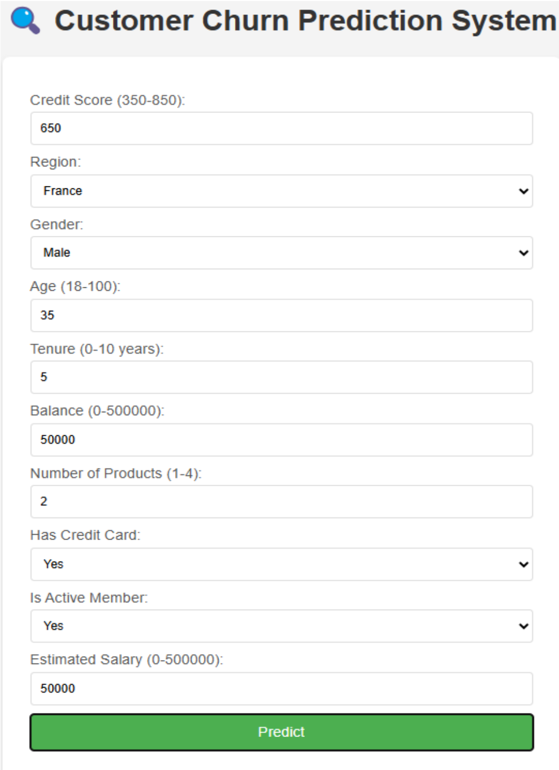

Fig. 2 The input interface. (Picture credit: Original)

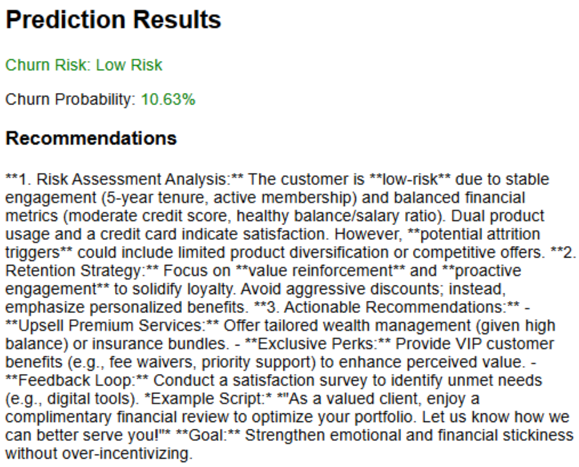

Fig. 3 Prediction results. (Picture credit: Original)

5. Conclusions

This study developed a customer churn prediction system based on a Flask web application that integrates multiple machine learning models to assess the likelihood of customer churn. The system processes customer data, applies preprocessing techniques, and utilizes various classifiers, including Random Forest, Support Vector Machine, and XGBoost, to predict churn probabilities. By integrating the DeepSeek API, the system further enhances its functionality by providing tailored retention strategies based on the predicted risk levels, thereby offering actionable insights for businesses.

Despite the model's effectiveness, there are some limitations. The model's performance largely depends on the quality and representativeness of the training data. If the dataset is biased or lacks diversity, the prediction results may not generalize well to new customers. Additionally, relying on a single dataset for training and testing might overlook potential variations in customer behavior across different contexts or time periods. Therefore, future research could consider using larger and more diverse datasets to improve the model's generalization ability and accuracy.

References

[1] Shaaban E, Helmy Y, Khedr A, et al. A proposed churn prediction model. International Journal of Engineering Research and Applications, 2012, 2(4): 693–697.

[2] Chang V, Hall K, Xu Q A, et al. Prediction of customer churn behavior in the telecommunication industry using machine learning models. Algorithms, 2024, 17(6): 231.

[3] Agarwal V, Taware S, Yadav S A, et al. Customer - churn prediction using machine learning. In: 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan. IEEE, 2022: 893–899.

[4] Jung S, Pramanik S. An efficient path computation model for hierarchically structured topographical road maps. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(5): 1029–1046.

[5] Wu W W. Choosing knowledge management strategies by using a combined ANP and DEMATEL approach. Expert Systems with Applications, 2008, 35(3): 828–835.

[6] Sun Y, Yuan N J, Wang Y, et al. Contextual intent tracking for personal assistants. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 273–282.

[7] Rudd N A, Tetteh D N A, Boateng R. A multimodal fusion learning model for predicting customer churn risk in financial services. arXiv preprint, arXiv:2312.01301, 2023.

[8] Varshneya S, Ledent A, Liznerski P, et al. Interpretable tensor fusion. arXiv preprint, arXiv:2405.04671, 2024.

[9] Improved decision tree, random forest, and XGBoost algorithms for customer churn prediction. The Science and Information Organization, 2021. Available: https://thesai.org/Downloads/ Volume15No12/ Paper_68-Improved_Decision_Tree_Random_Forest_and_XGBoost_Algorithms.pdf

[10] Gayathri. Predicting customer churn using XGBoost: A comprehensive guide. Medium, 2023. Available: https://medium.com/@gayathri.s.de/predicting-customer-churn-using-xgboost-a-comprehensive-guide-d2d45d412f3b

[11] Building churn predictor with Python, Flask, HTML and CSS. Medium, 2021. Available: https://medium.com/star-gazers/building-churn-predictor-with-python-flask-html-and-css-fbab760e8441