

# **Requirement Analysis And Specification**

## **1.1 Introduction**

Churn rate prediction is a vital analytical process employed by businesses across various industries, especially those with subscription-based or recurring revenue models. Churn rate refers to the percentage of customers who stop using a product or service within a given period. Predicting churn allows companies to anticipate and mitigate customer attrition, thereby fostering customer retention and maximizing revenue.

## **1.2 System Features And Functional Requirements:**

For this project, we can break it down into three main areas: data collection and preparation, data analysis and modeling, and predictive analytics and reporting.

- 1.) Data Collection and Preparation: The first step is to collect customer data from various sources such as CRM systems, billing systems, support tickets, etc. Once the data has been collected, it needs to be cleaned, transformed, and prepared for analysis. This may involve removing duplicates, handling missing values, encoding categorical variables, scaling numerical variables, etc.
- 2.) Data Analysis and Modeling: After preparing the data, the next step is to analyze it using statistical methods and machine learning algorithms to identify patterns and relationships between different variables. These insights can then be used to build predictive models that estimate the likelihood of customers churning based on their behavior, demographics, and other factors.
- 3.) Predictive Analytics and Reporting: Finally, the system should provide easy-to-understand reports and visualizations that allow users to monitor churn rates over time, compare them across segments, and track the effectiveness of retention strategies. Alerts can also be set up to notify users when churn rates exceed certain thresholds.

### **1.3 External Interfaces:**

In this section, we describe how the system interacts with external entities such as users and data sources. There are two types of interfaces to consider here: user interface and data interfaces.

- 1.) User Interface: Users interact with the system through a web-based dashboard that displays key metrics, charts, and graphs. They can filter the data by date range, customer segment, product, etc., and customize the layout and appearance of the dashboard. Access to the dashboard should be secure and restricted to authorized personnel only.
- 2.) Data Interfaces: The system needs to connect to various data sources to extract customer data. These connections can be established via APIs, FTP sites, databases, spreadsheets, or other means depending on the specific requirements. Data transfers should be encrypted and authenticated to ensure security and privacy.

### **1.4 Non-functional requirements:**

Non-functional requirements specify performance, reliability, usability, scalability, and other quality attributes of the system. Some examples for this project could be:

- 1.) Performance Requirements: The system should respond to user requests within 2 seconds, handle concurrent access by at least 100 users, and process large datasets (e.g., millions of records) efficiently.
- 2.) Security Requirements: The system must comply with industry standards for data protection, encryption, and authentication. Access control mechanisms should enforce role-based permissions and auditing trails should record all user activities.