

**Steering The Sports Industry**

Technical Assessment: **Customer Churn**

for the Role of:

**Data Scientist | MLE**

**Solution by**

**Motaseam Yousef**

Contents

[Problem Statement 3](#_Toc177931913)

[Research and Literature Review 4](#_Toc177931914)

[Articles 4](#_Toc177931915)

[Reference [1] 4](#_Toc177931916)

[Reference [2] 5](#_Toc177931917)

[Reference [3] 5](#_Toc177931918)

[Reference [4][5] 6](#_Toc177931919)

[Papers and research 9](#_Toc177931920)

[Reference [6]: A Comparison of Machine Learning Techniques for Customer Churn Prediction 9](#_Toc177931921)

[Reference [7]: Customer Churn Prediction System: A Machine Learning Approach 11](#_Toc177931922)

[GitHub and Kaggle 13](#_Toc177931923)

[Reference [8]: 13](#_Toc177931924)

[Reference [9]: 14](#_Toc177931925)

[Data Analysis 17](#_Toc177931926)

[EDA 21](#_Toc177931927)

[Preprocessing depends on the previous results 32](#_Toc177931928)

[Conclusion 39](#_Toc177931929)

[Future Work 40](#_Toc177931930)

[References 40](#_Toc177931931)

[Appendix 1 41](#_Toc177931932)

Table of figures

[Figure 1 models evaluation results [8] 14](#_Toc177932100)

[Figure 2 feature importance [8] 14](#_Toc177932101)

[Figure 3 churn for all regions 21](#_Toc177932102)

[Figure 4 regions shape 21](#_Toc177932103)

[Figure 5 models results 35](#_Toc177932104)

[Figure 6 cm for the Xgboost 36](#_Toc177932105)

[Figure 7 feature importance for RF 37](#_Toc177932106)

[Figure 8 feature importance for xgboost 37](#_Toc177932107)

[Figure 9 test API 39](#_Toc177932108)

# Problem Statement

Customer churn refers to the loss of customers who stop using a service or product. Identifying which customers are at risk of leaving is vital to improving retention rates. Accurately predicting churn can help businesses implement targeted strategies to reduce customer loss.

In this report, a data analysis will be conducted to better understand the underlying patterns and trends within the data. Following this, machine learning models will be developed to predict customer churn, providing valuable insights for optimizing customer retention and driving sales growth.

# Research and Literature Review

In this section, research will be conducted to explore how others have addressed customer churn problem and how they solve it. The research will examine preprocessing and data cleaning techniques, the features used, the models implemented, feature engineering methods, and any relevant ideas that contributed to achieving effective results.

## Articles

### Reference [1]

1. Definition:

Customer churn (attrition) refers to the loss of customers over a specific period. In subscription-based models like SaaS, tracking churn is essential to understanding business health.

2. Churn Rate Formula:

3. Types of Churn:

- Active (Voluntary) churn: Customers actively choose to leave, often due to dissatisfaction or better offers.

- Passive (Involuntary) churn: Customers unintentionally leave due to external factors like payment failures, which can be addressed with proactive solutions.

4. Importance:

High churn directly affects Monthly Recurring Revenue (MRR). Given the high Customer Acquisition Cost (CAC) in SaaS, it is more cost-efficient to retain customers than constantly acquiring new ones.

5. Reduction Strategies:

Improving customer satisfaction and addressing involuntary churn (e.g., payment issues) through tools and feedback loops are key tactics to reduce churn.

6. Retention Focus:

Prioritizing customer retention boosts MRR, while upselling to existing customers is more effective and profitable than acquiring new customers.

7. Benchmarks:

A churn rate of 3-5% monthly is considered good for SaaS companies, with lower rates supporting faster growth.

### Reference [2]

**Churn vs. Growth Rate**:

* **Churn Rate**: Tracks how many customers a business loses.
* **Growth Rate**: Tracks how many new customers a business gains.
* The balance between churn and growth shows if a business is expanding or shrinking.

**Pros of Churn Rate**:

* Clearly shows how well a business is keeping its customers.
* Helps spot problems like bad service or high costs.
* Reflects customer satisfaction—whether they stay or leave.

### Reference [3]

**Causes of Churn**:

* Poor customer service
* Misfit between product and customer needs
* Inadequate pricing
* Competition offering better products or services
* Seasonality in customer demand

**Reducing Churn**:

* Improve customer experience (CX)
* Educate customers about product usage
* Offer loyalty rewards and recognize valuable customers
* Use customer feedback to address potential issues early

### Reference [4][5]

**Best Practices to Reduce Churn**:

* Use **analytics** to track customer behavior and predict churn.
* Segment customers into **cohorts** based on behavior or demographics to understand loyalty trends.
* Ensure **product stickiness** by encouraging customers to use more features regularly.
* Provide educational materials and **incentives** to retain customers.
* Continuously evolve the product to meet changing customer needs. [4]

**How to Improve Churn Rate:**

* Improve Customer Service: Resolve issues quickly with better communication channels.
* Analyze Customer Experience: Identify pain points through surveys or feedback.
* Enhance Content Strategy: Keep customers engaged with helpful and relevant content. [5]

**Summary**

**- Churn rate Formula:**

**- Types of Churn:**

* Active (Voluntary) Churn: Customers leave by choice, often due to dissatisfaction or better alternatives.
* Passive (Involuntary) Churn: Customers leave unintentionally, often due to factors like payment issues.

**Importance of Churn:**

* High churn decreases Monthly Recurring Revenue (MRR).
* Customer Acquisition Cost (CAC) is higher than the cost of retaining customers, making retention more cost-effective.
* Growth vs. Churn: Growth tracks new customers, while churn tracks losses. The balance between these defines business expansion or shrinkage.

**Causes of Churn:**

* Poor customer service
* Product not meeting customer needs
* Pricing issues
* Competitors offering better alternatives
* Seasonality in customer demand

**Churn Reduction Strategies:**

* Improve customer service by resolving issues faster and through better communication.
* Address involuntary churn (e.g., payment failures) with proactive tools.
* Enhance customer experience (CX): Use feedback loops and analytics to address pain points.
* Educate customers on product usage and features to ensure they extract maximum value.
* Offer loyalty rewards and recognize valuable customers.
* Use analytics to predict churn and segment customers based on behavior to identify loyalty trends.

**Focus on Retention:**

* Retaining existing customers boosts MRR and reduces the cost of churn.
* Upselling to current customers is more profitable than acquiring new ones.
* Ensure product stickiness by encouraging customers to use more features regularly.

**Best Practices to Reduce Churn:**

* Segment customers by behavior or demographics to understand loyalty patterns.
* Continuously evolve the product to meet changing customer needs.
* Provide educational materials and incentives to retain customers.
* Use analytics to track customer behavior and identify early signs of churn.

## Papers and research

### Reference [6]: A Comparison of Machine Learning Techniques for Customer Churn Prediction

This paper compares various machine learning methods applied to customer churn prediction, particularly in the **telecommunications industry**. The study focuses on the performance of five classifiers: Artificial Neural Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM), Naïve Bayes (NB), and Logistic Regression (LR). It also explores the performance improvements offered by boosting techniques, particularly AdaBoost.

**Preprocessing**

Explanation of Excluding the Categorical "State" Variable:

The "state" variable is a categorical feature that represents the geographic location of the customer. In this paper, the authors excluded the "state" variable from the analysis because it was too specific and not necessary for their goal of creating a general model that can apply across different regions. In simpler terms, they wanted the model to focus on broader customer behavior patterns rather than location-specific information.

**Models and Techniques:**

1. Artificial Neural Networks (ANN):

- The number of neurons in the hidden layer varied between 5 and 45 in the experiments.

2. Support Vector Machines (SVM):

- Two types of kernels were used:

- Radial Basis Function (RBF)

- **Polynomial (POLY)**

3. Decision Trees (DT):

4. Naïve Bayes (NB):

5. Logistic Regression (LR):

**Boosting Algorithm:**

The AdaBoost algorithm was applied to improve the performance of ANN, SVM, and DT models. This technique iteratively adjusts the weights of incorrectly classified samples, enhancing the ability of the model to handle difficult cases.

**Results:**

- Boosted SVM with Polynomial Kernel (SVM-POLY + AdaBoost) performed the best, achieving nearly 97% accuracy and an F-measure over 84%.

- Boosting significantly improved the performance of the models, particularly in terms of the F-measure, which combines precision and recall.

### Reference [7]: Customer Churn Prediction System: A Machine Learning Approach

This paper introduces a Customer Churn Prediction (CCP) system tailored for the telecom industry, utilizing machine learning techniques to predict customer churn. The goal is to identify customers who are likely to leave (churn) and take proactive actions to retain them. The proposed model consists of several phases, including data preprocessing, feature selection, model training, and performance evaluation, with a focus on using ensemble methods to improve accuracy.

**Preprocessing and Data Cleaning**

Data preprocessing involved several key steps:

- Handling Missing Values: Missing data was handled during the preprocessing phase, though the specific method used (e.g., filling with mean/mode or dropping rows/columns) was not explicitly mentioned. However, it’s noted that non-relevant parameters and rows with missing data were likely dropped to ensure clean data.

- Feature Selection: The Gravitational Search Algorithm (GSA) was used to select the most significant features, reducing dimensionality and enhancing model performance. The fitness function in GSA focused on minimizing the error rate to select only the most impactful features.

- Class Balancing: Resampling techniques were applied to address class imbalance between churners and non-churners, improving model performance and ensuring accurate predictions.

**Modeling and Machine Learning Techniques**

Several machine learning models were employed in this study:

- Logistic Regression

- Naive Bayes

- Support Vector Machine (SVM)

- Decision Trees

- Random Forest

- Ensemble Techniques:

- AdaBoost

- XGBoost

- CatBoost

The models were trained on the dataset and evaluated using metrics like accuracy, precision, recall, and AUC score. Additionally, K-fold cross-validation (with K=5) was used to tune hyperparameters and prevent overfitting.

**Results and Evaluation**

The performance of each model was evaluated based on accuracy and the AUC (Area Under the Curve) score:

- AdaBoost: Achieved the highest accuracy of 81.71% with an AUC score of 84%.

- XGBoost: Achieved an accuracy of 80.8% and also an AUC score of 84%.

- CatBoost: Performed well with 81.8% accuracy and an AUC score of 82%.

## GitHub and Kaggle

### Reference [8]:

From this repository, they primarily test different ML models, and the results are shown in the image below. However, due to the imbalance of data, the models they built are quite weak.

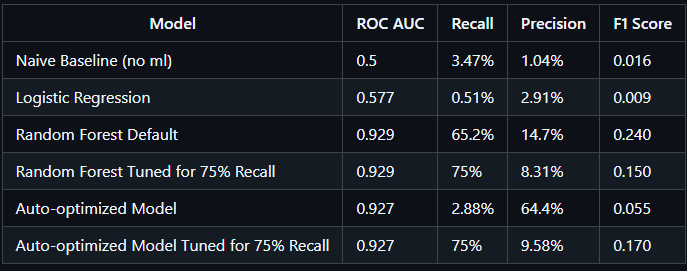


Figure 1 models evaluation results [8]

In the figure below, the feature importance shows that the most important feature is the "paid previous month."

A screenshot of a graph

Description automatically generated

Figure 2 feature importance [8]

### Reference [9]:

**Customer Churn Insights and Analysis**

**Key Fact:** Did you know that attracting a new customer costs five times as much as retaining an existing one? This highlights the importance of managing customer churn effectively.

**The Impact of Customer Churn:**

Increasing churn rates often indicate strong competition in the market. Detecting early signs of potential churn requires a comprehensive understanding of customer behavior. This involves analyzing various interactions, including:

* Store or branch visits
* Product purchase histories
* Customer service interactions
* Web-based transactions
* Social media activity

**Data Analysis Focus:**

The data exploration aims to address the following questions:

* What percentage of customers are churning versus those staying with active services?
* Are there any patterns in customer churn based on gender?
* Do customer preferences or patterns differ based on the type of service provided?
* Which service types are the most profitable?
* Which features and services generate the highest profit?
* Additional questions may arise as the analysis progresses.

**Machine Learning Model Testing:**

To enhance churn prediction accuracy, several machine learning algorithms were tested, including:

1. **Gradient Boosting Classifier**
2. **Logistic Regression**
3. **AdaBoost Classifier**

A voting ensemble method was implemented to combine the results of these models, improving overall prediction accuracy.

**Summary**

Preprocessing & Data Preparation:

* Handling Missing Data: Irrelevant parameters and rows with missing data were dropped to ensure clean data for analysis.
* Feature Selection:
  + Gravitational Search Algorithm (GSA): Used to select the most significant features, reducing dimensionality and improving model performance.
* Class Balancing: Resampling techniques were applied to address class imbalance between churners and non-churners, enhancing model accuracy.

Models & Techniques:

1. Artificial Neural Networks (ANN):
2. Support Vector Machines (SVM): Two types of kernels were tested:
   * Radial Basis Function (RBF)
   * Polynomial (POLY)
3. Decision Trees (DT)
4. Naïve Bayes (NB)
5. Logistic Regression (LR)
6. Random Forest
7. Ensemble Techniques:
   * AdaBoost
   * XGBoost
   * CatBoost

Best Algorithms:

* SVM with Polynomial Kernel + AdaBoost: Achieved the highest accuracy of 97% and an F-measure over 84%.
* AdaBoost: Consistently improved model performance, achieving the best overall accuracy of 81.71% in the second study.
* XGBoost and CatBoost also performed well, with accuracies of 80.8% and 81.8%, respectively, making them strong contenders for customer churn prediction.

**The most important feature is the "paid previous month."**

**Data Analysis Focus:**

The data exploration aims to address the following questions:

* What percentage of customers are churning versus those staying with active services?
* Are there any patterns in customer churn based on gender?
* Do customer preferences or patterns differ based on the type of service provided?
* Which service types are the most profitable?
* Which features and services generate the highest profit?

# Data Analysis

- The dataset contains 1,927,531 records and 164 columns.

**Column Naming and Descriptions:**

Personal Information:

- PERSON\_ID: A unique identifier for the individual.

- AGE: The age of the person in years.

- STATE: The person's membership or program status (e.g. ACTIVE).

- PERSON\_TYPE: The type or classification of the person (e.g., PRIVATE).

- GENDER: The person’s gender (e.g., M for Male, F for Female).

- CENTER\_REGION: The region of the center associated with the person (e.g., WR for Western Region).

- CITY: The city where the center or the person is located.

- TOT\_SUBS: Total number of subscriptions the person has.

- REJOIN\_CNT: The number of times the person rejoined after cancellation.

- RENEWAL\_CNT: The number of subscription renewals by the person.

- NEWSALE\_CNT: The number of new sales made to the person.

- NO\_OF\_CENTER: The total number of centers associated with the person.

- SUBS\_CREATION\_DATE: The date when the person's subscription was created.

- OUTAGE: Indicates if the service experienced any outages.

- SUBS\_DAYS: The total number of days the subscription has been active.

- OUTAGE\_PERC: The percentage of time the subscription was affected by outages.

- OUTAGE\_TILLNOW: The cumulative duration of service outages up to the current date.

Program Names:

- GX: Program name.

- PT: Program name.

- BOTH\_PROGRAM: Indicates if the person is enrolled in multiple programs.

- FT90\_P: Program name.

- NO\_PROGRAM: Indicates that the person is not currently enrolled in any programs.

Type of Center:

- FT\_CENTER: Indicates involvement with a full-time center.

- PRO\_CENTER: Indicates involvement with a professional center.

- PLUS\_CENTER: Indicates involvement with a plus center.

- XPRESS\_CENTER: Indicates involvement with an express center.

- POPUP\_CENTER: Indicates involvement with a pop-up center.

- JUNIOR\_CENTER: Indicates involvement with a junior center.

- HQ\_CENTER: Indicates involvement with the headquarters center.

- NO\_OF\_PRODUCTS: The number of products the person has purchased or subscribed to.

Packages:

- MONTH\_12\_PKG: Status of a 12-month subscription package.

- MONTH\_9\_PKG: Status of a 9-month subscription package.

- MONTH\_6\_PKG: Status of a 6-month subscription package.

- MONTH\_3\_PKG: Status of a 3-month subscription package.

- MONTH\_1\_PKG: Status of a 1-month subscription package.

- DAYS\_1\_PKG: Status of a 1-day subscription package.

Dates:

- Visits (YYYY/MM): The number of visits made by the person during a specific year and month.

- Subscription (YYYY/MM): The status of the person’s subscription during a specific year and month.

**Analysis**

- There are no duplicates in the data.

Question to deep understanding the data  
(PS: in my case I will make assumption to move forward with clean approach)

* + How is the STATE actually determined?
  + For the Region or CITY, how can we determine how many gyms are in that area?
  + For the Type of Center, does this refer to another gym or a private area within the same gym?
  + What do Packages actually mean? Are they product packages or subscription packages (possibly including tools, products, etc.)?
  + Regarding Visits and Subscriptions, it seems like these are new features that were added, as there are a lot of missing values or removed by masked for sensitivity of data (**I will dealing with them as is**).

Assumptions to move forward:

* + Assume the Packages is a subscription offer (This will be helpful in the Next best offer model)
  + I will check the CITY and CENTER\_REGION to focus the study on a specific area first, as the data is large.
  + Based on my experience, it's better to cluster data for large and diverse areas separately. Combining all at once could create outliers, which may cause models to fail.
  + For the churn Label I will assume the following from the STATE columns:
    - INACTIVE = Churned
    - ACTIVE = Not Churned
    - FROZEN = Maybe Churned (drop for now case study)
    - Future work: use other columns (like visit frequency, subscription history, outages) to strengthen your prediction, especially for ambiguous cases like FROZEN.

I will split the data based on the CENTER\_REGION. However, before doing this, I will drop the empty columns to focus more on the problem and reduce storage size.

The columns “Visits : 2019 / 01” to “Visits : 2023 / 12” totally missing -> so drop them  
The others column in visits and Subscription have a lot of missing values -> but for those which not missing they enough to make some studies on them  
may I will drop them in or make a feature from them for churn prediction model.

Now before splitting data for each region, check over all churn

The churn for this data for all regions together shown below:

A screenshot of a computer

Description automatically generated

Figure 3 churn for all regions

Split data for all regions results shown below the size of data for each regions

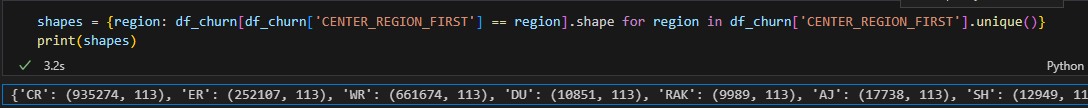


Figure 4 regions shape

Following for more details about the split

{'CR': (935274, 113), 'ER': (252107, 113), 'WR': (661674, 113), 'DU': (10851, 113), 'RAK': (9989, 113), 'AJ': (17738, 113), 'SH': (12949, 113)}

## EDA

In this section I will answer the following questions to understand data better

* + How much churn for each region?
  + For categorical data -> make counts to check our clients and the relation to the churn
  + For numerical data -> check the distribution and if there outliers
  + Correlation between features themselves
  + Take one region and check the other factors correlation and how they effect on churn (as Gender, age, ets)

**How much churn for each region?**

{'CR': 0.7585370704200053, 'ER': 0.8051184616055881, 'WR': 0.7972793248639057, 'DU': 0.7930144687125611, 'RAK': 0.6997697467213936, 'AJ': 0.7346375014094035, 'SH': 0.7206734110742142}

The values close to each other

Max in ER -> 80.5%

Min in RAK -> ~70%

**For categorical data -> make counts to check our clients** **and the relation to the churn**

**PERSON\_TYPE**

- Highest:

- ONEMANCORPORATE: Churn = 1.00 (Active: 0, Inactive: 1)

- STAFF: Churn = 0.973 (Active: 67, Inactive: 2,461)

- Note: Very high churn among STAFF, but their numbers are low compared to other categories.

- Note: FAMILY, STUDENT, PRIVATE: have same churn ~80%

- Lowest churn:

- CORPORATE: Churn = 0.266 (Active: 60,332, Inactive: 21,824)

- GUEST: Churn 0.613883 (Active: 356, Inactive:566)

**GENDER**

- Highest:

- F: Churn = 0.823 (Active: 87,718, Inactive: 407,763)

- Lowest:

- M: Churn = 0.762 (Active: 334,951, Inactive: 1,070,150)

Men less churn than females

**GX**

- Highest:

- No Showup: Churn = 0.822 (Active: 255,921, Inactive: 1,181,583)

- Lowest:

- GX Showup: Churn = 0.640 (Active: 166,748, Inactive: 296,330)

**PT**

- Highest:

- No Showup: Churn = 0.796 (Active: 360,896, Inactive: 1,406,160)

- Lowest:

- PT Showup: Churn = 0.537 (Active: 61,773, Inactive: 71,753)

**BOTH\_PROGRAM**

- Highest:

- No Take Programs: Churn = 0.833 (Active: 229,982, Inactive: 1,145,252)

- Note: People who don't take any programs have the highest churn, with large numbers.

- Lowest:

- both programs: Churn = 0.497 (Active: 35,834, Inactive: 35,422)

**FT90\_P**

- Highest:

- No Showup: Churn = 0.817 (Active: 305,141, Inactive: 1,363,845)

- Lowest:

- FT90 Showup: Churn = 0.493 (Active: 117,528, Inactive: 114,068)

**NO\_PRGRAM**

- Highest:

- Go’s Solo: Churn = 0.853 (Active: 191,058, Inactive: 1,106,997)

- Lowest:

- At least one program: Churn = 0.616 (Active: 231,611, Inactive: 370,916)

People how take programs less churn than others

**CENTER\_REGION\_FIRST**

- Highest:

- ER: Churn = 0.805 (Active: 49,131, Inactive: 202,976)

- Lowest:

- RAK: Churn = 0.700 (Active: 2,999, Inactive: 6,990)

**CENTER\_REGION\_CHANGE**

- Highest:

- False: Churn = 0.790 (Active: 374,437, Inactive: 1,408,450)

- Lowest:

- True: Churn = 0.590 (Active: 48,232, Inactive: 69,463)

People who change their regions less churn!

**CITY\_FIRST**

- Highest:

- Khamis Mushait: Churn = 0.864 (Active: 3,585, Inactive: 22,730)

- Hafar Al Batin: Churn = 0.839 (Active: 2,045, Inactive: 10,688)

- Lowest:

- Ar Rass: Churn = 0.007 (Active: 135, Inactive: 1)

- Sakaka: Churn = 0.000 (Active: 99, Inactive: 0)

- Jazan, Muzahmiyah, Majmaah: Churn= ~0.20

Numbers for ACTIVE and INACTIVE for them

Muzahmiyah (516 152),

Majmaah (1523 454),

Jazan (429 110)

Increasing the people -> increase the churn

**CITY\_CHANGE**

- Highest:

- False: Churn = 0.796 (Active: 352,837, Inactive: 1,375,804)

- Lowest:

- True: Churn = 0.594 (Active: 69,832, Inactive: 102,109)

People who change the city less churn!

**For numerical data -> check the distribution and if there outliers**

**Correlation between features themselves**

**1. AGE:**

- ACTIVE: Mean = 31.51, Median = 30, Min = 15, Max = 143.

- INACTIVE: Mean = 32.34, Median = 31, Min = 2, Max = 595.

- Note: Ages below 10 and above 70 should be cleaned (dropped) due to potential data errors.

**2. TOT\_SUBS:**

- ACTIVE: Mean = 3.49, Median = 3, Min = 1, Max = 34.

- INACTIVE: Mean = 1.89, Median = 1, Min = 1, Max = 44.

- Note: Outliers over 30 for both groups may require review for data anomalies.

**3. REJOIN\_CNT:**

- ACTIVE: Mean = 1.35, Median = 1, Min = 0, Max = 16.

- INACTIVE: Mean = 0.64, Median = 0, Min = 0, Max = 14.

- Note: Values over 10, particularly in the ACTIVE group, could be flagged for review.

**4. RENEWAL\_CNT:**

- ACTIVE: Mean = 1.26, Median = 0, Min = 0, Max = 33.

- INACTIVE: Mean = 0.33, Median = 0, Min = 0, Max = 43.

- Note: Renewal counts above 30 might indicate outliers in the data.

**5. NEWSALE\_CNT:**

- ACTIVE: Mean = 0.88, Median = 1, Min = 0, Max = 2.

- INACTIVE: Mean = 0.92, Median = 1, Min = 0, Max = 1.

- Note: No significant outliers, data appears consistent.

**6. FT\_CENTER:**

- ACTIVE: Mean = 2.12, Median = 1, Min = 0, Max = 32.

- INACTIVE: Mean = 1.09, Median = 1, Min = 0, Max = 44.

- Note: Values over 30 should be reviewed for possible outliers.

**7. PRO\_CENTER:**

- ACTIVE: Mean = 0.94, Median = 0, Min = 0, Max = 34.

- INACTIVE: Mean = 0.62, Median = 0, Min = 0, Max = 26.

- Note: Counts over 20 could be outliers and may need review.

**8. PLUS\_CENTER:**

- ACTIVE: Mean = 0.03, Median = 0, Min = 0, Max = 13.

- INACTIVE: Mean = 0.01, Median = 0, Min = 0, Max = 9.

- Note: No significant outliers.

**9. XPRESS\_CENTER:**

- ACTIVE: Mean = 0.34, Median = 0, Min = 0, Max = 34.

- INACTIVE: Mean = 0.13, Median = 0, Min = 0, Max = 24.

- Note: Values above 30 in ACTIVE and above 20 in INACTIVE should be checked for anomalies.

**10. POPUP\_CENTER & JUNIOR\_CENTER:**

- Both have minimal counts with no significant outliers or anomalies.

**11. HQ\_CENTER:**

- ACTIVE: Mean = 0.06, Median = 0, Min = 0, Max = 7.

- INACTIVE: Mean = 0.05, Median = 0, Min = 0, Max = 9.

- Note: Data is consistent with no significant outliers.

**12. NO\_OF\_PRODUCTS:**

- ACTIVE: Mean = 3.49, Median = 3, Min = 1, Max = 34.

- INACTIVE: Mean = 1.89, Median = 1, Min = 1, Max = 29.

- Note: Max values over 30 for ACTIVE should be reviewed as possible outliers.

**13. MONTH\_X\_PKG (12, 9, 6, 3, 1):**

- ACTIVE group generally has higher means and maximums than INACTIVE.

- Note: Extreme values, especially for MONTH\_12\_PKG and MONTH\_6\_PKG, with maxes up to 18, could indicate outliers.

**14. DAYS\_1\_PKG:**

- ACTIVE: Mean = 0.01, Median = 0, Min = 0, Max = 15.

- INACTIVE: Mean = 0.01, Median = 0, Min = 0, Max = 18.

- Note: Values around 15-18 could be flagged for review as potential anomalies.

**15. NO\_OF\_CENTER:**

- ACTIVE: Mean = 3.49, Median = 3, Min = 1, Max = 34.

- INACTIVE: Mean = 1.89, Median = 1, Min = 1, Max = 29.

- Note: Max values over 30 for ACTIVE and over 25 for INACTIVE should be reviewed.

**16. SUBS\_CREATION\_DATE:**

- ACTIVE: Mean = 70607786.59, Median = 60632074, Min = 20191201, Max = 687733052.

- INACTIVE: Mean = 38264798.36, Median = 20230904, Min = 20180101, Max = 586149358.

- Note: Large range of dates, values over 600000000 should be checked for data entry errors.

**17. OUTAGE:**

- ACTIVE: Mean = 470.18, Median = 139, Min = 0, Max = 4380.

- INACTIVE: Mean = 249.16, Median = 0, Min = 0, Max = 2452.

- Note: Max values over 4000 may represent outliers.

**18. SUBS\_DAYS:**

- ACTIVE: Mean = 652.67, Median = 443, Min = 1, Max = 6804.

- INACTIVE: Mean = 290.22, Median = 182, Min = -2, Max = 2720.

- Note: Negative values and extreme maxes (above 6000) should be cleaned as potential errors.

**19. OUTAGE\_PERC:**

- ACTIVE: Mean = 5.47, Median = 0.78, Min = -1071.34, Max = 2268.03.

- INACTIVE: Mean = 3.10, Median = 0, Min = -1717.53, Max = 2296.01.

- Note: Negative percentages and extreme positive values should be cleaned as outliers.

**20. OUTAGE\_TILLNOW:**

- ACTIVE: Mean = 71598.89, Median = 30349, Min = 0, Max = 1162008.

- INACTIVE: Mean = 61990.13, Median = 49877, Min = 0, Max = 952278.

- Note: Extremely high values over 1 million should be reviewed as potential outliers.

**Summary:**

* The **JUNIOR\_CENTER** should be dropped because it only contains one value (all 0's).
* **Age** must be cleaned to retain data only between 15 and 70.
* Since there are outliers in the data, it’s best to use a **Standard Scaler** to reduce their impact.
* For the correlation between **Centers**, **Packages**, and **Programs between each other for each class have low correlations, Keep All.**

**Correlation between features themselves**

|  |  |  |
| --- | --- | --- |
| Feature1 | Feature2 | Correlation |
| NO\_OF\_PRODUCTS | NO\_OF\_CENTER | 0.999999624 |
| SUBS\_CREATION\_DATE | NO\_OF\_PRODUCTS | 0.999999474 |
| NO\_OF\_CENTER | SUBS\_CREATION\_DATE | 0.999999349 |
| TOT\_SUBS | NO\_OF\_CENTER | 0.999719922 |
| TOT\_SUBS | NO\_OF\_PRODUCTS | 0.999719546 |
| SUBS\_CREATION\_DATE | TOT\_SUBS | 0.999719111 |
| NO\_OF\_PRODUCTS | OUTAGE\_TILLNOW | 0.858498297 |
| OUTAGE\_TILLNOW | NO\_OF\_CENTER | 0.858498177 |
| OUTAGE\_TILLNOW | TOT\_SUBS | 0.85834951 |
| OUTAGE\_TILLNOW | SUBS\_CREATION\_DATE | 0.858039053 |
| TOT\_SUBS | RENEWAL\_CNT | 0.836638909 |
| SUBS\_CREATION\_DATE | RENEWAL\_CNT | 0.83590151 |
| RENEWAL\_CNT | NO\_OF\_CENTER | 0.835890525 |
| RENEWAL\_CNT | NO\_OF\_PRODUCTS | 0.835890174 |
| MONTH\_3\_PKG | SUBS\_CREATION\_DATE | 0.813450113 |
| NO\_OF\_CENTER | MONTH\_3\_PKG | 0.813435644 |
| MONTH\_3\_PKG | NO\_OF\_PRODUCTS | 0.81343528 |
| TOT\_SUBS | MONTH\_3\_PKG | 0.812974736 |
| NO\_OF\_CENTER | REJOIN\_CNT | 0.801809497 |
| NO\_OF\_PRODUCTS | REJOIN\_CNT | 0.801809268 |
| REJOIN\_CNT | SUBS\_CREATION\_DATE | 0.801753571 |
| TOT\_SUBS | REJOIN\_CNT | 0.801503148 |
| TOT\_SUBS | SUBS\_DAYS | 0.799353976 |
| NO\_OF\_PRODUCTS | SUBS\_DAYS | 0.799292317 |
| SUBS\_DAYS | NO\_OF\_CENTER | 0.799291778 |
| SUBS\_DAYS | SUBS\_CREATION\_DATE | 0.799230807 |

**Recommendations:**

1. Drop `**NO\_OF\_CENTER**`: It is highly correlated with both `**NO\_OF\_PRODUCTS**` and `**TOT\_SUBS**`.

2. Drop `**SUBS\_CREATION\_DATE`:** It is highly correlated with `**NO\_OF\_PRODUCTS**`, `**TOT\_SUBS**`, and other features.

3. Monitor `OUTAGE\_TILLNOW` and `RENEWAL\_CNT`: They show moderate correlations and could be considered for removal if they don't add much predictive value.

**From the correlation with the target columns**

|  |  |
| --- | --- |
| **Feature name** | **STATE\_FIRST\_BINARY** |
| NEWSALE\_CNT | 0.053478196 |
| AGE | 0.032548945 |
| DAYS\_1\_PKG | -0.000503255 |
| MONTH\_1\_PKG | -0.017599959 |
| HQ\_CENTER | -0.021999517 |
| POPUP\_CENTER | -0.025440221 |
| OUTAGE\_PERC | -0.032582077 |
| PLUS\_CENTER | -0.043863169 |
| OUTAGE\_TILLNOW | -0.055280073 |
| MONTH\_9\_PKG | -0.088940346 |
| PRO\_CENTER | -0.093904436 |
| XPRESS\_CENTER | -0.132099413 |
| MONTH\_3\_PKG | -0.17000851 |
| OUTAGE | -0.182217435 |
| MONTH\_6\_PKG | -0.19661958 |
| REJOIN\_CNT | -0.23376083 |
| FT\_CENTER | -0.24490881 |
| MONTH\_12\_PKG | -0.281882261 |
| RENEWAL\_CNT | -0.288625368 |
| TOT\_SUBS | -0.324412408 |
| NO\_OF\_PRODUCTS | -0.324414576 |
| NO\_OF\_CENTER | -0.324415678 |
| SUBS\_CREATION\_DATE | -0.32480457 |
| SUBS\_DAYS | -0.369055 |

- The correlation with the target (`STATE\_FIRST\_BINARY`) for most features is very low, indicating weak linear relationships with the target.

- Keep `NEWSALE\_CNT` and `AGE`: These have the highest (although still weak) correlations with the target.

- Consider removing features like `**DAYS\_1\_PKG**`, `**MONTH\_1\_PKG**`, and `**HQ\_CENTER**`, as they show near-zero or negative correlations, which may not contribute to the model.

If the model **overfit** -> then good choice to drop them!

## Preprocessing depends on the previous results

- Columns to drop: `SUBS\_CREATION\_DATE`, `NO\_OF\_CENTER`, `JUNIOR\_CENTER`

- Reason: Dropping these columns to prevent data leakage.

- Filter `AGE`: Keep only ages greater than 15 and less than 60, and remove the others.

- Feature removal consideration:

- Consider removing features like `DAYS\_1\_PKG`, `MONTH\_1\_PKG`, and `HQ\_CENTER` due to their near-zero or negative correlations.

- If model overfits, dropping these features is a good choice.

Here’s your document with the points in a simple, short, and direct format:

**For categorical data:**

1. PERSON\_TYPE  
PRIVATE and CORPORATE: Only these two categories are kept due to low representation of other values.

Then PRIVATE -> 0 , CORPORATE->1

2. GENDER:

- M -> 0

- F -> 1

3. GX:

- No Showup -> 0

- Showup -> 1

4. PT:

- No Showup -> 0

- Showup -> 1

5. FT90\_P:

- No Showup -> 0

- Showup -> 1

6. NO\_PROGRAM:

- Go’s Solo -> 0

- At least one program -> 1

7. CENTER\_REGION\_CHANGE:

- False -> 0

- True -> 1

8. CITY\_CHANGE:

- False -> 0

- True -> 1

9. BOTH\_PROGRAM:

- drop cause we have data for each program!

10. CITY\_FIRST:

- 35 values -> Target Encoding -> "Mean of the target variable"

11. CENTER\_REGION\_FIRST: Target encoding

- Try at the first Target encoding -> name it as CENTER\_REGION

12. Target: STATE\_FIRST -> churn

- INACTIVE -> 0 -> churn

- ACTIVE -> 1 -> not churn

**Next and for the numerical features**

* **Initialize the scaler**: Set up the scaler for normalizing the data.  
  used -> Standard scaler cause not effect by the outliers.
* **Fit and transform the training data**: Calculate the necessary scaling parameters (mean and standard deviation) from the training data and apply the transformation.
* **Transform the test data**: Apply the same transformation to the test data to ensure consistent scaling across both datasets.

**Model to used and simple discussions about each**

From the researches, the best algorithms to this problem are:

* SVM with Polynomial Kernel: the data is huge this will not be good enough!
* AdaBoost
* XGBoost

So, I will start by Random forest, then check the XGBOOST and AdaBoost

Because the data is large this will take a long time!

After this I will tune the best model

The results shown below

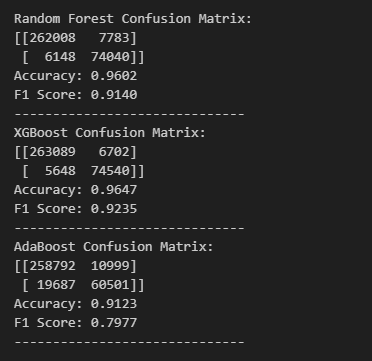


Figure 5 models results

Best is XGBOOST

This its confusion matrix for more details shown below:

A blue squares with white text

Description automatically generated

Figure 6 cm for the Xgboost

The features importance shown below, for RF and XGBoost cause this have better results

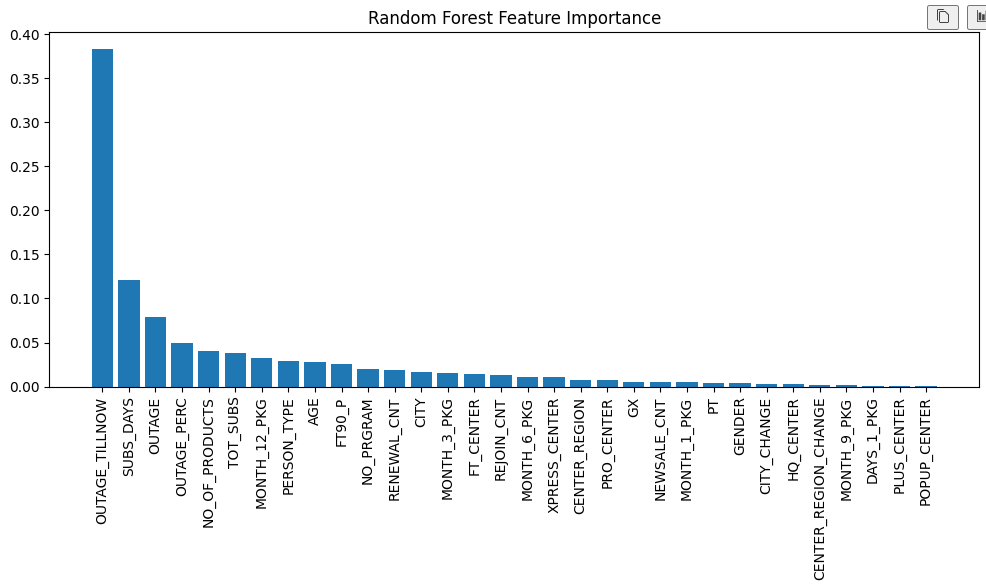


Figure 7 feature importance for RF

A graph of a number of blue bars

Description automatically generated with medium confidence

Figure 8 feature importance for xgboost

XGBoost Model:

- Top 10 important features: NO\_OF\_PRODUCTS, NO\_PRGRAM, OUTAGE\_TILLNOW, SUB5\_DAYS, PERSON\_TYPE, FT90\_P, OUTAGE, MONTH1\_12\_PKG, MONTH\_6\_PKG, NEWSALE\_CNT.

- Least 3 important features: AGE, PRO\_CENTER, PLUS\_CENTER.

Random Forest Model:

- Top 10 important features: OUTAGE\_TILLNOW, SUB5\_DAYS, OUTAGE\_PERC, NO\_OF\_PRODUCTS, TOT\_SUBS, MONTH1\_12\_PKG, PERSON\_TYPE, AGE, FT90\_P, NO\_PRGRAM.

- Least 3 important features: PLUS\_CENTER, REJOIN\_CNT, POPUP\_CENTER.

Both XGBoost and Random Forest have shown strong results in terms of accuracy and F1 score, but each model has unique strengths. XGBoost achieved an accuracy of 96.47% and an F1 score of 0.9235, while Random Forest has an accuracy of 96.02% and an F1 score of 0.9140. To leverage the strengths of both models, I will use a weighted ensemble approach, where each model's contribution is based on its performance. XGBoost will likely receive a slightly higher weight due to its better accuracy and F1 score, but both models will contribute to the final prediction.

The ensemble model size is 2GB, while the XGB only 2.4MG

Also, their results close -> so better to use the tuned XGB model!

Convert to Flask API, and test it using Postman  
results in following figure

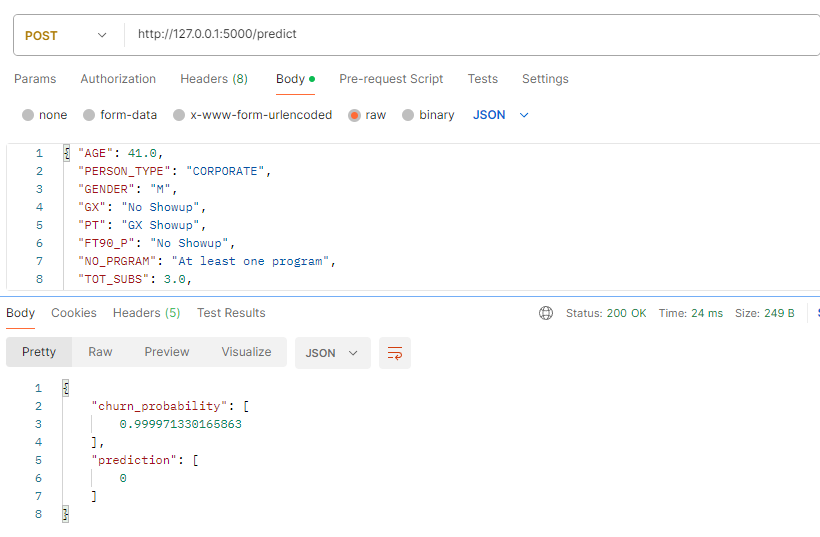


Figure 9 test API

The input and output in appendix 1

I made also a simple UI using streamlit library (How to run it and use it in the churn\_README.md file)

# Conclusion

For the given sample data, I followed the full machine learning (ML) lifecycle, which includes:

1. Understanding the Data: I began by thoroughly analyzing the dataset to gain a deep understanding of the features, distributions, relationships, and correlations. This step is critical to identifying patterns and insights that can guide the modeling process

2. Data Cleaning: Next, I performed data cleaning to handle missing values, remove outliers, and correct any inconsistencies in the dataset. This ensures that the data is of high quality and ready for the next steps.

3. Creating a Golden Layer: After cleaning the data, I created a "golden layer," which refers to a carefully curated version of the data that contains the most relevant and valuable features for the task at hand. This stage helps in reducing noise and focusing on the features that are most predictive.

4. Preprocessing: I then applied various preprocessing techniques, such as feature scaling, target encoding categorical variables, and splitting the data into training and test sets. Preprocessing is essential to ensure the model can effectively learn from the data.

5. Modeling: Using the preprocessed data, I built a machine learning model tailored to the prediction task the xgboost perform best. I experimented with different algorithms to find the one best suited for the problem, ensuring that it captures the underlying patterns in the data.

6. Model Tuning: To improve performance, I fine-tuned the model hyperparameters using techniques such as cross-validation and grid search. This step is key to optimizing the model and avoiding overfitting.

7. API Integration: After finalizing the model, I integrated it into an API to make predictions accessible to other applications. I implemented the API using Flask, which allows for easy interaction with the model.

8. UI Implementation with Streamlit: To provide a user-friendly interface, I built a simple web app using Streamlit. This app allows users to input data and receive predictions from the trained model.

9. Testing with Postman: Finally, I tested the API using Postman to ensure that it functions correctly, validating that the endpoints work as expected and that the predictions are accurate and timely.

# Future Work

1. Test to reduce features: Combine related features like programs, packages, and centers into singular features using PCA or custom code to improve model latency by simplifying the feature space.

2. Create region or city-specific models: Develop models for each region or city to potentially improve accuracy by accounting for localized behavior patterns.

3. Churn story model: Build a model that tracks customer behavior over time, predicting patterns such as "active for 20 days, then inactive for 60 days." This approach can provide deeper insights into the customer's lifecycle and help tailor targeted actions.

4. Incorporate new features: Add features like cost, purchase method, and branch number to study churn trends at a more granular level. These variables could reveal useful patterns to enhance the overall model's performance.

5. Leverage social listening: Implement sentiment analysis by monitoring social media mentions and reviews of your services and products. This can be used as an additional predictor of churn, as negative sentiment often correlates with customer dissatisfaction and potential churn.

6. Customer Segmentation with Clustering: Use clustering techniques (such as K-means or hierarchical clustering) to segment customers based on their behavior. Then apply churn models to each segment, allowing for more targeted predictions and interventions.

7. Time Series Modeling: Use time series models like ARIMA or LSTM to capture temporal patterns in customer behavior. This can help predict future churn based on past customer interactions over time.

8. Lifecycle Stage-Based Modeling: Different customers are at different stages of their lifecycle (e.g., new, active, at-risk). Create churn models specific to these stages for more refined predictions.

9. Behavioral Anomaly Detection: Incorporate anomaly detection algorithms to flag sudden changes in customer behavior, which could be early indicators of churn.

10. Survival Analysis: Use survival analysis techniques to estimate the time until a customer churns. This method allows you to focus on predicting the "time to churn" rather than just whether a customer will churn.

11. Cross-Sell/Upsell Features: Add features related to cross-sell and upsell opportunities, as customers who engage in additional purchases may have a lower likelihood of churning.

# References

[1] Paddle. (n.d.). *Customer churn: Understanding why customers leave & how to reduce churn*. Available at: <https://www.paddle.com/resources/customer-churn> (Accessed: 20 September 2024).

[2] Investopedia. (n.d.). *Churn rate*. Available at: <https://www.investopedia.com/terms/c/churnrate.asp> (Accessed: 20 September 2024).

[3] Qualtrics. (n.d.). *Customer churn: what it is and why it’s important*. Available at: <https://www.qualtrics.com/experience-management/customer/customer-churn/> (Accessed: 20 September 2024).

[4] Heap. (n.d.). *Master customer churn and massively improve retention*. Available at: <https://www.heap.io/topics/master-customer-churn-and-massively-improve-retention> (Accessed: 20 September 2024).

[5] Forbes. (n.d.). *Churn rate: What it is and how to reduce it*. Available at: <https://www.forbes.com/advisor/business/churn-rate/> (Accessed: 20 September 2024).

[6] Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G. and Chatzisavvas, K.Ch. (2015). *A comparison of machine learning techniques for customer churn prediction*. Thessaloniki, Greece: mSensis S.A., VEPE Technopolis.

[7] Lalwani, P., Mishra, M.K., Chadha, J.S. and Sethi, P. (2021). Customer churn prediction system: a machine learning approach. Springer-Verlag GmbH, part of Springer Nature. Received 19 June 2020, Accepted 12 January 2021.

[8] Alteryx. (n.d.). *Predict customer churn*. Available at: <https://github.com/alteryx/predict-customer-churn?tab=readme-ov-file> (Accessed: 20 September 2024).

[9] Prasad, B. (n.d.). *Customer churn prediction*. Available at: <https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction> (Accessed: 20 September 2024).

# Appendix 1

**Input JSON**

{ "AGE": 41.0,

  "PERSON\_TYPE": "CORPORATE",

  "GENDER": "M",

  "GX": "No Showup",

  "PT": "GX Showup",

  "FT90\_P": "No Showup",

  "NO\_PRGRAM": "At least one program",

  "TOT\_SUBS": 3.0,

  "REJOIN\_CNT": 0.0,

  "RENEWAL\_CNT": 2.0,

  "NEWSALE\_CNT": 1.0,

  "FT\_CENTER": 3.0,

  "PRO\_CENTER": 0.0,

  "PLUS\_CENTER": 0.0,

  "XPRESS\_CENTER": 0.0,

  "POPUP\_CENTER": 0.0,

  "HQ\_CENTER": 0.0,

  "NO\_OF\_PRODUCTS": 3.0,

  "MONTH\_12\_PKG": 0.0,

  "MONTH\_9\_PKG": 0.0,

  "MONTH\_6\_PKG": 2.0,

  "MONTH\_3\_PKG": 1.0,

  "MONTH\_1\_PKG": 0.0,

  "DAYS\_1\_PKG": 0.0,

  "OUTAGE": 4.0,

  "SUBS\_DAYS": 452.0,

  "OUTAGE\_PERC": 0.027,

  "OUTAGE\_TILLNOW": 70233.0,

  "CENTER\_REGION\_CHANGE": "False",

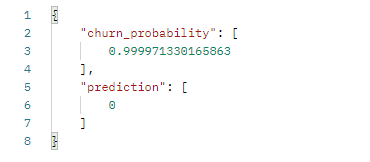
  "CITY\_CHANGE": "False",

  "CITY": "Riyadh",

  "CENTER\_REGION": "CR"

}

**Output**

****