REPORT-4

CUSTOMER CHURN PREDICTION AT BANK

Submitted by:

Date:18-7-2023

Steve Rodrigues
S4 MCA

7. SYSTEM COST ESTIMATION

7.1 INTRODUCTION

A project can only come together with all the necessary materials and labor, and those materials and labors cost money. Putting together a budget that keeps costs to a minimum, while maximizing the project's quality and scope can be challenging. This is why proper cost estimation is important. Cost estimation in project management is the process of forecasting the financial and other resources needed to complete a project within a defined scope. Cost estimation accounts for each element required for the project—from materials to labor—and calculates a total amount that determines a project's budget. An initial cost estimate can determine whether an organization greenlights a project, and if the project moves forward, the estimate can be a factor in defining the project's scope. If the cost estimation comes in too high, an organization may decide to pare down the project to fit what they can afford (it is also required to begin securing funding for the project). Once the project is in motion, the cost estimate is used to manage all of its affiliated costs in order to keep the project on budget.

There are two key types of costs addressed by the cost estimation process:

1. Direct costs: Costs associated with a single area, such as a department or the project itself. Examples of direct costs include fixed labor, materials, and equipment.

2. Indirect costs: Costs incurred by the organization at large, such as utilities and quality control.

7.2 FUNCTION POINT BASED ESTIMATION

The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies.

Allan J. Albrecht initially developed function Point Analysis in 1979 at IBM and it has been further modified by the International Function Point Users Group (IFPUG). FPA is used make estimate of the software project, including its testing in terms of functionality or function size of the software product. However, functional point analysis may be used for the test estimation of the product. The functional size of the product is measured in terms of the function point, which is a standard of measurement to measure the software application. FPs of an application is found out by counting the number and types of functions used in the applications.

Function point analysis is a method used to estimate the size and complexity of software development projects based on the functionality provided by the system. It quantifies the functionality in terms of function points, which are derived from the different features and functionalities of the software system. While function point estimation is relevant for software development projects, it is not directly applicable to predicting the price of used cars.

In this project, the estimation involves predicting the price of used cars based on factors like vehicle age, fuel type, model, and brand. This is a regression problem, where you aim to develop a model that can accurately estimate the price of a used car based on its attributes. The estimation is based on historical data and the relationships between the input features and the corresponding prices.

8. SYSTEM TESTING

8.1 INTRODUCTION

System testing is an essential phase in the software development lifecycle that focuses on evaluating the complete system to ensure its functionality, performance, and reliability. It involves testing the integrated components and interactions within the system to verify that they work seamlessly together and meet the specified requirements. System testing encompasses various types of tests, including functional testing to validate system features, performance testing to assess response times and scalability, security testing to identify vulnerabilities, and usability testing to gauge the user experience. By conducting comprehensive system testing, organizations can detect and address defects, ensure the system meets user expectations, and enhance its overall quality before deployment, minimizing the risk of critical failures and optimizing user satisfaction. Effective system testing is vital for building robust and dependable software systems in today's demanding and competitive technological landscape.

8.2 INTEGRATION TESTING

Integration testing plays a crucial role in the project of customer churn prediction at a bank. It is a phase in software testing where individual software modules are combined and tested as a group. Integration testing occurs after unit testing and before validation testing. The purpose of integration testing is to verify the functional, performance, and reliability requirements placed on major design items.

In the context of customer churn prediction, integration testing ensures that the different modules and functionalities of the system work together harmoniously when integrated. This includes modules such as data preprocessing, feature engineering, machine learning models, customer segmentation, and churn prediction algorithms. Integration testing validates that data flows correctly between these modules and that the integration points, data transfer, and communication between different modules are functioning as expected.

By conducting integration testing, potential issues and inconsistencies between the various components can be identified and resolved early on in the development process. It helps ensure the smooth operation and functionality of the entire churn prediction system, enhancing its overall reliability and performance. Integration testing also helps validate that the system meets the objectives of accurately predicting customer churn and providing actionable insights to the bank for customer retention strategies.

Overall, integration testing is an essential step in the customer churn prediction project, ensuring the seamless integration and proper functioning of the different components to achieve accurate and reliable churn predictions for the bank.

8.3 SYSTEM TESTING

System testing is a critical phase in the project of customer churn prediction at a bank. It is a type of software testing that evaluates the compliance of the complete integrated system with the corresponding requirements. In system testing, the integrated modules that have passed integration testing are taken as input.

In the context of customer churn prediction, system testing ensures that the entire system, including modules such as data preprocessing, feature engineering, machine learning models, customer segmentation, and churn prediction algorithms, functions seamlessly and meets the desired requirements. The goal of system testing is to detect any defects or irregularities within the integrated units and the system as a whole.

System testing is carried out based on the system requirement specifications and functional requirement specifications, or both. It tests the design, behavior, and customer expectations of the system. By performing system testing, potential issues such as incorrect data flow, functionality gaps, or compatibility problems can be identified and resolved before the system is deployed.

Additionally, system testing evaluates the performance, security, and usability of the customer churn prediction system. It ensures that the system delivers accurate churn predictions, effectively identifies customers at risk of churn, provides a user-friendly interface for data analysis and reporting, and facilitates smooth customer retention strategies.

Overall, system testing plays a vital role in guaranteeing the reliability, functionality, and quality of the customer churn prediction system at the bank, helping to achieve accurate predictions, optimize customer retention efforts, and enhance the overall customer experience.

9. SYSTEM MAINTENANCE

9.1 INTRODUCTION

The maintenance is an important phase of any system. Maintenance of the System should be done accurately and with specific care for proper running system. Maintenance involves the software industry captive, typing up the system resources. It means restoring something to its original condition. Maintenance involves a wide range of activities including correcting, coding and design errors, updating documentation and test data and upgrading user support. Maintenance is continued till the product is re-engineered or deployed to another platform. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

9.2 MAINTENANCE

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. Maintenance is the ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. The system should be maintained and upgraded according to technological advancements. lt ensures data integrity, data control and security. The system must be protected from fire and other natural calamities. The backup copies of the data must be maintained daily so that to prevent the loss of data due to various reasons. Security measures provided to prevent unauthorized access of the database at various levels.
Software maintenance is divided into the following three categories: -

• corrective maintenance

• Adaptive maintenance

• perfect maintenance

Corrective maintenance has to do with the removal of residual errors present in the product when it is delivered as well as errors introduced into the software during its maintenance.

Adaptive maintenance modifies the software to keep it up to date with its operative environment. It may be needed because of changes in the user requirements, changes in target platforms, or changes in external interfaces.

Perfective maintenance involves changing the software to improve some of its qualities The request to perfective maintenance may come directly from the software engineer, in order to improve the status of the product on the market, or they may come.

10. SYSTEM SECURITY MEASURES

10.1 INTRODUCTION

The security of a computer system is a crucial task. It is a process of ensuring the confidentiality and integrity of the OS. Security is one of most important as well as the major task in order to keep all the threats or other malicious tasks or attacks or program away from the computer's software system. A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of various malicious threats and unauthorized access.

The security of a system can be threatened via two violations:
Threat: A program that has the potential to cause serious damage to the system.

Attack: An attempt to break security and make unauthorized use of an asset.

Security violations affecting the system can be categorized as malicious and accidental threats. Malicious threats, as the name suggests are a kind of harmful computer code or web script designed to create system vulnerabilities leading to back doors and security breaches. Accidental Threats, on the other hand, are comparatively easier to be protected against. Example: Denial of Service DDoS attack.

Security can be compromised via any of the breaches mentioned:
Breach of confidentiality: This type of violation involves the unauthorized reading of data.

Breach of integrity: This violation involves unauthorized modification of data.

Breach of availability: It involves unauthorized destruction of data.

Theft of service: It involves the unauthorized use of resources.

Denial of service: It involves preventing legitimate use of the system. As mentioned before, such attacks can be accidental in nature.

10.2 OPERATING SYSTEM-LEVEL SECURITY

Operating system security (OS security) is the process of ensuring OS integrity, confidentiality and availability. OS security refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised. OS security encompasses many different techniques and methods which ensure safety from threats and attacks. OS security allows different applications and programs to perform required tasks and stop unauthorized interference. OS security may be approached in many ways, including adherence to the following:

• Performing regular OS patch updates

• Installing updated antivirus engines and software

• Scrutinizing all incoming and outgoing network traffic through a firewall

• Creating secure accounts with required privileges only (i.e., user management)

10.3 DATABASE LEVEL SECURITY

Database security refers to the various measures organizations take to ensure their databases are protected from internal and external threats. Database security includes protecting the database itself, the data it contains, its database management system, and the various applications that access it. Organizations must secure databases from deliberate attacks such as cyber security threats, as well as the misuse of data and databases from those who can access them.

In the last several years, the number of data breaches has risen considerably. In addition to the considerable damage these threats pose to a company’s reputation and customer base, there are an increasing number of regulations and penalties for data breaches that organizations must deal with, such as those in the General Data Protection Regulation (GDPR)—some of which are extremely costly. Effective database security is key for remaining compliant, protecting organizations’ reputations, and keeping their customers.

Security concerns for internet-based attacks are some of the most persistent challenges to database security. Hackers devise new ways to infiltrate databases and steal data almost daily organizations must ensure their database security measures are strong enough to withstand these attacks.

Some of these cyber security threats can be difficult to detect, like phishing scams in which user credentials are compromised and used without permission. Malware and ransomware are also common cyber security threats.

Another critical challenge for database security is making sure employees, partners, and contractors with database access don't abuse their credentials. These exfiltration vulnerabilities are difficult to guard against because users with legitimate access can take data for their own purposes. Edward Snowden's compromise of the NSA is the best example of this challenge. Organizations must also make sure users with legitimate access to database systems and applications are only privy to the information they need for work. Otherwise, there's greater potential for them to compromise database security.

There are three layers of database security: the database level, the access level, and the perimeter level. Security at the database level occurs within the database itself, where the data live. Access layer security focuses on controlling who is allowed to access certain data or systems containing it. Database security at the perimeter level determines who can and cannot get into databases. Each level requires unique security solutions.

10.4 SYSTEM-LEVEL SECURITY

System-level security refers to the architecture, policy and processes that ensure data and system security on individual computer systems. It facilitates the security of standalone and/or network computer systems/servers from events and processes that can exploit or violate its security or stature. System-level security is part of a multi-layered security approach in which information security (IS) is implemented on an IT infrastructure's different components, layers or levels. System- level security is typically implemented on end-user computer and server nodes.

It ensures that system access is granted only to legitimate and trusted individuals and applications. The key objective behind system-level security is to keep system secure, regardless of security policies and processes at other levels. If other layers or levels are breached, the system must have the ability to protect itself. Methods used to implement system-level security are user/ID login credentials, antivirus and system-level firewall applications.

11. FUTURE ENHANCEMENT AND SCOPE OF THE FURTHER DEVELOPMENT

11.1 INTRODUCTION

Many applications (and wider database analytics systems) will go through what the industry likes to call the software application development lifecycle (SDLC) when they are first built. The SDLC describes the process of planning, analysis, design, build, test and implementation. But the lifecycle has further steps into maintenance, enhancement, and progression. Vendors are now appearing that are specifically describing themselves as application enhancement and maintenance specialists. User demands today change so rapidly, it makes sense for organizations to look for ways to extend functionality rather than go through rip-and-replace cycles, which are, ultimately, always costly.

The future enhancement and scope for further development of the customer churn prediction system at the bank hold immense potential for improving its effectiveness and delivering additional value. Some key areas for enhancement include incorporating enhanced feature engineering techniques to capture more comprehensive customer insights, exploring advanced machine learning algorithms for improved accuracy, developing real-time churn monitoring capabilities, implementing automated retention strategies, refining customer segmentation and targeting, integrating with CRM systems, establishing robust performance monitoring mechanisms, expanding to new channels and platforms, integrating with feedback and complaint systems, and fostering collaboration between data scientists and business units. These enhancements will enable the bank to proactively identify churn risks, personalize retention strategies, optimize customer engagement, and ultimately improve customer satisfaction, retention rates, and profitability. Regular updates and adaptations to changing customer behaviors and market dynamics are essential for sustained success.

11.2 MERITS OF THE SYSTEM

The system for predicting the price of used cars offers several merits that can benefit usersand stakeholders:

1. **Improved Customer Retention:** The customer churn prediction system enables the bank to identify customers who are likely to churn in advance. By implementing targeted retention strategies, such as personalized offers, proactive customer service, and tailored engagement campaigns, the bank can effectively reduce customer attrition and improve overall customer retention rates.

2. **Cost Savings:** By focusing on retaining existing customers, the bank can minimize the costs associated with acquiring new customers. Customer acquisition can be an expensive process, involving marketing campaigns, lead generation, and onboarding efforts. The churn prediction system helps allocate resources more efficiently by prioritizing retention efforts and reducing the need for costly acquisition activities.
3. **Enhanced Customer Experience:** The system enhances the overall customer experience by enabling the bank to understand customer preferences, needs, and behaviors. With this information, the bank can personalize interactions, offer relevant products and services, and provide proactive support, ultimately leading to higher customer satisfaction and loyalty.
4. **Time and Cost Savings:** The customer churn prediction system provides significant time and cost savings for the bank. By accurately identifying customers at risk of churn, the bank can focus its resources on targeted retention efforts, rather than investing time and money in acquiring new customers. This leads to more efficient resource allocation and cost-effective customer management.
5. **Improved Decision-Making:** The system enhances the bank's decision-making capabilities by providing valuable insights into customer behavior and churn patterns. With access to accurate churn predictions and related customer data, the bank can make informed decisions on retention strategies, product offerings, and customer engagement initiatives. This leads to more effective decision-making and better outcomes for the bank.
6. **User-Friendly Interface:** The customer churn prediction system features a user-friendly interface that is easy to navigate and understand. It allows bank personnel to access and analyze churn-related information effortlessly. This intuitive interface enables quick identification of at-risk customers and facilitates the implementation of targeted retention measures. The user-friendly nature of the system enhances user satisfaction and promotes efficient utilization of the system.

11.3 LIMITATIONS OF THE SYSTEM

1. **Data Availability and Quality**: The accuracy and effectiveness of the churn prediction system heavily rely on the availability and quality of data. Limited or inconsistent data, incomplete customer profiles, and data privacy constraints can pose challenges in accurately predicting churn and implementing effective retention strategies. Ensuring

data completeness, accuracy, and privacy compliance is essential for the system's success.

2. **Model Accuracy and Adaptability:** While the churn prediction model may demonstrate a high level of accuracy in predicting churn for existing customers, it may face challenges when applied to new or evolving customer segments. Changes in customer behavior, market dynamics, or external factors may require continuous monitoring and adjustment of the model to maintain its effectiveness.
3. **Retention Strategy Implementation:** The system's effectiveness also depends on the bank's ability to effectively implement retention strategies based on the churn predictions. Challenges may arise in coordinating and executing personalized retention efforts, ensuring timely and relevant interventions, and measuring the impact of these strategies on customer retention.

11.4 FUTURE ENHANCEMENT OF THE SYSTEM

potential future enhancement for the system is to implement a feedback loop that continuously collects user feedback and transaction data. By leveraging this feedback and data, the system can refine its predictive algorithms, improve accuracy over time, and provide personalized recommendations based on user preferences and historical patterns. This enhancement would enable the system to adapt and evolve with changing market dynamics and user needs, ultimately enhancing the user experience and the effectiveness of the churn prediction capabilities.