THOMPSON RIVERS UNIVERSITY

An Individual Participant Data Meta-Analysis of the Machine Learning Models in Predicting Customer Churn

By

Khanh Van Tran

A GRADUATE PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

KAMLOOPS, BRITISH COLUMBIA

August, 2024

SUPERVISOR

Dr. Jabed Tomal

Dr. Md Erfanul Hoque

ABSTRACT

Predicting customer churn is a critical task for businesses aiming to retain customers and maintain profitability. This research adopts an individual participant data meta-analysis (IPD-MA) approach to evaluate the effectiveness of various machine learning models in predicting customer churn across multiple publicly available datasets. This methodology facilitates a robust comparison and validation of predictive models by integrating raw data from different studies. The study employs a two-stage approach: first, individual datasets are analyzed to obtain machine learning performance metrics; second, these aggregated metrics are combined using fixed-effect and random-effect meta-analysis models. The results reveal significant variability in model performance across different datasets, with ensemble methods like Catboost, Lightgbm, and Gradient Boosting consistently outperforming other models, achieving the highest average AUCs of 0.9036, 0.9000, and 0.8936, respectively. The study also highlights the importance of considering dataset-specific characteristics and model capabilities, as well as the necessity of accounting for heterogeneity in meta-analyses. This research makes several key contributions, including methodological advancements in applying IPD-MA to machine learning, and a comprehensive evaluation of model performance. The findings offer a valuable reference for selecting and optimizing machine learning models in various industrial applications, guiding future research and practical implementations.

Key Words: Customer Churn, Machine Learning, Meta-Analysis, Individual Participant Data, Fixed-Effect Model, Random-Effect Model.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my two supervisors, Dr. Jabed Tomal and Dr. Md Erfanul Hoque for teaching, supervising, guiding, and mentoring me in the entire process of my graduation project, which indeed has helped me to shape my research properly and overcome lots of obstacles.

An additional warm thank you goes to the instructors in the MSc Data Science program at Thompson Rivers University: Dr. Roger Yu, Dr. Mohamed Tawhid, Dr. Becky Wei Lin, Dr. Mateen Shaikh, and Dr. Mila Kwiatkowska.

And I'd like to thank my family for the love and support they've shown me every step of the way. Their unwavering faith in me helped carry me through all this.

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Problem Statement 2

1.3 Research Objectives 2

1.4 Research Questions 3

1.5 Significance of The Study 3

1.6 Structure of The Project 4

2 Background **5**

2.1 The Importance of Customer Churn Prediction 5

2.2 Machine Learning in Churn Prediction 6

2.3 Machine Learning Algorithms 8

2.3.1 Logistic Classification 8

2.3.2 The K-Nearest Neighbors (KNN) 9

2.3.3 Support Vector Machines (SVM) 10
2.3.4 Naive Bayes 11
2.3.5 Decision Trees 11
2.3.6 Ensemble Classifiers 12
2.4 Model Evaluation 15
2.4.1 Stratified K-Fold Cross-Validation 15
2.4.2 Model Performance Metrics 16
2.5 Meta-Analysis 19

3 Data **21**

4 Methodology **26**

4.1 Individual Dataset - Exploratory Data Analysis and Data Processing . . 27
4.1.1 Exploratory Data Analysis (EDA) 27
4.1.2 Data Preparation 32
4.2 Individual Dataset - ML Models Evaluation 36
4.2.1 Predictive Modeling 36
4.2.2 Classifiers Performance Evaluation 37
4.2.3 Hyperparameter Tuning 37
4.3 Aggregate Effect Size and Standard Error 38
4.4 Estimating Pooled Effect Size and Confidence Interval 39

4.4.1 Fixed-Effect Model 39
4.4.2 Random-Effect Model 41
4.5 Residual Heterogeneity Estimates 42
4.5.1 Test for Heterogeneity 43
4.5.2 Fixed-Effect Model 43
4.5.3 Random-Effect Model 44

5 Results **46**

5.1 Aggregated Data 46
5.1.1 Performance Metrics by ML Model 46
5.1.2 Performance Metrics by Dataset 47
5.2 Fix and Random Effect Model Results 48
5.2.1 Meta-Analysis by ML Model 49
5.2.2 Meta-Analysis by Dataset 50
5.3 Residual Heterogeneity Estimates 53
5.3.1 Residual Heterogeneity Estimates by ML Model 53
5.3.2 Residual Heterogeneity Estimates by Dataset 53
5.4 Feature Importance 54

6 Discussion and Conclusion **63**

6.1 Discussion 63

6.1.1 Variability in Model Performance 63

6.1.2 Top Performing Models and Datasets 64

6.1.3 Fixed-Effect vs. Random-Effect Models 65

6.1.4 Feature Importance 65

6.1.5 Implications for Practice 67

6.1.6 Contributions of This Research 67

6.1.7 Limitations and Future Research 68

6.2 Conclusion 69

A Source Codes **75**

List of Figures

2.1 Stratified 5-Fold Cross Validation. 16

2.2 Confusion matrix. 17

2.3 The ROC-AUC. 18

4.1 Two-stage IPD-MA Workflow 27

4.2 Target Distribution. 29

4.3 Categorical Features Distribution in Credit Card Dataset. 30

4.4 Numerical Features Distribution in Bank Dataset. 31

4.5 Correlations Among Numerical Features in Telcom Churn Dataset. . . . 32

4.6 The training class samples before and after apply SMOTE. 35

5.1 Fix Effect Model Forest Plot by ML. 49

5.2 Random Effect Model Forest Plot by ML. 50

5.3 Fix Effect Model Forest Plot by Dataset. 51

5.4 Random Effect Model Forest Plot by Dataset. 52

5.5 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Bank Dataset. 55

5.6 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Credit Card Dataset. 56

5.7 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for E-Commerce Dataset. 57

5.8 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Employee Dataset. 58

5.9 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Internet Dataset. 59

5.10 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Cell2Cell dataset. 59

5.11 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Membership dataset. 60

5.12 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Nigeria Telecom dataset. 60

5.13 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the SA Wireless dataset. 61

5.14 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Telco Europa Dataset. 61

5.15 Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Telecom Dataset. 62

List of Tables

3.1 Summary of Churn Datasets 21

4.1 Summary of Internet Churn Dataset 28

5.1 Mean AUC, Standard Deviation, and Standard Error for Each Model . . 47

5.2 Mean AUC, Standard Deviation, and Standard Error for Each Dataset . 48

Chapter 1

Introduction

1.1 Background and Motivation

In today's fast-paced business landscape companies are increasingly turning to data driven strategies to maintain their edge. Customer churn is when a customer stops purchasing products and services from a company. Customers are hard to acquire and might be very expensive for an organization. If a customer does not use your firm's products and services in the long run, it will inevitably lead to the bad fate of an organization. This is why customer retention is also as important as customer acquisition. Without appropriate attention to customer churn management, it can result in heavy monetary losses, reduced profitability, and loss of visibility in the market. Therefore, organizations started relying on predictive analytics or machine learning approaches to predict or avoid customer churn.

Machine learning (ML) technologies have revolutionized approaches, providing new ways to forecast and analyze customer churn. ML models use algorithms to analyze datasets and uncover patterns that may not be evident using traditional statistical methods. However, the effectiveness of these models can vary depending on the characteristics and industry specifics.

Given the availability of datasets and diverse ML techniques, there is a strong demand to combine these resources to evaluate the performance of these models thoroughly. This project utilizes an individual participant data meta-analysis a statistical technique that reanalyzes raw data from multiple studies to assess how reliable and effective ML models in predicting customer churn across different datasets and industries.

1.2 Problem Statement

Despite the critical insights provided by individual studies on ML models for churn prediction, there is substantial variability in their outcomes. This variation may be influenced by factors such as dataset characteristics, industry-specific dynamics, and methodological approaches. This research aims to mitigate these inconsistencies through an IPD-MA, offering a deeper and more accurate understanding of how different models perform across varied contexts. By integrating data from multiple sources, this study seeks to establish a clearer understanding of the generalizability and limitations of current machine learning approaches to churn prediction.

1.3 Research Objectives

This study is structured around several key objectives designed to address the complexities of machine learning in churn prediction:

1. Comprehensive Model Evaluation: To quantitatively assess the performance metrics of various machine learning models across 11 public datasets in predicting customer churn.

2. Cross-industry Applicability: To investigate the efficacy of these models across different industry sectors, identifying how contextual variables influence predictive success.

3. Feature Relevance Analysis: To explore which variables are consistently significant predictors of churn across multiple models and datasets, highlighting critical factors influencing customer retention.

1.4 Research Questions

1. Which machine learning models most accurately predict customer churn across diverse datasets?
2. Are there universally significant predictors of churn, or do these vary significantly across different models and datasets?
3. To what extent do machine learning models maintain their predictive accuracy across various industry sectors?
4. How can machine learning models be optimized to improve churn prediction based on the insights gained from the meta-analysis?

1.5 Significance of The Study

The significance of this study lies in its potential to:

- Bridge Knowledge Gaps: By providing an analysis of machine learning models across multiple datasets, this study aims to fill existing gaps in the literature regarding the comparative effectiveness and limitations of these models.
- Enhance Business Strategies: The findings will aid businesses in refining their customer retention strategies by adopting the most effective predictive models tailored to their specific industry conditions.

- Contribute to Academic Knowledge: This research contributes to the academic field by demonstrating the application of IPD-MA in machine learning, potentially setting a precedent for future research methodologies.

1.6 Structure of The Project

This research project is structured as follows:

- Chapter 2: Background - This chapter comprehensively reviews existing studies on machine learning models for churn prediction, outlining theoretical underpinnings and methodological advancements.
- Chapter 3: Data - Describes data sources used in the study.
- Chapter 4: Methodology - Describes the meta-analytical approach, model selection, and statistical methods employed in the study.
- Chapter 5: Results - Analyze the results and discuss the implications of findings in the context of existing research and industry practices.
- Chapter 6: Discussion and Conclusion - discuss and conclude with a summary of key findings, discuss limitations, and suggest directions for future research.

Chapter 2

Background

This chapter explains the importance of customer churn prediction. It also provides a summary of the literature review about the application of machine learning in the area of churn prediction. In addition, a brief description of the machine learning algorithms and techniques is also included.

2.1 The Importance of Customer Churn Prediction

Predicting customer churn plays a crucial role in Customer Relationship Management (CRM). The main goal of the CRM is to build and sustain lasting relationships with customers. This approach is important in many sectors, particularly to subscription-based service firms, such as those in telecommunications, insurance, banking, and online services (Geiler et al., 2022). Given that such companies rely on consistent and regular membership fees, it is imperative to curb customer switching behavior to maintain sustainable profits. Consequently, the precise prediction of customers likely to churn has become a paramount objective in the industry.

2.2 Machine Learning in Churn Prediction

Churn prediction study has gained popularity in recent years, due to its substantial impact on corporations and businesses. Machine learning has been widely studied and applied in churn prediction across several industries such as telecommunications, human resources, finance, and online services.

Several case studies have been published in the telecommunications industry. Lalwani et al. (2022) utilized a range of predictive models, including logistic regression, Naive Bayes, support vector machine, random forest, decision trees, boosting, and ensemble approaches, to forecast customer turnover in the telecom sector. K-fold cross-validation was utilized for hyperparameter tuning and to mitigate overfitting. The Adaboost and XGBoost classifiers demonstrated the best accuracies among the built models, with 81.71% and 80.8%, respectively. Ahmad et al. (2019) created a churn prediction model utilizing machine learning techniques on a big data platform to help telecom operators forecast which customers are likely to churn. The dataset was acquired from SyriaTel telecommunications firm. The paper shows that XGBoost algorithm achieved the most optimal outcomes for predicting churn. In addition, Ullah et al. (2019) introduced a churn prediction model that utilizes classification and clustering approaches to identify consumers who are likely to churn and to understand the underlying variables contributing to customer turnover in the telecommunications industry. The model employed the Random Forest technique to classify churn and leveraged cosine similarity to categorize churn clients. The process involved the utilization of feature selection techniques and an attribute-selected classifier algorithm to detect and determine the key elements contributing to churn. The assessment of the suggested approach showcased greater churn categorization and client profiling, facilitating customer retention tactics, promotion suggestions, and improved marketing campaigns. A case study by Qureshi et al. (2013) on churn prediction in the mobile communication market. It describes the use of regression analysis, decision trees, artificial neural networks, and logistic regression to predict potential churners. The study used a dataset from the Customer DNA website, where usage

data for 106,000 customers was provided over three months, along with total usage by the customers. Dealing with the problem of class imbalance in the dataset, the authors check how the different machine learning algorithms performed regarding real usage by the customer, while demonstrating that the decision trees were the best classifiers for identifying potential churners.

In the human resource (HR) management sector, Sisodia et al. (2017) built a model for predicting employee churn rate based on HR analytics dataset, using five different machine learning algorithms, namely, linear support vector machine, decision tree classifier, random forest, k-nearest neighbor, and naïve bayes classifier. The authors evaluated the correlation between attributes, generated a histogram to contrast left employees with various factors, and proposed strategies to optimize employee attrition in organizations.

Rahman and Kumar (2020)'s paper focuses on predicting customer churn in a commercial bank using efficient data mining methods. It discusses data transformation and classification techniques such as k-nearest neighbor, support vector machine, decision tree, and random forest. The paper results show that oversampling improves the accuracy of decision trees and random forest classifiers, while support vector machine is not suitable for large amounts of data. The study also analyzes customer behavior to explore the likelihood of churn and compares the performance of different models, finding that the Random Forest model after oversampling achieves higher accuracy.

Several published papers also try to provide a generalizability of ML in churn prediction. García et al. (2017) suggested a more comprehensive literature review. The authors explain several phases involved in churn prediction analysis, including data collecting, feature selection, model implementation, and assessment techniques and metrics. Their survey closes with suggestions derived from the existing body of information. Geiler et al. (2022) focused on churn prediction in businesses and explored the performance of various supervised and semi-supervised learning methods and sampling approaches on publicly available datasets. The study suggests an ensemble approach should be used for churn prediction.

Given the diversity of methods and the varying results across different studies, there is a compelling need for a meta-analytical approach to churn prediction. Individual Participant Data Meta-Analysis stands out as an advantageous method for this purpose. By integrating data from multiple studies, IPD-MA allows for more comprehensive analysis and interpretation of churn prediction techniques across different datasets and industries. Furthermore, IPD-MA supports the application of uniform statistical methods across different studies, providing a more reliable and consistent framework for evaluating the efficacy of machine learning techniques in churn prediction. This approach is particularly valuable in the context of churn prediction, where variations in industry-specific factors and model implementations can significantly influence the effectiveness of predictive strategies.

2.3 Machine Learning Algorithms

2.3.1 Logistic Classification

Logistic regression is the simplest machine learning algorithm for binary as well as multiclass classification. It estimates the probability that an instance belongs to one of the classes as a function of input features using the logistic function (Eq. 2.1).

$$P(y = 1|\mathbf{x}) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_k x_k}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_k x_k}} \tag{2.1}$$

where:

- $P(y = 1|\mathbf{x})$ is the probability that the outcome y is 1 given the input features $\mathbf{x}$.
- $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ is the vector of input features data.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are the model parameters.

- e is the base of the natural logarithm.

During training, the model parameters are optimized using techniques such as maximum likelihood estimation and gradient descent (Witten and James, 2013, Géron, 2022). It is very efficient to compute and highly interpretable.

2.3.2 The K-Nearest Neighbors (KNN)

KNN is a machine learning method used for both regression and classification tasks. KNN utilizes distance measurements to predict the most probable value of the target feature. The Euclidean distance is often used in KNN and it is calculated using Equation 2.2 (Witten and James, 2013). The predicted class in categorization is the most popular class among the k neighbors. The optimal k nearest neighbors are identified for the given case based on the cross-validation. This method is non-parametric. It is a categorization method based solely on examples, utilizing existing data without generalization. It is called a lazy learning algorithm since its stages and operations are executed during the query. The algorithm does not require any preparation. It is effective with low-dimensional data but less so with high-dimensional data. To use KNN for high-dimensional data, we may use principle component analysis before implementing KNN.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.2}$$

where:

- $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between points $\mathbf{x}$ and $\mathbf{y}$.
- $\sum_{i=1}^{n}$ denotes the summation over all dimensions from 1 to n.
- x_i and y_i are the coordinates of points $\mathbf{x}$ and $\mathbf{y}$ in the i-th dimension, respectively.
- $(x_i - y_i)^2$ is the squared difference between the i-th coordinates of the two points.

2.3.3 Support Vector Machines (SVM)

SVM is a supervised machine learning technique that has been widely used for classification purposes. It searches the hyperplane that accurately divides the majority of the training data into two classes, while it may incorrectly categorize a small number of observations. This is the optimal solution to the following optimization problem (Witten and James, 2013).

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_N,M}{\text{maximize}} \; M \tag{2.3}$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1 \tag{2.4}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \xi_i) \tag{2.5}$$

$$\xi_i \geq 0, \quad \sum_{i=1}^{n} \xi_i \leq C \tag{2.6}$$

where:

- M is the width of the margin.
- $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients of the hyperplane
- $\epsilon_1, \ldots, \epsilon_N$ individual observations that allow fall on the incorrect side of the margin or the hyperplane.
- $x_1, \ldots, x_n$ set of n training observations
- $y_1, \ldots, y_n$ associated class labels
- C is a nonnegative tuning parameter.

SVM can perform both linear and nonlinear classification. It can find a nonlinear separator, also known as a functional, by transforming the data into a higher-dimensional

space. The algorithm then uses linear classification by selecting the support vectors to define the decision boundary or hyperplane. The SVM algorithm can utilize kernel functions, such as linear, polynomial, and radial basis function (RBF) to handle both linearly and nonlinearly separable data (Géron, 2022).

2.3.4 Naive Bayes

Naive Bayes is a type of probabilistic classifier model, which is based on the Bayes theorem. The Naive part comes from the assumption that observation characteristics are conditionally independent given the class label. It is stated mathematically as Equation 2.7 (Witten and James, 2013). Multiple class predictions are made feasible by probabilistic classifiers. Based on conditional probability, the decision is made.

$$Pr(Y = k \mid X = \mathbf{x}) = \frac{\pi_k \times f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)}{\sum_{l=1}^{K} \pi_l \times f_{l1}(x_1) \times f_{l2}(x_2) \times \cdots \times f_{lp}(x_p)} \tag{2.7}$$

for k = 1,...,K

where:

- Pr($Y = k \mid X = \mathbf{x}$) is posterior probability that an observation X belongs to k class.
- π_k is prior probability that a randomly chosen observation comes from the k class.
- f_{kj} is the density function of the jth predictor among observations in the kth class.

2.3.5 Decision Trees

Decision trees are powerful and versatile techniques used for classification and regression. It is commonly referred to as the Classification and Regression Trees (CART) method. The method is capable of handling data with large dimensionality and can process both

numerical and categorical data. It operates by recursively partitioning the dataset into subsets based on the most informative features (Witten and James, 2013). Each internal node of the tree represents a decision point where a feature is evaluated, and each leaf node corresponds to a class label or a regression value. Decision trees are constructed using various criteria such as Gini impurity or information gain, to optimize the splitting process (Géron, 2022). They are interpretable models that facilitate human understanding of decision-making processes in complex datasets.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2 \tag{2.8}$$

where:

- G_i is gini score of node i-th
- $p_{i,k}$ is the training instance ratio of class k in the i-th node.

2.3.6 Ensemble Classifiers

Ensemble method can help to minimize the variance and bias of separate models or classifiers by merging their predictions, resulting in more accurate and reliable models (Latha and Jeeva, 2019). There are three commonly used ensemble learning methods, namely bagging, boosting, and stacking, which can be utilized to enhance the machine learning process. In this section, we will delve into the details of each method, including its working nature, characteristics regarding data generation, training of baseline classifiers, and suitable fusion methods. We will also cover the advantages, disadvantages, and implementation challenges associated with each method.

Bagging

The bagging method, also referred to as bootstrap aggregating, is a data-driven algorithm that involves creating multiple subsets of data from the original dataset (Breiman,

1996). Bagging aims to generate diverse predictive models by adjusting the distribution of training datasets, where even small changes in the training data set can lead to significant changes in the model predictions. Bagging reduces variance, eliminates overfitting, and performs well on high-dimensional data. However, it also has some drawbacks such as being computationally expensive, having high bias, and reducing the interpretability of models (Bühlmann and Yu, 2002). The Random Forests algorithm is a notable example of the bagging technique (Breiman, 2001). Implementing the bagging method presents several challenges, such as determining the optimal number of base learners and subsets, the maximum number of bootstrap samples per subset, and the fusion method for integrating the outputs of the base classifiers using various voting methods. In summary, the bagging method employs parallel ensemble techniques with no data dependency, and the fusion methods depend on different voting methods to generate predictions. This approach generates B different bootstrapped training data sets. The algorithm is then trained on the b_{th} bootstrapped training set to predict a point x. The following equation is the bagging function:

$$f(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f_b(\mathbf{x}) \tag{2.9}$$

where $f_b(\mathbf{x})$ is a weak learner

Boosting

The boosting approach is a sequential procedure in which each succeeding model attempts to correct the prior model's errors (Freund, Schapire, et al., 1996). Boosting employs multiple weak learners in a highly adaptive way, where each model in the sequence is fitted while giving more importance to observations that the previous models in the sequence handled poorly. Boosting can be used for both regression and classification problems and includes algorithms such as Adaptive Boosting (AdaBoost), Stochastic Gradient Boosting (SGB), and Extreme Gradient Boosting (XGB) (Freund et al., 2003, J. H. Friedman, 2001,and J. Friedman et al., 2000). Several studies have utilized various

types of boosting, including AdaBoost for noise detection and speech feature extraction, and XGB for fake news classification (Sun et al., 2016, Asbai and Amrouche, 2017, and Haumahu et al., 2021). Boosting provides interpretability and helps reduce variance and bias in machine learning ensembles. However, the drawback of boosting is that each classifier must correct errors in the predecessors, and scaling sequential training can be challenging. Additionally, boosting is computationally costly, vulnerable to overfitting, and slower to train than bagging. To summarize, boosting is an ensemble learning technique that uses a sequential approach, where multiple learners learn sequentially with data dependency, and fusion methods rely on various voting methods. The boosting function is shown as follows:

$$f(x) = \sum_i \lambda_i g_i(x) \tag{2.10}$$

Where several classifiers $g_i(x)$ create a strong classifier $f(x)$. The inclusion of the shrinkage parameter λ in the model slows down the process, which enables a wider range of differently shaped trees to be utilized to address the residuals.

Stacking

The stacking method is a powerful model ensembling technique that combines multiple predictive models to generate a new model, also known as a meta-model (Džeroski and Ženko, 2004). The stacking model architecture consists of two or more base models (level 0 models) and a meta-model that integrates the predictions of the base models (level 1 models). The base models are fit on the training data and their predictions are compiled, while the meta-model learns how to best combine these predictions. One of the key benefits of stacking is its ability to provide a deeper understanding of the data, leading to increased precision and effectiveness in predictions. However, a major challenge with stacking is overfitting, which can occur when there are many predictors that all predict the same target. Additionally, multi-level stacking can be both data and time-intensive, as each layer adds multiple models. As the amount of available data

grows exponentially, computation time complexity becomes an issue, and highly complex models may take months to run (Xiong et al., 2021). Another challenge with stacking is interpreting the final model, as well as identifying the appropriate number and baseline models that can be relied upon to generate better predictions from the dataset when designing a stacking ensemble from scratch. The problem of multi-label classification also poses issues such as overfitting and the curse of dimensionality due to the high dimensionality of the data (Elnagar et al., 2020). To sum up, stacking is a parallel ensemble method that creates baseline learners simultaneously without any data dependency. The meta-learning method determines the fusion techniques. Although stacking can be very successful, it poses several challenges such as overfitting, complexity, and interpretability, which must be taken into account when using it. The stacking model is as follows:

$$f_s(x) = \sum_{i=1}^{n} a_i f_i(x) \tag{2.11}$$

Stacking makes predictions from several models $(f_1, f_2, \ldots, f_n)$ to build a new model, where the new model is used to make predictions on the test dataset. Stacking aims to improve a model's predictive ability. To put it simply, stacking involves combining the predictions of multiple models by assigning weights to each prediction and adding them together by a linear combination of weights a_i.

2.4 Model Evaluation

2.4.1 Stratified K-Fold Cross-Validation

Since the dataset is highly skewed, stratified K-fold cross-validation is used to evaluate ML models. In the K-fold cross-validation, the original dataset has first been divided into K-folds. Each fold has an equal number of training cases and an equal number of test cases. For model training, the train model is built based on K-1 folds as a training

set and tested with the K-th test fold.

This process has been repeated K times. Each fold takes a turn to be the test set. Finally, the K-time modeled performance has been summarized. The major disadvantage with the K-fold cross-validation design is that when there is a severe class imbalance between classes in the original data set, the size of one class in the training fold is much less than the number of cases in the other class of the original data set (Pedregosa et al., 2011). The consequence of that is that the performance outcome in the validation evaluation is too optimistic and in favor of the model.

Therefore, stratified K-fold cross-validation is recommended in this case, and each fold should be stratified in the same way as the original data set. This can ensure that the number of each class in every fold is the same as the class distribution of the original data set. Figure 2.1 Stratified 5-fold cross-validation (Adapted from Pedregosa et al., 2011)

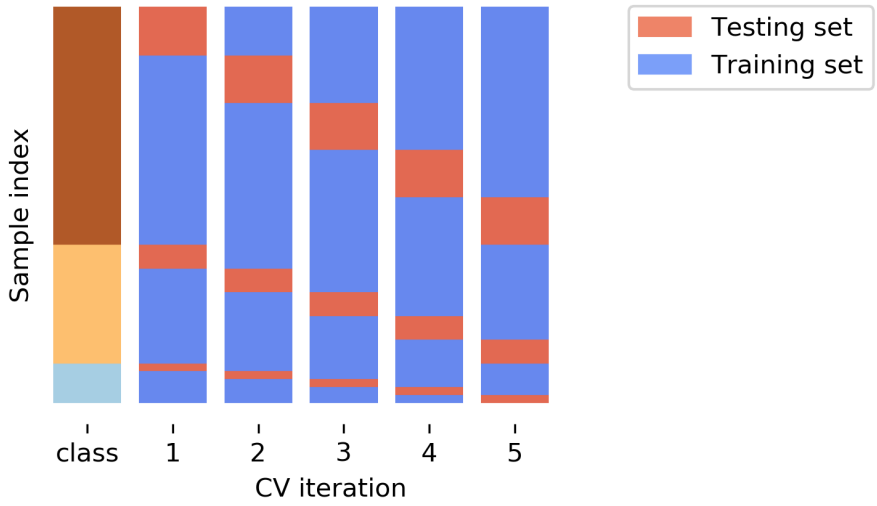

Figure 2.1: Stratified 5-Fold Cross Validation.

2.4.2 Model Performance Metrics

The performance of the models developed in this study is assessed using the confusion matrix and its derived metrics, namely accuracy, precision, recall, specificity, and F1

score. Figure 2.2 demonstrates the confusion matrix represented with actual and predicted outcome categories plotted against its axis (Adapted from Bryan, 2020).

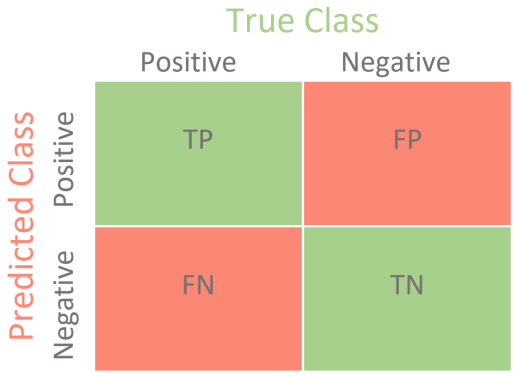

Figure 2.2: Confusion matrix.

Accuracy

Equation 2.12 calculates accuracy as the proportion of correctly predicted classes over the total number of instances in the dataset (Witten and James, 2013). Accuracy is used to measure the overall performance of a model but can be misleading in imbalanced datasets

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.12}$$

Precision

Precision shows the proportion of positive predictions that are truly positive (Witten & James, 2013). It is basically a ratio of correctly positively labeled to all positively labeled and can be calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

Recall

Recall has other names Sensitivity or True Positive Rate. It is a measure of the percentage of actual positive classes that are predicted as positive (Witten & James, 2013). It can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

F1 Score

The F1 Score is a measure that is calculated from precision and recall (Equation 2.15). It is often a preferred metric over accuracy when data is unbalanced.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.15}$$

ROC-AUC

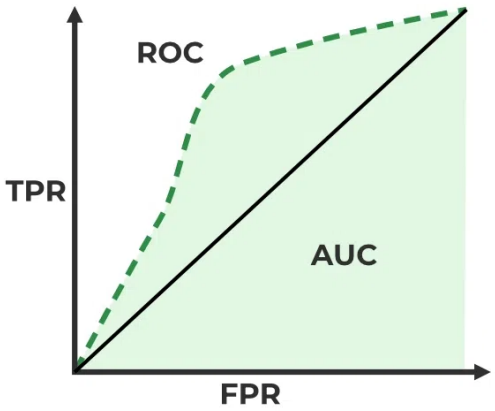

Figure 2.3: The ROC-AUC.

The ROC AUC curve is an assessment metric for classification at various discrimination thresholds. The Receiver Operator Characteristic (ROC) is the probability curve, and AUC is the area under the ROC curve that estimates the degree of separability. It indicates how well the model can distinguish the two classes. The higher the AUC, the better the model (Witten and James, 2013). AUC is calculated by plotting the True

Positive Rate (TPR)/Sensitivity on the y-axis versus the False Positive Rate (FPR)/(1-Specificity) on the x-axis (Fig. 2.3).

2.5 Meta-Analysis

Meta-analysis is a statistical method created to combine and analyze findings, from scientific studies that explore the same topic. The term meta-analysis was introduced by statistician Gene V. Glass in the 1970s (Glass, 1976). Glass described meta-analysis as the examination of a set of study results from individual research to amalgamate the discoveries. In the late 1970s and 1980s, researchers such as Thomas D. Cook, Donald T. Campbell, and Frank L. Schmidt further advanced the methodology of meta-analysis by introducing techniques for systematically merging findings evaluating variations among study outcomes, and addressing potential biases (Harrer et al., 2021). This era also witnessed the introduction of effect size as a measure for comparing results across studies regardless of the measurement scales employed in each study. From the 1980s to the 1990s meta-analysis became widely embraced in fields including medicine, education, social sciences, and ecology. Noteworthy organizations like The Cochrane Collaboration (established in 1993) and the Campbell Collaboration actively promote systematic reviews and meta-analyses in healthcare and social policy.

However, meta-analyses that use published aggregate data can be ineffective in some scenarios. In medical care, a weighted average may not be useful when relative treatment effect estimates are heterogeneous. Identifying whether impact modification affects treatment effects across clinical subgroups is crucial in such cases. Published aggregate data can be used to explore treatment effect modifiers, but when summary data (e.g., mean age) are similar across trials, it lacks power. More significantly, published aggregate data cannot account for subject-level factors, making it biased for effect modification research. When aggregate data are not accessible, inadequately reported, determined, and presented differently across studies (e.g., odds ratio vs relative risk), and more likely

to be reported (and in more detail) when statistically or clinically significant, further issues arise (Riley et al., 2021). This is why researchers are increasingly using individual participant data meta-analysis (Riley et al., 2010). These meta-analyses incorporate raw data from each relevant article (preferably discovered via a systematic review). By obtaining IPD of individual trials, subject- and study-level variability in treatment effect may be separated. This may assist investigate effect modification (Simmonds et al., 2005) and consistently control for confounding factors. Access to IPD may also improve data quality, standardize definitions and analyses, obtain complete follow-up data on all randomized participants, combine studies with different follow-up times, and analyze multiple outcomes (Riley et al., 2010, Riley et al., 2010, Simmonds et al., 2005).

Chapter 3

Data

This study uses 11 available public churn datasets. Table 3.1 shows the datasets' name, source, and number of data records as well as the number of features in each dataset.

Table 3.1: Summary of Churn Datasets

Dataset Names	**Sources**	**Entries**	**Features**
Telecom Customer Churn	MavenAnalytics	7043	38
Internet Service Churn	Kaggle	72274	11
Bank Customer Churn Records	Kaggle	10000	18
Credit Card Churn	Kaggle	10127	21
E-commerce Churn	Kaggle	5630	20
Employee Churn	Kaggle	1070	35
Telco Europa	Kaggle	190776	20
Telcom Cell2Cell	Kaggle	71047	70
Membership Subscription	Kaggle	10362	15
Wireless Telecom South Asia	Kaggle	2000	14
Nigeria Telecoms Churn	Kaggle	1401	16

The first dataset that I examined is related to telecom customer churn. It can be

found on the Maven Analytics website. It consists of two CSV formatted tables. The Customer Churn table contains details of 7,043 customers from a telecommunications company in California during the quarter of 2022. Each entry in this dataset corresponds to a customer and includes information about their demographics, location, tenure services subscribed to, and customer status among other relevant details. Additionally, the Zip Code Population table offers population estimates for the California zip codes mentioned in the Customer Churn table.

Another dataset used for analysis is the internet service churn dataset available on Kaggle. This dataset compiles customer data from an internet service provider with a total of 72,274 records containing 11 attributes. These characteristics range, from indicators like whether customers have subscribed to services such as TV and movie packages to complex variables like how long they have been subscribed (subscription age) the average amount they are billed (bill avg) and metrics related to service quality (service failure count download avg and upload avg). There are instances of missing data in attributes such as remaining contract details and internet speed measures (download avg and upload avg) which means that some cleaning or filling in of data is needed before further analysis can be done. The target feature is the churn attribute, which indicates whether customers are continuing with or ending their services.

The other dataset named bank customer churn records which accessible on Kaggle. This dataset contains 10,000 records and 18 features, outlining the characteristics of bank clients. It covers information, financial details, and patterns of service usage including name, location, gender, credit score, account balance, and number of banking products used. The dataset includes factors related to customer satisfaction and engagement levels such, as the Satisfaction Score, Complaints, and Exited statuses. It also covers details like Card Type and Points Earned which may indicate how effective customer loyalty programs are. There are no missing data in the dataset making the preprocessing phase simple.

This study also utilizes the credit card churn dataset available on Kaggle. This

dataset has 10,127 entries with 23 attributes, offering insights into credit card customers' profiles. The dataset contains both categorical variables, like Marital Status and Income Category, and continuous variables such as Customer Age and Credit Limit. It also includes demographic information like age, gender, and education level along with behaviors such as credit limit usage and total transaction amount. Additionally, it tracks changes in customer status through features like Attrition Flag.

The dataset e-commerce churn is also used for this research and can be found on Kaggle. It comprises 5,630 entries and 20 columns that represent customer attributes in an e-commerce environment. This dataset contains a mix of numerical and categorical data such as customer ID, gender, login device, payment methods, and cashback amount. Some columns have missing values suggesting the need for data cleaning before analysis. The dataset includes features like product purchase counts, customer demographics, and transaction specifics that could aid in understanding customer behavior and predicting churn in an e-commerce setting. The presence of a range of values, in product category and transaction columns implies that the dataset captures a diverse array of customer interactions and preferences.

Employee attrition data is also included in this study. This dataset is available on Kaggle, which comprises 1,470 entries with 35 features. These features are categorical and numerical, offering an overview of the employees. It contains the employee demographic features like age and gender, job specifics such as department, role, and monthly earnings, and satisfaction indicators like work environment, job fulfillment, and work-life balance. The target variable is Attrition, signaling whether employees are still part of the organization or have moved on.

Another dataset included in this study is Telco Europa. This dataset is accessible on Kaggle, having 190,776 entries with 20 attributes. These features include both categorical and continuous variables. These features include basic service usage metrics and extend to geographical and technical dimensions. Notable categorical variables like cni_customer which represents customer identifiers, and churn, a binary indicator show-

ing whether a customer has discontinued the service or not. Continuous variables include days_life representing the tenure of the customer in days, device_technology indicating the type of device technology used, and various metrics related to call minutes such as tot_min_call_out and avg_min_in_3 for the average incoming call minutes over the last three months. Other important variables are min_plan and price_plan, indicating the minutes and cost of the customer's plan, respectively. Additionally, it features technical attributes for both data and voice services, such as tec_ant_data, state_data, city_data, tec_ant_voice, state_voice, and city_voice.

The Telcom Cell2Cell dataset is another dataset that was used in this study. This dataset is downloaded from the Kaggle website. It provided by Duke University, consists of 71,047 records and 70 attributes focused on customer churn within a telecommunications framework. The dataset captures a wide array of information, including basic churn indicators, detailed usage metrics such as minutes of use, charges for overages and roaming, and customer service interactions including call quality metrics like dropped and blocked calls. Additionally, it provides insights into customer demographics and equipment usage, featuring variables that describe age, device usage duration, and possession of web-capable devices.

The Membership Subscription dataset was also used in this project. This dataset contains 10,362 entries, detailing various aspects of membership within an organization, encapsulated in 15 attributes. These include the membership number, annual fees, and detailed member demographics such as marital status, gender, annual income, and occupation. Key attributes further encompass the type of membership package, the number of additional members, and the respective payment mode. Each entry also tracks the membership life-cycle through start and end dates and the membership status, providing a rich basis for analyzing membership retention.

The South Asian Wireless Telecom Churn dataset comprises 2,000 records with 14 attributes that detail various aspects of customer usage and interaction with a telecom service provider. It includes metrics such as network age, total revenue, and revenue

specifics from SMS and data usage, alongside total data volumes. Notable features include categorical variables like user type and favorite services for different months, and continuous variables such as the number of calls made, revenue from on-net and off-net calls, and the count of customer complaints. Furthermore, the dataset tracks changes in customer status through the Class attribute, which indicates whether customers have continued or discontinued their services.

The `Nigeria Telecoms Churn` dataset encompasses 1,401 entries with 16 attributes, detailing the telecom usage and interactions of customers in Nigeria. It includes comprehensive metrics such as network age, customer tenure, total expenditure over two months, SMS and data spending, and data consumption. The dataset further assesses customer engagement through metrics like the total number of unique calls, spending on calls within and outside the network, and the number of complaint calls to the service center. Additionally, it captures the network technology type preferred by customers across two months and their favored competitor networks during the same period. A pivotal aspect of this dataset is the Churn Status, which indicates whether customers have remained with or left the service provider, providing critical insights for churn analysis and customer retention strategies.

Following are my methodology and result chapters. Please write the discussion and conclusion chapter:

Chapter 4

Methodology

This research adopts an individual participant data meta-analysis approach to evaluate the effectiveness of various machine learning models in predicting customer churn across multiple datasets. This methodology allows for a more nuanced analysis by integrating raw data from different studies, thus facilitating a comprehensive comparison and validation of predictive models under varying conditions. Since the public datasets have a large variation in the number of features and feature names, the two-stage approach in IDP-MA is applied in this project. In the first stage, each study (dataset) analyzes the IPD separately to obtain aggregate data. This includes ML performance effect estimates and standard errors. In the second stage, the aggregate data obtained from the first stage are then combined in a standard meta-analysis model. In this project, both common-effect and random-effects model will be applied. The figure 4.1 shows the workflow of this project research methodology. In which, the first stage analysis processes is represented in blue box while second stage is in the green box.

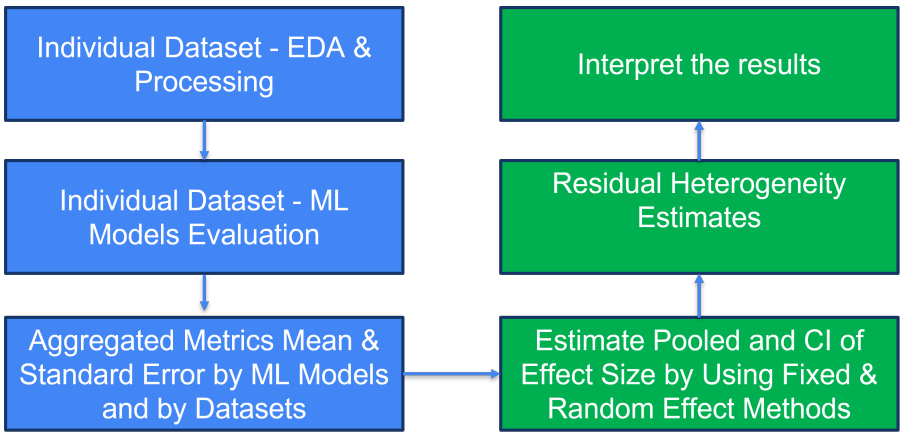

Figure 4.1: Two-stage IPD-MA Workflow

4.1 Individual Dataset - Exploratory Data Analysis and Data Processing

4.1.1 Exploratory Data Analysis (EDA)

This EDA process was similarly applied to all 11 datasets, with visualizations and statistical analyses tailored to the specific characteristics of each dataset. The EDA process included creating summary data tables for each dataset to identify unique values, missing values, NaN values, duplicated entries, and data types. It also involved visualizing the distribution of the target variable to identify class imbalances and analyzing the distribution of categorical features to understand customer preferences and service uptake. Additionally, the distribution of numerical features was examined to assess data range, central tendency, and dispersion. Finally, the correlation matrix was evaluated to identify multicollinearity and guide feature selection. These steps ensure a robust exploratory analysis, providing valuable insights that inform the predictive modeling and strategic decision-making processes.

We begin by creating a summary data table for each dataset. The summary data table is a valuable tool in the exploratory data analysis phase, providing a concise overview

Table 4.1: Summary of Internet Churn Dataset

Feature	Unique	Missing	NaN	Duplicated	Dtypes
reamining_contract	247	21572	21572	0	float64
download_avg	2856	381	381	0	float64
upload_avg	802	381	381	0	float64
id	72274	0	0	0	int64
is_tv_subscriber	2	0	0	0	int64
is_movie_package_subscriber	2	0	0	0	int64
subscription_age	1110	0	0	0	float64
bill_avg	179	0	0	0	int64
service_failure_count	19	0	0	0	int64
download_over_limit	8	0	0	0	int64
churn	2	0	0	0	int64

of the dataset's characteristics. These tables highlight the number of unique values, missing values, NaN values, duplicated entries, and data types for each feature. This information is crucial for identifying categorical versus continuous features, guiding the handling of missing data, and ensuring appropriate data types for analysis and modeling. For instance, the table 4.1 reveals that the reamining_contract feature has a significant number of missing values, necessitating imputation or exclusion, while binary features like is_tv_subscriber are confirmed as categorical. Additionally, understanding the extent of missing data and identifying features with high cardinality or duplicates helps prioritize data cleaning efforts and informs feature engineering decisions.

The distribution of the target variable is a crucial aspect of EDA as it provides valuable insights into the balance or imbalance within the dataset. This information is essential for understanding the dataset's characteristics and informing subsequent modeling decisions. Figure 4.2 effectively illustrates the varying distributions of target variables across 11 different datasets, providing clear examples of where imbalances exist. For instance, the Telecom Customer Status plot in figure 4.2 shows a significant difference

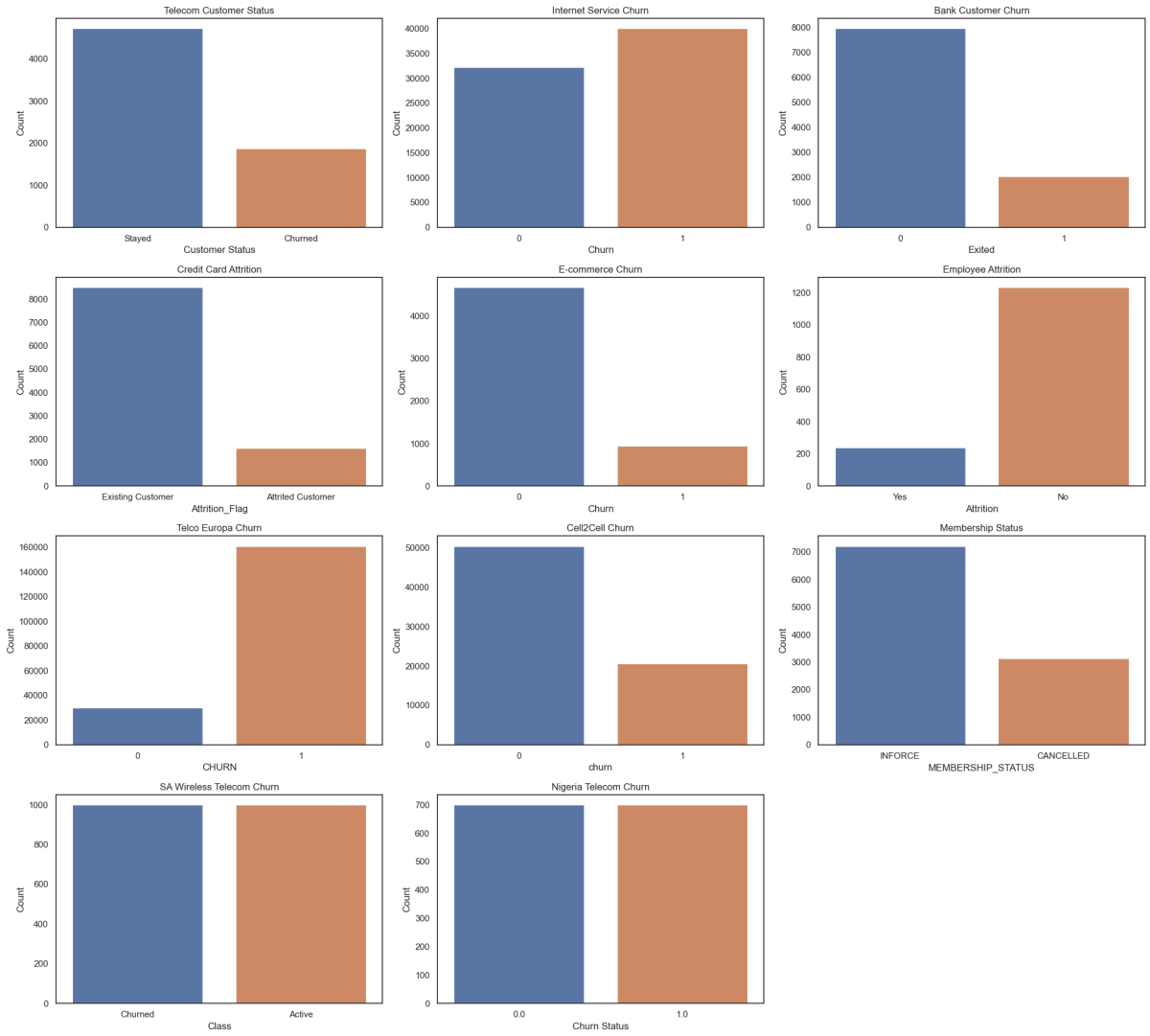

Figure 4.2: Target Distribution.

between the number of customers who stayed versus those who churned, indicating a class imbalance. Identifying such imbalances is critical because they can bias the predictive model, leading to poor performance in predicting the minority class. Similarly, the Credit Card Attrition plot reveals an imbalance with more existing customers than attrited customers, highlighting the need for strategies to handle such disparities in data. Understanding these imbalances helps in selecting appropriate techniques like oversampling, undersampling, or synthetic data generation (e.g., SMOTE) to balance the classes and choosing evaluation metrics like precision, recall, F1-score, and ROC-AUC, which are more informative for imbalanced datasets.

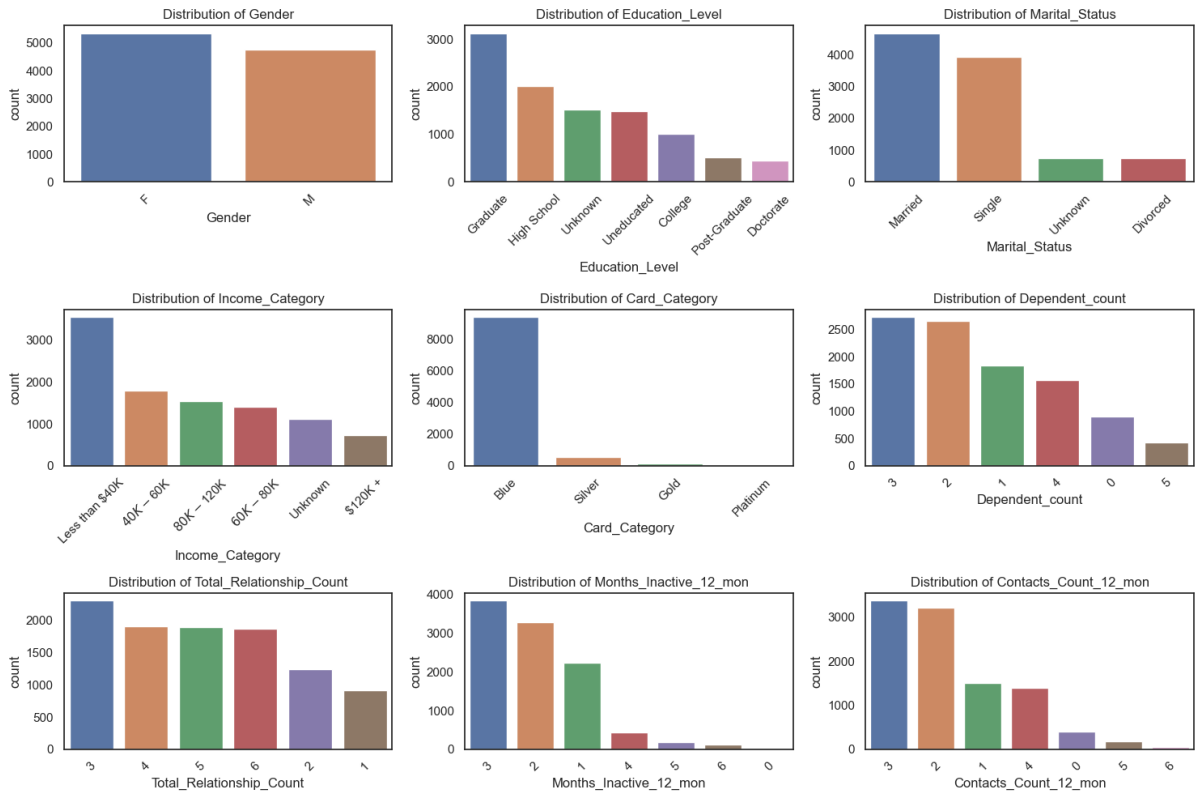

Figure 4.3: Categorical Features Distribution in Credit Card Dataset.

Next, the distribution of categorical features is crucial in exploratory data analysis as it provides valuable insights into the composition and characteristics of the dataset. Understanding these distributions aids in feature engineering and selection, allowing me to decide which features to include in their models and how to preprocess them effectively. Figure 4.3 displays the distribution of several categorical features from the credit card dataset. Each subplot provides a count plot for a specific categorical feature, revealing key patterns and distributions within the data. For instance, the Distribution of Gender subplot shows a nearly balanced distribution between female and male customers, with a slight predominance of females. While, the Distribution of Education Level indicates that most customers are graduates, followed by high school graduates and those with unknown educational backgrounds, while the least represented are doctorate holders.

The distribution of numerical features is an essential part of EDA. This analysis provides a comprehensive understanding of the dataset's range, central tendency, and

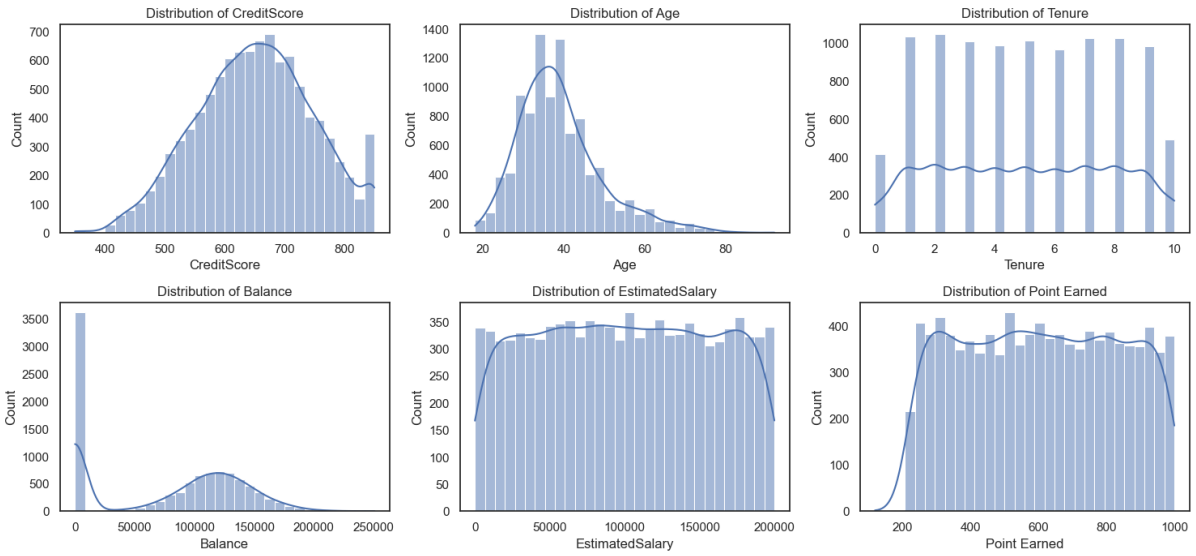

Figure 4.4: Numerical Features Distribution in Bank Dataset.

dispersion. For example, histograms and Kernel Density Estimation plots can reveal whether the data is normally distributed, skewed, or contains outliers, which is crucial for deciding on the appropriate statistical methods and data transformations to apply. This step ensures that the numerical data is accurately represented and ready for further analysis and modeling. Figure 4.4 displays the distribution of several numerical features from the bank customer dataset. The Distribution of CreditScore subplot shows an approximately normal distribution, centered around a mean value with a slight skew towards higher scores, indicating most customers have credit scores between 500 and 800, peaking around 700. The Distribution of Age is right-skewed, with the majority of customers aged between 30 and 50.

The heatmap, which visualizes the correlation matrix, is another powerful tool in EDA. It helps in identifying multicollinearity among numerical features by displaying the correlation coefficients between pairs of variables. High correlations indicate redundancy, which can affect the performance of predictive models. For instance, if two features are highly correlated, one might be dropped or transformed to avoid multicollinearity issues in the model. The heatmap provides a clear and immediate visual representation of these relationships, making it easier to spot and address potential problems. By understanding

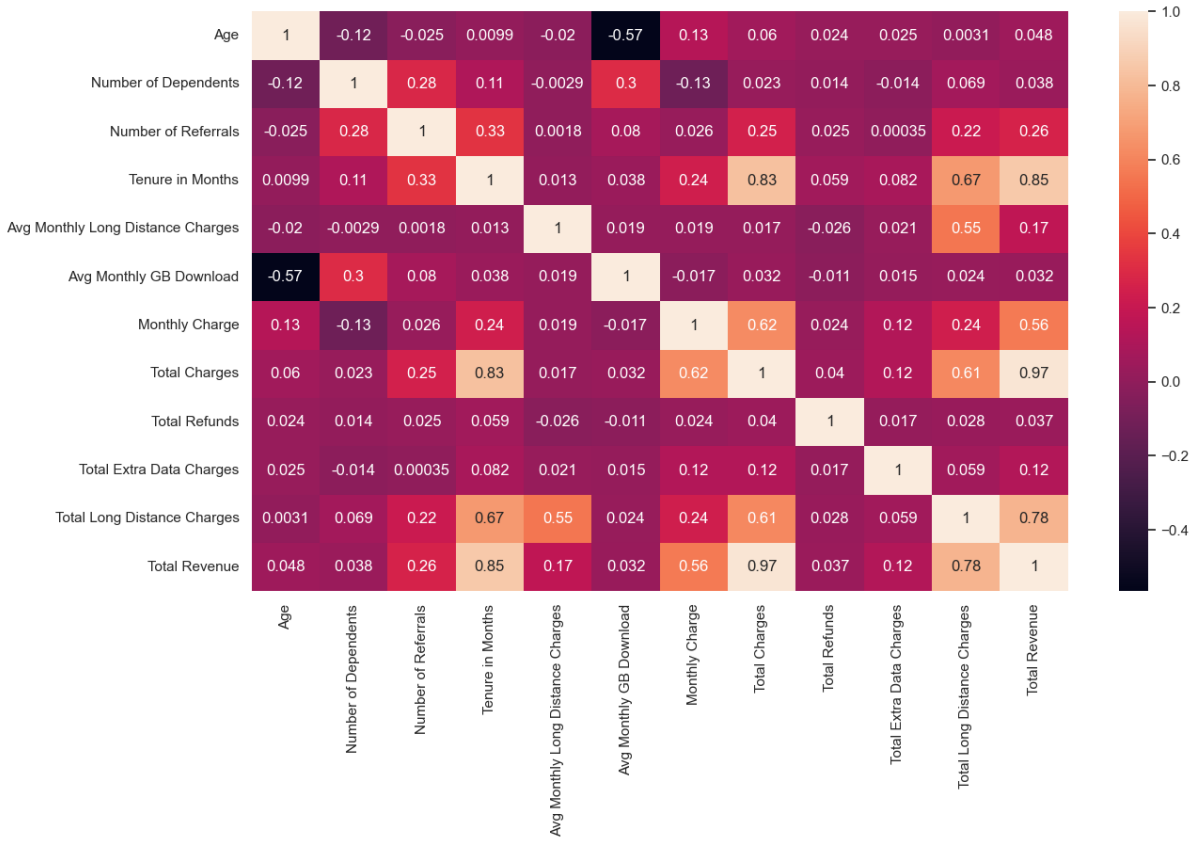

Figure 4.5: Correlations Among Numerical Features in Telcom Churn Dataset.

the correlations, data scientists can make informed decisions about feature selection and engineering, ensuring that the models are both efficient and effective. Figure 4.5 shows the correlation coefficients among numerical variables in the telecommunications data set. It can be seen that there is a strong correlation between total revenue and total charges, with a correlation of 0.97.

4.1.2 Data Preparation

Data preparation is the next step after data collection and before machine learning model building in this workflow. It includes data splitting, missing value treatment, scaling numerical variables, encoding categorical features, imbalanced data handling, etc. All these are vital steps for data preparation before training a machine learning algorithm.

Data Splitting

Splitting the dataset into train and test sets is done first in the data preparation to avoid any potential data leakage during this step. Data leakage occurs when training models can access information outside of the training dataset (Shabtai et al., 2012). This can lead to overestimating models' performance during the training process. Once the model is deployed in production, its performance can decrease significantly. The predictions turn out to be far less reliable than the performance of the model during training would suggest. Steps can be taken to eliminate what is called data leakage so that the performance during training is a better reflection of what the model can do with data it has not seen already.

Since the telecom customer churn dataset is unbalanced data, stratified splitting was applied to ensure that the class distribution of training and test sets is similar (Joseph and Vakayil, 2022). The initial dataset was split into 75 percent and 25 percent for the train and test sets, respectively. The random state is set to make sure the experiment result is reproducible.

Handling Missing Values

First, some non-informative columns, namely, customer id, city, zip code, latitude, and longitude were dropped, assuming that these variables simply do not offer predictive power. In addition, direct churn-related columns, churn category, and churn reason were also removed because they are directly related to churn and only available after customer churn. If these two columns were included in the model, it would introduce a very optimistic model for churn prediction.

The process continues with the imputation of missing values for categorical variables. As pointed out in the previous exploratory data analysis, for service-related features like multiple lines, internet type, and online security, among others, we will fill the missing entries with No because these customers do not have a service. By replacing missing

categorical data with meaningful labels, we avoid introducing bias arising from dropping rows that contain missing data, which may lead to the loss of important information.

For a continuous variable such as average monthly long distance charges or average monthly GB download, the imputation method used is zero filling. The rationale is that these customers do not subscribe to phone or internet service.

Scaling Numerical Variables

The numerical columns were scaled using min-max scaling. In this study, the MinMaxScaler function from the sklearn.preprocessing module is used to implement the scaling task. It calculates the difference between each value and the minimum feature value and then divides it by the difference between the maximum and minimum feature values. This scaling process transformed all the numerical features into a range between 0 and 1, which would help many machine learning algorithms that are very sensitive to the scale of the input features (Alshdaifat et al., 2021).

Encoding Categorical Variables

In the encoding phase, binary categorical variables such as gender, married, phone service, and paperless billing are manually mapped to numeric binary values, with a straightforward mapping of female and yes to 1, and male and no to 0. This binary encoding transforms the categorical data into a format suitable for machine learning algorithms, which require numerical input.

The remaining categorical variables undergo one-hot encoding, a process that converts categorical variable values into a form that could be provided to machine learning algorithms to do a better job at prediction. The OneHotEncoder creates a binary column for each category and returns a sparse matrix or dense array (Pedregosa et al., 2011). By employing one-hot encoding, the model can interpret the data without falsely attributing

order or priority where it does not exist (Seger, 2018).

Data Sampling

The SMOTE technique was implemented to see whether it could help minimize class imbalance problems in the churn dataset or not. SMOTE should apply only to the training set to prevent leaking information during the training process. Which is essential for the model's performance evaluation.

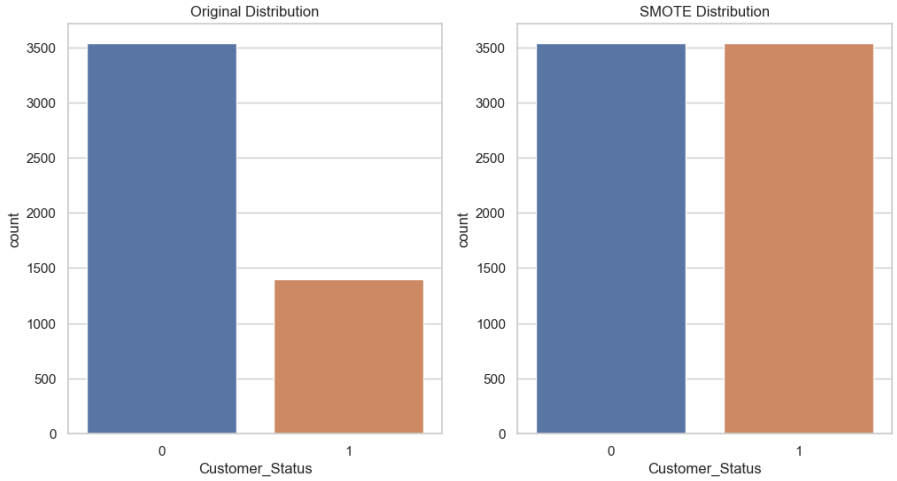

Figure 4.6: The training class samples before and after apply SMOTE.

Applying SMOTE to the training set can be done using a library such as imbalanced-learn in Python. The minority class is oversampled so that the training set contains synthetic samples that are nearest to the samples of the minority class. After training on the newly balanced training set, the model is assessed on the untouched testing set to test its ability to generalize to new data. Figure 4.6 shows the target class distribution with and without applying the SMOTE technique.

4.2 Individual Dataset - ML Models Evaluation

4.2.1 Predictive Modeling

In this study, a large number of machine learning models were implemented to find which model was best suited to the research tasks. Many of these models are ensembles that consist of other machine learning algorithms.

First, logistic regression is selected for the model benchmark. Then the support vector machine was also included. This model works well in high dimensions without scaling. For comparability purposes, KNN, gaussian naive bayes, decision trees, and random forest classifiers were selected. Among these algorithms, KNN is a simple model that also works effectively in classification. The Gaussian Naive Bayes classifier is a naturally probabilistic approach known for its overall good performance in probabilistic classification. The decision tree is an easy-to-interpret model, and the random forest is robust and deals with overfitting.

As an additional benchmark, several boosting algorithms were also implemented, including gradient boosting, AdaBoost, and bagging classifiers. These ensemble models are built from many weak learner models into strong ones. A multi-layer perceptron (MLP) classifier, which is an artificial neural network, was also employed. This model is capable of inferring highly non-linear relationships. Moreover, this study includes some highly sophisticated ensemble methods, such as XGBoost and LightGBM, that have proven to be very fast and highly competitive in many machine learning competitions.

Each of the models was initialized with a random seed to ensure that the reproduction of the results was reproducible. The model's performance is evaluated using multiple different metrics: accuracy, precision, recall, and F1 score, to give the best evaluation available. The final model will be decided upon by weighing all the performance measures as well as other considerations such as interpretability and computational efficiency.

4.2.2 Classifiers Performance Evaluation

All the models were evaluated by stratified k-fold cross-validation with 5 splits. This approach ensures that the number of samples for each class is about the same in each split. Such splitting can better evaluate the model performance than a single train-test split, especially for imbalanced data. The metrics of model evaluation are accuracy, precision, recall, F1 score, and ROC-AUC.

For each step of the cross-validation, the initial training data was split into training and validation sets. Then the model was trained on the training set and evaluated on the validation set, and the obtained score for each metric was recorded. This process was repeated for all training-validation splits, and the obtained set of scores was then averaged across folds to get a more stable estimate of the model's performance.

After cross-validation, the models were retrained on the entire initial training set and evaluated on the unseen test set to assess generalization performance. The models were then evaluated on the test set using the same metrics used in the previous evaluation.

4.2.3 Hyperparameter Tuning

GridSearchCV and RandomisedSearchCV were used for hyperparameter tuning. This process was aimed at optimizing the hyperparameters to improve the model's performance. GridSearchCV was used to do an exhaustive search over the parameter grid for each classifier. For each classifier, it evaluates all possible combinations of parameters and picks the best. While, RandomizedSearchCV used a fixed number of parameters randomly sampled. It is less intensive to compute than GridSearchCV, but sometimes it can find a good approximation of the best parameters.

Once the tuning was done, the best parameters for each classifier were determined and re-trained on the entire training dataset. Finally, the retrained models were assessed on the unseen test dataset, where the ability to generalize can be estimated.

4.3 Aggregate Effect Size and Standard Error

Once the performances of various machine learning algorithms on individual datasets were obtained, aggregate metrics were derived for meta-analysis. This approach provides a comprehensive understanding of the algorithms' performances across multiple datasets and allows for a robust comparison.

Aggregate Effect Size

The aggregate effect size is a critical metric in meta-analysis, offering a consolidated measure of the performance of each ML algorithm across all datasets. This was achieved by calculating the average performance metric of each ML algorithm across all datasets. The mean AUC will be used as an aggregated effect size for the next step meta-analysis. This aggregate measure helps in understanding the overall efficacy of each algorithm and identifying which performs best on average. Additionally, the average performance of all ML algorithms on each individual dataset was computed, providing insights into which dataset presented the greatest challenges or the easiest conditions for the algorithms.

Standard Error

Standard error (SE) plays a crucial role in understanding the variability and reliability of the aggregate effect size. It measures the precision of the average performance estimates. For each ML algorithm's aggregate performance, the SE was calculated to quantify the dispersion of performance metrics across the datasets. This involves computing the standard deviation of the performance metrics for each algorithm and dividing it by the square root of the number of datasets. Similarly, the SE for the average performance of all ML algorithms on each individual dataset was calculated. This helps in assessing the consistency of the algorithms' performances on a particular dataset and the robustness of the aggregate metrics.

Calculating Aggregate Performance for Each ML Algorithm Across All Datasets

1. For each ML algorithm, the performance metrics from all datasets were collected.
2. The mean performance metric for each algorithm was computed.
3. The standard deviation of these performance metrics was calculated.
4. The SE for each algorithm's average performance was derived using the formula:

$$SE = \frac{\text{Standard Deviation}}{\sqrt{N}} \tag{4.1}$$

where N is the number of datasets.

Calculating Aggregate Performance for All ML Algorithms on Each Dataset

1. For each dataset, the performance metrics of all ML algorithms were collected.
2. The mean performance metric for each dataset was computed.
3. The standard deviation of these performance metrics was calculated.
4. The SE for each dataset's average performance was derived using the same formula:

$$SE = \frac{\text{Standard Deviation}}{\sqrt{N}} \tag{4.2}$$

where N is the number of evaluated ML models.

4.4 Estimating Pooled Effect Size and Confidence Interval

4.4.1 Fixed-Effect Model

The fixed-effect model assumes that the effect size is consistent across all studies or datasets, and any observed variation is just due to sampling error. The following section

describes the steps and equations to estimate the pooled effect size and its confidence interval (CI) from the aggregated mean AUC and standard error.

1. **Compute the Weights:**

   - Calculate the variance of each effect size:

   $$v_i = SE_i^2 \tag{4.3}$$

   - Assign a weight (w_i) to each effect size, which is the inverse of its variance:

   $$w_i = \frac{1}{v_i} \tag{4.4}$$

   - The weight reflects the precision of each effect size, giving more importance to effect sizes with smaller standard errors.

2. **Calculate the Pooled Effect Size:**

   - The pooled effect size ($\hat{AUC}$) is a weighted average of the individual effect sizes:

   $$\hat{AUC} = \frac{\sum_{i=1}^{k} w_i \hat{AUC}_i}{\sum_{i=1}^{k} w_i} \tag{4.5}$$

   where k is the number of datasets or studies.

3. **Estimate the Variance of the Pooled Effect Size:**

   - The variance of the pooled effect size ($V_{\hat{AUC}}$) is calculated as:

   $$V_{\hat{AUC}} = \frac{1}{\sum_{i=1}^{k} w_i} \tag{4.6}$$

4. **Compute the Standard Error of the Pooled Effect Size:**

   - The standard error (SE) of the pooled effect size is the square root of its variance:

   $$SE_{\hat{AUC}} = \sqrt{V_{\hat{AUC}}} \tag{4.7}$$

5. **Construct the Confidence Interval:**

- The 95% confidence interval for the pooled effect size is calculated using the standard error and the critical value from the standard normal distribution (usually 1.96 for a 95% CI):

$$CI_{95\%} = \hat{AUC} \pm 1.96 \times SE_{\hat{AUC}} \tag{4.8}$$

4.4.2 Random-Effect Model

The random-effects model assumes that the effect sizes vary across studies or datasets due to real differences in effects, as well as sampling error. This model incorporates both within-study and between-study variability to provide a more generalizable estimate of the pooled effect size and its confidence interval (CI). The following section describes the process and equations to estimate the pooled effect size and its CI from aggregated mean AUC and standard error.

1. **Compute the Weights:**

   - Calculate the variance of each effect size:

$$v_i = SE_i^2 \tag{4.9}$$

   - Estimate the between-study variance (τ^2) using the DerSimonian and Laird method (DerSimonian and Laird, 1986, Harrer et al., 2021):

$$\tau^2 = \max\left(0, \frac{Q - (k-1)}{C}\right) \tag{4.10}$$

   where

$$Q = \sum_{i=1}^{k} w_i^* (\hat{AUC}_i - \hat{AUC}_w)^2 \tag{4.11}$$

$$w_i^* = \frac{1}{v_i} \tag{4.12}$$

$$\hat{AUC}_w = \frac{\sum_{i=1}^{k} w_i^* \hat{AUC}_i}{\sum_{i=1}^{k} w_i^*} \tag{4.13}$$

$$C = \sum_{i=1}^{k} w_i^* - \frac{\sum_{i=1}^{k} (w_i^*)^2}{\sum_{i=1}^{k} w_i^*} \tag{4.14}$$

- Compute the random-effects weights (w_i):

$$w_i = \frac{1}{v_i + \tau^2} \tag{4.15}$$

2. **Calculate the Pooled Effect Size:**

   - The pooled effect size ($\hat{AUC}$) is a weighted average of the individual effect sizes:

$$\hat{AUC} = \frac{\sum_{i=1}^{k} w_i \hat{AUC}_i}{\sum_{i=1}^{k} w_i} \tag{4.16}$$

   where k is the number of datasets or studies.

3. **Estimate the Variance of the Pooled Effect Size:**

   - The variance of the pooled effect size ($V_{\hat{AUC}}$) is calculated as:

$$V_{\hat{AUC}} = \frac{1}{\sum_{i=1}^{k} w_i} \tag{4.17}$$

4. **Compute the Standard Error of the Pooled Effect Size:**

   - The standard error (SE) of the pooled effect size is the square root of its variance:

$$SE_{\hat{AUC}} = \sqrt{V_{\hat{AUC}}} \tag{4.18}$$

5. **Construct the Confidence Interval:**

   - The 95% confidence interval for the pooled effect size is calculated using the standard error and the critical value from the standard normal distribution (usually 1.96 for a 95% CI):

$$CI_{95\%} = \hat{AUC} \pm 1.96 \times SE_{\hat{AUC}} \tag{4.19}$$

4.5 Residual Heterogeneity Estimates

Residual heterogeneity refers to the variability in effect sizes that remains unexplained after accounting for within-study variance in meta-analysis. This section outlines the

methodology for estimating residual heterogeneity in both fixed-effect and random-effect models.

4.5.1 Test for Heterogeneity

The test for heterogeneity examines whether the variability observed among study estimates is greater than what would be expected by chance alone. In this context, the null hypothesis (H_0) is that all studies share a common effect size, suggesting no significant heterogeneity. The alternative hypothesis (H_A), on the other hand, proposes that there is significant heterogeneity, indicating that the variability among study estimates is not solely due to sampling error but also due to true differences in effect sizes.

The Q statistic is used to test for heterogeneity, and a significant Q statistic, particularly with a low p-value, supports the rejection of the null hypothesis, confirming the presence of heterogeneity among the studies. The Q-statistic follows a chi-square distribution with $k - 1$ degrees of freedom, where k is the number of studies. A significant Q-statistic (p-value ¡ 0.05) indicates the presence of heterogeneity.

4.5.2 Fixed-Effect Model

In a fixed-effect model, it is assumed that there is a single true effect size that is common across all studies, and any observed variation in effect sizes is due solely to within-study sampling error. However, if there is unexplained heterogeneity, it can be assessed using the following steps:

1. **Calculate the Q-statistic:** The Q-statistic tests for heterogeneity by comparing the observed variability in effect sizes to what would be expected by chance alone. It is calculated as:

$$Q = \sum_{i=1}^{k} w_i (A\hat{U}C_i - A\hat{U}C_w)^2 \tag{4.20}$$

where w_i is the weight assigned to each effect size (the inverse of the variance), $\hat{AUC}_i$ is the observed effect size for study i, and $\hat{AUC}_w$ is the weighted average effect size.

2. **Calculate the I^2 Statistic:** The I^2 statistic quantifies the proportion of total variation in effect sizes that is due to heterogeneity rather than chance. It is calculated as:

$$I^2 = \frac{Q - (k-1)}{Q} \times 100\% \tag{4.21}$$

An I^2 value greater than 50% typically indicates substantial heterogeneity.

3. **Calculate the H^2 Statistic:** The H^2 statistic represents the ratio of total variability to sampling variability. It is calculated as:

$$H^2 = \frac{Q}{k-1} \tag{4.22}$$

H^2 is a measure of heterogeneity, with values greater than 1 indicating the presence of variability beyond what would be expected by sampling error alone.

4.5.3 Random-Effect Model

In a random-effect model, it is assumed that there is not a single true effect size, but rather a distribution of true effect sizes across studies. The model accounts for both within-study variance and between-study variance, often referred to as tau-squared (τ^2). The steps to estimate residual heterogeneity in a random-effect model are as follows:

1. **Estimate Between-Study Variance (τ^2):** The DerSimonian and Laird method is commonly used to estimate the between-study variance. It is calculated as:

$$\tau^2 = \max\left(0, \frac{Q - (k-1)}{C}\right) \tag{4.23}$$

where

$$Q = \sum_{i=1}^{k} w_i^* (\hat{AUC}_i - \hat{AUC}_w)^2 \tag{4.24}$$

$$w_i^* = \frac{1}{v_i} \tag{4.25}$$

$$A\hat{U}C_w = \frac{\sum_{i=1}^{k} w_i^* A\hat{U}C_i}{\sum_{i=1}^{k} w_i^*} \tag{4.26}$$

$$C = \sum_{i=1}^{k} w_i^* - \frac{\sum_{i=1}^{k} (w_i^*)^2}{\sum_{i=1}^{k} w_i^*} \tag{4.27}$$

2. Compute the random-effects weights (w_i):

$$w_i = \frac{1}{v_i + \tau^2} \tag{4.28}$$

3. **Compute the Random-Effects Weights:** Adjust the weights to account for between-study variability:

$$w_i = \frac{1}{v_i + \tau^2} \tag{4.29}$$

where v_i is the within-study variance.

4. **Calculate the I^2 and H^2 Statistics:** Similar to the fixed-effect model, I^2 is used to quantify the proportion of total variation due to heterogeneity, and H^2 represents the ratio of total variability to sampling variability.

$$I^2 = \frac{Q - (k-1)}{Q} \times 100\% \tag{4.30}$$

$$H^2 = \frac{Q}{k-1} \tag{4.31}$$

Chapter 5

Results

5.1 Aggregated Data

This section presents the aggregated results of the mean AUC, standard deviation (SD), and standard error (SE) for each machine learning (ML) model and each dataset. The results provide a general understanding of the performance metrics across various ML models and datasets, highlighting both the models' overall effectiveness and variability. This data will be used as input data for the second stage of meta-analysis.

5.1.1 Performance Metrics by ML Model

Table 5.1 provides the mean AUC, standard deviation, and standard error for each ML model. These metrics reflect the aggregated performance of each model across all datasets.

Catboost achieves the highest mean AUC (0.9036) among the ML models, indicating its strong overall performance across the datasets. Lightgbm (0.9000), Gradient Boosting (0.8936), and Random Forest (0.8891) also demonstrate high mean AUCs, suggesting

Table 5.1: Mean AUC, Standard Deviation, and Standard Error for Each Model

Model	**Mean AUC**	**SD**	**SE**
AdaBoost	0.8836	0.0808	0.0244
Bagging	0.8682	0.0964	0.0291
Catboost	0.9036	0.0836	0.0252
Decision Tree	0.7609	0.1291	0.0389
GaussianNB	0.7882	0.0708	0.0214
Gradient Boosting	0.8936	0.0794	0.0239
KNeighbors	0.7845	0.1085	0.0327
Lightgbm	0.9000	0.0860	0.0259
Logistic Regression	0.8345	0.0859	0.0259
MLP	0.8745	0.0929	0.0280
Random Forest	0.8891	0.0884	0.0266
XGboost	0.8918	0.0923	0.0278

their effectiveness in handling diverse datasets. The Decision Tree model shows the lowest mean AUC (0.7609) and the highest standard deviation (0.1291), indicating significant variability in its performance. The high standard error (0.0389) for the Decision Tree further confirms this variability.

5.1.2 Performance Metrics by Dataset

Table 5.2 summarizes the mean AUC, standard deviation, and standard error for each dataset. These metrics reflect the aggregated performance of all ML models on each dataset.

The Internet dataset has the highest mean AUC (0.9508) with a relatively low standard deviation (0.0456) and standard error (0.0132), indicating consistent performance across ML models. Similarly, the Credit Card dataset also exhibits a high mean AUC

Table 5.2: Mean AUC, Standard Deviation, and Standard Error for Each Dataset

Dataset	**Mean AUC**	**SD**	**SE**
Telcom	0.9042	0.0476	0.0137
Internet	0.9508	0.0456	0.0132
Bank	0.8242	0.0570	0.0164
Credit Card	0.9517	0.0491	0.0142
E-Commerce	0.9467	0.0557	0.0161
Employee	0.8125	0.0885	0.0256
Telco Europa	0.8942	0.0763	0.0220
Cell2Cell	0.8167	0.0587	0.0169
Membership	0.6700	0.0413	0.0119
SA Wireless Telcom	0.8000	0.0644	0.0186
Niger Telcom	0.8458	0.0799	0.0231

(0.9517), demonstrating the models' strong performance on this dataset. On the other hand, the Membership dataset has the lowest mean AUC (0.6700), suggesting that the models performed less effectively on this dataset. The Employee dataset shows the highest standard deviation (0.0885) and standard error (0.0256), indicating higher variability in model performance.

5.2 Fix and Random Effect Model Results

This section presents the results of the meta-analysis conducted using both fixed-effect and random-effect models for each machine learning model and each dataset. The forest plots generated from these analyses are discussed to provide a comprehensive understanding of the performance metrics and their variability across ML models and datasets.

5.2.1 Meta-Analysis by ML Model

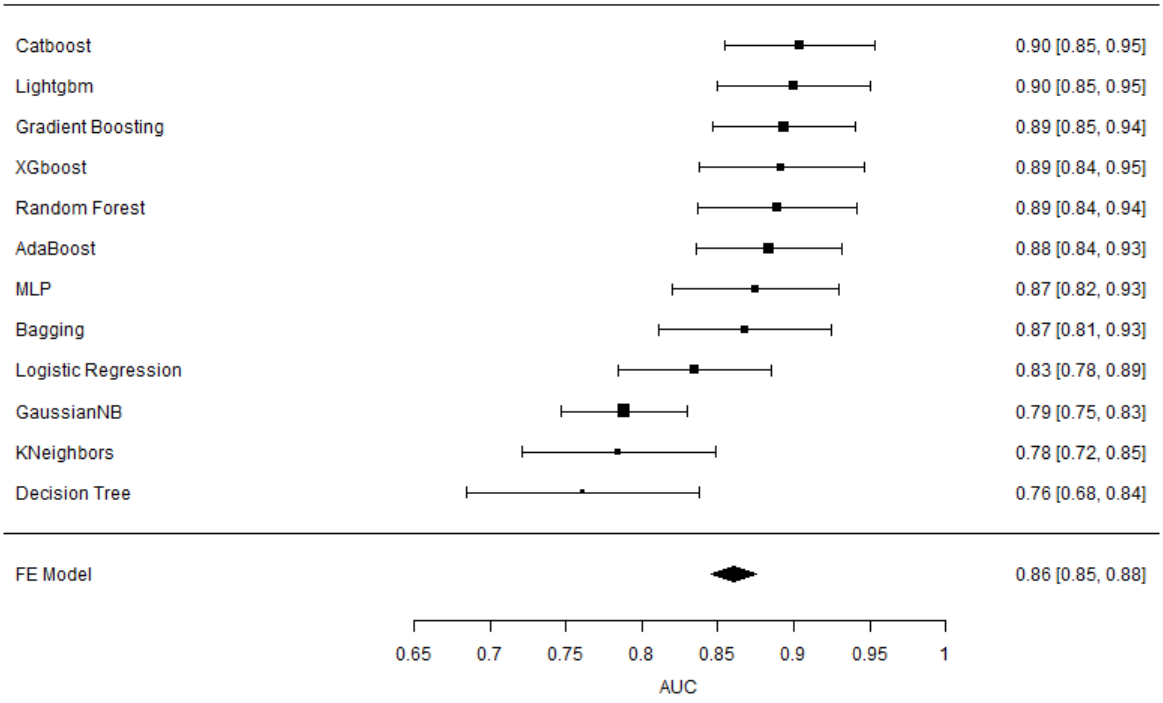

Figure 5.1: Fix Effect Model Forest Plot by ML.

The fixed-effect model forest plot (Figure 5.1) indicates that Catboost and Lightgbm have the highest mean AUCs (both around 0.90), with relatively narrow confidence intervals, suggesting consistent high performance across datasets. The Decision Tree model has the lowest mean AUC (0.76) with a wide confidence interval, indicating significant variability in performance. The overall fixed-effect model estimate for ML models is 0.86 [0.85, 0.88], representing the combined performance across all models. These results indicate a high average performance (AUC) with narrow confidence intervals, suggesting precise estimates. However, the significant heterogeneity suggests that the fixed-effect model may not fully account for the variability across studies.

The random-effect model forest plot (Figure 5.2) shows similar results, with Catboost

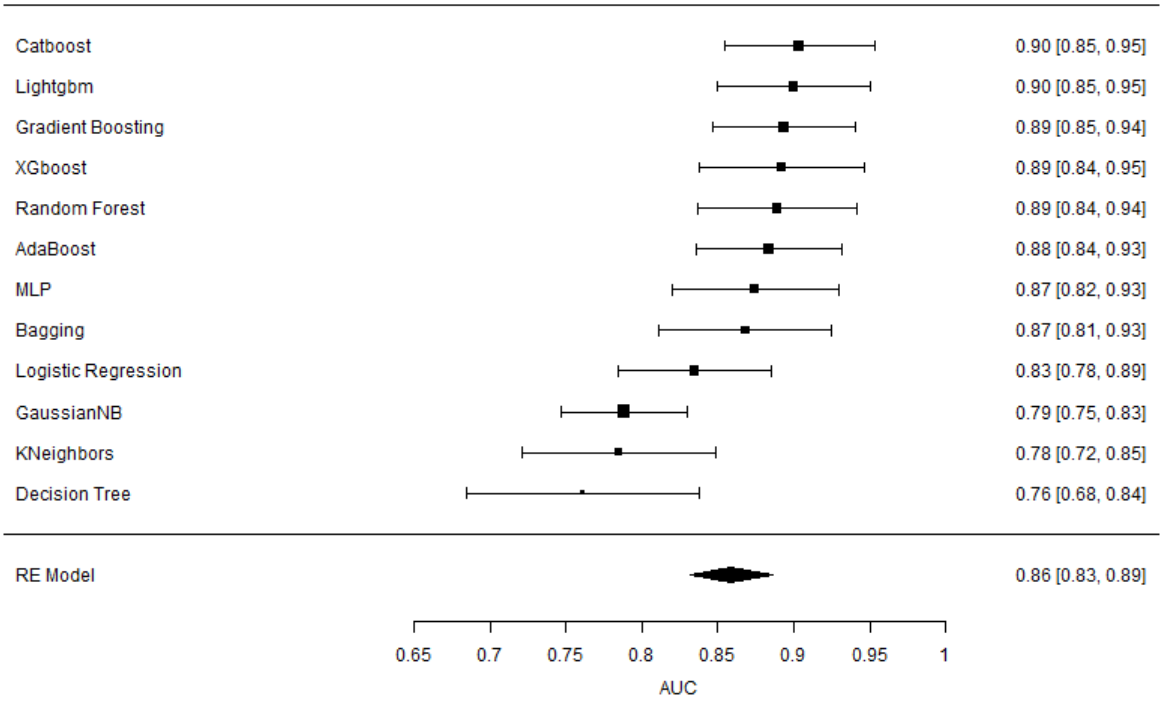

Figure 5.2: Random Effect Model Forest Plot by ML.

and Lightgbm still performing the best. The overall random-effect model estimate is 0.86 [0.83, 0.89], slightly lower than the fixed-effect model, reflecting the incorporation of between-model variability. The Decision Tree model continues to show the lowest performance with considerable variability.

5.2.2 Meta-Analysis by Dataset

The fixed-effect and random-effect models were applied to the aggregated performance data for each dataset. The forest plots for these models are shown in Figures 5.3 and 5.4

The fixed-effect model forest plot (Figure 5.3) shows that the Credit Card, Internet, and E-Commerce datasets have the highest mean AUCs, all around 0.95. These datasets

demonstrate narrow confidence intervals, indicating consistent performance across different ML models. The Membership dataset has the lowest mean AUC (0.67), with a narrower confidence interval, indicating less variability in performance but consistently lower AUC values. The overall fixed-effect model estimate is 0.85 [0.84, 0.86], representing the combined performance across all datasets.

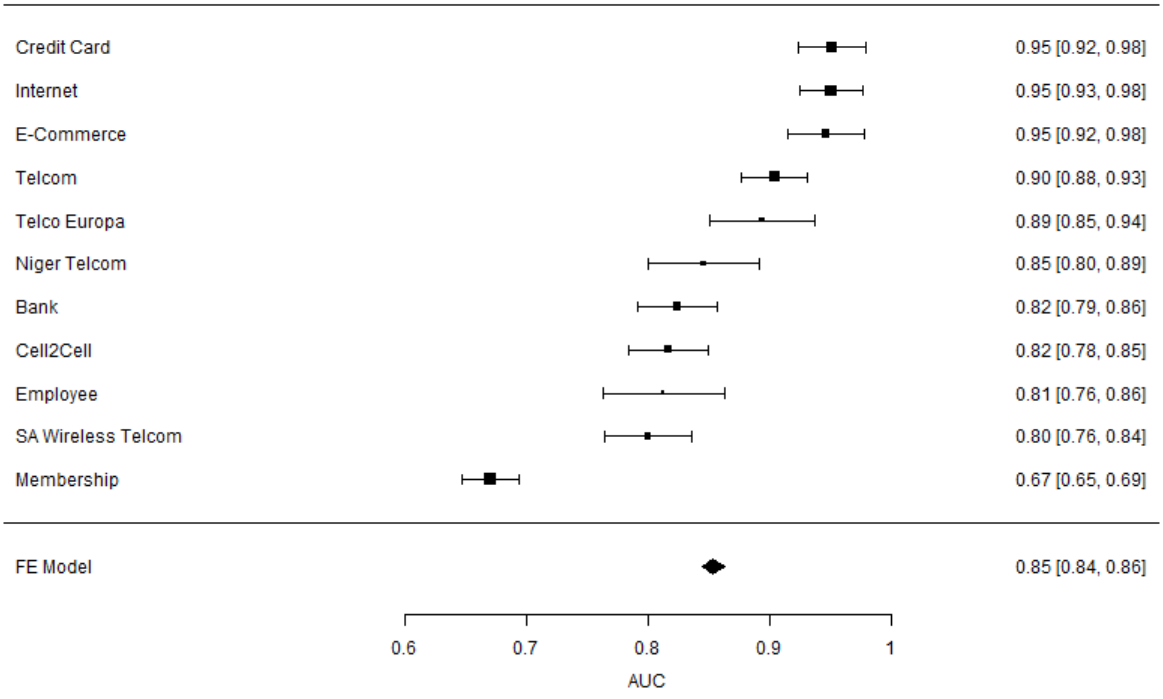

Figure 5.3: Fix Effect Model Forest Plot by Dataset.

These results indicate a high average performance (AUC) with narrow confidence intervals, suggesting precise estimates. The very high heterogeneity suggests that the fixed-effect model may not fully account for the variability across datasets.

The random-effect model forest plot (Figure 5.4) provides a similar view but accounts for between-study variability. The overall random-effect model estimate is slightly higher at 0.86 [0.81, 0.91], reflecting the incorporation of heterogeneity among datasets. The

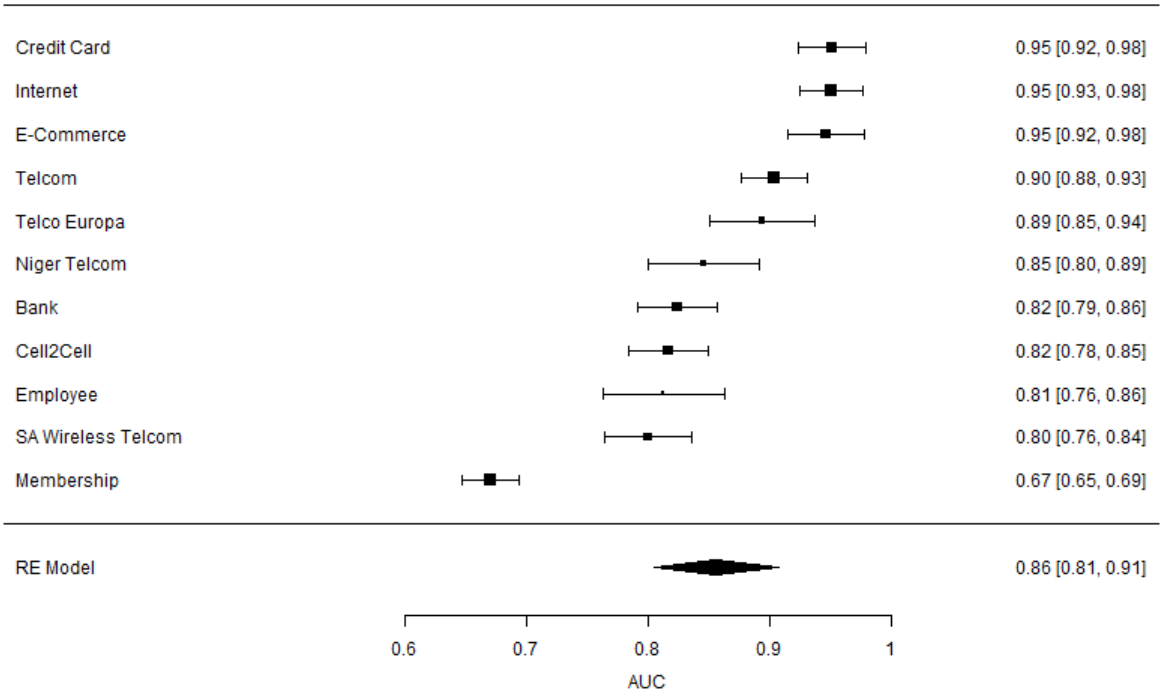

Figure 5.4: Random Effect Model Forest Plot by Dataset.

Credit Card and Internet datasets maintain high mean AUCs with slightly wider confidence intervals, indicating some variability in model performance across these datasets.

In general, the meta-analysis results highlight the variability in performance across different ML models and datasets. Among the ML models, Catboost, Lightgbm, and Gradient Boosting are the top performers, whereas the Decision Tree model shows less consistent results. The Credit Card, Internet, and E-Commerce datasets generally exhibit high AUCs, while the Membership dataset poses more challenges. These insights provide a clear understanding of how different datasets and models perform, guiding future model selection and optimization efforts.

5.3 Residual Heterogeneity Estimates

5.3.1 Residual Heterogeneity Estimates by ML Model

For the fixed-effect model analysis, which assumes a common effect size across all studies, the Q statistic was calculated to be 35.2159 with 11 degrees of freedom, resulting in a p-value of 0.0002. This significant result indicates substantial heterogeneity among the studies, leading to the rejection of the null hypothesis. The I^2 statistic, which quantifies the proportion of total variability due to heterogeneity, was found to be 68.76%, indicating a considerable level of heterogeneity. The H^2 statistic, which measures total variability relative to sampling variability, was calculated as 3.20, further suggesting notable heterogeneity.

In the random-effect model analysis, which accounts for both within-study and between-study variability, similar patterns of heterogeneity were observed. The I^2 statistic remained high at 68.57%, and the H^2 value was 3.18. Additionally, the tau-squared (Tau^2) value, representing the total amount of heterogeneity, was estimated at 0.0016 with a standard error of 0.0010, and the square root of Tau^2 (Tau) was 0.0395. The Q statistic was consistent with the fixed-effect model, reinforcing the conclusion of significant heterogeneity.

5.3.2 Residual Heterogeneity Estimates by Dataset

In the analysis by dataset using the fixed-effect model, a high degree of heterogeneity was evident. The Q statistic was exceptionally high at 408.4644 with 10 degrees of freedom, yielding a p-value of less than 0.0001, indicating significant heterogeneity among the datasets. The I^2 statistic was calculated to be 97.55%, highlighting an extremely high level of heterogeneity. The H^2 statistic was 40.85, suggesting that the observed variability greatly exceeds what would be expected due to sampling variability alone.

The random-effect model analysis produced similar findings. The I^2 statistic was slightly lower at 96.44%, still indicating substantial heterogeneity. The H^2 value was 28.13, again suggesting significant variability beyond sampling error. The Tau^2 value was estimated at 0.0072 with a standard error of 0.0033, and the square root of this value (Tau) was 0.0846. The Q statistic remained consistent with the fixed-effect model, further confirming the presence of significant heterogeneity across the datasets.

In summary, the residual heterogeneity estimates, as measured by I^2, H^2, and the Q statistic, indicate that both the ML model and dataset analyses exhibit substantial heterogeneity. This suggests that the observed variability among the studies is due to true differences in effect sizes rather than mere sampling error.

5.4 Feature Importance

The feature importance analysis was conducted using the top three performing machine learning models: Catboost, LightGBM, and Gradient Boosting, across multiple datasets. This analysis aimed to identify the key drivers of customer churn. Figures 5.5 to 5.15 illustrate the top 10 most important features as determined by each model for various datasets.

In the Catboost model, significant patterns were observed across the datasets. For instance, in the Bank dataset, features like Age, Balance, and CreditScore were identified as the most critical, emphasizing the role of financial stability and customer demographics in predicting churn. Similarly, the Credit Card dataset highlighted the importance of transaction behavior, with Total_Trans_Ct and Total_Trans_Amt emerging as top features. In the E-Commerce dataset, features such as Tenure and NumberOfAddress were crucial, suggesting that customer loyalty and geographic stability significantly influence churn. The Employee dataset's key features were Age and OverTime_No, pointing to the impact of employee demographics and work patterns on churn. For the Internet dataset, the most important features were Remaining_contract and bill_avg, underscoring the

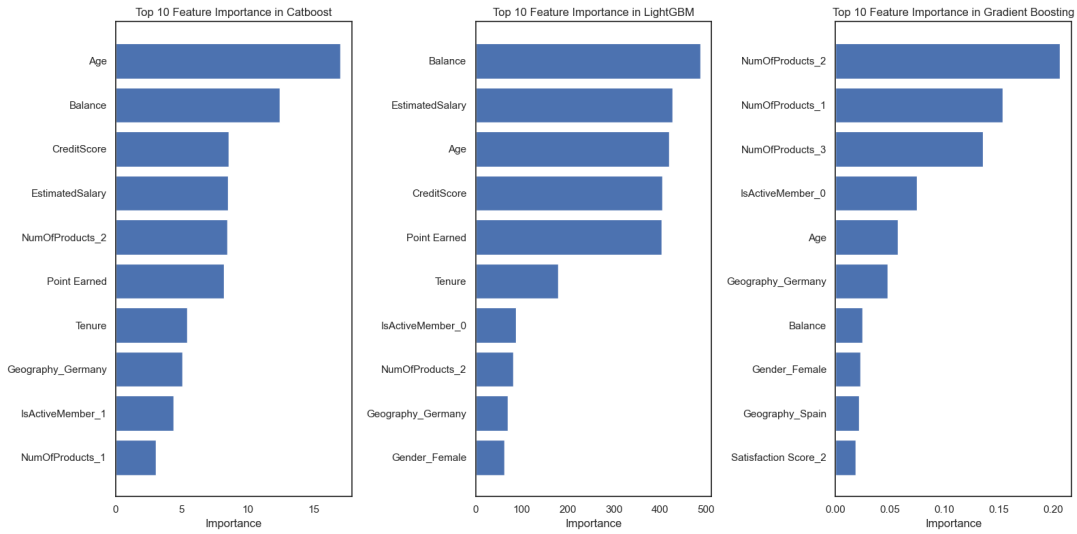

Figure 5.5: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Bank Dataset.

significance of contract terms and billing amounts in customer retention. In the Membership dataset, the analysis revealed that features like MEMBERSHIP_TERM_YEARS, ANNUAL_FEES, and MEMBER_ANNUAL_INCOME were the top predictors, suggesting that financial stability and membership tenure are key factors influencing churn. The Nigeria Telecom dataset emphasized the importance of usage and spending patterns, with Total Spend in Months 1 and 2 of 2017 and network_age emerging as significant features. The SA Wireless dataset showed that Aggregate_Total_Rev, Aggregate_SMS_Rev, and network_age were critical in predicting churn, pointing to the role of revenue and usage behavior. Other datasets, such as Telco Europa and Telecom, consistently showed that Tenure and ContractMonthtoMonth were significant, indicating the importance of contract types and customer retention over time.

In the LightGBM model, financial and usage-related features were consistently important across different datasets. For example, in the Bank dataset, Balance and EstimatedSalary emerged as top predictors, highlighting financial factors' influence on churn. The Credit Card dataset identified Total_Trans_Amt and Total_Amt_Chng_Q4_Q1 as leading indicators, pointing toward the significance of transaction changes and amounts in churn prediction. The E-Commerce dataset emphasized the role of incentives and logistics, with

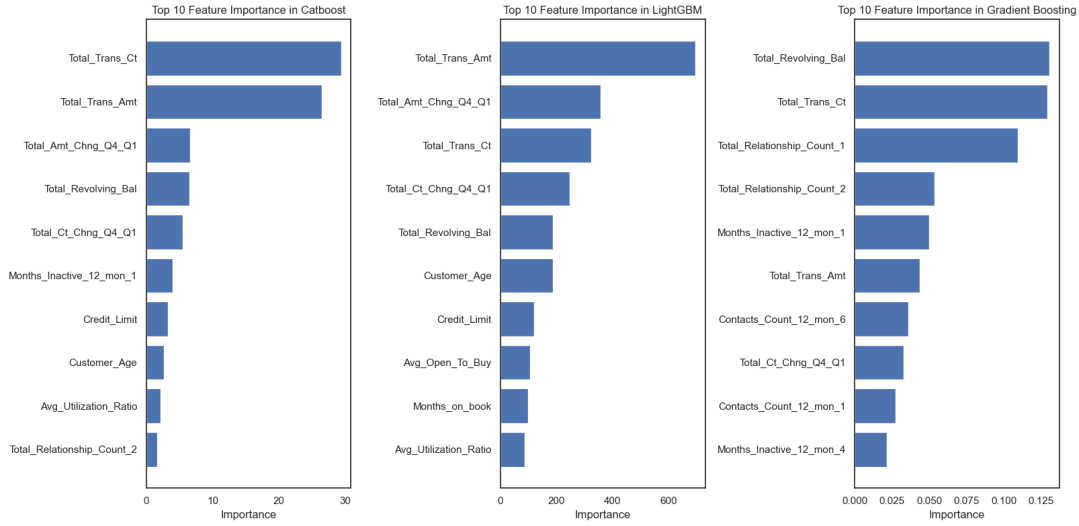

Figure 5.6: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Credit Card Dataset.

CashbackAmount and WarehouseToHome identified as top features. In the Employee dataset, financial compensation, represented by features such as DailyRate and MonthlyIncome, was crucial in predicting employee churn. For the Internet dataset, billing and subscription duration were significant, with Bill_avg and subscription_age identified as the most important features. In the Cell2Cell dataset, changem and mou emerged as top predictors, highlighting the significance of customer behavior and usage patterns. The Membership dataset identified ANNUAL_FEES and MEMBER_ANNUAL_INCOME as leading indicators, suggesting that financial aspects are crucial in predicting membership churn. In the Nigeria Telecom dataset, features such as Total Onnet spend and Total Data Consumption were identified as significant, pointing towards the importance of usage patterns in churn prediction. The SA Wireless dataset highlighted the role of aggregate revenue and data usage, with Aggregate_SMS_Rev and network_age emerging as top features. In both Telco Europa and Telecom datasets, Monthly Charge and Age were the top features, suggesting that billing and customer demographics are pivotal in churn prediction.

The Gradient Boosting model also revealed critical insights into the drivers of churn across various datasets. In the Bank dataset, the focus was on product usage, with Nu-

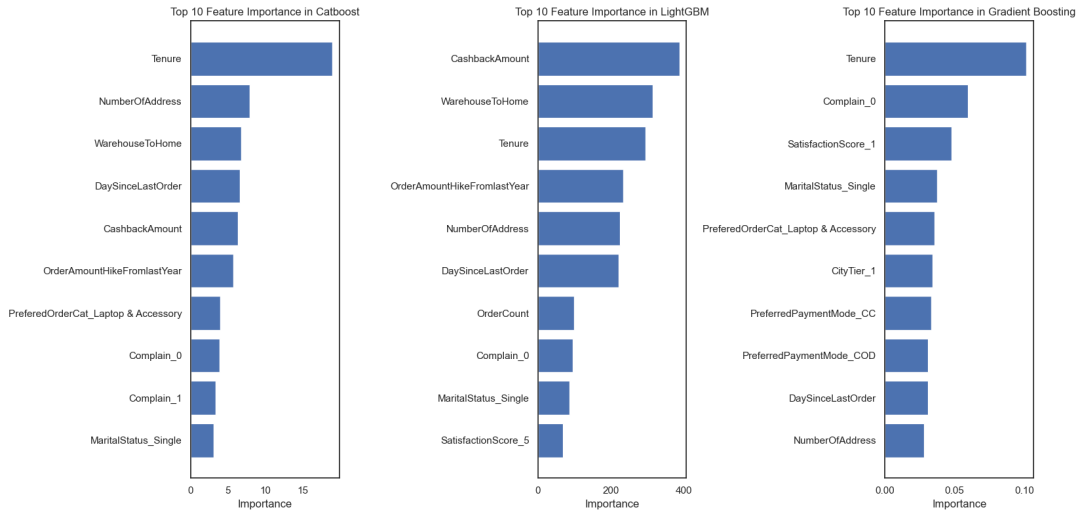

Figure 5.7: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for E-Commerce Dataset.

mOfProducts_2 and NumOfProducts_1 emerging as the most important features. The Credit Card dataset highlighted the significance of revolving balances and transaction counts, with Total_Revolving_Bal and Total_Trans_Ct identified as top features. In the E-Commerce dataset, customer loyalty and complaints were significant predictors, with Tenure and Complain_0 as key features. The Employee dataset emphasized job roles and work-life balance, with JobRole_Manufacturing Director and WorkLifeBalance_1 identified as critical features. The Internet dataset highlighted the importance of contract terms and usage patterns, with remaining_contract and download_avg emerging as top features. In the Cell2Cell dataset, the focus was on retention efforts, with retcalls_0 and retcalls_1 emerging as the most important features, indicating the importance of retention calls in predicting churn. The Membership dataset emphasized the importance of package types and occupation codes, with MEMBERSHIP_PACKAGE_TYPE-A and MEMBER_OCCUPATION_CD_2.0 identified as key features. In the Nigeria Telecom dataset, the analysis highlighted the significance of competitor networks, with Most Loved Competitor network in Month 2_Weematel and Month 2_Zintel emerging as top predictors. The SA Wireless dataset showed the importance of aggregate revenue and favorite network features, with sep_fav_a_ufone and sep_fav_a_telenor identified as critical. Similar patterns were observed in the Telco Europa and Telecom datasets, where

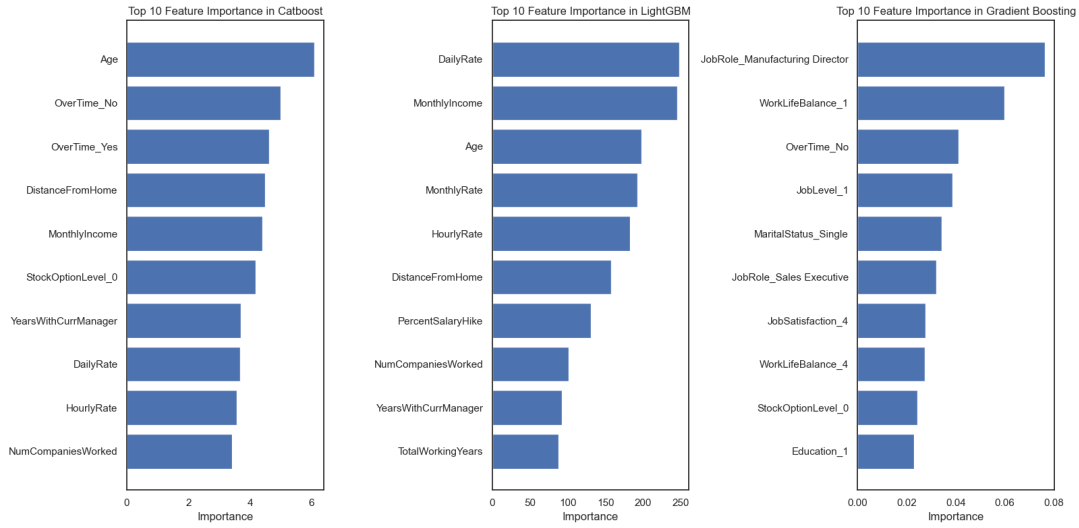

Figure 5.8: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Employee Dataset.

Contract_Month-to-Month and Number of Dependents were crucial in predicting churn.

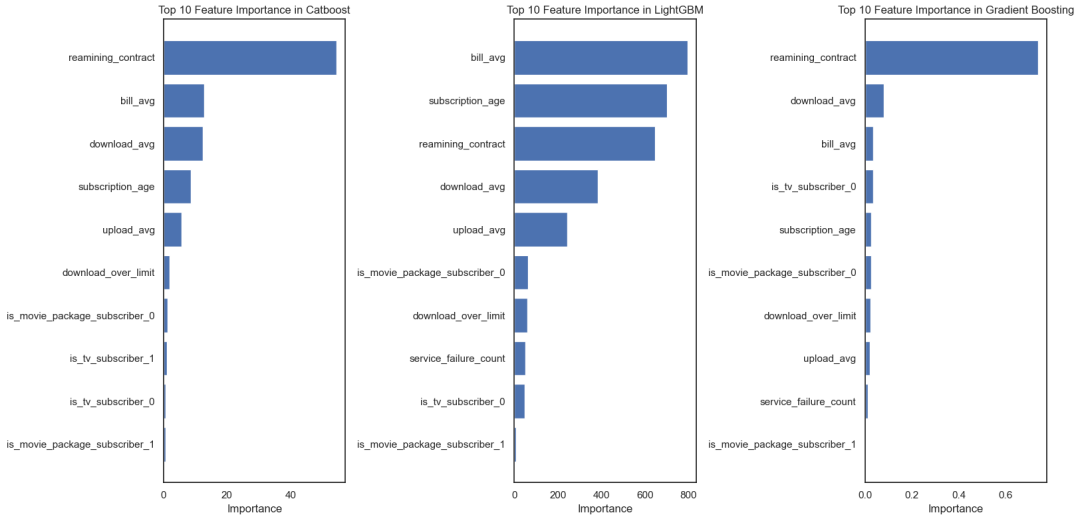

Figure 5.9: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Internet Dataset.

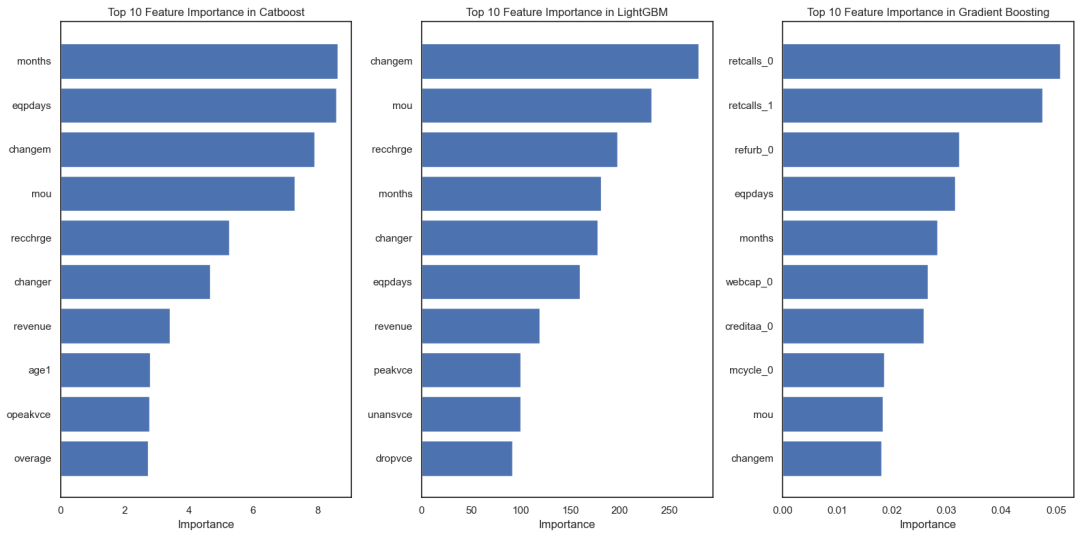

Figure 5.10: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Cell2Cell dataset.

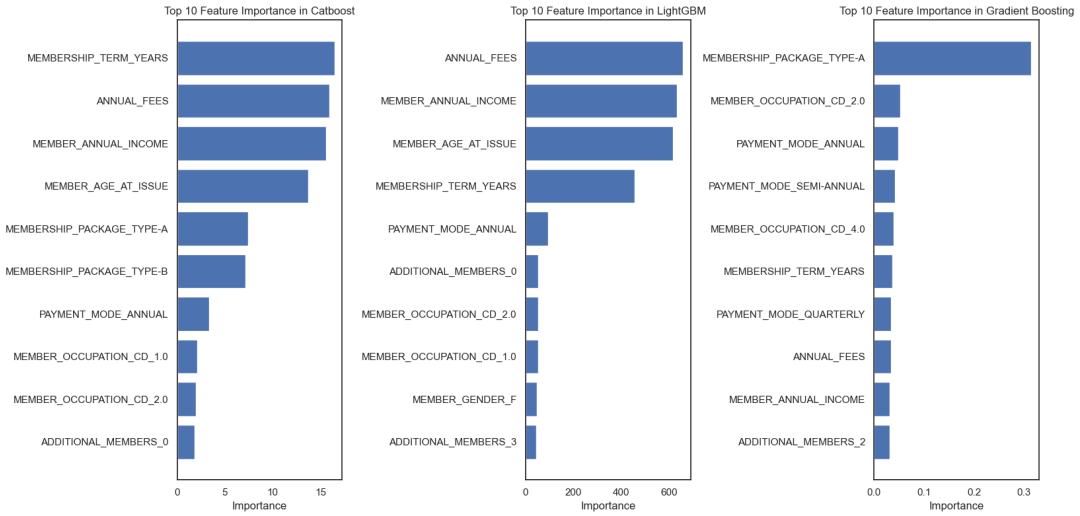

Figure 5.11: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Membership dataset.

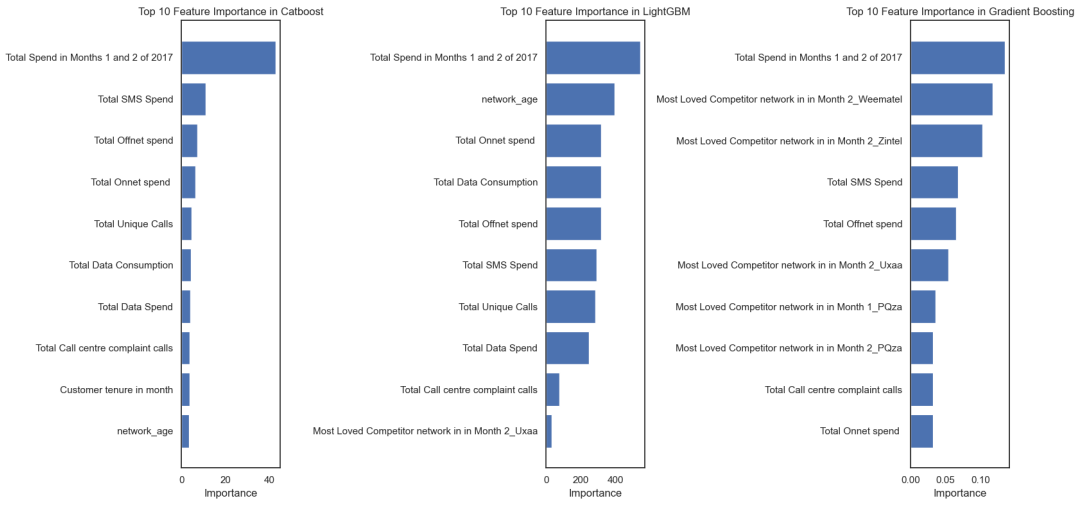

Figure 5.12: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the Nigeria Telecom dataset.

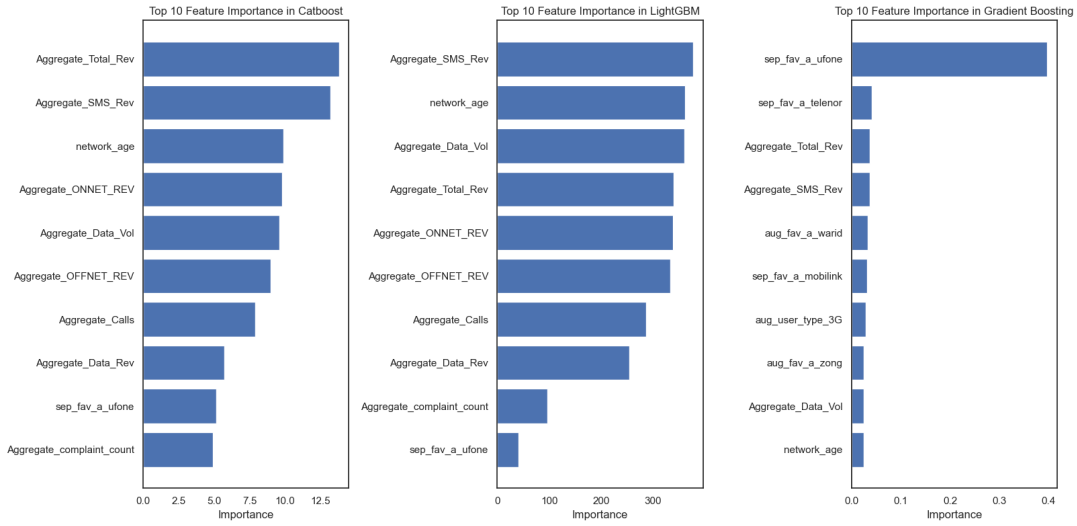

Figure 5.13: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting models for the SA Wireless dataset.

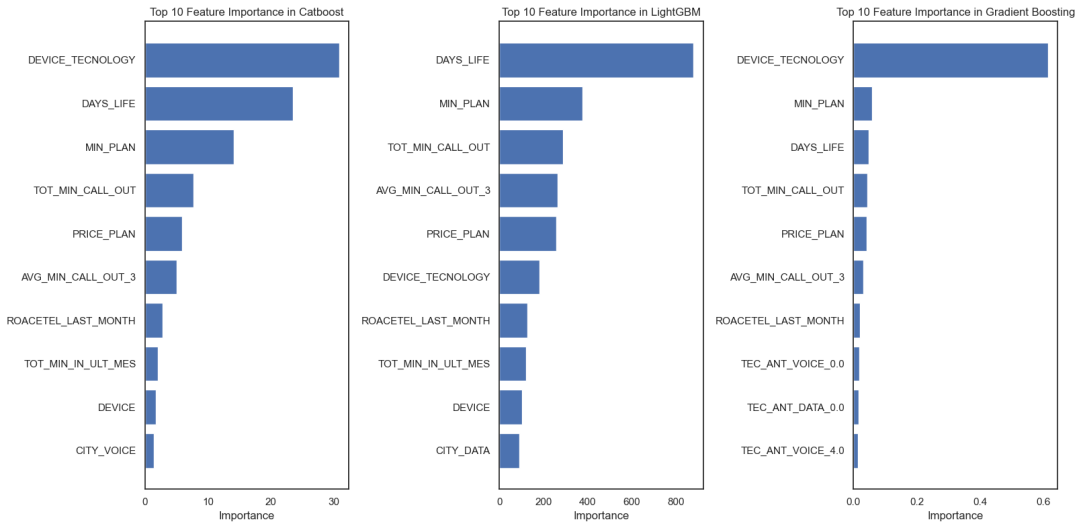

Figure 5.14: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Telco Europa Dataset.

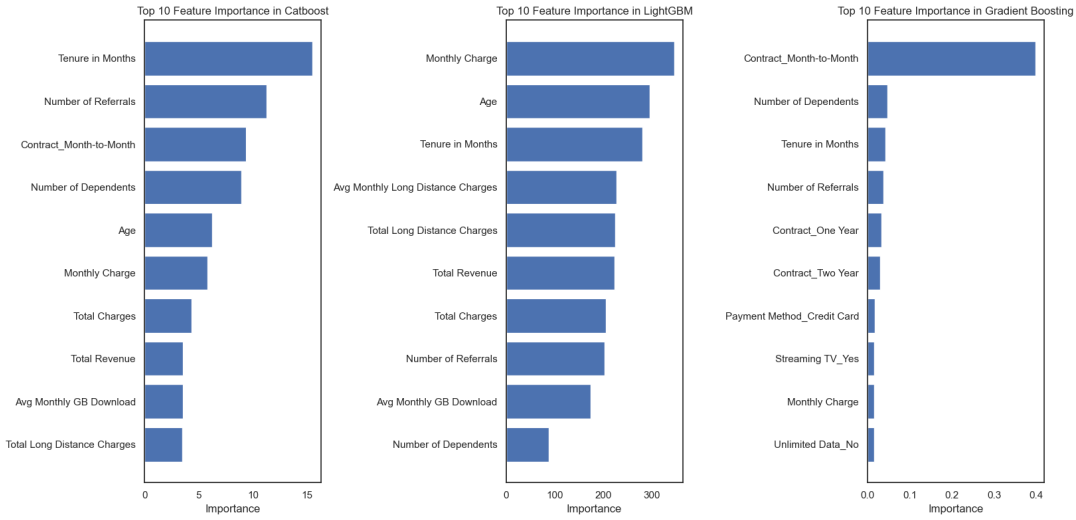

Figure 5.15: Top 10 Feature Importance in Catboost, LightGBM, and Gradient Boosting for Telecom Dataset.

Chapter 6

Discussion and Conclusion

6.1 Discussion

The results of this study provide a comprehensive understanding of the performance of various machine learning models in predicting customer churn across multiple datasets. By adopting an individual participant data meta-analysis approach, this research has been able to integrate raw data, facilitating a nuanced comparison and validation of predictive models under varying conditions.

6.1.1 Variability in Model Performance

The aggregated results reveal significant variability in the performance of machine learning models across different datasets and models. The fixed-effect model's high heterogeneity (I^2 of 68.76% by ML model and 97.55% by dataset) indicates substantial variability not accounted for by this model. This high heterogeneity suggests that the differences in effect sizes are not solely due to sampling error but also due to genuine differences in dataset characteristics or model capabilities.

The standard deviations and standard errors across different models and datasets also highlight this variability. For instance, the Employee dataset exhibited the highest standard deviation (0.0885) and standard error (0.0256), indicating significant variability in model performance. This suggests that some datasets are inherently more challenging for churn prediction, likely due to their unique characteristics or data quality issues.

6.1.2 Top Performing Models and Datasets

Among the machine learning models, Catboost, Lightgbm, and Gradient Boosting consistently demonstrated high performance, with mean AUCs close to or above 0.90. These models showed relatively narrow confidence intervals, indicating consistent performance across various datasets. Their high performance can be attributed to their sophisticated ensemble techniques, which combine multiple weak learners to create a strong predictive model. These methods are particularly effective in handling complex relationships within the data and mitigating overfitting.

Conversely, the Decision Tree model exhibited the lowest mean AUC and the highest variability, suggesting its limited effectiveness in handling diverse datasets compared to more sophisticated ensemble methods. The inherent simplicity of the Decision Tree algorithm, while offering interpretability, also leads to higher susceptibility to overfitting and poor generalization on unseen data.

In terms of datasets, the Internet and Credit Card datasets achieved the highest mean AUCs, around 0.95, with low standard deviations and standard errors. This consistency indicates that the models performed well on these datasets, likely due to their inherent characteristics that align well with the models' capabilities. Factors such as well-defined features, balanced target variables, and higher data quality might contribute to the superior performance observed on these datasets.

On the other hand, the Membership dataset posed the most significant challenge, with the lowest mean AUC and higher variability, indicating that the models struggled

to effectively predict churn in this dataset. This could be due to several reasons, such as the presence of more noise, fewer relevant features, or a higher degree of class imbalance that complicates the learning process.

6.1.3 Fixed-Effect vs. Random-Effect Models

The comparison between fixed-effect and random-effect models provides further insights. The fixed-effect model, which assumes a common effect size across studies, showed high precision but failed to account for the observed heterogeneity adequately. This model is useful when the studies are assumed to be functionally identical and the only variability is due to within-study sampling error. However, in real-world applications, this assumption is often unrealistic due to genuine differences in datasets and study conditions.

In contrast, the random-effect model, which incorporates between-study variability, offered slightly lower mean AUC estimates with wider confidence intervals but provided a more realistic assessment by acknowledging the variability across datasets and models. This model is more appropriate when the datasets are not identical and there is a need to generalize the findings beyond the included studies.

The random-effect model's higher pooled effect size estimate for the dataset analysis (0.86 [0.81, 0.91]) compared to the fixed-effect model (0.85 [0.84, 0.86]) underscores the importance of considering heterogeneity in meta-analyses. This approach offers a more generalizable understanding of model performance, essential for practical applications in varied real-world scenarios.

6.1.4 Feature Importance

The feature importance analysis across various datasets using Catboost, LightGBM, and Gradient Boosting revealed consistent patterns and highlighted specific variables that are critical in predicting customer churn.

Demographic Factors: Features such as Age, Tenure, and Membership_Term_Years appeared consistently across multiple datasets, underscoring the importance of demographic factors in predicting churn. This suggests that understanding the age distribution and customer loyalty can provide valuable insights into customer retention strategies.

Financial Indicators: Financial-related features, including Balance, Estimated-Salary, and Annual_Fees, were significant across various datasets. These findings highlight the role of financial stability in customer decision-making, where customers with higher financial stability may be less likely to churn. This indicates the importance of considering customers' financial health when designing retention strategies.

Usage Patterns: Usage-related features, such as Total_Trans_Ct, Total_Revolving_Bal, and Total Spend in Months 1 and 2 of 2017, were critical indicators of churn. These results suggest that customers who engage more frequently with the service or product are less likely to churn. Monitoring these patterns can help identify at-risk customers and design interventions to improve customer engagement and retention.

Contractual and Behavioral Factors: Features related to contracts and customer behavior changes, such as Contract_Month-to-Month and changes in service usage (e.g., changer and changem), were vital in predicting churn. These factors highlight the importance of flexible contract terms and monitoring customer behavior to reduce churn rates. Companies should consider offering flexible contract options and closely monitoring customer behavior to preemptively address potential churn.

Incentives and Support: Features like CashbackAmount and Retention Calls (retcalls) pointed to the significance of customer incentives and support in retention strategies. This suggests that companies should implement more personalized incentives and enhance customer support to improve retention. By focusing on these factors, businesses can better align their offerings with customer needs and preferences, thereby reducing churn.

Overall, this analysis provides valuable insights into the factors driving customer

churn across different industries and datasets. It also demonstrates the robustness of the top-performing models (Catboost, LightGBM, and Gradient Boosting) in identifying these critical features. These findings can be used to refine customer retention strategies, focusing on the key drivers of churn identified in this study.

6.1.5 Implications for Practice

The insights gained from this meta-analysis have several practical implications. Firstly, the consistent high performance of ensemble methods like Catboost, Lightgbm, and Gradient Boosting suggests that these models should be prioritized in customer churn prediction tasks. Their ability to handle complex data structures and mitigate overfitting makes them well-suited for diverse datasets encountered in real-world applications.

Secondly, the significant variability in model performance across different datasets highlights the need for dataset-specific strategies. For instance, datasets like Membership, which pose more challenges, may benefit from additional preprocessing steps such as feature engineering, advanced imputation techniques for missing values, and more sophisticated methods for handling class imbalance.

Furthermore, the results underscore the importance of using random-effect models in meta-analyses involving heterogeneous datasets. By accounting for between-study variability, these models provide a more accurate and generalizable estimate of effect sizes, which is crucial for informing practical decisions and ensuring the robustness of predictive models in diverse settings.

6.1.6 Contributions of This Research

This research makes several significant contributions to the field of machine learning and customer churn prediction:

- **Methodological Advancement**: By employing an IPD-MA approach, this study integrates raw data from multiple datasets, allowing for a more nuanced and robust comparison of machine learning models. This methodological advancement can be applied to other domains where meta-analysis of machine learning performance is required.

- **Comprehensive Evaluation**: The study provides a thorough evaluation of multiple machine learning models across a diverse set of datasets, highlighting the strengths and weaknesses of each model. This comprehensive evaluation aids in selecting the most suitable models for specific types of data and prediction tasks.

- **Handling Heterogeneity**: The research demonstrates the importance of accounting for heterogeneity in meta-analyses by comparing fixed-effect and random-effect models. This contribution is crucial for developing more reliable and generalizable predictive models in varied real-world scenarios.

- **Practical Insights**: The findings offer practical insights for practitioners in selecting and optimizing machine learning models for customer churn prediction. The detailed analysis of model performance across different datasets provides a valuable reference for tackling similar prediction tasks in industry.

- **Future Research Directions**: The study identifies key areas for future research, such as exploring additional datasets, refining models, and incorporating more advanced machine learning techniques. These directions can guide further advancements in the field and improve predictive accuracy and generalizability.

6.1.7 Limitations and Future Research

While this study provides valuable insights, it also has several limitations. The reliance on publicly available datasets means that the findings may not fully generalize to proprietary datasets with different characteristics. Additionally, the study focused on a limited set

of machine learning models and did not explore the full spectrum of possible algorithms and their hyperparameters.

Future research should aim to include a broader range of datasets, especially those from different industries and contexts, to validate the findings further. Additionally, exploring more advanced and hybrid machine learning models, as well as automated machine learning (AutoML) techniques, could yield even better performance and more robust insights.

6.2 Conclusion

This study has demonstrated the effectiveness of the IPD-MA approach in evaluating the performance of machine learning models for predicting customer churn across multiple datasets. The findings highlight the importance of considering dataset characteristics and model capabilities in such analyses. Key takeaways from this research include:

- **High-Performing Models**: Catboost, Lightgbm, and Gradient Boosting emerged as the top-performing models, consistently achieving high AUCs across various datasets. These models should be prioritized for churn prediction tasks in future applications.
- **Dataset Challenges**: The variability in model performance across datasets underscores the need for tailored approaches. The Internet and Credit Card datasets demonstrated high model performance, whereas the Membership dataset posed significant challenges, highlighting the necessity for dataset-specific strategies.
- **Meta-Analysis Models**: The comparison between fixed-effect and random-effect models emphasized the importance of accounting for heterogeneity in meta-analyses. The random-effect model provided a more comprehensive understanding of model performance by incorporating between-study variability.

Overall, this research contributes to the field by providing a robust methodological framework and actionable insights for improving churn prediction models. Future work should focus on exploring additional datasets and refining models to further enhance predictive accuracy and generalizability. By leveraging the findings of this study, practitioners can make informed decisions in selecting and optimizing machine learning models for effective customer churn prediction.

Bibliography

Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, *6*(1), 1–24.

Alshdaifat, E., Alshdaifat, D., Alsarhan, A., Hussein, F., & El-Salhi, S. M. F. S. (2021). The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance. *Data*, *6*(2), 11.

Asbai, N., & Amrouche, A. (2017). Boosting scores fusion approach using front-end diversity and adaboost algorithm, for speaker verification. *Computers & Electrical Engineering*, *62*, 648–662.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Bryan, S. (2020). Weighting confusion matrices by outcomes and observations.

Bühlmann, P., & Yu, B. (2002). Analyzing bagging. *The annals of Statistics*, *30*(4), 927–961.

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled clinical trials*, *7*(3), 177–188.

Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine learning*, *54*, 255–273.

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing & Management*, *57*(1), 102121.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, *4*(Nov), 933–969.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. *icml*, *96*, 148–156.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337–407.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232.

García, D. L., Nebot, À., & Vellido, A. (2017). Intelligent data analysis approaches to churn as a business problem: A survey. *Knowledge and Information Systems*, *51*(3), 719–774.

Geiler, L., Affeldt, S., & Nadif, M. (2022). A survey on machine learning methods for churn prediction. *International Journal of Data Science and Analytics*, *14*(3), 217–242.

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow.* " O'Reilly Media, Inc."

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational researcher*, *5*(10), 3–8.

Harrer, M., Cuijpers, P., Furukawa, T., & Ebert, D. (2021). *Doing meta-analysis with r: A hands-on guide*. Chapman; Hall/CRC.

Haumahu, J., Permana, S., & Yaddarabullah, Y. (2021). Fake news classification for indonesian news using extreme gradient boosting (xgboost). *IOP Conference Series: Materials Science and Engineering*, *1098*(5), 052081.

Joseph, V. R., & Vakayil, A. (2022). Split: An optimal method for data splitting. *Technometrics*, *64*(2), 166–176.

Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2022). Customer churn prediction system: A machine learning approach. *Computing*, 1–24.

Latha, C. B. C., & Jeeva, S. C. (2019). Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, *16*, 100203.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Qureshi, S. A., Rehman, A. S., Qamar, A. M., Kamal, A., & Rehman, A. (2013). Telecommunication subscribers' churn prediction model using machine learning. *Eighth international conference on digital information management (ICDIM 2013)*, 131–136.

Rahman, M., & Kumar, V. (2020). Machine learning based customer churn prediction in banking. *2020 4th international conference on electronics, communication and aerospace technology (ICECA)*, 1196–1201.

Riley, R. D., Lambert, P. C., & Abo-Zaid, G. (2010). Meta-analysis of individual participant data: Rationale, conduct, and reporting. *Bmj*, *340*.

Riley, R. D., Tierney, J. F., & Stewart, L. A. (2021). *Individual participant data meta-analysis: A handbook for healthcare research*. John Wiley & Sons.

Seger, C. (2018). An investigation of categorical variable encoding techniques in machine learning: Binary versus one-hot and feature hashing.

Shabtai, A., Elovici, Y., Rokach, L., Shabtai, A., Elovici, Y., & Rokach, L. (2012). *Data leakage detection/prevention solutions*. Springer.

Simmonds, M. C., Higginsa, J. P., Stewartb, L. A., Tierneyb, J. F., Clarke, M. J., & Thompson, S. G. (2005). Meta-analysis of individual patient data from randomized trials: A review of methods used in practice. *Clinical Trials*, *2*(3), 209–217.

Sisodia, D. S., Vishwakarma, S., & Pujahari, A. (2017). Evaluation of machine learning models for employee churn prediction. *2017 international conference on inventive computing and informatics (icici)*, 1016–1020.

Sun, B., Chen, S., Wang, J., & Chen, H. (2016). A robust multi-class adaboost algorithm for mislabeled noisy data. *Knowledge-Based Systems*, *102*, 87–102.

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: Analysis of machine learning techniques

for churn prediction and factor identification in telecom sector. *IEEE access*, *7*, 60134–60149.

Witten, D., & James, G. (2013). *An introduction to statistical learning with applications in r*. springer publication.

Xiong, Y., Ye, M., & Wu, C. (2021). Cancer classification with a cost-sensitive naive bayes stacking ensemble. *Computational and Mathematical Methods in Medicine*, *2021*.

Appendix A

Source Codes

The data and code for this project is stored in the GitHub Repository below,

https://github.com/khanhgeo/DASC6910-GraduateProject/tree/main