PERFORMANCE ANALYSIS OF CUSTOMER ATTRITION PREDICTION USING MACHINE LEARNING TECHNIQUES.

A PHASE 1 REPORT

Submitted by

AAMEER KHAN S [201501001]
HARINI MURUGAN[201501013]
NITHISSHKRISHNA KS [201501034]

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

RAJALAKSHMI ENGINEERING COLLEGE
ANNA UNIVERSITY : CHENNAI 600 025
NOVEMBER, 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“Performance Analysis of Customer Attrition Prediction using Machine Learning Techniques”** is the bonafide work of “**Aameer khan S [201501001], Akshara Sri L [201501005], Nithisshkrishna K S[201501034]”** who carried out the project work under my supervision.

SIGNATURE
Dr. N. SRINIVASAN
PROFESSOR AND
HEAD OF THE DEPARTMENT
B.Tech Artificial Intelligence and
Machine Learning,
Rajalakshmi Engineering College,
Thandalam, Chennai – 602 105.

SIGNATURE
Mrs. R. ANITHA
SUPERVISOR
B.Tech Artificial Intelligence and
Machine Learning,
Rajalakshmi Engineering College,
Thandalam, Chennai – 602 105.

Submitted for Project Viva-Voce Examination held on ______________________________.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the almighty for being with us through every walk of life. It is our privilege to express our sincerest thanks to our respected Chairman **Mr. S. Meganathan, B.E., F.I.E.,** and beloved Chairperson **Dr. (Mrs.) Thangam Meganathan, M.A., M.Phil., Ph.D.,** and beloved Vice- chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** for providing us with the requisite infrastructure and extending support in all endeavors.

Our heartfelt thanks to **Dr. S. N. Murugesan, M.E., Ph.D.,** our Principal for his kind support and resources provided to complete our work in time. We deeply express our sincere thanks to **Dr. N. Srinivasan, Ph.D.,** Head of the Department, Department of Artificial Intelligence and Machine Learning for his encouragement and continuous support to complete the project in time. We are glad to express our sincere thanks and regards to our supervisor **Mrs. R. Anitha,** Assistant Professor, Department of Artificial Intelligence and Machine Learning, and to our coordinator **Dr. P. Indira Priya, M.E., Ph.D.,** Professor, Department of Artificial Intelligence and Machine Learning for their guidance and suggestion throughout the course of the project.

Finally we express our thanks for all teaching, not teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

The research project titled "Performance Analysis of Customer Attrition Prediction using Machine Learning Techniques" explores the utilization of a web application built on a robust big data architecture, specifically leveraging Spark, to extract and analyze telecom data for insights into customer attrition. In response to the growing threat of customer attrition across industries, the study employs machine learning techniques, including Logistic Regression, K-means clustering and other machine learning algorithms, to assess their effectiveness in predicting customer churn. The project focuses on the hard and soft voting for classification algorithms, aiming to improve customer attrition prediction. The approach involves enhancing customer service, introducing loyalty programs, and adjusting pricing strategies based on the derived insights. The primary objective is to forecast the percentage of customers likely to leave using hard voting, clustering techniques, and regression algorithms such as K-means and logistic regression. By developing and implementing these advanced analytics techniques, the project aims to provide organizations with actionable insights to proactively address customer churn, ultimately contributing to improved customer retention strategies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DT	-	Decision Tree
ROC	-	Receiver Operating Characteristic
SVM	-	Support Vector Machine
BIRCH	-	Balanced Iterative Reducing and Clustering using Hierarchies
LR	-	Logistic Regression
ML	-	Machine Learning
AUC	-	Area Under the Curve
UI	-	User Interface
RF	-	Random Forest

CHAPTER 1

INTRODUCTION

1.1 PROJECT DEFINITION

The main objective of this project is to conduct a comprehensive analysis of the customers' needs while using the power of the big data platform. At a time when data is being generated at an unprecedented scale, organizations are facing the challenge of organizing and extracting insights from large amounts of data. Managing the big data platform is becoming important to manage, organize and derive useful conclusions from these big data in an efficient way. One aspect of this research is to combine "hard" and "soft" voting. Hard voting usually includes structured information, such as business records and demographic data, while soft voting includes less structured information such as customer surveys and social media impressions. By combining these two types of data, this project aims to create different things that contribute to the end. The combination of hardware and software data will facilitate deeper understanding.

To achieve this goal, this project will use an advanced machine learning method. Regression analysis, for example, will help to understand the relationship between different customer characteristics and the likelihood of fraud. K-means clustering will divide customers into different groups based on their behavior, enabling a targeted loyalty strategy. In addition, other classification algorithms will be used to analyze customer bias patterns. Combining these algorithms will reveal valuable insights and patterns within the data, allowing businesses to better understand why customers are frustrated. Big data technology and sophisticated analytics systems aim to empower organizations to make data-driven decisions. This project aims to provide companies with the insights needed to improve customer experience, reduce churn, and maintain competitive advantage in today's rapidly changing business environment. This system helps to increase the value of the customers, optimize the distribution of resources and make data-driven decisions, thus contributing to the success and longevity of the business.

1.2 NEED FOR PROPOSED SYSTEM

The proposed system for analyzing customer characteristics through logistic regression and K-means clustering addresses the critical needs of today's business environment. Most of all, it allows companies to gain a deeper understanding of customer behavior and trends. With the complexity of customer data growing, organizations need more advanced methods to understand the many reasons for customer churn. By optimizing logistic regression for individual-level prediction and K-means clustering for customer segmentation, the proposed method provides a customer perspective. This concept has many important aspects for identifying and understanding the factors that affect the conflict, allowing organizations to make better and more effective decisions.

Second, the proposed system addresses the need for preventive measures. Using predictive analytics, the system provides a way to predict potential events and engage in tailored containment strategies. Combining this predictive power with aggregated K-means allows the development of segment-specific strategies, recognizing that different groups of customers may have reasons for opting out. This is important in a competitive market where customer loyalty and satisfaction are paramount.

Finally, the proposed system is compatible with the increasing demand for data-based decision making and efficient resource allocation. As organizations collect more data than ever, they need tools and processes to gain actionable insights and effectively deploy their resources. Logistic regression and cumulative K-means provide a powerful analytical method for transforming raw data into meaningful insights.

By tracking trends and trends and segmenting customers based on their behavior, businesses can better focus efforts and allocate resources where they will have the greatest impact. This data-driven process improves operational efficiency and ultimately improves customer satisfaction and profitability, making it a vital necessity for business today.

1.3 APPLICATION OF PROPOSED SYSTEM

Performance Analysis of Customer Attrition using Machine Learning Techniques can potentially improve customer service, offering loyalty programs, or adjusting pricing strategies in various industries. Some of the examples are:

1. **Healthcare:** Our project can be used in Healthcare in which providers can segment patients based on their engagement with healthcare services by identifying reasons for patient attrition, such as long wait times, appointment scheduling issues, or communication problems. Improve patient experience and appointment management it also predicts patient churn and develop patient-centric strategies to enhance healthcare service quality and patient experience.

2. **Personalize customer retention campaigns:** Once businesses have identified the customers who are at risk of churning and the factors that contribute to churn, they can use this information to personalize their customer retention campaigns. This can be done by targeting customers with specific offers, discounts, or educational resources that are relevant to their individual needs.

For example, a subscription box company could use customer churn attribution to identify customers who are at risk of churning due to low product engagement. The company could then target these customers with a personalized email campaign that

highlights the benefits of their subscription and offers them exclusive discounts on products that they are likely to be interested in.

3. **Telecommunications:** A telecommunications company could use customer churn attribution to identify customers who are at risk of churning due to factors such as high service costs, poor customer service, or lack of new product offerings. The company could then target these customers with special offers or discounts, or with personalized customer service outreach.

4. **Retail:** Retailers can segment customers based on purchase history and behavior. Understand why customers stop shopping, e.g., product availability, price changes, or competition. Offer personalized discounts or product recommendations to retain customer

5. Subscription Services

The system can be used to Companies offering subscription services to segment customers based on usage and subscription plans. Determine why subscribers cancel, e.g., pricing, content quality, or customer support. Adjust pricing strategies or content offerings to reduce churn.

These are just a few examples. By combining the logistic regression and K-means clustering it helps organizations gain insights into why customers or users are churning, allowing them to take proactive measures to improve customer retention and satisfaction. It enables data-driven decision-making and more targeted customer retention strategies .

1.4 FUNCTIONAL DESCRIPTION

The functional description of a customer churn attribution system that integrates logistic regression and K-means clustering, the initial stages are data collection and preprocessing. This involves the compilation of diverse customer data, spanning demographics, transaction histories, and behavioral patterns. A crucial aspect of this phase is feature engineering, where relevant features are used to enhance the system's understanding of customer dynamics. Subsequently, the data undergoes a sophisticated analysis and segmentation using clustering algorithms, including K-means clustering, BIRCH, Agglomerative clustering, and Affinity propagation. The effectiveness of these algorithms through metrics such as inertia, Silhouette score, Davies-Bouldin Index, and Calinski-Harabasz Index, ensuring the optimal grouping of customers based on shared attributes and behaviors. Following the segmentation, logistic regression models are tailored for each customer segment. These models predict the likelihood of churn by leveraging insights derived from the segmented behaviors. The analysis of model coefficients is attributing churn to specific factors, why customers may be inclined to leave. Businesses can craft targeted retention strategies, ranging from personalized offers to elevated customer support experiences, all designed to effectively mitigate churn. The system's ongoing monitoring capabilities further enable businesses to stay in customer dynamics and adjust strategies accordingly. The true power of this system lies in its capacity to empower businesses to take decisive, data-driven actions. It facilitates the optimization of resource allocation, directing efforts towards areas with the highest potential impact on customer retention. As a result, customer satisfaction is bolstered, leading to increased retention rates and enhanced profitability. Moreover, the system provides robust reporting and visualization features, translating complex analytical results into accessible insights for stakeholders within the organization. Through clear and concise reporting, businesses can facilitate informed decision-making and collaboration among various departments. In the training and validation phase, the dataset is split to ensure robust model performance. Evaluation metrics are meticulously applied to gauge the efficacy of the models, providing a quantitative measure of their predictive capabilities. In essence, this comprehensive customer churn attribution system serves as a strategy for businesses, guiding them towards a proactive, customer approach that ultimately stands their position in the market.

CHAPTER 2

LITERATURE REVIEW

2.1 Random Forest

Xiancheng Xiahou et.al, [1], Describes methodology employed in this research involving the utilization of the random forest algorithm. B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM methodology employed in this research involves the utilization of the random forest algorithm. Random Forest algorithm was employed as a robust and efficient feature selection method. Random Forest is renowned for its high classification accuracy, resilience to noise and outliers, and its ability to generalize well across various domains including business management, economics, finance, and biological sciences.Given the dataset's considerable dimensionality of 17 variables, the challenge was to determine the optimal number of features (M) to include in the predictive model. To address this, the Out-of-Bag (OOB) error was utilized as a metric for feature selection. During the construction of each tree within the Random Forest, distinct bootstrap samples were employed for the training set, allowing for the calculation of the OOB error. Surprisingly, as the number of randomly selected features changed. It was seen that the distinctions in the OOB mistake rates were minimal.This suggested that the choice of the feature count (M) did not significantly impact the model's performance. Consequently, the decision was made to select four features in each iteration, resulting in a relatively low OOB error.This suggested that the choice of the feature count (M) did not significantly impact the model's performance. Consequently, the decision was made to select four features in each iteration, resulting in a relatively low OOB error. Four variables were identified as crucial for predicting customer churn: "Night Buy," "PM Buy," "Night PV," and "PM PV." These variables were considered as the key indicators for predicting customer loss in the churn prediction model.

2.2 Adaboost and XGboost

Praveen Lalwani et.al,[19], Describes the data pre-processing and feature analysis is performed. In the third phase, feature selection is taken into consideration using the gravitational search algorithm. Next, the data has been split into two parts: train and test set in the ratio of 80% and 20% respectively. In the prediction process, most popular predictive models have been applied, namely, logistic regression, naive bayes, support vector machine, random forest, decision trees, etc. on train set as well as boosting and ensemble techniques are applied to see the effect on accuracy of models. In addition, K-fold cross validation has been used over train sets for hyperparameter tuning and to prevent overfitting of models. Finally, the obtained results on the test set have been evaluated using a confusion matrix and AUC curve. It was found that Adaboost and XGboost Classifiers give the highest accuracy of 81.71% and 80.8% respectively. The highest AUC score of 84%, is achieved by both Adaboost and XGBoost Classifiers which outperforms over others.

Nagaraju Jajam et.al,[6], describes a model that plays a significant role in the churn classification process, aiming to accurately determine the likelihood of customer churn from the given dataset. To achieve this, a deep learning framework is utilized, incorporating an attention layer that enhances the understanding of churn classification accuracy In addition, assuming the semantic meaning of input data involves understanding the underlying information and context within the data. To train such a model, a crucial step is the generation of labels that signify whether a customer has churned or not. However, it's acknowledged that these assignments of churn labels can be somewhat subjective, as the determination of churn often depends on various factors and interpretations.

2.3 Ensemble Learning

B. Prabadevi et.al, [8], Describes this paper, the proposed Sampling-based Stack Framework, known as SS-IL, offers a novel approach to churn prediction. This framework leverages ensemble learning to enhance the performance of classifiers. Ensemble learning is a powerful technique that combines the outputs of multiple base classifiers to make a final classification decision. Stacking, a specific form of ensemble learning, employs several base learners, often referred to as level-0 learners, who are trained using the same training dataset. What sets the SS-IL framework apart is its utilization of varied training datasets for the level-0 classifiers. The objective here is to broaden the range of attributes considered and facilitate the accumulation of valuable information within the ensemble through the use of sampling techniques. This diversification in training data is aimed at improving the overall predictive capabilities of the framework. This meta-learner learns the combination weights for all the decision probabilities provided by the base-level classifiers, thereby enabling it to classify instances effectively. The success of a stacked ensemble like SS-IL hinges on the promotion of information gain from the features employed in training the meta-learner via the level-0 base learners. In essence, this framework is rooted in the rationale that the diversity in training data and the combined wisdom of multiple classifiers enhance predictive accuracy and robustness. It's worth noting that while this content discusses the SS-IL framework in the context of churn prediction, it also hints at potential applications in the medical field, specifically for pneumothorax diagnosis and monitoring in clinical settings. By saving time and potentially improving patient care, the framework showcases its versatility and utility across different domains, emphasizing its significance beyond just predictive analytics.

2.4 Support Vector Machine

Saran Kumar A et.al, [14], Describes the abundance of available data often necessitates a process of classification, grouping this data into various categories or types, such as sound, video, and text designs. This characterization is fundamental for viable information mining, which envelops a scope of functionalities like grouping, segregation, affiliation, and bunching, and that's only the tip of the iceberg. Numerous complete frameworks are intended to give a set-up of information mining functionalities inside a solitary stage (Neha and Vikram, 2015). One notable classification technique is the Support Vector Machine (SVM), which excels in handling linear permutations of subsets within a training dataset. SVM aims to find a maximum margin separating hyperplanes in a high-dimensional feature space, particularly useful when dealing with nonlinearly separable information highlights (Nadeem, Umar, and Shahzad, 2018). This method effectively organizes data based on the most significant characteristics, even in scenarios where the vectors are nonlinearly separable. In the SVM system, a few key parts assume essential parts: M: Represents the number of samples in the training dataset. Xi: Denotes vector support when the value of ai is greater than 0. 'X': Represents an unidentified vector sample. δ (delta): Serves as a threshold or margin. (ai): Parameter derived from solving a convex quadratic programming problem related to linear constraints. In practice, various kernel functions are employed, such as the Polynomial kernel and Gaussian radial basis functions (RBF), to transform data into higher-dimensional spaces, allowing for more effective separation of classes. The threshold (δ) is another parameter determined by selecting any 'i' where ai is greater than 0, and it satisfies the Karush–Kuhn–Tucker condition (Burges, 1998). In summary, SVM is a powerful classification technique that maximizes the margin between data points in high-dimensional space. It is particularly useful when dealing with complex and nonlinearly separable data, making it a valuable tool in data mining and classification tasks.

2.5 GWO-KELM Algorithm.

Deepthi Das et.al,[7], "An Effective Machine Learning Model For Customer Attrition Prediction In Motor Insurance Using GWO-KELM Algorithm", is developed this study is to predict the behaviors of the customers and to classify the churners and non-churners at an earlier stage. The Motor Insurance sector dataset consists of 20,000 records with 37 attributes collected from the machine learning industry. The missing values of the records are analyzed and explored via Expectation Maximization algorithm that categorizes the collected data based on the policy renewals. Then, the behavior of the customers are also investigated, so as to ease the construction process training classifiers. With the help of Naive Bayes algorithm, the behaviors of the customers on the upgraded policies are examined. Depending on the dependency rate of each variable, a hybrid GWO-KELM algorithm is introduced to classify the churners and non-churners by exploring the optimal feature analysis. Experimental results have proved the efficiency of the hybrid algorithm in terms of 95% prediction accuracy; 97% precision; 91% recall & 94% F-score.

2.6 Random Forest

O. F. Seymen et al,[1], where the paper is based on churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector proposed a churn prediction model that uses classification, as well as, clustering techniques to identify the churn customers and provides the factors behind the churning of customers in the telecom sector. Feature selection is performed by using information gain and correlation attribute ranking filter. The proposed model first classifies churn customers data using classification algorithms, in which the Random Forest (RF) algorithm performed well with 88.63% correctly classified instances. Creating effective retention policies is an essential task of the CRM to prevent churners. After classification, the proposed model segments the churning customer's data by categorizing the churn customers in groups using cosine similarity to provide group-based retention offers. This paper also identified churn factors that are essential in determining the root causes of churn. By knowing the significant churn factors from customers' data, CRM can improve productivity, recommend relevant promotions to the group of likely churn customers based on similar behavior patterns, and excessively improve marketing campaigns of the company. The proposed churn prediction model is evaluated using metrics, such as accuracy, precision, recall, f-measure, and receiving operating characteristics (ROC) area. The results reveal that our proposed churn prediction model produced better churn classification using the RF algorithm and customer profiling using k-means clustering. Furthermore, it also provides factors behind the churning of churn customers through the rules generated by using the attribute-selected classifier algorithm.

2.7 Ensemble Learning.

Krist et.al,[6], describes this study in which investigates whether churn prediction is a valuable option in the CRM palette of the online gambling companies. Using real-life data of poker players at bwin, single algorithms, CART decision trees and generalized additive models are benchmarked to their ensemble counterparts, random forests and GAMes. The results show that churn prediction is a valuable strategy to identify and profile those customers at risk. Furthermore, the performance of the ensembles is more robust and better than the single models. An alternative in customer relationship management (CRM) to analyze customer retention. Therefore it is essential to efficiently retain gamblers. Therefore it is essential to efficiently retain gamblers.

2.8 Machine Learning Techniques

Granberg et.al,[10], Defines this thesis based on the assumption that early signs of churn can be detected by the temporal changes in customer behavior where this paper is published in 2020. Recurrent neural networks and more specifically long short-term memory (LSTM) and gated recurrent unit (GRU) are suitable contenders since they are designed to take the sequential time aspect of the data into account. Random forest (RF) and stochastic vector machine (SVM) are machine learning models that are frequently used in related research. The problem is solved through a classification approach, and a comparison is done with implementations using LSTM, GRU, RF, and SVM. According to the results, LSTM and GRU perform similarly while being slightly better than RF and SVM in the task of predicting customers that will churn in the coming six months, and that all models could potentially lead to cost savings according to simulations (using non-official but reasonable costs assigned to each prediction outcome). Predicting the time until churn is a more difficult problem and none of the models can give reliable estimates, but all models are significantly better than random predictions.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM:

The existing system for analyzing customer churn attribution is a multifaceted process that involves several key components and data-driven approaches. It typically begins with data collection, where historical customer data, encompassing demographics, transaction records, and customer interactions, is gathered from various sources. This collected data is then subjected to thorough preprocessing, which includes data cleaning, normalization, and feature engineering to create relevant attributes for analysis. Churn, often defined based on specific criteria like the absence of customer activity over a defined period, is central to the analysis. Machine learning models, such as logistic regression, are developed and trained on the preprocessed data to predict and understand churn patterns.

The second phase of the existing system involves churn attribution, where the models are employed to identify and attribute churn to specific factors or variables. This step helps organizations gain insights into the primary drivers of customer attrition, leading to the formulation of actionable retention strategies. Continuous monitoring and iteration play a pivotal role in the system, allowing organizations to adapt and refine their models and strategies as new data becomes available. Overall this project aims to provide a solution for business organizations to overcome the challenges of loyalty problems or adjusting pricing strategies and it is often more expensive to acquire new customers than it is to retain existing customers. By reducing churn, businesses can save money on customer acquisition costs.

3.2 PROPOSED SYSTEM:

The proposed system represents a web application designed to address the challenges of handling extensive industry data effortlessly and managing vast volumes of industrial data. With the support of Apache Spark, this application ensures the smooth and it streamlines the database maintenance process, enabling organizations to handle large datasets with ease. The project embarks on a journey of comprehensive performance analysis. Beyond its database management function, this project also aspires to conduct in-depth performance analysis on the data. Its primary objective is to identify potential customer churn by employing a variety of clustering algorithms, including Logistic Regression, Decision Tree, K-means Clustering, and Random Forest. These algorithms, known for their predictive power, will play a pivotal role in distinguishing customers at risk of churn within the industry.

Moreover, the project seeks to quantify the extent of customer churn by determining the percentage of customers likely to churn. This percentage serves as a vital metric for businesses to assess the impact of attrition on their operations and revenues. To ensure the utmost accuracy and reliability of these predictive models, the system will undertake a rigorous evaluation process. This evaluation involves the fusion of both hard voting and soft voting derived from Logistic Regression, Decision Tree, K-means Clustering, and Random Forest. By combining the strengths of these algorithms, the system will not only enhance the precision of churn prediction but also provide valuable insights that can guide strategic decision-making.

In conclusion, the proposed web application, powered by Apache Spark, addresses the pressing need for seamless management of large-scale industrial data while simultaneously offering a sophisticated solution for customer churn prediction and analysis. Through a combination of advanced clustering algorithms and comprehensive evaluation techniques, this system aims to provide organizations with the tools they need to proactively manage customer attrition and optimize their overall business performance.

3.3 FEASIBILITY STUDY:

The project involves Evaluating the viability and advantages of developing a one-time prediction model to predict the churn of a particular customer is the task of a feasibility study for a single customer churn prediction. It necessitates analyzing the quality and accessibility of the data, thinking through privacy and legal ramifications, projecting expenses and possible profits, determining how well it aligns with organizational objectives, and carrying out a proof of concept. In order to inform the decision on whether to proceed with the prediction model implementation, this feasibility study aims to ascertain whether the resources and expertise required to make a single prediction for this customer's churn are justified by the expected benefits, such as customer retention and improved relationships.

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

Hardware requirements for customer churn prediction products include a multi-core CPU with sufficient processing power, ample RAM (8-16GB minimum), SSD storage for data and models, GPUs for deep learning (optional), high-speed network bandwidth, and redundancy mechanisms. Cloud-based platforms like AWS, Azure, or GCP offer scalable and flexible resources. Additionally, monitoring and maintenance systems are essential, ensuring compliance with industry standards and data security protocols.

4.2 SOFTWARE REQUIREMENTS

Software requirements for a customer churn prediction product involve a diverse set of tools and technologies. You'll need programming languages such as Python for data analysis and machine learning. Data preprocessing and ETL tasks can be handled with Pandas, Apache Spark, and SQL databases. Machine learning frameworks like Scikit-Learn, TensorFlow, or PyTorch are essential for model development. Feature engineering libraries, web frameworks like Flask or FastAPI for serving predictions, and containerization tools like Docker are crucial for deployment. Monitoring, logging, and database management solutions ensure system reliability. Version control with Git and cloud services for scalability round out the requirements. Security, compliance, testing, and documentation tools are also necessary for a robust churn prediction system.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE DIAGRAM:

The architecture diagram for the "Performance Analysis of Customer attrition prediction using Machine Learning Techniques" project is a comprehensive visual representation that outlines the various components, algorithms, and data flows involved in the system. This diagram serves as a blueprint for understanding how the project operates and how different elements interact to achieve the project's objectives.

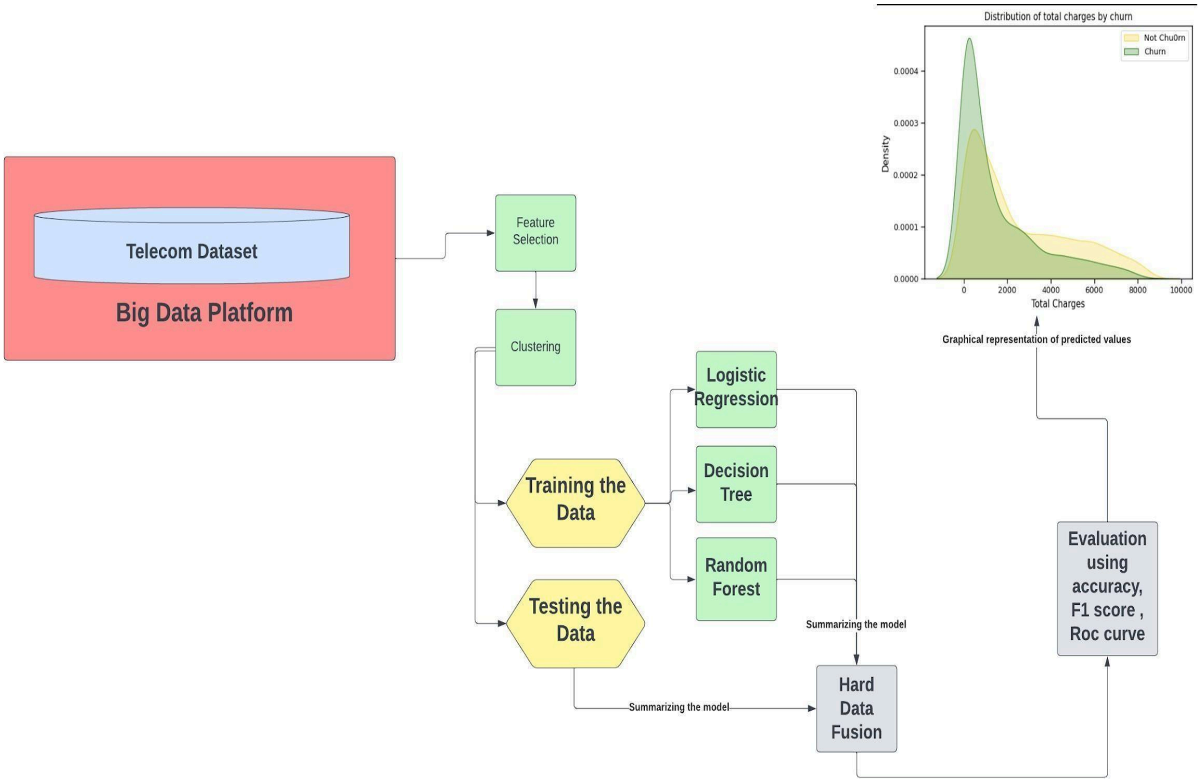


Figure 5.1 System Architecture diagram

The above image shows a system architecture diagram for a big data platform for telecom dataset analysis. The system is designed to handle large volumes of telecom data, such as call detail records (CDRs), network traffic data, and customer demographic data. The system architecture is divided into two main components: data processing and visualization and reporting. The data processing component is responsible for preprocessing the telecom data, performing feature selection, and training and testing machine learning models. The preprocessing step cleans and prepares the data for analysis. The feature selection step identifies the most important features in the dataset for predicting the target variable. This step helps to reduce the dimensionality of the data and improve the performance of the machine learning models. The model training step trains machine learning models on the preprocessed data. The models are trained to predict the target variable, such as customer churn or fraud. The model testing step evaluates the performance of the trained machine learning models on a held-out test set. This step helps to identify the best model for deployment. The visualization and reporting component is responsible for visualizing and reporting the results of the machine learning analysis. The data visualization step visualizes the preprocessed data and the results of the machine learning analysis. This step helps to identify patterns and trends in the data. The reporting step generates reports that summarize the results of the data analysis. These reports can be used to make informed business decisions.

5.2 DATA FLOW DIAGRAM

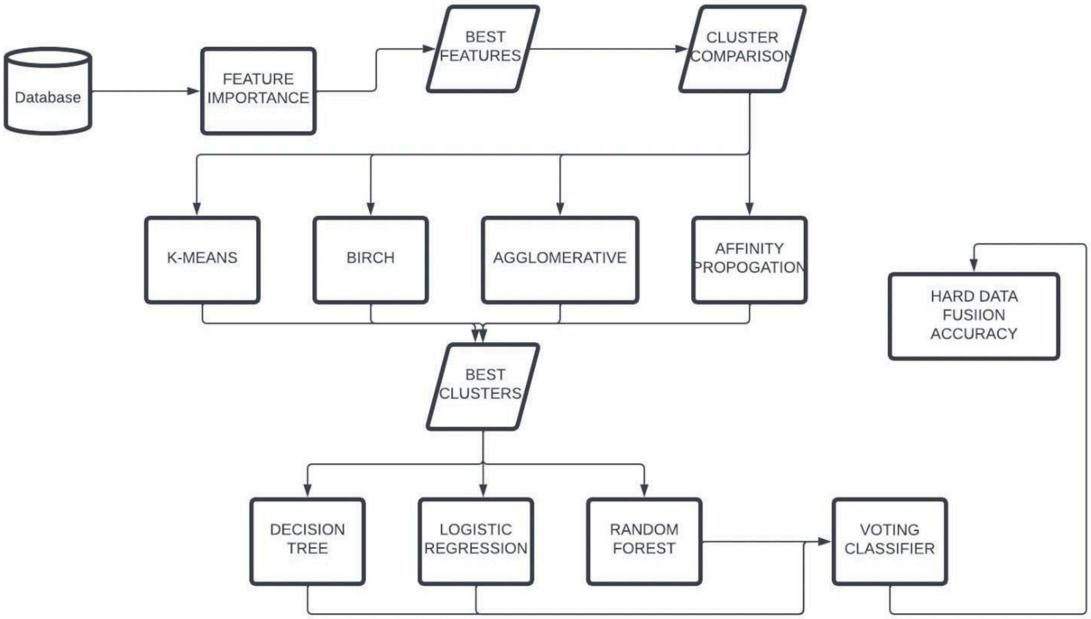


Figure 5.2 Data flow diagram

The image shows a block diagram of a machine learning system for data clustering. The system starts with a database of data, which is then preprocessed to remove noise and normalize the features. The preprocessed data is then fed into a feature importance algorithm, which identifies the most important features for clustering. The next step is to compare different clustering algorithms, such as K-means, BIRCH, agglomerative clustering, and affinity propagation. The best clustering algorithm is then selected and used to cluster the data. The clustered data is then evaluated using a hard voting accuracy metric. The best clusters are then identified and used to train a decision tree, logistic regression, random forest, or voting classifier.

5.3 MODULE DESCRIPTION:

5.3.1 MODULE 1

DATA COLLECTION AND FEATURE IMPORTANCE:

The goal is to collect and preprocess data on customer churn in order to train logistic regression and K-means clustering models to predict customer churn. Where Extracting features is important. We are going to collect data from spark where we will be pulling the data from spark and the second way to collect from spark is by Extract, transform, and load (ETL) the data. Once the data has been collected, it needs to be preprocessed which involves removing duplicate records, handling missing values, converting categorical variables to numerical variables from which the best eight columns are selected for the further process. We can find feature importance using the Random Forest algorithm, typically by examining the feature_importances_ attribute of a trained Random Forest model. Feature importance will provide the importance scores for each feature in your dataset.Sort the feature importance values in descending order. The features with the highest importance scores are the most important features and we are taking K most important features and then selected features undergo a clustering process.

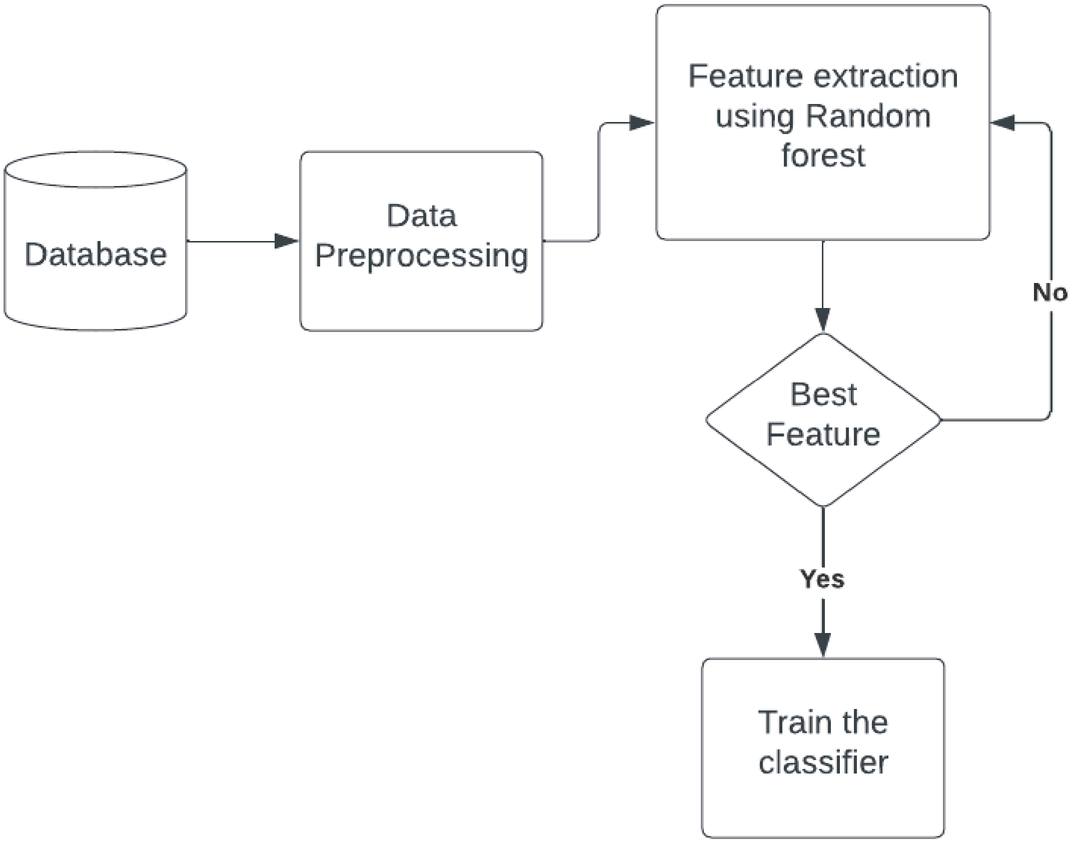


Figure 5.3.1.1 Data flow diagram Module 1

5.3.2 MODULE 2 CLUSTER COMPARISON:

The features with the highest importance are selected and passed to the clustering algorithms to determine the best clustering technique . Clustering algorithms such as K-means clustering, BIRCH, Agglomerative clustering and Affinity propagation, The metrics of these algorithms are evaluated based on the inertia, Silhouette score, Davies-Bouldin Index, Calinski-Harabasz Index. The best clustering technique is used in the regression algorithms for further analysis.

MODEL TRAINING AND OPTIMIZATION :

Model training and optimization for customer churn prediction is a crucial and iterative process. It begins with data collection and preprocessing, where historical customer data is cleaned and prepared. The dataset is then divided into training, validation, and test sets. Feature engineering is undertaken to create meaningful predictors. Model selection follows, with various machine learning algorithms considered. Hyperparameter tuning fine-tunes the model for optimal performance. The chosen model is then trained on the training data and evaluated using validation metrics. If the initial model performance is unsatisfactory, iteration through algorithm selection, feature engineering, and hyperparameter optimization is necessary. Optional ensemble methods can enhance performance. Regularization and cross-validation ensure model robustness.

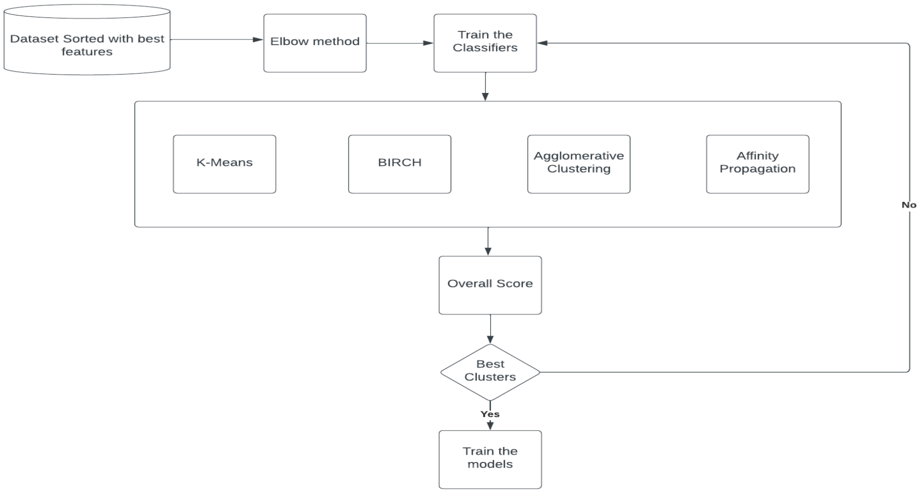


Figure 5.3.2.1 Data flow diagram Module 1

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION:

In conclusion, the purpose of this kind of study in the telecom industry is to assist businesses in increasing their profits. It is well known that one of the most significant revenue streams for telecom firms is churn prediction. This paper reviewed techniques to build an application using big data technology to predict the percentage of customer attrition. The integration of big data technology and machine learning for predicting customer attrition holds great promise for the telecommunications industry. We recognized the importance of preprocessing and preparing the data for analysis. By identifying relevant attributes and customer behavior, we aimed to capture the essential factors influencing customer churn. Furthermore, using techniques like logistics regression and K-means clustering enables extracting meaningful attributes from high-volume telecom data and delivering relevant information to machine learning algorithms to predict customer churn and take action to prevent it. Overall, the techniques reviewed have the potential to create a well-balanced machine learning model to help the industry to prevent loss from customer churning.These techniques have the potential to make a significant positive impact on the industry's bottom line and customer satisfaction levels. By accurately identifying customers at risk of churning and tailoring retention efforts to individual needs, telecom companies can enhance customer satisfaction, reduce churn-related revenue loss, and ultimately strengthen their competitive position in the market. We hope that this research contributes to a deeper understanding of customer attrition prediction and inspires further exploration and innovation in the field of telecommunications and big data analytics.

7.2 FUTURE ENHANCEMENT:

The “Performance Analysis of Customer Attrition Prediction using Machine Learning Techniques” has further expanded its capabilities and impact. Here are some potential areas for improvement and development:

However, there are several areas for future Enhancement that could be made to improve the proposed system. As technology advances, there will be new opportunities to improve and extend the capabilities of using real-time data to improve model performance. Customer churn prediction models can be updated in real time as new customer data becomes available. This can help to improve the accuracy of the model and identify at-risk customers sooner by using time series analysis. It can be used to identify patterns in customer behavior over time. This information can then be used to improve the accuracy of customer churn predictions.

Secondly by using more features. The more features that are used to train the logistic regression model, the more accurate the predictions will be. This could include features such as customer demographics, purchase history, customer support interactions, and social media activity. Successful implementation of this project has the potential to improve the quality of life and help businesses to identify new opportunities to grow their business. For example, if a business finds that a large number of customers are churning due to a lack of certain features, the business can invest in developing those features to attract and retain new customers. Also by reducing customer churn and improving customer retention, businesses can increase their revenue.

APPENDIX

A.1.1.1 SAMPLE CODE

FEATURE SELECTION:

```
import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler, OneHotEncoder from
sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline from tabulate import tabulate
# Load your dataset

data = pd.read_csv("/Users/aameerkhan/Desktop/testdata1.csv") data =
data.drop(['customerID'], axis = 1)
# Convert 'TotalCharges' to numeric, handling errors='coerce' to convert invalid
parsing to NaN
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')

# Convert columns from object to numeric, handling errors='coerce' to convert
invalid parsing to NaN
columns_to_convert = ['StreamingMovies', 'DeviceProtection', 'PaperlessBilling',
'OnlineBackup', 'Contract', 'Partner']
for column in columns_to_convert:

data[column] = data[column].astype('category').cat.codes

# Convert 'tenure' column to int

data['tenure'] = data['tenure'].astype(int)

# Separate features (X) and target variable (y)

X = data.drop(columns=['Churn']) y = data['Churn']
```

```
y = data['Churn']
# Identify numerical and categorical columns
numerical_cols = X.select_dtypes(include=['float64', 'int64']).columns
categorical_cols = X.select_dtypes(include=['object']).columns
# Define preprocessing for numerical and categorical data
# (You can customize these transformers based on your dataset)
numerical_transformer = StandardScaler()
# Standardize numerical features
categorical_transformer = OneHotEncoder(handle_unknown='ignore')
# One-hot encode categorical features
# Create a RandomForestClassifier model
model = RandomForestClassifier(random_state=42)
# Create a pipeline that includes preprocessing and the classifier
pipeline = Pipeline(steps=[('preprocessor', preprocessor),('classifier', model)])
# Train the model
pipeline.fit(X, y)
feature_importances = model.feature_importances_
# Create a dictionary to store feature names without prefixes and their importances
grouped_feature_importances = {}
# Assuming categorical_transformer is your OneHotEncoder after fitting # Assuming numerical_cols and categorical_cols are pandas Index objects
numerical_cols_list = list(numerical_cols)
```

categorical_cols_list = list(categorical_cols)

# Concatenate numerical and categorical column names feature_names = numerical_cols_list + categorical_cols_list grouped_feature_importances = {}

# Extract feature names without prefixes and group them by original column names

for feature_name, importance in zip(feature_names, feature_importances):

column_name = feature_name.split('_')[0] # Get the original column name if
column_name not in grouped_feature_importances:
grouped_feature_importances[column_name] = importance
else:

# If the original column name already exists, add the importance score to the existing value

grouped_feature_importances[column_name] += importance from tabulate import tabulate

# Prepare data for tabulate

table_data = [[key, f"{value:.4f}"] for key, value in grouped_feature_importances.items()]

# Print the tabulated output

table_headers = ["Feature", "Importance"]

table = tabulate(table_data, headers=table_headers, tablefmt='grid')

CLUSTER COMPARISON:

import streamlit as st import pandas as pd
from sklearn.cluster import KMeans, Birch, AgglomerativeClustering,

AffinityPropagation

```
from sklearn.metrics import silhouette_score, davies_bouldin_score from
sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression from sklearn.metrics import
accuracy_score
from sklearn.ensemble import RandomForestClassifier, VotingClassifier from
sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor from
sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error import numpy as np
st.title("Clustering and ML Model Evaluation App")
agglomerative_inertia += cluster_inertia
agglomerative_silhouette = silhouette_score(X, agglomerative_labels)
agglomerative_db_index = davies_bouldin_score(X, agglomerative_labels)
agglomerative_calinski_harabasz = calinski_harabasz_score(X) agglomerative_score
= agglomerative_inertia + agglomerative_silhouette + (1 /
agglomerative_db_index) + agglomerative_calinski_harabasz # Perform Affinity
Propagation Clustering
affinity_propagation = AffinityPropagation() affinity_propagation_labels =
affinity_propagation.fit_predict(X)
affinity_propagation_inertia = len(affinity_propagation.cluster_centers_indices_)
affinity_propagation_silhouette = silhouette_score(X, affinity_propagation_labels)

affinity_propagation.fit_predict(X)
affinity_propagation_inertia = len(affinity_propagation.cluster_centers_indices_)
affinity_propagation_silhouette = silhouette_score(X, affinity_propagation_labels)
affinity_propagation_db_index = davies_bouldin_score(X)

clustering_model = None if best_cluster_technique == 'K-means':
```

```
# Upload CSV file

import streamlit as st

# Maximum allowed file size in bytes (1GB)

max_file_size = 1 * 1024 * 1024 * 1024

uploaded_file = st.file_uploader("Upload a CSV file", type=["csv"]) if uploaded_file:
# Check the file size

if len(uploaded_file.getvalue()) > max_file_size:

st.error("Error: File size exceeds the allowed limit (1GB). Please upload a smaller
else:
st.success("File uploaded successfully!") if uploaded_file is not None:
data = pd.read_csv(uploaded_file) # Display the uploaded data st.write("Uploaded
Data:") st.write(data)
# Check if 'Churn' column exists in the dataset

if 'Churn' in data.columns

data = data.drop(columns=['customerID'])

# Extract features and target variable

X = data[['tenure', 'MonthlyCharges', 'TotalCharges']] y = data['Churn']
# Perform clustering n_clusters = 3
kmeans = KMeans(n_clusters=n_clusters) kmeans_labels = kmeans.fit_predict(X)
# ... (perform other clustering methods as needed)
kmeans_inertia = kmeans.inertia
# Perform BIRCH Clustering

birch = Birch(n_clusters=n_clusters) birch_labels = birch.fit_predict(X) birch_inertia
= 0.0
for cluster_id in range(n_clusters): cluster_points = X[birch_labels == cluster_id]
cluster_center = birch.subcluster_centers_[cluster_id]
```

```
cluster_inertia = np.sum(np.square(np.linalg.norm(cluster_points - cluster_center
birch_inertia += cluster_inertia
birch_silhouette = silhouette_score(X, birch_labels) birch_db_index =
davies_bouldin_score(X, birch_labels) birch_calinski_harabasz =
calinski_harabasz_score(X, birch_labels)
birch_score = birch_inertia + birch_silhouette/rch_db_index)birch_calinski_harabasz

# Perform Agglomerative Clustering

agglomerative = AgglomerativeClustering(n_clusters=n_clusters)
agglomerative_labels = agglomerative.fit_predict(X) cluster_centers = []
for cluster_id in range(agglomerative.n_clusters_): cluster_points =
X[agglomerative_labels == cluster_id] cluster_center = cluster_points.mean(axis=0)
cluster_centers.append(cluster_center)
cluster_centers = np.array(cluster_centers) agglomerative_inertia = 0.0for cluster_id
in range(agglomerative.n_clusters_): cluster_points = X[agglomerative_labels ==
cluster_id]

cluster_center = cluster_centers[cluster_id]

cluster_inertia = np.sum(np.square(np.linalg.norm(cluster_points - cluster_center)
affinity_propagation.fit_predict(X)
affinity_propagation_inertia = len(affinity_propagation.cluster_centers_indices_)
affinity_propagation_silhouette = silhouette_score(X, affinity_propagation_labels)
affinity_propagation_db_index = davies_bouldin_score(X)
data = {

'Technique': ['K-means', 'BIRCH', 'Agglomerative', 'Affinity Propagation'], 'Score':
[kmeans_score, birch_score, agglomerative_score]
scores_df = pd.DataFrame(data)

# Choose the clustering method with the highest score

# Get cluster assignments based on the best clustering technique
# ... (perform clustering and determine the best clustering technique)
best_cluster_technique = scores_df.loc[scores_df['Score'].idxmax(), 'Technique']
print("Best Clustering Technique:")
print(best_cluster_technique)
# Check the best clustering technique and proceed accordingly
cluster_points = X[agglomerative_labels == cluster_id] cluster_center =
cluster_centers[cluster_id]
cluster_inertia = np.sum(np.square(np.linalg.norm(cluster_points - cluster_center)
agglomerative_inertia += cluster_inertia
agglomerative_silhouette = silhouette_score(X, agglomerative_labels)
agglomerative_db_index = davies_bouldin_score(X, agglomerative_labels)
agglomerative_calinski_harabasz = calinski_harabasz_score(X) agglomerative_score
= agglomerative_inertia + agglomerative_silhouette + (1 /
agglomerative_db_index) + agglomerative_calinski_harabasz # Perform Affinity
Propagation Clustering
affinity_propagation = AffinityPropagation() affinity_propagation_labels =
affinity_propagation.fit_predict(X)
affinity_propagation_inertia = len(affinity_propagation.cluster_centers_indices_)
affinity_propagation_silhouette = silhouette_score(X, affinity_propagation_labels)
affinity_propagation_db_index = davies_bouldin_score(X)
data = {
'Technique': ['K-means', 'BIRCH', 'Agglomerative', 'Affinity Propagation'], 'Score':
[kmeans_score, birch_score, agglomerative_score]
scores_df = pd.DataFrame(data)
# Choose the clustering method with the highest score
# Get cluster assignments based on the best clustering technique
# ... (perform clustering and determine the best clustering technique)
```

```
best_cluster_technique = scores_df.loc[scores_df['Score'].idxmax(), 'Technique']
print("Best Clustering Technique:")
print(best_cluster_technique)
# Check the best clustering technique and proceed accordingly
clustering_model = Noneif best_cluster_technique == 'K-means': clustering_model =
KMeans(n_clusters=n_clusters)
elif best_cluster_technique == 'BIRCH': clustering_model =
Birch(n_clusters=n_clusters)
elif best_cluster_technique == 'Agglomerative':
clustering_model = AgglomerativeClustering(n_clusters=n_clusters) elif
best_cluster_technique == 'Affinity Propagation':
clustering_model = AffinityPropagation() cluster_labels =
clustering_model.fit_predict(X) # Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(cluster_labels.reshape(-1, 1), y, test_
size= 0.2, random_state=42)
# Initialize LabelEncoder
label_encoder = LabelEncoder()
# Encode target variable for ML models y_train_encoded =
label_encoder.fit_transform(y_train) y_test_encoded =
label_encoder.transform(y_test)
# Initialize and train the Logistic Regression model
logreg = LogisticRegression() logreg.fit(X_train, y_train_encoded) y_pred_logreg =
logreg.predict(X_test)
accuracy_logreg = accuracy_score(y_test_encoded, y_pred_logreg) # Initialize and
train the Random Forest Classifier random_forest = RandomForestClassifier()
random_forest.fit(X_train, y_train_encoded)
y_pred_rf = random_forest.predict(X_test)
```

# Initialize and train the Decision Tree Regressor decision_tree_reg = DecisionTreeRegressor() decision_tree_reg.fit(X_train, y_train_encoded)
y_pred_dt_reg = decision_tree_reg.predict(X_test)

mse_dt_reg = mean_squared_error(y_test_encoded, y_pred_dt_reg)

# Display results

st.write("Best Clustering Technique: ", best_cluster_technique) st.write("Logistic Regression Accuracy: ", accuracy_logreg) st.write("Random Forest Accuracy: ", accuracy_rf)
st.write("Mean Squared Error for Decision Tree Regression: ", mse_dt_reg)

# Initialize individual models

logreg = LogisticRegression() random_forest = RandomForestClassifier()
decision_tree = DecisionTreeClassifier()
# Create a voting classifier with hard voting

voting_classifier = VotingClassifier(estimators=[('lr', logreg), ('rf', random_forest), decision_tree)], voting='hard')

kmeans_silhouette = silhouette_score(X, kmeans_labels) kmeans_db_index = davies_bouldin_score(X, kmeans_labels) kmeans_calinski_harabasz = calinski_harabasz_score(X, kmeans_labels)
kmeans_score = kmeans_inertia + kmeans_silhouette + (1 / kmeans_db_index)
# Fit the voting classifier on the training data

voting_classifier.fit(X_train, y_train)

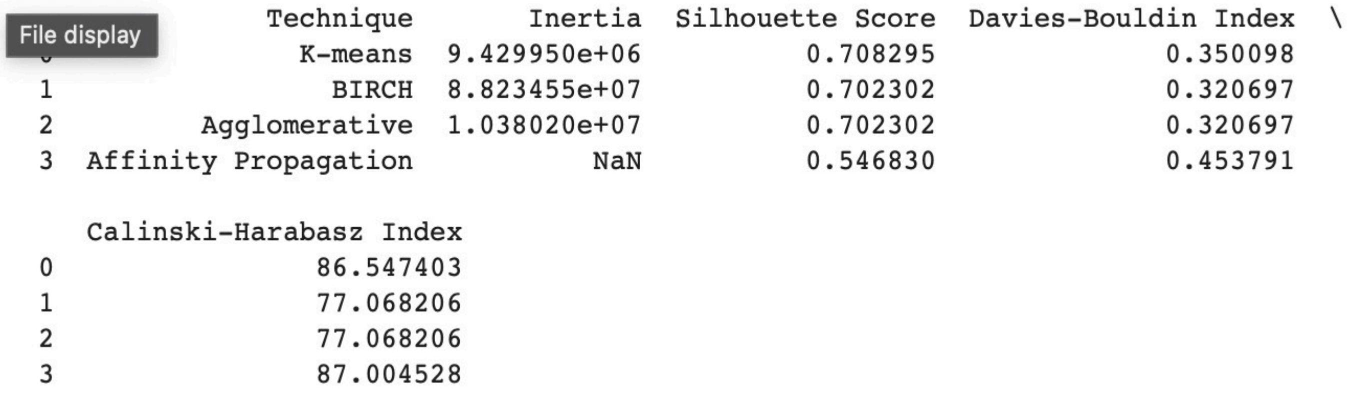
# Make predictions on the test set y_pred_voting = voting_classifier.predict(X_test)
# Calculate accuracy for the voting classifier
accuracy_voting = accuracy_score(y_test, y_pred_voting)

# Display the accuracy

st.write("Accuracy with Hard Data Fusion (Voting Classifier):", accuracy_voting)

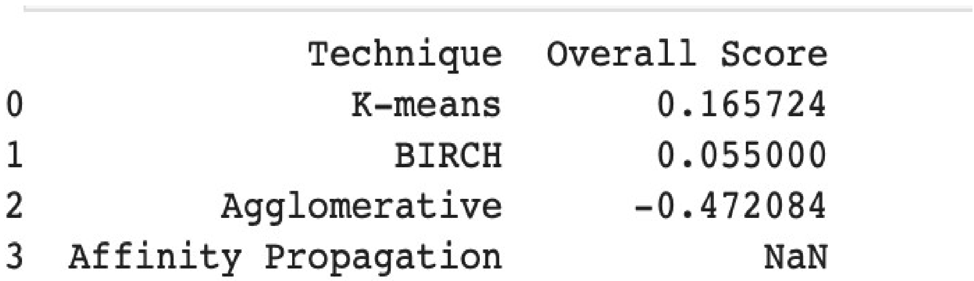
A.1.2 SCREENSHOTS

A.1.2.1 CLUSTER COMPARISON SCORE



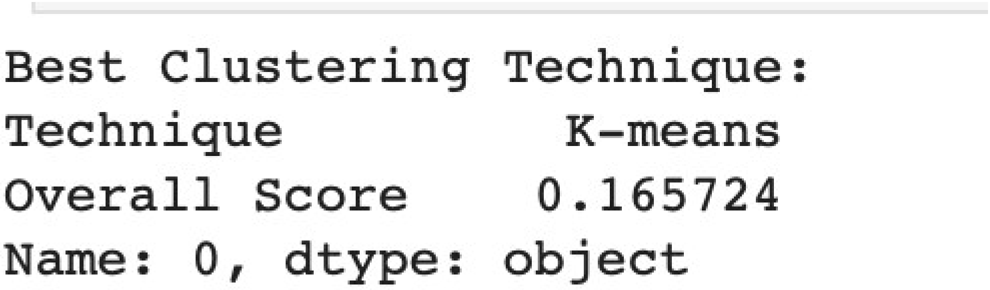
```
                Technique        Inertia  Silhouette Score  Davies-Bouldin Index  \
0                 K-means   9.429950e+06          0.708295              0.350098
1                   BIRCH   8.823455e+07          0.702302              0.320697
2           Agglomerative   1.038020e+07          0.702302              0.320697
3    Affinity Propagation            NaN          0.546830              0.453791

   Calinski-Harabasz Index
0                86.547403
1                77.068206
2                77.068206
3                87.004528
```

A.1.2.2 OVERALL SCORE

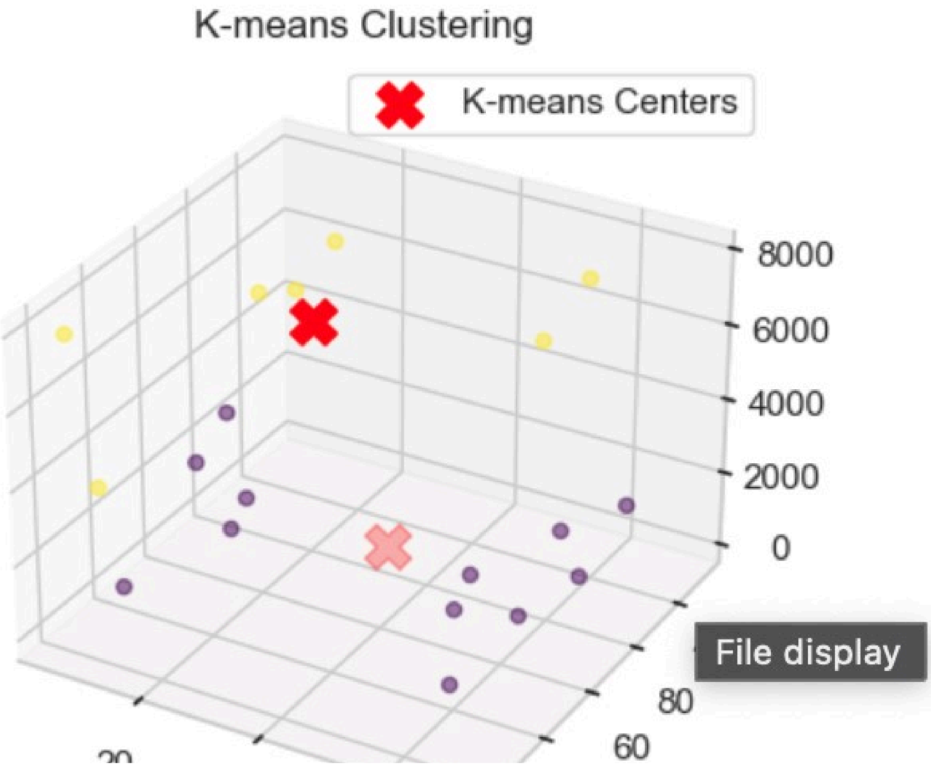


```
              Technique  Overall Score
0               K-means       0.165724
1                 BIRCH       0.055000
2         Agglomerative      -0.472084
3  Affinity Propagation            NaN
```

A.1.2.3 BEST CLUSTERING TECHNIQUES



```
Best Clustering Technique:
Technique              K-means
Overall Score         0.165724
Name: 0, dtype: object
```



```
Logistic Regression Accuracy with Best Clustering Technique: 0.75
```

REFERENCES

1. Xiancheng Xiahou and Yoshio Harada , "B2C E-Commerce Customer Churn Prediction Based K-Means and SVM.

2. Burez J., & Van den Poel, D “Crm at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services”, Expert Systems with Applications 32,277–288.

3. Ledro, C., Nosella, A., & Vinelli, A. (2022). Artificial intelligence in customer relationship management: literature review and future research directions. Journal of Business & Industrial Marketing, 37(13), 48-63.

4. Jain, H., Khunteta, A., & Srivastava, S. (2020). Churn prediction in telecommunication using logistic regression and logit boost. Procedia Computer Science, 167, 101-112.

5. Khulood Ebrah, Selma Elnasir “Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms”.Journal of Computer and Communications > Vol.7 No.11, November 2019.

6. Nagaraju Jajam, Nagendra Panini Challa, Kamepalli S.L.Prasanna “Arithmetic Optimization With Ensemble Deep Learning SBLSTM-RNN-IGSA Model for Customer Churn Prediction” in IEEE vol11.

7. Soumi De, Prabu.P” A Sampling-Based Stack Framework for Imbalanced Learning in Churn Prediction in IEEE vol 10.

8. Prabadevi.B, Shalini.R, Kavitha.B.R (2023). Customer Churning analysis using machine learning algorithms. In International Journal of Intelligent Networks.

9. M. Alizadeh, D. S. Zadeh, B. Moshiri and A. Montazeri, "Development of a Customer Churn Model for Banking Industry Based on Hard and Soft Data Fusion," in IEEE Access, vol. 11, pp. 29759-29768, 2023 doi:10.1109/ACCESS.2023.3257352

10. Anand, M., Shaukat, I., Kaler, H., Narula, J., & Rana, P. S. Hybrid Model for the Customer Churn Prediction

11. Zadoo, A., Jagtap, T., Khule, N., Kedari, A., & Khedkar, S. (2022, May). A review on churn prediction and customer segmentation using machine learning. In 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON) (Vol. 1, pp. 174-178). IEEE.

12. PM, U., & Balaji, N. V. (2019). Analysing Employee attrition using machine learning. Karpagam Journal of Computer Science, 13, 277-282.

13. Abdulsalam Sulaiman Olaniyi , Arowolo Micheal Olaolu , Bilkisu Jimada- Ojuolape , Saheed Yakub Kayode,,"Customer Churn Prediction in Banking Industry Using K-Means and Support Vector Machine Algorithm. In International Journal of Multidisciplinary Sciences and Advanced Technology Vol 1 No 1 (2020) 48–54.

14. A. S. Kumar and D. Chandrakala, "A survey on customer churn prediction using machine learning techniques", *Int. J. Comput. Appl.*, vol. 154, no. 10, pp. 1-4, 2016.

15. R. A. de Lima Lemos, T. C. Silva and B. M. Tabak, "Propension to customer churn in a financial institution: A machine learning approach", *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11751-11768, Jul. 2022.

16. A. K. Ahmad, A. Jafar and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform", *J. Big Data*, vol. 6, no. 1, pp. 1-24, Dec. 2019.

17. A. De Caigny, K. Coussement and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees", *Eur. J. Oper. Res.*, vol. 269, no. 2, pp. 760-772, Sep. 2018.

18. V. V. Saradhi and G. K. Palshikar, "Employee churn prediction", *Expert Syst. Appl.*, vol. 38, no. 3,pp. 1999-2006, Mar. 2011.

19. P. Lalwani, M. K. Mishra, J. S. Chadha and P. Sethi, "Customer churn prediction system: A machine learning approach", *Computing*, vol. 104, pp. 271-294, Feb. 2022.

20. O. F. Seymen, O. Dogan and A. Hiziroglu, "Customer churn prediction using deep learning", *Proc. 12th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, pp. 520-529, Apr. 2021..

21. C. Kirui, L. Hong, W. Cheruiyot and H. Kirui, "Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining", *Int. J. Comput. Sci. Issues*, vol. 10, pp. 165, Mar2020.