Comparative Analysis of Machine Learning Techniques in Predicting Customer Churn and Profiling

Supervisor: Steven Bergner

Student: Sijia Cai (301471111), Zhi Zheng (301473984)

1. Introduction

Banks and financial institutions are constantly striving to retain customers and maintain a strong relationship with them. Customer retention is crucial for these institutions' profitability and long-term success. Identifying potential customer churn and profiling customers are key aspects to better understand and cater to their needs, ensuring customer satisfaction and loyalty[1].

Predicting customer churn involves analyzing various factors that influence a customer's decision to leave the bank, such as dissatisfaction with services, better offers from competitors, or changing personal financial circumstances. By leveraging historical data and employing machine learning techniques, banks can develop predictive models to identify customers at risk of churning. Early detection of potential churn enables banks to take timely actions, such as offering personalized incentives, improving customer support, or addressing specific pain points, ultimately increasing retention rates[2].

Customer profiling, on the other hand, involves segmenting customers based on their demographics, behaviors, and financial patterns[3]. This process allows banks to better understand their customers, identify high-value segments, and develop targeted marketing strategies. By tailoring their services and offers to specific customer segments, banks can enhance customer satisfaction and loyalty, resulting in higher revenue and reduced marketing costs.

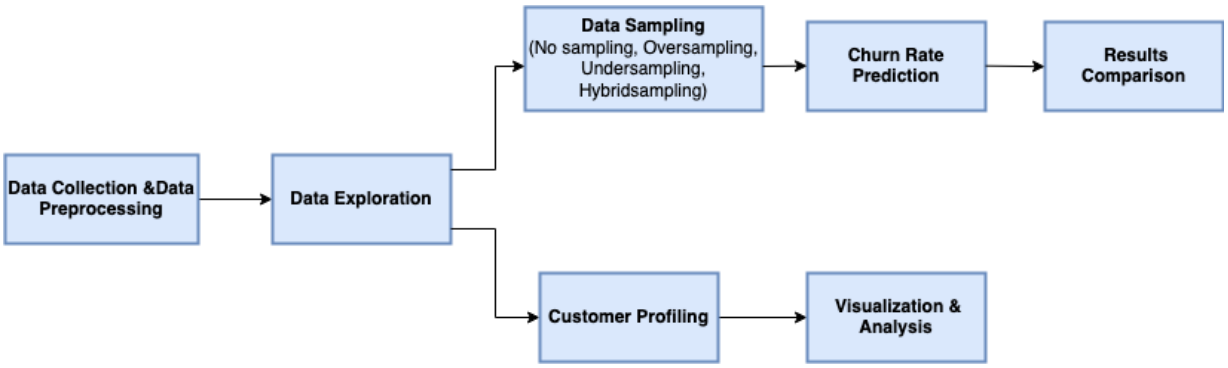
Throughout this project, we will explore various big data techniques to identify the most effective approach to predict customer churn and create customer profiles. We will also evaluate our models' performance using appropriate metrics to ensure the reliability and accuracy of our predictions. We hope to gain insights into the applications of machine learning techniques in customer retention and segmentation in the banking industry. We believe that our findings may provide some value to banks and financial institutions in their efforts to enhance their customer-centric strategies.

2. Problem Statement

In this project, we aim to address several key questions related to predicting customer churn and customer profiling using bank data.

1. Which supervised learning model (Logistic Regression, Random Forest, Decision Tree, or Extreme Gradient Boosting, Artificial Neural Network) performs the best in predicting customer churn rates when combined with various sampling methods (no sampling, oversampling, undersampling, and hybrid sampling)?
2. How can we effectively identify the most relevant features for predicting customer churn, and what insights can we gain from analyzing these features?
3. What is the optimal number of clusters for customer profiling using the k-means algorithm, and how can we determine this number using methods such as the elbow method and silhouette score?
4. After clustering customers using the k-means algorithm, how do the identified customer segments differ in terms of their characteristics, and what specific patterns or trends can be observed within each segment that may inform the development of personalized services?

3. Data Science Pipeline



4. Data Preprocessing

Our dataset has been obtained from Kaggle, and is titled “Churn for Bank Customers”. (https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers?select=churn.csv)
Before modeling, we removed three irrelevant information: RowNumber, Customer Id, and Surname.
We used OrdinalEncoder to encode non-numeric features into integer arrays and KNNImputer for missing value imputation. Since the integer representation of OrdinalEncoder is not directly available for all scikit-learn estimators, we add OneHotEncoder to complete the encoding of non-numeric features. Then, We normalize numeric features by using the StandardScaler function.

5. Methodology

5.1 Data sampling

In this dataset, with 20% of churned customers (labeled as 1) and 80% of non-churned customers (labeled as 0), imbalanced data can lead to model bias towards the majority class and misleading performance metrics.Therefore, we need to use sampling techniques to

address the data imbalance issue and improve the model's performance on the minority class. We will conduct a comparison of several sampling techniques.

Oversampling

Oversampling methods aim to balance class distribution by increasing the number of samples in the minority class. In this project, we explored and compared the following oversampling techniques:

1. Random Oversampling: This method involves randomly selecting instances from the minority class and duplicating them to increase their representation in the dataset. It is a straightforward approach but may lead to overfitting due to the simple replication of instances.
2. Synthetic Minority Over-sampling Technique (SMOTE)[4]: SMOTE is an advanced oversampling technique that generates synthetic instances of the minority class by interpolating between the feature values of existing instances and their k-nearest neighbors.
3. Adaptive Synthetic Sampling (ADASYN)[5]: ADASYN is another oversampling technique that generates synthetic instances adaptively based on the distribution of the minority class. It calculates the number of synthetic instances to generate for each minority instance by considering the class distribution of its k-nearest neighbors.

Undersampling

Undersampling methods aim to balance class distribution by reducing the number of samples in the majority class. In this project, we explored and compared the following undersampling techniques:

1. Random Undersampling: This is the simplest approach, where instances from the majority class are randomly selected and removed from the dataset to balance the class distribution.
2. Tomek Links[6]: This method identifies pairs of instances belonging to different classes that are nearest neighbors to each other. The majority class instances that form Tomek Links are considered noisy and removed from the dataset.
3. Neighborhood Cleaning Rule (NCR)[7]: NCR combines two rules that remove redundant and ambiguous instances from the majority class. The first rule, Condensed Nearest Neighbor (CNN), selects a subset of instances from the majority class that cannot be classified correctly, considering them relevant for learning. The second rule, Edited Nearest Neighbors (ENN), removes ambiguous instances using a k-NN approach. A majority class instance misclassified by its neighbors is removed from the dataset, while a minority class instance misclassified by its majority class neighbors implies the deletion of those majority neighbors.

Hybrid sampling

Hybrid sampling techniques involve the use of both oversampling and undersampling methods to address class imbalance in datasets. By simultaneously increasing the number of samples in the minority class and reducing the number of samples in the majority class,

these techniques aim to create a more balanced class distribution. In this project, we explored and compared the following combined sampling techniques:

1. SMOTEENN (SMOTE + Edited Nearest Neighbors)[8]: This approach combines the synthetic oversampling method SMOTE with the undersampling technique ENN. SMOTE is used to generate synthetic instances of the minority class, while ENN is employed to remove ambiguous instances from both the majority and minority classes.
2. SMOTETomek (SMOTE + Tomek Links)[9]: SMOTETomek is another combined sampling technique that combines SMOTE with Tomek Links. SMOTE generates synthetic instances for the minority class, and Tomek Links identifies and removes noisy majority class instances that are nearest neighbors to minority class instances.

By implementing and comparing these sampling techniques, we aim to identify the most suitable approach for addressing the class imbalance problem and enhancing the predictive performance of our model on the minority class.

5.2 Feature Selection

First, we determine whether each feature follows a normal distribution. Then, we use Pearson correlation coefficient for normally distributed features and Spearman correlation coefficient for no-normally distributed features. To avoid multicollinearity issues, we filter out features with a correlation coefficient greater than 0.8 or less than -0.8[10][11].

Next, we calculate the mutual information between each feature variable and the target variable to measure the dependencies between variables. We filter out feature variables with zero mutual information, which implies that these variables have high dependence on the target variable.

Through the above method, we found there is no-normal distribution feature in the dataset, and then we used the Spearman correlation coefficient to filter out Gender_Female, whose correlation coefficient is -1. By mutual information values, we removed CreditScore, Tenure, and Geography_Spain without sampling. When we use different sampling methods, the features that need to be removed will also be different.

5.3 Modeling

Our models can be categorized into two types: supervised and unsupervised. Supervised models include logistic regression (LR), random forest (RF), decision tree (DT), extreme gradient boosting (XGBoost), and artificial neural network(ANN). The unsupervised category consists of the K-means algorithm. Supervised learning is used to accomplish the first task: predicting churn rates. Unsupervised learning is applied to the second task: clustering customers to facilitate the analysis of different customer segments.

For each supervised model, we evaluated the performance using both default parameters and tuned parameters. For tuning parameters, we use cross-validation to evaluate how different values of the same parameter change the performance of the model to find the optimal value for this parameter. Then, we carry out the same steps for each parameter of the model, and adjust the model to the optimal parameters.

In the task of predicting churn rate, we conducted extensive experiments by combining various supervised models with different sampling techniques. Ultimately, we identified the best approach for predicting churn rate on imbalanced datasets by comparing their results.

In the customer clustering task, we employed the k-means algorithm, and determined the optimal number of clusters using the elbow method and silhouette score. We then analyzed the characteristics of each group.

6. Evaluation and Metrics

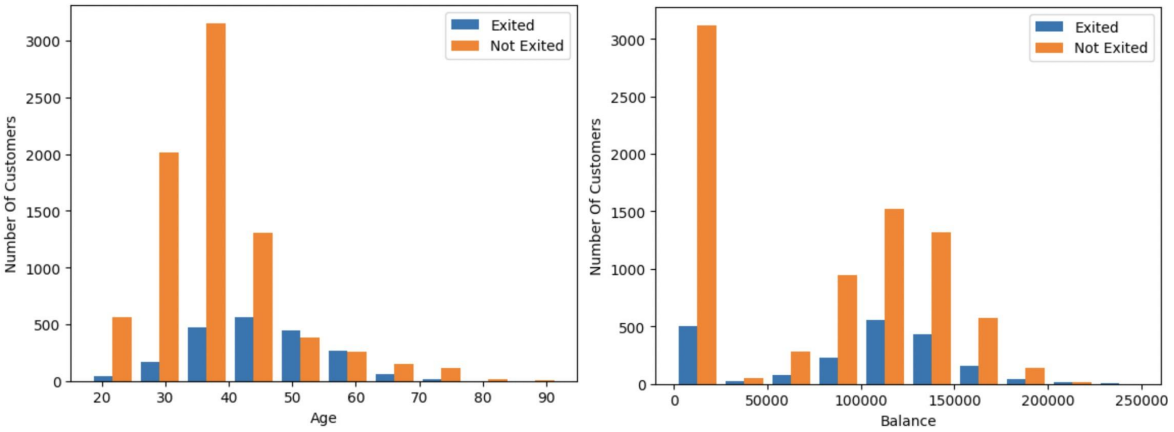
We used the following metrics to evaluate model performance：

1. **Accuracy**(**ACC**): Accuracy is the proportion of samples that the model correctly predicts out of the total samples. It is a simple and intuitive metric, but it can be misleading in imbalanced datasets[12].
2. **Receiver Operating Characteristic(ROC)**: The ROC curve is a plot of the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR).
3. **Recall**: Also known as the True Positive Rate (TPR), recall is the proportion of actual positive samples that are correctly predicted as positive. Recall measures the ability of a model to capture positive samples.
4. **F1_Score**: The F1 score is the harmonic mean of Precision and Recall. It takes values between 0 and 1, with higher values indicating better model performance. The F1 score is particularly suitable for imbalanced datasets, as it takes into account both Precision and Recall.
5. **Negative Predictive Value(NPV)**: NPV is the proportion of actual negative samples that are correctly predicted as negative. It measures the accuracy of a model when predicting negative samples.

7. Results and Analysis

7.1 Data Exploration

We visually show the relationship between the exited status and age, gender, geography, balance, credit score, number of products (see Figure. 1).



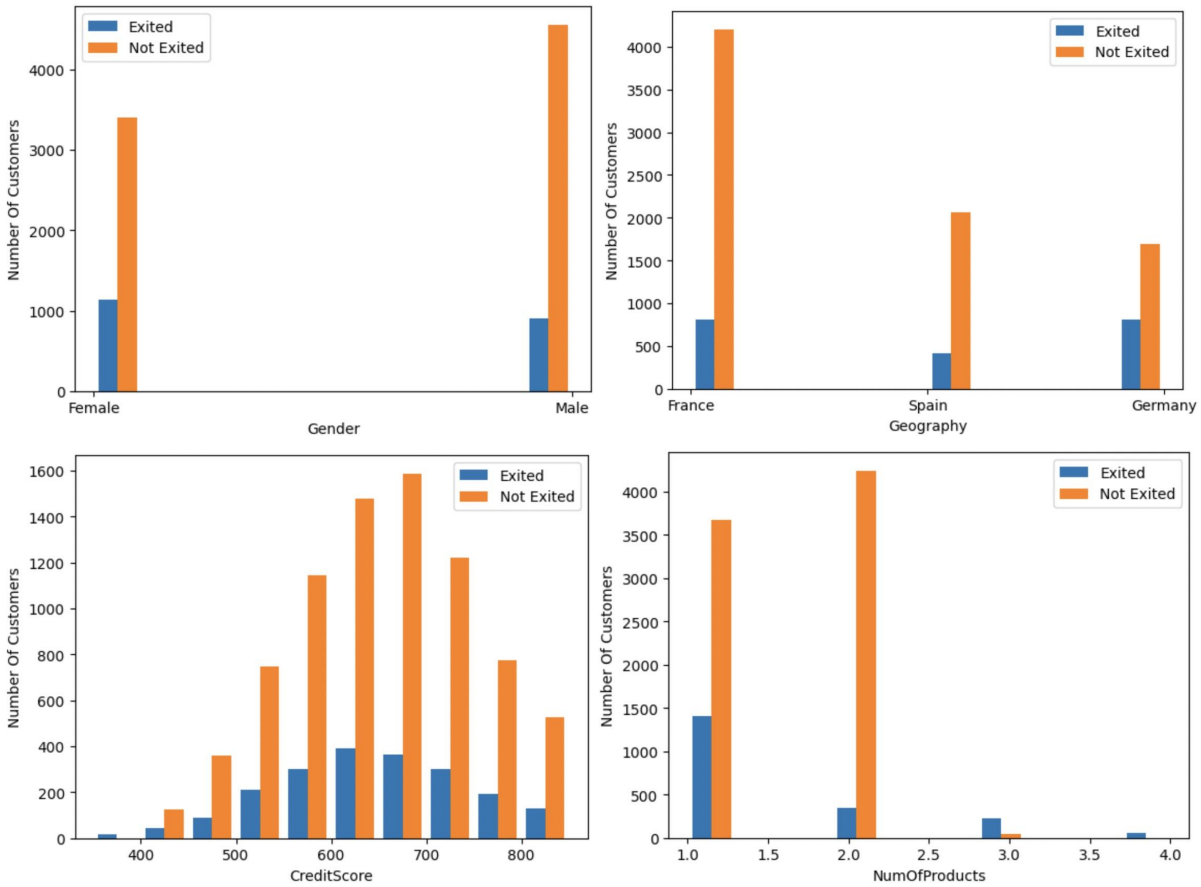


Figure. 1 Data Analysis Visualization

From the figure presented, we can observe that customers under 40 years of age tend to stay with the bank, while those between 50 and 80 years of age are more likely to leave. The bank has a significant number of customers with balances below 50,000, and the majority of them choose to stay. However, the highest number of customers who choose to leave the bank falls within the balance range of 100,000 to 150,000.

In terms of gender, the bank has more male customers than female customers. Interestingly, female customers display less loyalty compared to males. Geographically, the bank serves the highest number of customers from France. Spanish customers appear to be the most loyal, while German customers, although fewer in number, exhibit the least loyalty.

Analyzing the credit score graph reveals that the decision of customers to stay or leave the bank is not strongly influenced by their credit scores. The largest customer segment consists of those who chose two products, and among them, the lowest proportion of customers exited the bank.

7.2 Churn Rate Prediction

For churn rate prediction, we used logistic regression (LR), random forest (RF), decision tree (DT), extreme gradient boosting (XGBoost), and artificial neural network(ANN) models. We conducted experiments on the models with both default parameters and tuned parameters and various sampling methods to explore the best prediction approach.

The results of each model with no sampling and different oversampling methods are shown in Table. 1. In the table, '_def' represents models with default parameters, while '_tun' represents models with tuned parameters.

	Model	ACC	ROC	Recall	F1_Score	NPV
No Sampling	LR_def	0.80950	0.775464	0.189189	0.287850	0.823718
	LR_tun	0.80400	0.779254	0.095823	0.165957	0.810015
	DT_def	0.79250	0.678585	0.486486	0.488286	0.869048
	DT_tun	0.86100	0.843411	0.503686	0.595930	0.882490
	RF_def	0.85600	0.843817	0.469287	0.570149	0.875648
	RF_tun	0.86200	0.842613	0.439803	0.564669	0.871404
	XGBoost_def	0.85200	0.846062	0.481572	0.569767	0.877254
	XGBoost_tun	0.86550	0.857972	0.454545	0.579030	0.874434
	ANN	0.80450	0.758374	0.68059	0.586243	0.911081
Random	LR_def	0.70276	0.773319	0.691777	0.699460	0.698403
(oversmapling)	LR_tun	0.70747	0.774058	0.698682	0.704877	0.703886
	DT_def	0.65380	0.653798	0.442561	0.561082	0.608120
	DT_tun	0.76930	0.860004	0.742624	0.762980	0.755662
	RF_def	0.73854	0.845063	0.550534	0.678005	0.673358
	RF_tun	0.76868	0.832649	0.788449	0.773161	0.779739
	XGBoost_def	0.74200	0.832906	0.614564	0.704317	0.692846
	XGBoost_tun	0.76711	0.854226	0.690521	0.747791	0.731628
	ANN	0.76208	0.762084	0.68801	0.743051	0.728267
SMOTE	LR_def	0.723478	0.791786	0.723792	0.723564	0.723618
(oversmapling)	LR_tun	0.731952	0.795641	0.731952	0.731952	0.731952
	DT_def	0.773384	0.773384	0.712492	0.75869	0.743705
	DT_tun	0.804143	0.888129	0.804143	0.804143	0.804143
	RF_def	0.849655	0.926157	0.788449	0.839853	0.811521
	RF_tun	0.813559	0.896633	0.778406	0.806766	0.792962
	XGBoost_def	0.897363	0.954156	0.870056	0.894482	0.876786
	XGBoost_tun	0.858443	0.932565	0.801632	0.849917	0.821871
	ANN	0.762084	0.762084	0.68801	0.743051	0.728267
ADASYN	LR_def	0.694161	0.754313	0.695864	0.697987	0.688085
(oversmapling)	LR_tun	0.698486	0.757744	0.695255	0.700797	0.69055
	DT_def	0.762434	0.763549	0.692822	0.747621	0.724646
	DT_tun	0.797652	0.869652	0.791971	0.799018	0.789149
	RF_def	0.842138	0.916719	0.790146	0.835638	0.805305
	RF_tun	0.797652	0.88656	0.787713	0.798151	0.786675
	XGBoost_def	0.898672	0.957809	0.868613	0.896985	0.872717
	XGBoost_tun	0.869323	0.939873	0.832725	0.866182	0.840116
	ANN	0.762084	0.762084	0.68801	0.743051	0.728267

Table. 1 Models Performance with No Sampling & Oversampling

From Table. 1, we can observe the performance of various models under no sampling and different oversampling methods:

(a) No Sampling: All models have higher ACC but lower Recall and F1 score. This might be due to the imbalanced dataset, which causes the models to perform poorly in predicting the less frequent class[13].

(b) Random Oversampling: Most models have improved ACC, Recall and F1 score, but some models (such as DT_def and RF_def) still have relatively low Recall and F1 score. This might be because random oversampling can lead to overfitting issues.

(c) SMOTE: Most models achieve a better balance between ACC and Recall. In particular, the XGBoost_def model achieves relatively high levels on all metrics.

(d) ADASYN: Similar to SMOTE, most models achieve a better performance. With the XGBoost_def model, all metrics also reach relatively high levels.

In general, SMOTE and ADASYN sampling methods seem to perform better in this problem, especially when using the XGBoost model, where both ACC and Recall reach higher levels.

The results obtained using the undersampling techniques are shown in the Table. 2.

	Model	ACC	ROC	Recall	F1_Score	NPV
No Sampling	LR_def	0.80950	0.775464	0.189189	0.287850	0.823718
	LR_tun	0.80400	0.779254	0.095823	0.165957	0.810015
	DT_def	0.79250	0.678585	0.486486	0.488286	0.869048
	DT_tun	0.86100	0.843411	0.503686	0.595930	0.882490
	RF_def	0.85600	0.843817	0.469287	0.570149	0.875648
	RF_tun	0.86200	0.842613	0.439803	0.564669	0.871404
	XGBoost_def	0.85200	0.846062	0.481572	0.569767	0.877254
	XGBoost_tun	0.86550	0.857972	0.454545	0.579030	0.874434
	ANN	0.80450	0.758374	0.68059	0.586243	0.911081
Random	LR_def	0.708845	0.774010	0.702703	0.707046	0.706311
(undersmapling)	LR_tun	0.711302	0.776069	0.702703	0.708798	0.707729
	DT_def	0.676904	0.676904	0.673219	0.675709	0.675610
	DT_tun	0.734644	0.826748	0.759214	0.741007	0.746770
	RF_def	0.775184	0.844690	0.766585	0.773234	0.770531
	RF_tun	0.778870	0.856513	0.766585	0.776119	0.772182
	XGBoost_def	0.762899	0.831089	0.766585	0.763770	0.764851
	XGBoost_tun	0.782555	0.858436	0.761671	0.777917	0.771226
	ANN	0.762084	0.762084	0.68801	0.743051	0.728267
Tomek Links	LR_def	0.804015	0.786082	0.243243	0.347979	0.822068
(undersmapling)	LR_tun	0.804015	0.786042	0.243243	0.347979	0.822068
	DT_def	0.793450	0.70163	0.540541	0.529483	0.872703
	DT_tun	0.856841	0.845668	0.471744	0.586260	0.869301
	RF_def	0.859482	0.852952	0.513514	0.611111	0.877475
	RF_tun	0.865821	0.869892	0.511057	0.620896	0.877914
	XGBoost_def	0.854200	0.855748	0.528256	0.609065	0.879548
	XGBoost_turn	0.868991	0.878101	0.513514	0.627628	0.878825
	ANN	0.766304	0.765059	0.692908	0.744495	0.738359
NCR	LR_def	0.799726	0.828321	0.447174	0.554878	0.813896
(undersmapling)	LR_tun	0.799726	0.828302	0.449631	0.556231	0.814416
	DT_def	0.806584	0.764219	0.668305	0.658596	0.870067
	DT_tun	0.858025	0.886976	0.702703	0.734275	0.888582
	RF_def	0.861454	0.898825	0.675676	0.731383	0.881402
	RF_tun	0.870370	0.910301	0.660934	0.740028	0.878735
	XGBoost_def	0.858711	0.892441	0.685504	0.730366	0.883742
	XGBoost_tun	0.868313	0.910038	0.663391	0.737705	0.879082
	ANN	0.765815	0.764533	0.692908	0.743975	0.738359

Table. 2 Models Performance with No Sampling & Undersampling

From Table. 2, we can observe the performance of various models under different undersampling methods:

(a) Random Undersampling: The tuned Random Forest (RF_tun) and the tuned XGBoost (XGBoost_tun) perform relatively well across all metrics.

(b) Tomek Links Undersampling: Most models perform well on ACC metrics, but both Recall and F1 scores are poor. We thought that this might be due to the Tomek Links mechanism, which only removes a small number of borderline majority class samples. This may not be sufficient to significantly improve the class imbalance.

(c) Neighborhood Cleaning Rule (NCR) Undersampling: The tuned Random Forest (RF_tun) and the tuned XGBoost (XGBoost_tun) achieve relatively good performance across all metrics.

Overall, the undersampling methods perform relatively poorly compared to over-sampling methods, and are only better than no sampling. The performance of Tomek Links and NCR is lower than that of random undersampling.

Finally, we can observe the results of hybrid sampling from Table. 3:

	Model	ACC	ROC	Recall	F1_Score	NPV
No Sampling	LR_def	0.80950	0.775464	0.189189	0.287850	0.823718
	LR_tun	0.80400	0.779254	0.095823	0.165957	0.810015
	DT_def	0.79250	0.678585	0.486486	0.488286	0.869048
	DT_tun	0.86100	0.843411	0.503686	0.595930	0.882490
	RF_def	0.85600	0.843817	0.469287	0.570149	0.875648
	RF_tun	0.86200	0.842613	0.439803	0.564669	0.871404
	XGBoost_def	0.85200	0.846062	0.481572	0.569767	0.877254
	XGBoost_tun	0.86550	0.857972	0.454545	0.579030	0.874434
	ANN	0.79858	0.779837	0.723618	0.705450	0.858156
SMOTEENN	LR_def	0.791343	0.867995	0.826458	0.819753	0.760784
(hybridsmapling)	LR_tun	0.792568	0.868836	0.827881	0.820874	0.762512
	DT_def	0.849735	0.853166	0.830014	0.863805	0.792715
	DT_tun	0.871784	0.923627	0.895448	0.889124	0.856305
	RF_def	0.896693	0.961899	0.896871	0.908829	0.865741
	RF_tun	0.868518	0.944144	0.888336	0.885816	0.848309
	XGBoost_def	0.928134	0.977632	0.931721	0.937053	0.909348
	XGBoost_tun	0.890976	0.95895	0.881223	0.902732	0.84955
	ANN	0.858646	0.864548	0.808933	0.863576	0.795575
SMOTETomek	LR_def	0.723668	0.795147	0.724619	0.72393	0.724094
(hybridsmapling)	LR_tun	0.725254	0.795538	0.724619	0.725079	0.724968
	DT_def	0.787119	0.787119	0.737944	0.776109	0.76141
	DT_tun	0.822018	0.882116	0.831218	0.82364	0.828054
	RF_def	0.860406	0.930446	0.813452	0.853529	0.829466
	RF_tun	0.828363	0.908856	0.796954	0.822797	0.808955
	XGBoost_def	0.900698	0.955544	0.866117	0.897141	0.874777
	XGBoost_tun	0.873731	0.940271	0.829949	0.86795	0.843641
	ANN	0.860215	0.860215	0.808933	0.852659	0.826707

Table. 3 Models Performance with No Sampling & Hybrid Sampling

(a) SMOTEENN :The models trained with SMOTEENN demonstrate an overall improvement in performance, particularly in terms of Recall, which is crucial for imbalanced datasets. XGBoost_def stands out with the highest ACC , ROC , Recall, F1_Score and NPV among the all models. The RF_def and RF_tun models also show strong performance.

(b) SMOTETomek: The models using SMOTETomek sampling also show improved Recall compared to the No Sampling models but are not as strong as the SMOTEENN models. XGBoost_def has the best performance among the SMOTETomek models.

Both combined sampling methods significantly improved the accuracy of the Random Forest and XGBoost models. From the above three tables, we can easily see that XGBoost with default parameters is the best model if we use SMOTEENN hybrid sampling.

Feature Importance

Since our experiments show that XGBoost has the best performance, we use the feature importance parameter of the XGBoost model to look at the most and least important

features, and the difference in feature importance under different oversampling and undersampling methods.

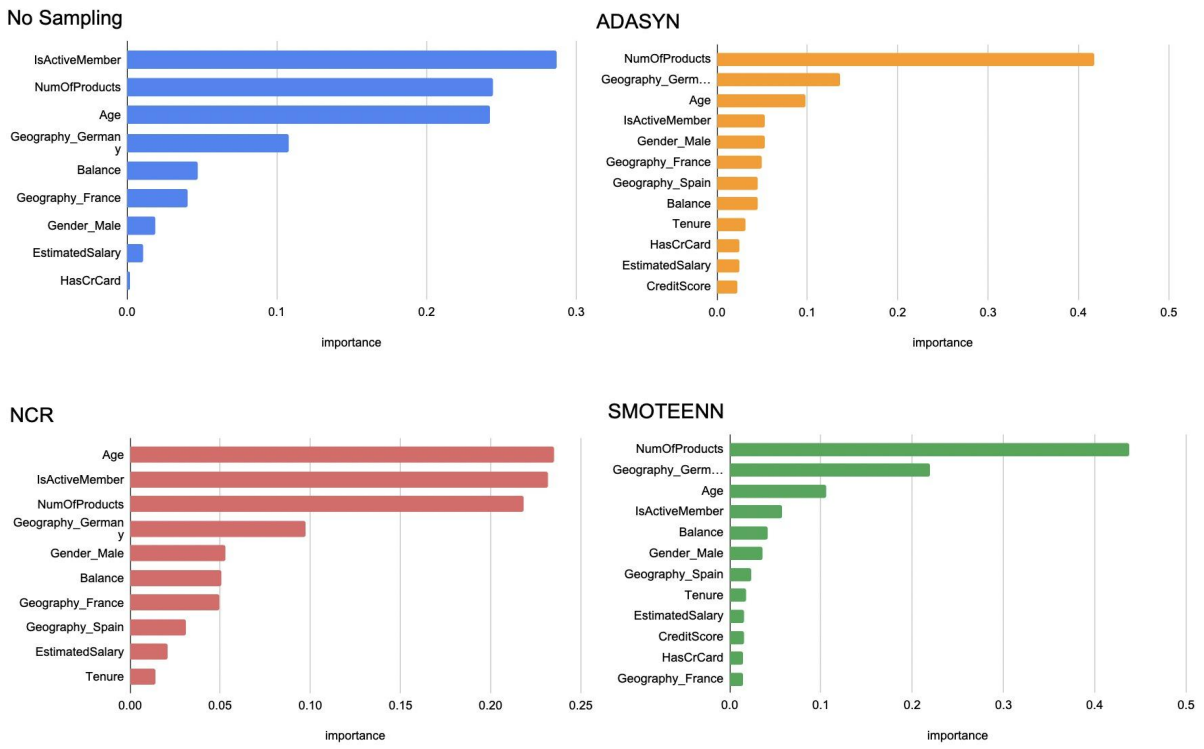


Figure. 2 Feature Importance

From Figure. 2, it can be observed that the importance of features varies under different sampling methods. This highlights the fact that the choice of sampling techniques can influence the relative importance of features in the modeling process.

For the ADASYN sampling method, the top three important features are NumOfProducts, Geography_Germany, and Age. Surprisingly, the importance of CreditScore and EstimatedSalary are the lowest among the features. This finding indicates that, when using ADASYN sampling, the model relies more on the number of products, geographic location, and age of the customers rather than their credit score and estimated salary to predict churn.

For the NCR sampling method, the top three important features are Age, IsActiveMember, and NumOfProducts. Conversely, the importance of Tenure and EstimatedSalary are the lowest among the features. This demonstrates that, when using NCR sampling, the model prioritizes factors such as customer age, active membership status, and the number of products they have in predicting churn. Meanwhile, it assigns less significance to the length since customers joined the bank (Tenure) and their estimated salary.

For the SMOTEENN sampling method, the top three important features are NumOfProducts, Geography_Germany, and Age, which is similar to the results obtained using ADASYN. The importance of Geography_France and CreditScore are lowest. This suggests that, when employing the SMOTEENN sampling method, the model emphasizes the importance of the number of products, customer location in Germany, and age in predicting churn, while giving less weight to the customer's credit score, estimated salary, and their location in France.

7.3 Customer Profiling and Analysis

After performing churn prediction, we conducted customer profiling to gain deeper insights into the characteristics and behavior of the bank's customers. Customer profiling plays a crucial role in real-world banking, as it allows banks to understand their clients better and tailor their products and services to cater to the specific needs and preferences of different customer segments.

By segmenting the customers into distinct groups based on their shared attributes, companies can devise targeted marketing campaigns, improve customer retention rates, and enhance overall customer satisfaction. Furthermore, customer profiling enables companies to identify high-value customers and focus their resources on retaining and expanding these relationships. Ultimately, a comprehensive understanding of the customer base allows companies to make informed decisions, optimize their product offerings[14].

In this project, we used the k-means algorithm for customer segmentation and delved into the characteristics of each customer group, with the aim of offering valuable insights and perspectives.

Utilizing the K-means algorithm, we segment the customers into 6 groups, with the mean values for each group presented in Table. 4. The number of customers in each group is shown in the Figure. 3.

We can analyze the characteristics of the different groups by looking at the mean value of the features in each group. The following analysis is performed for each clustering group:

a. **High-value, high churn risk customer group (corresponding to cluster 0 in the table):** This group has the second highest balance as well as the second highest churn rate. We believe that the potential value and churn risk of this group is high and the bank needs to adopt appropriate marketing strategies to increase customer satisfaction and loyalty.

	CreditScore	Age	Tenure	Balance	NumOfProdu	EstimatedSalary	Churn Rate
cluster							
0	649.97	39.42	5.050152	120259.668	1.527356	99905.03396	0.278
1	651.77	39.2	5	59862.0925	1.573003	100734.1075	0.212
2	650.06	38.3	5.049401	63546.2849	1.517254	100174.2525	0.127
3	653.09	40.15	4.965633	119145.966	1.511316	102446.4241	0.376
4	650.99	38.65	5.057637	63352.8337	1.512968	98425.68768	0.131
5	649.19	38.77	4.950022	60322.6702	1.547545	99564.25276	0.203

Table. 4 Customer Segmentation

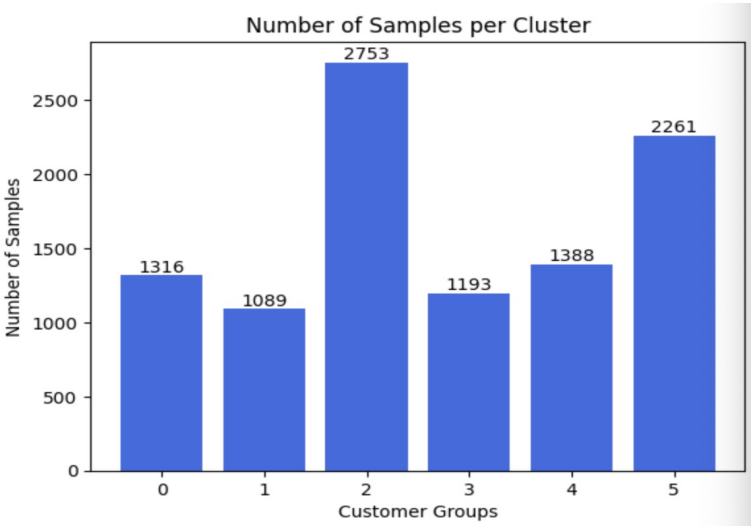


Figure. 3 Size of Each Customer Group

b. **Active customers group(corresponding to cluster 1 in the table)**: This group contains the smallest number of customers and the highest number of products. We believe that this group represents an active user base with a greater willingness to use the bank's products. As a result, the bank may consider introducing them to some new offerings.
c. **Loyal and youngest customers group(corresponding to cluster 2 in the table)**: This group is the largest in size and has the lowest churn rate, as well as the youngest age profile and a substantial estimated salary income. We believe that this group represents high-potential and high-loyalty customers. It is worthwhile for the bank to adopt appropriate marketing strategies to maintain customer engagement and retention.
d. High value, high churn risk customers group(corresponding to cluster 3 in the table): This group of customers has the highest average credit score, the highest age, the second-highest balance, and the highest estimated salary, yet they also have the highest churn rate. These customers are considered high-value and should be the bank's top priority among all groups. The bank should take proactive measures to reduce the churn rate of this user segment.
e. **High loyalty customer group(corresponding to cluster 4 in the table)**: This group of customers has the highest tenure and a relatively low churn rate. The bank should focus on maintaining the loyalty of these customers as its objective.
f. **Emerging customers group(corresponding to cluster 5 in the table)**: This group of customers has the lowest credit scores and the shortest tenure. These customers are relatively new and still have untapped market potential to be explored.

8. Lessons Learned

1. Handling imbalanced datasets: This project highlighted the importance of addressing class imbalance in datasets and evaluating the performance of models with appropriate metrics, rather than relying solely on accuracy.

2. Model selection and tuning: Experimenting with various models, along with different sampling methods, allowed us to understand their strengths and weaknesses in the context of predicting customer churn.
3. Feature engineering and preprocessing: We learned the significance of proper feature engineering, including handling categorical variables and scaling, which can significantly impact model performance.
4. Clustering for customer profiling: This project demonstrated the power of unsupervised learning, particularly k-means clustering, in uncovering hidden customer segments and providing insights into their distinct characteristics, which can help inform targeted marketing strategies.

9. Future Works

In future work, we plan to collect more data, and explore advanced sampling techniques to improve our models' performance. Additionally, we will investigate alternative clustering algorithms and ensemble methods to provide more accurate insights. Finally, we aim to develop targeted intervention strategies to retain high-risk customers and tailor marketing efforts towards specific customer segments.

10. Summary

In conclusion, this project has successfully applied various machine learning techniques to predict customer churn and perform customer profiling using a bank dataset. By comparing the performance of different models and sampling methods, we identified the most effective approaches for dealing with imbalanced data. We can conclude that XGBoost with default parameters and SMOTEENN sampling is the best model to predict customer churn with imbalanced datasets.The insights gained from clustering customers have shed light on the distinct characteristics of various customer segments.

References

[1]Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, *38*(12), 15273–15285. https://doi.org/10.1016/j.eswa.2011.06.028

[2]Glady, N., Baesens, B., & Croux, C. (2008). Modeling churn using customer lifetime value. European Journal of Operational Research. doi:10.1016/j.ejor.2008.06.027.

[3]Palaniappan, S., Mustapha, A., Mohd Foozy, C. F., & Atan, R. (2017). Customer Profiling using Classification Approach for Bank Telemarketing. JOIV : International Journal on Informatics Visualization, 1(4-2), 214–217. https://doi.org/10.30630/joiv.1.4-2.68

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic Minority Over-sampling Technique. The Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[5]Haibo He, Yang Bai, Garcia, E. A., & Shutao Li. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 10, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[6]Two Modifications of CNN. (1976). IEEE Transactions on Systems, Man, and Cybernetics, SMC-6(11), 769–772. https://doi.org/10.1109/TSMC.1976.4309452

[7]Kubat, M., & Matwin, S. (1997, July). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml* (Vol. 97, No. 1, p. 179).

[8] Fernandez, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. The Journal of Artificial Intelligence Research, 61, 863–905. https://doi.org/10.1613/jair.1.11192

[9]Geiler, L., Affeldt, S., & Nadif, M. (2022). An effective strategy for churn prediction and customer profiling. Data & Knowledge Engineering, 142, 102100. https://doi.org/10.1016/j.datak.2022.102100

[10]Statistical Analysis of Economic Data[Course lecture notes]: CHAPTER 8: MULTICOLLINEARITY. SFU. https://www.sfu.ca/~dsignori/buec333/lecture%2016.pdf

[11] Ameeruddin Mohammed. (2023). How to drop out highly correlated features in Python?[Tutorial]. Projectpro. https://www.projectpro.io/recipes/drop-out-highly-correlated-features-in-python

[12]Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. ACM International Conference Proceeding Series; Vol. 148: Proceedings of the 23rd International Conference on Machine Learning; 25-29 June 2006, 233–240. https://doi.org/10.1145/1143844.1143874

[13] Batista, G., Prati, R., & Monard, M. (2004). A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6(1), 20–29. https://doi.org/10.1145/1007730.1007735

[14]Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018, April). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP conference series: materials science and engineering* (Vol. 336, p. 012017). IOP Publishing.

Appendix

The tools we used in this project are as follows:

Pandas	scikit-learn	numpy	matplotlib
seaborn	XGBoost	imbalanced-learn	