Amsterdam University of Applied Sciences

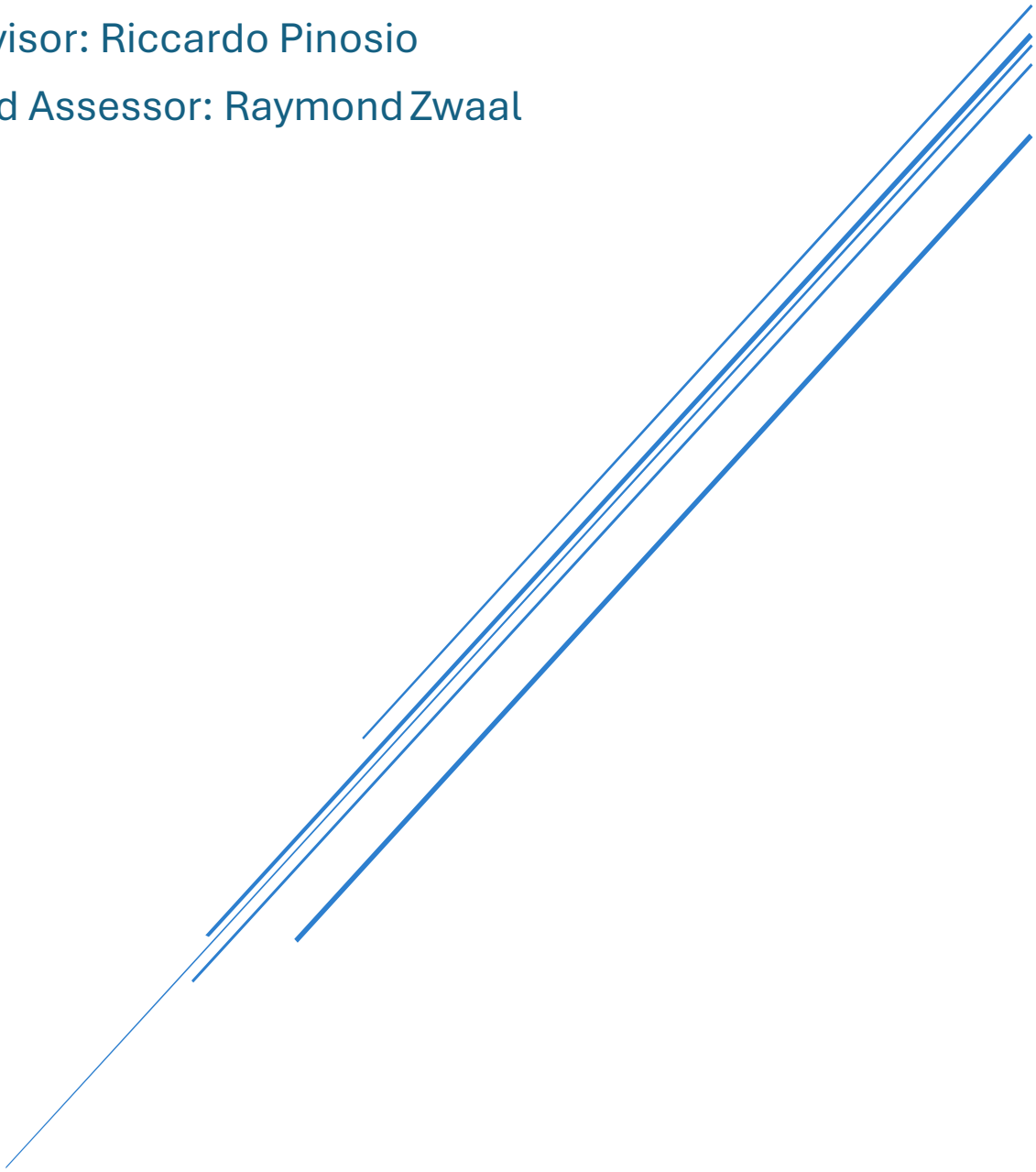
COUNTERFACTUAL EXPLANATIONS FOR MODEL DISTILLATION

Dide Uzun

500923796

Supervisor: Riccardo Pinosio

Second Assessor: Raymond Zwaal



Amsterdam University of Applied Sciences
Master Digital Driven Business/ FinTech Track

Personal Information

Name: Dide Uzun
Student Number: 500923796
Phone Number: +31 6 85887264
Email: dide.uzun@hva.nl

Education Information

School: Amsterdam University of Applied Sciences
Department: Master Digital Driven Business/ FinTech Track
Address: Fraijlemaborg 133, 1102 CV, Amsterdam
Supervisor: Riccardo Pinosio
Second Assessor: Raymond Zwaal
Period: March 2024 – June 2024

Table of Contents

1. Introduction

Machine learning has been recognized as a critically important field for many years (Jordan, 2015), aiding organizations and businesses to make smarter choices. Thanks to advances in processors, smarter learning algorithms, and easier access to large datasets (Linardatos, Papastefanopoulos, & Kotsiantis, 2020) the development and deployment of complex models have become more efficient and widespread. Machine learning techniques are now essential to a wide array of everyday applications (Pawelczyk, 2021) and the increasing reliance on these complex models in high-stake sectors, such as finance and healthcare (Verma, 2020), emphasizes the importance of a deeper understanding of model behavior (Watson, 2020).

Even though machine learning models have a lot of potential to improve decision-making processes, their predictions can profoundly affect companies' and individuals' lives, potentially leading to outcomes that are unfavourable or unfair for the end users (Pawelczyk, 2021). In the lending sector, companies utilize machine learning algorithms to evaluate the financial profiles of potential customers, specifically analyzing metrics such as income, credit score, and the probability of default. These factors play a crucial role in determining whether to approve or reject a loan application. This decision-making process significantly affects the financial stability of the lending institution and the economic well-being of the customer. Additionally, these lending companies are responsible for providing explanations for their final decisions. Therefore, clarifying the link between the input and the outcome of machine learning models, in a manner that is comprehensible to humans, is crucial for creating reliable and trustworthy systems (Verma, 2020).

The challenge of explainability (Stepin, 2021) in machine learning has been a subject of considerable discussion and an important quality measurement for its systems. This issue arises as enhanced predictive capabilities are typically achieved through increased model complexity, consequently reducing their understandability and explainability. Numerous strategies have been explored in the literature to mitigate this shortfall in explainability, occasionally at the cost of trading interpretability for accuracy (Guidotti R. M., 2018). Despite these efforts, most explanations provided only explain the foundation behind specific outcomes and often fail to offer sufficient insight into how to influence those results (Guidotti R. , 2022).

Counterfactual explanations are one of the insightful tools in machine learning, employed to interpret a model's behavior through the supply of alternative scenarios (Guidotti R. , 2022). These explanations demonstrate how minor adjustments to the input can yield opposing outcomes, being particularly beneficial for interpreting decisions rendered by complex models. Counterfactual explanations clarify the basis of decisions by specifying which changes in the input variables could lead to different outcomes (Guidotti R. , 2022). This method enables entities, such as lending companies, to provide detailed justifications for their final decisions. Essentially, they generate "what-if" scenarios that enhance human understanding and transparency.

1.1. Problem Statement and Research Question

The counterfactual method has a compelling feature where it provides a set of observations that are near the model's decision boundary, suggesting the feasibility of employing counterfactual explanations in model distillation. Model distillation, also referred to as knowledge distillation, is a technique in machine learning where a smaller, simpler model (the "student" model) is trained to mimic the behavior of a larger, more complex model (the "teacher" model) (Gou, 2021). In other words, model distillation is a technique where simpler models are trained to replicate the predictions and knowledge of more complex models (Tan, 2018). This leads to the main research question of this paper.

Main Research Question:

How can counterfactual explanations be effectively utilized in model distillation for classification models to enhance both interpretability and accuracy?

In recent years, the field of machine learning has witnessed significant advancements, yet the complexity and opacity of many high-performing models, such as random forest, pose substantial challenges in terms of interpretability and transparency. Model distillation has emerged as a promising technique to address these challenges, allowing complex models to transfer knowledge to simpler, more interpretable models without significant loss of accuracy. However, the integration of counterfactual explanations into this process remains underexplored. Counterfactual explanations offer a unique approach to understanding model behavior by illustrating how changes in input data can alter predictions. This research seeks to fill this gap by investigating how counterfactual explanations can be harnessed within model distillation to not only preserve but

potentially enhance the interpretability and accuracy of classification models. Through this exploration, the study aims to provide a novel framework that leverages the strengths of both counterfactual explanations and model distillation, contributing to the development of more transparent and reliable machine learning models.

1.2. Research Objectives

To address the main research question, this study is guided by several key objectives that span the development, implementation, and evaluation of counterfactual explanations within the framework of model distillation. Each objective is designed to systematically explore the potential of counterfactual explanations to enhance the interpretability and accuracy of classification models.

Investigating Adaptive or Dynamic Methods for Generating Counterfactual Explanations: The first objective involves a comprehensive investigation into adaptive or dynamic methods for generating counterfactual explanations. This involves identifying adaptive techniques that can efficiently produce relevant counterfactuals in varying contexts. The aim is to identify methods that can generate meaningful counterfactuals in real-time, adjusting to changes in the input data or model parameters. By achieving this, the research seeks to ensure that counterfactual explanations are not only accurate but also relevant and applicable across different scenarios, enhancing their utility in practical applications.

Selecting Appropriate Methods and Generating Counterfactual Explanations for Model Distillation: This objective focuses on the careful selection and generation of counterfactual explanations that can be effectively used in the model distillation process. It involves evaluating different counterfactual generation methods to determine their suitability for various types of models and classification tasks. The goal is to identify those methods that best preserve the predictive accuracy of the original complex models while improving interpretability.

Investigating the Distribution of Counterfactual Explanations Concerning the Decision Boundary of a Machine Learning Classifier: Understanding the relationship between counterfactual explanations and the decision boundaries of machine learning classifiers is crucial for enhancing model interpretability. This objective aims to map and analyze how counterfactuals are distributed in relation to the decision boundaries. By doing so, the research seeks to uncover patterns and insights into the decision-making processes of classifiers. This analysis will help identify areas where the model's predictions might be less certain or more prone to errors, providing valuable information for improving model reliability and trustworthiness.

Employing Counterfactual Explanations as Synthetic Datasets for Model Distillation: The final objective is to employ counterfactual explanations to create synthetic datasets that can be used in the model distillation process. This innovative approach aims to leverage the explanatory power of counterfactuals to enhance the interpretability of distilled models. By generating synthetic datasets composed of counterfactual examples, the research seeks to produce distilled models that not only replicate the performance of their complex counterparts but also offer clearer and more understandable predictions. This approach has the potential to transform decision-making scenarios by providing models that stakeholders can trust and comprehend, ultimately bridging the gap between high performance and interpretability in machine learning.

Together, these objectives aim to create a comprehensive framework for integrating counterfactual explanations into model distillation, thereby advancing the state of the art in machine learning interpretability and accuracy. Through systematic exploration and rigorous evaluation, this research endeavors to contribute significantly to the development of more transparent, reliable, and effective classification models.

1.3. Sub-Questions

To comprehensively explore the main research question and achieve the outlined objectives, this study addresses several sub-research questions that delve into specific aspects of counterfactual explanations and their application in model distillation.

1. *What are the state-of-the-art computational frameworks for generating counterfactual explanations?*
2. *How do the generated counterfactual explanations distribute around the decision boundaries?*
3. *How does the inclusion of counterfactual explanations generated by a complex non-explainable model in the training dataset influence the accuracy of an intrinsically explainable model?*
4. *How can the proposed model distillation method using counterfactual explanations be demonstrated for high-stake use cases such as credit risk and insurance prediction?*

These sub-research questions are designed to guide the investigation and offer a structured approach to unraveling the complexities involved. Firstly, the study aims to identify the state-of-the-art computational frameworks for generating counterfactual explanations, which is crucial for selecting the appropriate counterfactual generator model for this research. Secondly, it examines how these generated counterfactual explanations are distributed around the decision boundaries of classifiers, providing insights into their effectiveness and potential biases. Thirdly, the research investigates the impact of including counterfactual explanations generated by a complex, non-explainable model in the training dataset on the accuracy of an intrinsically explainable model, such as logistic regression, thus assessing the practical benefits of this integration. Finally, the study aims to demonstrate the proposed model distillation method using counterfactual explanations in high-stake use cases such as credit risk and insurance prediction, showcasing its applicability and value in critical real-world scenarios. These sub-research questions collectively contribute to a deeper understanding of how counterfactual explanations can enhance model distillation, ultimately improving both interpretability and accuracy in machine learning applications.

1.4. Outline

This section presents a concise overview of the thesis structure. Chapter 1 introduces the problem statement, research question, objectives, and sub-questions. Chapter 2 reviews the literature on counterfactual explanations and their integration into model distillation. Chapter 3 describes the methodology, covering research design, data collection from various datasets, model training, and counterfactual generation using the DICE framework. Chapter 4 analyzes the results and evaluates tool performance across different models. Chapter 5 discusses the findings in relation to the initial research questions and provides recommendations. The final chapter concludes the thesis, summarizes the insights gained, and discusses the limitations and potential future research directions. Appendices and references provide supplementary materials and documentation of sources.

2. Theoretical Framework & Literature Review

The field of machine learning has seen rapid advancements, particularly in the development and application of complex models such as deep neural networks. Despite their impressive performance, these models often suffer from a lack of interpretability (Guidotti R. M., 2018), posing challenges to their use in decision-critical areas. To address these challenges, various techniques have been proposed, with model distillation and counterfactual explanations emerging as promising approaches.

2.1. Counterfactual Explanations

In the literature, particularly in recent years, counterfactual explanations have become a prominent topic. Counterfactual explanations offer a powerful tool for understanding model decisions by illustrating how changes in input features can lead to different outcomes (Ferrario, 2022).

Early works by Wachter, Mittelstadt, and Russell (2017) introduced formal frameworks for generating counterfactuals, highlighting their potential for enhancing transparency and user trust in automated systems. Recent studies have expanded on these foundations, exploring methods to efficiently generate counterfactuals, assessing their robustness, and applying them in diverse contexts such as healthcare, finance, and legal decision-making.

Riccardo Guidotti's recent study (Guidotti R. , 2022) highlights that counterfactual explanations are essential for interpretable machine learning, especially in the context of black-box models (Guidotti, 2022). These explanations demonstrate what changes need to be made to an instance to achieve a desired outcome, helping users understand model decisions (Guidotti, 2022). The paper provides a thorough literature review and benchmarking of various counterfactual explanation methods to help select the most appropriate ones for different applications. The paper focuses on ten desirable properties of these explanations while benchmarking these methods. It inspects them on various fronts such as validity, minimality, similarity, plausibility, diversity, actionability, causality, efficiency, stability, and fairness. It provides a detailed taxonomy and categorizes existing counterfactual explainers based on their strategies, properties, and capabilities. It emphasizes that no single method guarantees all desirable properties simultaneously. By categorizing and benchmarking various methods, it serves as a guide to selecting appropriate counterfactual explainers for specific needs, acknowledging the trade-offs between different desirable properties.

Another important paper dives into the domains of both contrastive and counterfactual explanations (Stepin, 2021), concerning their foundational theories, operational frameworks, and their practical applications,

pointing out the difference between theoretical concepts and real-world applications. This paper also introduces a taxonomy aimed at guiding future research, like offering a structured approach to navigating the complex landscape.

In one paper, researchers introduce CARLA (Pawelczyk, 2021), a Python library and a benchmarking tool for comparing various counterfactual explanation methods. It emphasizes the growing importance of counterfactual explanations for model transparency and the need for actionable feature changes. The benchmarking framework and the detailed analysis of different methods can serve as a valuable resource for the research, aiding in the identification of suitable counterfactual explanation techniques for distilling complex models.

One paper provides comprehensive review analyses of 350 publications related to counterfactual explanations, categorizing them based on the need for explanations. (Verma, 2020). It provides a deep understanding of the current state of research in counterfactual explanations, including methodologies, challenges, and evaluation metrics. This review not only offers an extensive overview of the generation methods and their applications but also highlights their limitations and proposes potential research directions.

Another study introduces the DICE method which stands out by generating diverse counterfactual explanations, focusing on achieving a balance between diversity and the proximity of the generated counterfactuals to the original instance (Mothilal, 2020). The DICE approach also takes into account practical considerations such as feasibility and user constraints. Its effectiveness is demonstrated through evaluations of multiple datasets, indicating its superiority over existing methods.

Another key paper compares machine learning models to logistic regression for predicting loan defaults in Europe, using a dataset of 12 million residential mortgages (Barbaglia, 2023). It compares the performance of machine learning models to traditional logistic regression, finding ML algorithms to significantly outperform logistic regression in predicting defaults. It showcases the application of ML algorithms in a real-world context, specifically in forecasting loan defaults which can give important insights into the credit risk aspect of this research.

One paper discusses the durability of counterfactual explanations in machine learning models as they are retrained over time (Ferrario, 2022). The study highlights a significant issue where retraining models may invalidate previously valid counterfactual explanations, weakening the works made based on those explanations. The authors propose "counterfactual data augmentation" as a method to enhance the stability of these explanations across model updates.

Again, another paper by Riccardo Guidotti provides an extensive review of various methods aimed at interpreting and explaining machine learning models that are otherwise "black box" in nature (Guidotti R. M., 2018). This survey classifies the main problems related to explaining black box models, categorizes approaches based on the type of explanations they provide, and discusses the desirability of certain characteristics in interpretable models.

The document "A Survey of Algorithmic Recourse" provides a comprehensive overview of algorithmic recourse within the domain of Explainable Artificial Intelligence (Karimi, 2022). It lays a solid foundation for understanding the key concepts, challenges, and future directions in generating counterfactual explanations. It provides a robust framework for exploring algorithmic recourse, aiming to enhance the transparency and interpretability of distilled models through counterfactual reasoning.

Another research paper explores the concept of providing counterfactual explanations to understand automated decisions without delving into the complexities of the algorithm's internal mechanics (Wachter, 2017). This paper addresses the challenge of distilling complex model decisions into understandable insights for individuals, hence contributing to the research on model distillation and counterfactual explanations.

Collectively, these studies navigate the complex domain of counterfactual explanations, spanning from conceptual frameworks to applied uses.

2.2. Integration of Counterfactuals in Model Distillation

While both model distillation and counterfactual explanations have been extensively studied independently, their integration remains relatively underexplored. Preliminary research suggests that incorporating counterfactual explanations into the distillation process can potentially enhance the predictive power of the student model (Liu, 2023).

In certain applications, counterfactuals prove to be truly beneficial only when they present an actionable alternative (Guidotti R. , 2022)however in this research counterfactual explanations will be used as new data points to create a synthetic dataset therefore actionability won't be a hard requirement. Nevertheless, the validity, plausibility, and stability of the generated counterfactuals still should be considered.

The current literature identifies several challenges in this emerging area. These include the computational complexity of generating counterfactuals, the trade-offs between interpretability and accuracy, and the difficulty of ensuring that counterfactuals are both feasible. Despite these challenges, the integration of counterfactual explanations in model distillation could present significant opportunities for advancing machine learning interpretability (Jeanneret, 2024).

In summary, the current state of the literature highlights the potential of combining model distillation with counterfactual explanations to enhance the interpretability and accuracy of machine learning models. This theoretical framework sets the stage for the subsequent sections, which will delve deeper into specific methodologies, empirical findings, and practical applications of this integrated approach. In the research methodology section of this paper, additional explanations of the theoretical framework will be provided as necessary to enhance understanding and clarity of the methodologies employed.

3. Research Methodology & Algorithm Development

The research methodology outlines the systematic process used to conduct this study in order to address the research objectives and answer the sub-research questions. The focus of this thesis is on using counterfactual explanations for model distillation. Specifically, this involves generating counterfactuals using a random forest model, integrating them into the original dataset, and subsequently retraining simpler models like logistic regression, decision trees, and shallower random forests to enhance their performance. The following sections elaborate on the key components of this tool.

3.1. Research Design

This study employs a quantitative research design, utilizing computational experiments to evaluate the effectiveness of counterfactual explanations in model distillation. The approach is experimental, involving the generation, augmentation, and analysis of datasets.

3.2. Data Collection

The initial step in our methodology involves selecting three distinct datasets suitable for classification tasks. These datasets were chosen for their diversity in features, domains, and class distributions, which helps in the robust evaluation of the model distillation process. The datasets selected are:

- Census Adult Income Dataset (Kohavi, 1996)
- South German Credit Dataset (South German Credit, 2019)
- Banking Dataset - Marketing Targets

3. 2. 1. Census Adult Income Dataset

The Census Adult Income dataset, also known as the "Adult" dataset or the "Census Income" dataset (Becker, 1996), is a widely used dataset for machine learning tasks, particularly classification problems. It was originally extracted from the 1994 U.S. Census Bureau data and is often used to predict whether an individual's income exceeds $50,000 per year based on various demographic and employment-related attributes. In this research, this dataset has been used as the base dataset. The dataset is sourced from the UCI Machine Learning Repository.

Characteristics:

The dataset contains 48,842 instances and 14 attributes, with one target attribute. These attributes include categorical, numerical and continuous features. This dataset initially contains 3620 NaN values appearing in the 'workclass', 'occupation', and 'native-country' columns. After these values are removed, the dataset comprises a total of 45,222 instances. The following details about the features of the dataset reflect the values after the NaN values have been excluded.

Features:

For a comprehensive overview, of each feature's name, type, and description is provided in Appendix A, Table 15. Additionally, statistical descriptions of the numerical columns in the dataset, including details such as the mean, minimum and maximum values, and standard deviation in Appendix A in Table 16. For a visual representation of the distribution of these features, which can aid in a better understanding of the data's characteristics, consult the documentation submitted with this project. These additional resources provide insights that complement the results of this research.

Target Variable:

Table 1: Adult Income Dataset- Target Variable Description

Feature Name	Feature Type	Feature Description
Income	Categorical	This feature categorizes individuals based on whether their annual income is less than or equal to $50,000, or greater than $50,000. Of the dataset, 34,014 instances, or 75.2%, report incomes below $50,000. Conversely, 11,208 instances, or 24.8%, have incomes exceeding $50,000. While there is some imbalance in this distribution, it is moderate and reflects a significant division within the dataset, showcasing a substantial variance in income levels among the individuals surveyed.

Preprocessing:

Raw data often contains errors, inconsistencies, and missing values that can negatively impact model performance. Data preprocessing addresses these issues by cleaning the data, handling missing values, and correcting errors, leading to more accurate and reliable models (Alexandropoulos, 2019). Upon importing the dataset and reviewing its features, the initial step involved cleaning the target variable. The data type of the target variable was converted to an integer, '0' for incomes below $50,000 and '1' for incomes above $50,000.

Given that the 'Education' column duplicated the information in the 'Education-num' column, it was deemed redundant and therefore excluded from further analysis to streamline the dataset. The next phase of cleaning involved removing rows containing NaN values, which were represented by question marks within the dataset. This purification ensured that only complete data entries were retained for analysis. Subsequently, datasets for the independent variables (X) and the dependent variable (y) were prepared. The 'native-country' column was omitted from the X set due to its high imbalance, which could skew the model's performance.

The final step in the preprocessing was to split the dataset into training and testing sets, with the test size set at 30% and the training set at 70% of the total dataset. It was crucial to verify that both the training and testing datasets maintained a similar distribution of the target variable, approximately 25% to 75%, to ensure consistency and reliability in the model evaluation phase.

3.2.2. South German Credit Dataset

The South German Credit dataset, also known as the "German Credit" dataset (South German Credit, 2019), is widely used for credit risk modelling and binary classification tasks. This dataset is frequently utilized in machine learning and statistical analysis to predict whether a customer poses a good or bad credit risk based on a variety of attributes. It is publicly available from the UCI Machine Learning Repository.

Characteristics:

The South German Credit dataset contains 1,000 instances and 20 features and these features are a mixture of categorical, numerical, and continuous features. This dataset does not have any NaN values. The dataset typically contains several qualitative and quantitative, and due to its comprehensive nature and the relevance of the attributes to credit scoring, this dataset is frequently used in research papers and educational exercises related to machine learning.

Features:

The comprehensive overview, of each feature's name, type, and description is provided in Appendix B, Table 17. Additionally, statistical descriptions of the numerical columns in the dataset, including details such as the mean, minimum and maximum values, and standard deviation in Appendix A in Table 18. For a visual representation of the distribution of these features, which can aid in a better understanding of the data's

characteristics, consult the documentation submitted with this project. These additional resources provide insights that complement the results of this research.

Target Variable:

Table 2: South German Credit Dataset- Target Variable Description

Feature Name	Feature Type	Feature Description
Creditworthiness (Good or Bad)	Binary	The target variable is used to classify loan applicants into two distinct categories of credit risk. It serves as the primary outcome of interest for predictive models that aim to assess the likelihood of applicants fulfilling their credit obligations. The classification into 'good' or 'bad' credit risks supports financial institutions in making informed decisions about whom to grant credit. 70% of the instances have the target value of good and the rest have bad.

3.2.3. Banking Dataset - Marketing Targets

The Bank Marketing Dataset (Moro, 2012) is a widely recognized dataset within the data science community, often used to model and analyse marketing strategies for banking products, particularly term deposits. This dataset originates from a Portuguese banking institution and was used in direct marketing campaigns (phone calls) to determine if a customer would subscribe to a term deposit, which is a fixed-term investment product typically offering higher interest rates than regular savings accounts. It contains 45,211 instances and 17 attributes, a mixture of categorical and numerical variables.

Features:

For a comprehensive overview, of each features' name, type, and description see Appendix C, Table 19. Additionally, statistical descriptions of the numerical columns in the dataset, including details such as the mean, minimum and maximum values, and standard deviation in Appendix C in Table 20. For a visual representation of the distribution of these features, which can aid in a better understanding of the data's characteristics, consult the documentation submitted with this project. These additional resources provide insights that complement the results of this research.

Target Variable:

Table 3: Banking Dataset- Target Variable Description

Feature Name	Feature Type	Feature Description
Y Subscription to term deposit	Binary	The target variable addresses the question: "Has the client subscribed to a term deposit?" It is categorized into two responses: 'yes' and 'no'. Out of the total instances, 39,922 are labelled as 'no', and 5,289 as 'yes', representing a distribution of approximately 88% to 12%, respectively.

3. 4. Model Training

3.4.1. Data Preprocessing

Data preprocessing is a critical step in the machine learning pipeline that involves preparing raw data for further processing and analysis. This process includes cleaning, transforming, and organizing data to ensure that it is in the best possible condition for training machine learning models. All the datasets are cleaned before the independent and dependent dataset splits. This section will focus on the transformation and organization of the datasets, inside the model training.

Effective data preprocessing significantly enhances the performance and reliability of ML algorithms (Huang J. L., 2015). Prior to training the model, the dataset undergoes comprehensive preprocessing to ensure that the features are in a suitable format for the classifier models. For all the models used in this research, same normalization, standardising and encoding techniques are used.

Categorical Data Transformation

In this research, categorical variables are transformed using OneHotEncoder, a method that converts categorical data into binary columns for each category. Each category within a row is marked with a 1 in its corresponding column, while all others are set to 0. This technique ensures that no ordinal relationship is inferred among categories, making the data suitable for algorithms that treat numerical inputs as having an inherent order, such as linear regression, neural networks, and SVMs.

However, OneHotEncoder can lead to challenges, particularly with features that have many categories, by significantly increasing the dataset's dimensionality. This can cause computational inefficiencies and memory issues, and often results in a sparse matrix filled with zeros, which some algorithms handle inefficiently.

Despite these drawbacks, OneHotEncoder is a crucial preprocessing tool in machine learning (Seger, 2018). It ensures categorical variables are suitably formatted for numerical algorithms, enhancing model performance, and maintaining data integrity.

Numerical Data Transformation

Numerical variables are scaled using StandardScaler, a prevalent data preprocessing technique (Ahsan, 2021) in machine learning that standardizes features by giving them zero mean and unit variance. This process, known as Z-score normalization, involves subtracting the mean and dividing by the standard deviation for each feature. This normalization is crucial because it improves the performance and training stability of machine learning algorithms.

Many algorithms, particularly those that utilize gradient descent (e.g., logistic regression, neural networks), benefit from feature standardization as it facilitates faster convergence (Ahsan, 2021). Unlike Min-Max scaling, which is sensitive to outliers because it relies on minimum and maximum values, Standard Scaler minimizes the impact of outliers by using the mean and standard deviation for scaling. However, it is important to note that extreme outliers can still influence the scaling process. Standardizing features ensures that all features contribute equally to the model, preventing any single feature with a larger range from dominating the learning process and potentially leading to biased outcomes.

Employment of Preprocessing in the Models

In the preprocessing phase for all models, separate pipelines for categorical and numerical data are established and combined using a Column Transformer. This setup provides several key advantages. Firstly, it streamlines the workflow by encapsulating data preprocessing and model training into a unified process. This integration simplifies the codebase, improving both readability and maintainability.

A crucial benefit of using pipelines is their ability to ensure consistent preprocessing across both training and test datasets, preventing data leakage and enhancing the model's ability to generalize to new data. Pipelines also enhance the reproducibility and flexibility of the model training process, allowing for easy experimentation with different preprocessing techniques and model configurations.

Additionally, by automating the preprocessing and modelling steps, pipelines reduce the risk of errors during manual data handling and ensure accurate transformations at every stage, thereby maintaining the integrity of the data processing (Huang Y. C., 2019).

In summary, the strategic integration of pipelines with the Column Transformer significantly boosts the efficiency, consistency, and robustness of the model training and evaluation process. This approach ensures meticulous processing of all features before they enter the model, optimizing overall predictive performance. All models in this research utilize pipelines for training.

3.2.2. Classification Models

In this section, the methodology and rationale behind the selected classification models are described, along with their use in training and evaluation. The primary objective is to use a robust predictive model that can classify instances accurately based on the provided features, to generate counterfactual explanations to essentially employ them for model distillation.

Random Forest

For a counterfactual generation, Random Forest model with default parameters has been chosen. Random Forest is a popular and effective model for classification tasks due to its unique ensemble approach and numerous advantages in handling various types of data (Rodriguez-Galiano, 2012), as well as its ability to manage large datasets with high dimensionality (Díaz-Uriarte, 2006). This makes it particularly suitable for

high-dimensional data such as datasets that have a lot of categorical variables and one hot encoded, as in this research. It also provides measures of variable importance, which helps in identifying the most significant features for classification tasks (Breiman, 2001). Random Forest is forceful to noise and performs well even when most predictive variables are noise (Gislason, 2006), which is the case with few of the Adult Income dataset's features.

The Random Forest classifier is initialized with specified parameters for the number of estimators (n_estimators) and maximum depth (max_depth). These parameters will be adjusted to create shallower Random Forest models with fewer estimators and restricted tree depth to eventually conduct experiments where counterfactual explanations are eused to improve shallower trees.

Logistic Regression

Logistic Regression (LR) is a statistical fitting model that is widely used and considered relatively explainable (Tahirovic, 2023). Several studies (Ayer, 2010) have shown that LR is a valuable tool in model predictions.

The Logistic Regression model is constructed within a comprehensive pipeline framework that incorporates preprocessing steps such as one-hot encoding and standardization, in addition to the classifier itself. The implementation relies on methods provided by the 'sklearn' library, which will be further elaborated upon in the "Tools and Techniques" section of this document.

The operational flow of the pipeline begins by taking the pre-cleaned training dataset, denoted as X_fit along with the corresponding target variable y_fit. These inputs are then processed through a preprocessor that is explained previously. Subsequent to preprocessing, the model is fit to these data.

Upon completion of the training phase, the model is tested using a dataset that has been separated in the beginning, to predict the target variable. This evaluation is undertaken to determine the efficacy of the model under study conditions.

Decision Tree

Decision tree classifiers are considered one of the most popular methods for data classification due to their clear and understandable representation of decision-making processes (Charbuty, 2021).

The Decision Tree model is established within a comprehensive pipeline framework that integrates preprocessing steps such as one-hot encoding and standardization, alongside the classifier itself.

The operational flow of the pipeline initiates with the pre-cleaned training dataset, labelled as X_fit, and the corresponding target variable, y_fit. These inputs are first processed through a previously described preprocessor. After preprocessing, the Decision Tree model is trained on these data.

Once the training phase is complete, the model undergoes testing with a dataset that was segregated at the outset to predict the target variable. This evaluation phase is conducted to assess the effectiveness of the model under the specified study conditions.

3.2.3. Model Performances

To thoroughly assess the performance of the models, two principal metrics are computed: the F1 Score and the ROC AUC Score. These metrics are particularly vital in the context of binary classification problems involving imbalanced datasets, as encountered in this research. They provide a more nuanced understanding of model effectiveness, capturing both the precision and recall of the classifier, thus offering a holistic view of its performance. This dual metric approach ensures a balanced evaluation, addressing potential biases and validating the robustness of the model.

F1 Score

The F1 Score is defined as the harmonic mean of precision and recall, providing a balanced measure of the model's accuracy. Precision is calculated as the ratio of correctly predicted positive observations to the total predicted positives. On the other hand, recall, also known as sensitivity, is the ratio of correctly predicted positive observations to all actual positives. In the calculation of the F1 Score, the harmonic mean is employed rather than the arithmetic mean because it places greater emphasis on the lower of the two values. Consequently, a high F1 Score is attained only if both precision and recall are high.

In scenarios involving imbalanced datasets, relying solely on accuracy can be misleading. This is due to the possibility of a model predominantly predicting the majority class and still achieving high accuracy rates. The

use of the F1 Score is particularly beneficial as it ensures that the model also performs effectively in predicting the minority class, which is crucial in many applications such as fraud detection.

ROC AUC Score

The Area Under the Receiver Operating Characteristic Curve (ROC AUC) is employed to evaluate a model's capability to differentiate between positive and negative classes. The ROC curve illustrates the true positive rate (recall) versus the false positive rate at various threshold settings. The AUC score quantifies the entire two-dimensional area underneath the ROC curve. A model with an AUC score of 1.0 is considered perfect, indicating flawless discrimination, whereas a score of 0.5 suggests that the model lacks discriminative power and performs no better than random guessing.

In contrast to accuracy, the ROC AUC score is not contingent upon a specific classification threshold. It assesses the model's performance across all possible thresholds, offering a comprehensive view of its effectiveness. This score is particularly valuable in contexts with imbalanced datasets, as it measures the model's ability to distinguish between the positive and negative classes without being biased towards the majority class. Essentially, the ROC AUC score provides an overall evaluation of the model's discriminative ability.

Results

After the calculation of F1 Score and ROC AUC score, the performance metrics are recorded as a row, and a summary data frame of the model's performance is created. This summary includes the model's name, iteration details, sample size that is used for the counterfactual generation, number of counterfactuals generated per sample, the size of the training dataset, F1 score, ROC AUC score, and the standard deviation of these two scores through chosen number of iterations.

3. 5. Generating Counterfactuals

3.5.1. DICE Framework

In this research, the DICE (Diverse Counterfactual Explanations) framework (Mothilal, 2020) has been used for generating counterfactual explanations. This framework is specifically designed to elucidate machine learning classifiers by producing diverse and feasible counterfactual explanations. DICE primarily provides post-hoc explanations after a machine learning model has issued its predictions which is a Random Forest model with default parameters in this research. These explanations constitute hypothetical scenarios illustrating how an instance could have elicited a different prediction by modifying certain features. Typically, they aid users in comprehending and acting upon these predictions.

However, in the context of this research, DICE-generated explanations are utilized to construct a synthetic dataset. This dataset is intended to potentially enhance the predictive capabilities of more interpretable machine learning algorithms, such as logistic regression.

According to (Mothilal, 2020), effective counterfactual explanations need to be both feasible and diverse. Feasibility ensures that the proposed changes are practical and possible within the user's context and constraints. Diversity in counterfactual explanations is equally important as it provides a range of potential changes that could lead to a different outcome. This variety helps users explore multiple ways to achieve a desired result, offering more actionable insights.

The DICE framework ensures diversity among generated counterfactuals using Determinantal Point Processes (Mothilal, 2020). This technique allows DICE to create a set of diverse counterfactual examples. Additionally, DICE constructs an optimization problem that balances the proximity and sparsity of the counterfactuals to the original instance with the need for diversity (Mothilal, 2020). This balance ensures that the generated counterfactuals are not only varied but also closely related to the original instance, making them more realistic. These metrics help in generating actionable counterfactuals. Users can input their domain knowledge and constraints to guide the generation process, ensuring that certain features remain unchanged.

For evaluation, DICE proposes quantitative metrics to assess the validity, diversity, and proximity of the generated counterfactuals. It also uses a 1-nearest neighbour model trained on these counterfactuals to approximate the local decision boundary of the machine learning model.

3. 5. 2. Counterfactual Generation

In this section, the process used for generating counterfactual explanations utilizing the DICE framework is detailed. The steps involve the creation of a data object, model object, DICE model instantiation, and the

generation of counterfactuals. This methodology provides a structured approach to generating diverse and feasible counterfactuals, aiding in the interpretability of machine learning models.

Data Sampling

For the generation of counterfactual explanations, samples from the fitting dataset were selected. In this research, subsets constituting 10%, 20%, 30%, and 40% of the dataset were employed. For each subset, various experiments were conducted by generating 1, or 2 counterfactuals per sample.

In the scope of this study, counterfactuals are crafted to represent the opposite class of the original instance. For instance, if a sample's target variable is assigned a value of 0, the generated counterfactuals will bear a target variable of 1. A notable issue arises when a random sample is taken from the fitting data; it is likely to retain the same distribution as the original dataset. Consequently, as the number of counterfactuals increases, the initial distribution of the target variable becomes increasingly distorted. This alteration of distribution could potentially impact the overall balance and representativeness of the dataset, affecting the conclusions drawn about the model's performance and the effectiveness of the counterfactual explanations.

Consequently, the initial step in the counterfactual generation process involves selecting a sample that exhibits the exact opposite distribution of the target variable. For example, in the Adult Income dataset, where 75% of the instances have a target value of 0 and the selected sample for random counterfactual generation should ideally consist of a data frame which only have 25% of the instances with the target value of 0. This inverse sampling approach ensures that when counterfactuals are generated (assuming one counterfactual per sample is created), they mirror the distribution of the original fitting dataset. This methodological adjustment is crucial for maintaining the representational balance of the target variable, ensuring that the effects of the generated counterfactuals on the dataset are consistent with the original data characteristics.

The process of creating a dataset with an opposite class distribution begins by identifying all columns in the input dataset, labelled as X_fit. An analysis of the original class distribution is then conducted, involving counting the instances of each class and calculating their proportions. Based on this data, the required number of samples for each class to achieve the desired opposite distribution is determined. These samples are carefully selected from the fitting data to reflect the new, counterposed distribution of the target variable. Once the necessary samples are collected from each class, they are combined to form the final dataset. To ensure balance and eliminate any potential order bias, the dataset is shuffled. This is a crucial step to prevent any skewed distribution from affecting the model's learning and evaluation phases, ensuring that the dataset accurately represents the new distribution intended for analysis.

3. 5. 3. Employed DICE Model

In this section, the process used for generating counterfactual explanations utilizing the DICE framework is detailed. There are 4 steps involved, the creation of a data object, model object, DICE model instantiation, and the generation of counterfactuals.

The initial step involves the creation of a Data object using the DICE library, which encapsulates the dataset along with information about continuous features and the outcome variable. In this context, X_fit is used to represent the input data frame containing the predictor variables, while y_fit is the series containing the target variable. The method assigned is utilized to combine these into a single data frame. The parameter 'continuous_features' is a list identifying which features in the dataset are continuous. The 'outcome_name' parameter specifies the column representing the outcome variable.

Subsequently, a model object is created to encapsulate the pre-trained machine learning model. In this research, 'model' was the pre-trained random forest classifier, which is utilized with default parameters. The specification of backend="sklearn" indicates that the model is implemented using the scikit-learn library. This step is crucial as it facilitates the integration of the DICE framework with the underlying machine learning model, thereby enabling the generation of interpretable counterfactual explanations.

The next step involves the creation of an instance of the DICE model by integrating the data and model objects. In this instantiation, method="random" specifies that the counterfactual generation method employed is random. The DICE framework supports various methods for generating counterfactuals, and in this instance, a random generation method is chosen to produce diverse counterfactual examples.

Finally, counterfactual explanations are generated using the DICE model, under the name of 'e1'. Here, X_fit is the input data for which counterfactuals are to be generated. The parameter 'total_CFs' specifies the number

of counterfactual explanations desired. Setting ‘desired_class="opposite"’ indicates that the generated counterfactuals should result in an opposite outcome class compared to the current prediction. The ‘features_to_vary’ parameter defines which features are allowed to change during the counterfactual generation process.

After the generation of counterfactuals, the subsequent phase involves their visualization, storage, and preparation for further analysis. This process begins with saving the counterfactuals within a data frame format to facilitate easier inspection and comprehension. A for loop is employed to iterate through each sample, appending its respective counterfactuals to the data frame. Once all counterfactuals from each sample have been successfully appended, the data frame's index is reset. This reset is performed to ensure that each entry has a unique index, aligning with the length of the fitting dataset, which helps in maintaining organized and accessible data. Following the restructuring of the data frame, it is saved as a CSV file. Finally, for further analysis, this new data frame is divided into two new datasets: one for independent variables and one for the dependent variable.

3. 5. 4. Running an Experiment

After defining the counterfactual generation function, the primary function for running an experiment is designed to conduct experiments with various sample sizes, numbers of counterfactuals (CFs), and iterations.

The process is initiated by creating an empty data frame designed to store the generated counterfactuals. Subsequently, the default Random Forest (RF) and a simpler model are executed on the original dataset, with their performance metrics consolidated into the data frame. This establishes a baseline for comparing the outcomes of future experiments. Additionally, in scenarios involving multiple iterations, these two models are executed repeatedly as the number of iterations.

Following the establishment of this baseline, two nested loops are initialized to iterate over varying sample sizes and the number of counterfactuals to be generated. Initially, empty lists are created to store the performance metrics across multiple iterations. Subsequently, the function responsible for generating counterfactuals is invoked, which processes the sample size and the specified number of counterfactuals per sample to return X_fit_cf and y_fit_cf, comprising newly generated instances. These new frames facilitate the creation of a second set of datasets, which incorporate both original and counterfactual data, thereby enabling experiments that integrate both datasets.

The third loop commences, iterating over the number of iterations. Within this loop, the function for the simple model is activated using both the original data and the expanded dataset. The F1 and AUC scores from both experiments are recorded in a list to calculate the average scores across multiple iterations. Furthermore, the performance metrics of the models are appended to the initially created data frame, which originally contained metrics from models run solely on the original data.

Upon completion of the third loop, the lists containing the F1 and AUC scores from each iteration are utilized to compute the means and standard deviations of these scores across multiple iterations. These calculated performance metrics are then appended to a distinct dataset, which was initialized prior to the nested loops. This dataset contains iterated performance metrics of both the default Random Forest and the simple model, effectively documenting the evolution of model performance over successive iterations.

The algorithms used for these analyses function like a tool, enabling any user to conduct the same analyses across various datasets. This tool allows for customization of key parameters including sample size, the number of counterfactuals per sample, and the number of iterations, providing flexibility to adapt to different data requirements and research objectives.

3. 6. Tools and Technologies

In this research, a variety of tools and technologies were employed to facilitate data processing, model development, evaluation, and counterfactual generation. This section provides a detailed overview of the tools, technologies, and environments utilized in this project. By utilizing these tools and technologies, the research ensured strong data handling, healthy model development, and thorough evaluation of results. The combination of powerful libraries, interactive environments, and efficient hardware enabled the research of the project objectives.

3.6.1. Programming Languages and Environments

Python: Python was the primary programming language used for this project. Its extensive libraries and ease of use made it ideal for data manipulation, machine learning, and statistical analysis.

Jupyter Notebook: Jupyter Notebook provided an interactive environment for writing and running Python code. It was particularly useful for iterative development, visualization, and the documentation of the workflow.

3. 6. 2. Libraries for Data Processing and Analysis

pandas: Used for data manipulation and preprocessing. pandas provided robust data structures and data analysis tools.

NumPy: Utilized for numerical operations and array manipulations, enhancing the efficiency of data processing.

scikit-learn: A comprehensive machine learning library used for implementing various models and preprocessing techniques. Specific modules and classes included:

- preprocessing: For scaling and encoding data.
- model_selection: For splitting datasets into training and testing sets.
- ensemble: For implementing the Random Forest classifier.
- linear_model: For implementing the Logistic Regression model.
- tree: For implementing the Decision Tree classifier.
- metrics: For evaluating model performance.

Matplotlib: Used for creating visualizations to explore data and present results.

DICE (Diverse Counterfactual Explanations): Used for generating counterfactual explanations. DICE helps in understanding model predictions and improving model interpretability.

Warnings: Used to handle and suppress warnings during the development process to ensure cleaner output.

3. 6. 3. Hardware

Computations for this research were performed on standard workstations or cloud-based environments, depending on the dataset size and complexity. The primary hardware specifications used are as follows:

Processor: 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz, 2.42 GHz

RAM: 12.0 GB

System Type: 64-bit operating system, x64-based processor

3. 6. 4. Version Control and Collaboration

GitHub: GitHub was used for version control and collaboration. It facilitated efficient tracking of changes, issue management, and collaborative development. The project repository was hosted on GitHub, ensuring that all code, data, and the documentation were versioned and accessible to collaborators.

4. Results, Analyses, and Algorithm Performance

This section provides a detailed analysis of the results obtained from the application of Logistic Regression, Decision Tree, and Simple Random Forest (RF) models. The models were initially trained with the original data, and their performance metrics were recorded. Subsequently, the models were trained with counterfactual (CF) data and a combination of CF and original data. This comprehensive analysis allows for a comparison of the results, investigating the effectiveness and reliability of the tools under different scenarios.

4.1. Interpretation of the Algorithm's Output

The primary models evaluated in this research included Logistic Regression, Decision Tree, and a Simplified Random Forest (RF). To ensure a comprehensive and thorough analysis, each of these models was meticulously trained and tested across three distinct datasets. The performance of each model was evaluated using several metrics, specifically the F1 score and the ROC AUC score.

To facilitate a clear and detailed understanding of the models, the results for each model will be systematically presented across all datasets. Additionally, to further elucidate the models' capabilities, a dedicated section

will discuss additional experiments conducted exclusively with the Adult Income dataset. This segmentation ensures that the findings are both comprehensive and accessible, allowing for a nuanced interpretation of the models' performances across different data scenarios.

The counterfactual generation process was methodically repeated five times for each combination of sample size and number of counterfactuals. Once the counterfactuals were generated, the models intended for distillation were subsequently trained on two distinct datasets: one containing only the counterfactual data and another combining both the original and counterfactual data.

The process of generating counterfactuals was consistently repeated five times across various sample sizes and numbers of counterfactuals. The performance metrics from these experiments were meticulously recorded to facilitate a detailed comparison between the models trained on different datasets. Additionally, the means and standard deviations of the performance metrics from these five iterations were documented in another report to ensure the results were robust and not merely attributable to variability or chance. This structured approach provides a comprehensive evaluation of how counterfactual data influences model performance, enhancing the reliability and interpretability of the findings. The performance metrics of all iterations can be found in the documentation submitted with this report.

4.1.1. Logistic Regression Distillation

Logistic regression distillation employs a complex model, specifically a random forest, to enhance a simpler model, such as logistic regression. The goal is to transfer the sophisticated insights from the complex model to the simpler one, aiming to boost its performance while reducing computational demands and increasing interpretability. In this research, the random forest is used to generate a synthetic dataset that is then applied to refine and improve the logistic regression model. Initially, for each dataset, the Random Forest and Logistic Regression models were trained on the original dataset, and their performance metrics were documented for future reference. During the generation of the counterfactuals, the DICE model was set to permit changes across all features, ensuring robust and extensive testing of model adaptability.

In this section, all the results are meticulously presented in tabular form for clarity and precision. However, for a visual representation of the outcomes, particularly concerning the AUC score plots, please see Appendix D.

Census Adult Income Dataset

For the Adult Income dataset specifically, various sample sizes representing 10, 20, 30, and 38 percent of the dataset were selected, corresponding to 3166, 6331, 9495, and 12000 samples, respectively. At each of these sample sizes, either one or two counterfactual explanations were generated, each with the target variable switched to the opposite class.

Table 4:

Logistic Regression Trained on dataset with CFs - Average of 5 iterations – Adult Income Dataset

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31655	0,6677	0,0009	0,7697	0,0007
LR default on original data	5	0	0	31655	0,3979	0,0000	0,6188	0,0000
Models Trained on CFs + Original data								
LR sample: 10% CF:1	5	3166	1	34821	0,3438	0,0035	0,5998	0,0011
LR sample: 10% CF:2	5	3166	2	37987	0,3145	0,0013	0,5911	0,0004
LR sample: 20% CF:1	5	6331	1	37986	0,3130	0,0013	0,5906	0,0004
LR sample: 20% CF:2	5	6331	2	44317	0,2854	0,0011	0,5818	0,0004
LR sample: 30% CF:1	5	9495	1	41150	0,3111	0,0302	0,5904	0,0102
LR sample: 30% CF:2	5	9495	2	50645	0,2610	0,0031	0,5741	0,0010

LR sample: 38% CF:1	5	12000	1	43655	0,3878	0,0030	0,6155	0,0010
LR sample: 38% CF:2	5	12000	2	55655	0,3809	0,0014	0,6131	0,0005
Models Trained on Only CFs								
LR sample: 10% CF:1	5	3166	1	3166	0,1519	0,0163	0,5408	0,0047
LR sample: 10% CF:2	5	3166	2	6332	0,1582	0,0078	0,5426	0,0023
LR sample: 20% CF:1	5	6331	1	6331	0,1639	0,0031	0,5443	0,0009
LR sample: 20% CF:2	5	6331	2	12662	0,1546	0,0056	0,5416	0,0016
LR sample: 30% CF:1	5	9495	1	9495	0,1620	0,0085	0,5438	0,0025
LR sample: 30% CF:2	5	9495	2	18990	0,1645	0,0042	0,5446	0,0012
LR sample: 38% CF:1	5	12000	1	12000	0,3718	0,0075	0,6099	0,0029
LR sample: 38% CF:2	5	12000	2	24000	0,3722	0,0045	0,6100	0,0018

Note: The term "Fitting Data" refers to the size of the dataset (X_fit) utilized for training the model. For Logistic Regression samples, the fitting data includes the original fitting data combined with the product of the sample size and the number of counterfactuals (CF number). The terms "F1 std" and "AUC std" denote the standard deviations of the F1 scores and AUC scores, respectively, across different iterations.

To accurately interpret the results from the logistic regression models re-trained with counterfactuals, it is essential to compare these outcomes to the baseline performance established by models trained on the original dataset. The baseline RF model recorded an F1 score of 66.77% and an AUC of 76.97%, significantly surpassing the LR model, which scored an F1 of 39.79% and an AUC of 61.88%.

Upon incorporating counterfactuals into the dataset at varying proportions and varying counterfactuals per sample, a general trend of decreased performance was observed in the logistic regression outcomes. Most configurations resulted in AUC scores that were lower than those achieved by the original LR model. This decline was particularly marked when 30% of the sample included two counterfactuals per sample, resulting in the lowest AUC score of 57.41%. A similar downward trend was noted in F1 scores, where all configurations with counterfactuals failed to reach the performance level of the original LR model. The largest sample size (38%) was the closest to approaching baseline AUC levels, although it still fell short. These findings suggest that the addition of counterfactuals generally had a negative impact, potentially due to the introduction of noise or misalignment with the logistic regression model's decision boundaries. The counterfactuals might have been too divergent from the true data distribution or could have led to overfitting, which detracted from the model's accuracy.

For the logistic regression models trained exclusively on counterfactual data, a significant performance decline was evident when compared to the baseline models. The F1 scores for models trained solely on counterfactuals were notably lower across all setups, starting as low as 15.19% for models trained with 10% of the data consisting of one counterfactual per sample. The highest AUC score was observed at a 38% sample size, with two counterfactuals per sample, reaching 60%. Despite this, the model still did not meet the F1 score from the baseline LR model. Although there was a slight upward trend in AUC with larger sample sizes, the scores remained well below those of the baseline model.

These results indicate a drastic reduction in performance when logistic regression models are trained with counterfactuals, suggesting that counterfactuals do not provide a comprehensive representation of the original data's distribution, not enough to improve model performance. This issue might stem from the lack of feature diversity or coverage that is essential for capturing the full spectrum of the data landscape, highlighting areas for potential improvement in the utilization of counterfactuals for model training.

South German Credit Dataset

The study used the South German Credit dataset, selecting sample sizes that accounted for 10, 20, and 30% of the dataset, corresponding to 100, 200, and 300 samples, respectively. At each sample size, one or two counterfactual explanations were constructed, each of which shifted the target variable to the opposite class.

Table 5

Logistic Regression Trained on dataset with CFs - Average of 5 iterations – South German Credit dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	700	0,8582	0,0063	0,7080	0,0176
LR default on original data	5	0	0	700	0,8279	0,0000	0,6896	0,0000
Models Trained on CFs + Original data								
LR sample: 10% CF:1	5	100	1	800	0,8332	0,0033	0,6950	0,0042
LR sample: 10% CF:2	5	100	2	900	0,8255	0,0060	0,6859	0,0064
LR sample: 20% CF:1	5	200	1	900	0,8326	0,0064	0,6945	0,0098
LR sample: 20% CF:2	5	200	2	1100	0,8359	0,0041	0,6997	0,0084
LR sample: 30% CF:1	5	296	1	996	0,8376	0,0056	0,7064	0,0079
LR sample: 30% CF:2	5	296	2	1292	0,8391	0,0069	0,6973	0,0124
Model Trained on Only CFs								
LR sample: 10% CF:1	5	100	1	100	0,8145	0,0092	0,6634	0,0102
LR sample: 10% CF:2	5	100	2	200	0,8033	0,0107	0,6632	0,0118
LR sample: 20% CF:1	5	200	1	200	0,8207	0,0092	0,6551	0,0126
LR sample: 20% CF:2	5	200	2	400	0,8253	0,0103	0,6683	0,0076
LR sample: 30% CF:1	5	296	1	296	0,8295	0,0107	0,6804	0,0178
LR sample: 30% CF:2	5	296	2	592	0,8347	0,0092	0,6676	0,0250

To interpret the results of logistic regression models trained with the South German Credit dataset, both with counterfactuals added to the original data and models trained solely on counterfactuals. Random Forest (RF) on original data achieved an F1 score of 85.82% and an AUC of 70.80%. Logistic Regression (LR) on original data recorded an F1 score of 82.79% and an AUC of 68.96%.

Slight improvements in F1 scores were seen when counterfactuals were added to original data, with the highest F1 score being 83.91% (30% CF:2). This score is slightly above the baseline logistic regression model. AUC scores varied slightly with the addition of counterfactuals, showing minor fluctuations but generally remaining close to the baseline. The highest AUC recorded was 70.64% (30% CF:1), which is an improvement over the baseline LR model. These results suggest that, for this dataset, adding counterfactuals to the original dataset can enhance or at least maintain model performance, indicating potential alignment with decision boundaries or beneficial variance introduced by the counterfactuals.

For models trained solely on counterfactuals, there was a reduction in AUC scores compared to the baseline models and those trained on a combination of original and counterfactual data. The highest F1 score achieved by models trained only on counterfactuals was 83.47% for the configuration with 30% counterfactuals and two per sample. Although this score is impressive, it still does not reach the benchmark set by the baseline Random Forest model, though it slightly exceeds that of the Logistic Regression model. Regarding AUC scores, models trained exclusively on counterfactuals showed lower performance, with the highest recorded at 68.04% for 30% counterfactuals with one per sample. This score, while below the baseline Logistic Regression model, comes very close, indicating generated counterfactuals were somewhat good at capturing the behavior of original dataset.

The analysis of the models trained with counterfactuals suggests that while adding counterfactuals to the original data can yield competitive or slightly improved results, training exclusively on counterfactuals leads to

inferior model performance. This indicates that counterfactuals, though useful in certain contexts, may not provide a comprehensive basis for model training when used alone.

Banking Dataset -Marketing Targets

The analysis focused on the Banking dataset, where sample sizes representing 10, 20, and 30 percent of the dataset were chosen, equating to 3164, 6328, and 9492 samples, respectively. It is crucial to note that for the 30% sample, the distribution of the target value was not maintained due to the insufficient number of instances in the minority class. To achieve the desired sample size, all available instances of the minority class were gathered, supplemented by additional instances from the majority class. At each of these sample sizes, either one or two counterfactual explanations were generated, with the target variable of each counterfactual altered to the opposite class.

Table 6:

Logistic Regression Trained on datasets with CFs - Average of 5 iterations – Banking dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31647	0,5081	0,0047	0,6929	0,0019
LR default on original data	5	0	0	31647	0,4509	0,0000	0,6612	0,0000
Models Trained on CFs + Original data								
LR sample: 10% CF:1	5	3164	1	34811	0,4013	0,0014	0,6343	0,0006
LR sample: 10% CF:2	5	3164	2	37975	0,3588	0,0012	0,6147	0,0006
LR sample: 20% CF:1	5	6328	1	37975	0,4140	0,0044	0,6408	0,0022
LR sample: 20% CF:2	5	6328	2	44303	0,3975	0,0044	0,6325	0,0020
LR sample: 30% CF:1	5	9492	1	41139	0,4581	0,0029	0,6640	0,0015
LR sample: 30% CF:2	5	9492	2	50631	0,4573	0,0039	0,6639	0,0020
Model Trained on Only CFs								
LR sample: 10% CF:1	5	3164	1	3164	0,1312	0,0185	0,5340	0,0053
LR sample: 10% CF:2	5	3164	2	6328	0,1256	0,0010	0,5322	0,0003
LR sample: 20% CF:1	5	6328	1	6328	0,3306	0,0093	0,6037	0,0040
LR sample: 20% CF:2	5	6328	2	12656	0,3327	0,0055	0,6049	0,0023
LR sample: 30% CF:1	5	9492	1	9492	0,4678	0,0037	0,6809	0,0021
LR sample: 30% CF:2	5	9492	2	18984	0,4650	0,0019	0,6786	0,0013

First, baseline measurements were established using a Random Forest model, which achieved an F1 score of 50.81% and an AUC of 69.29%, and a Logistic Regression model, which recorded an F1 score of 45.09% and an AUC of 66.12%.

When logistic regression models were trained on a mixture of counterfactuals and original data, a general decline in performance was observed across all metrics in comparison to the baseline logistic regression model. With 10% of the data consisting of counterfactuals, the AUC score decreased to 63.43% and 61.47 % for models trained with one and two counterfactuals, accompanied by a corresponding decline in F1 scores. However, increasing the sample number improved the performance slightly and reached scores close to baseline logistic regression which was trained on the original data. In the case of 30 percent sample size, 1 CF generation achieved 66.40% which is a small bit higher than the baseline, and same for experiment with the 2 CFs per sample where it reached AUC score of 66.39%. Yet, these scores are still cannot reach to the baseline Random Forest.

Furthermore, the models trained exclusively on counterfactuals from 10 and 20% sample sizes displayed significantly lower performance metrics, with F1 scores ranging from 13.12% to 33.27%, and AUC scores hovering just below 61%. These markedly reduced metrics signal a severe performance degradation, underscoring the inadequacy of using only 10 and 20% of the dataset for generating counterfactuals. Such small sample sizes fail to capture the comprehensive data characteristics essential for effective model training. It is logically anticipated that training models on a synthetic dataset derived from merely 10 to 20 percent of the original dataset would result in lower performance scores, given the limited data diversity and volume required to accurately model complex real-world phenomena.

Interestingly, the models trained on a sample size of 30% exhibited a slight improvement over the baseline model in terms of AUC scores. When trained with one counterfactual per sample, the model achieved an AUC score of 68.09%, and with two counterfactuals per sample, it registered a score of 67.86%. This enhancement in performance suggests that a larger dataset comprising 30% of the original data may provide a more vigorous and representative sample for training, enabling the model to better generalize and capture underlying patterns compared to the original dataset. This finding indicates that increasing the proportion of the dataset used for generating counterfactuals can potentially enhance model accuracy and predictive reliability.

Summary of Logistic Regression Distillation

In this section, the effects of counterfactual explanations being integrated into the training datasets of logistic regression model was investigated. By employing the DICE framework, counterfactuals were generated that switched the target variable to the opposite class, and their influence on model performance was examined across three distinct datasets: Census Adult Income, South German Credit, and Banking datasets.

For Census Adult Income dataset, it was found that a generally negative impact on performance was observed when logistic regression models were re-trained with added counterfactuals. Despite counterfactuals being added at varying proportions (10% to 38%), most configurations resulted in lower F1, and AUC scores compared to the baseline models trained on the original dataset. Notably, when logistic regression models were trained exclusively on counterfactuals, significant declines in performance were observed, suggesting that counterfactuals alone might not provide a comprehensive representation of the dataset, potentially due to overfitting or misalignment with the true data distributions.

In contrast to the Adult Income dataset, it was observed that slight improvements or maintained performance levels were achieved when counterfactuals were added to the original South German Credit dataset in logistic regression models. The highest F1 score slightly exceeded the baseline RF, indicating that counterfactuals might align well with the model's decision boundaries or introduce beneficial variability in certain contexts. However, reduced performance was again noted when logistic regression models were trained solely on counterfactuals, affirming that a diverse training set is necessary to avoid performance drops, even when counterfactuals are used. It is also important to note that the high standard deviations in AUC scores indicate significant variability in model performance across different iterations or datasets. This suggests that the model may not consistently achieve similar accuracy levels, potentially due to underlying data inconsistencies of the generated counterfactuals.

For the Banking dataset, a mixed impact was observed with the addition of counterfactuals. While decreased performance was noted with smaller sample sizes (10% and 20%), an expansion to a 30% sample size saw slight improvements over the baseline model, particularly in AUC scores. This suggests a potential threshold effect where a larger proportion of counterfactuals might be necessary to capture sufficient feature diversity and data characteristics for effective model training. Interestingly, for the banking dataset, logistic regression (LR) models trained exclusively on counterfactuals, constituting 30% of the dataset, achieved higher scores than those trained on a combination of original data and counterfactuals. This outcome suggests that in certain scenarios, the counterfactual-only dataset may better align with the underlying decision boundaries of the model, or it might highlight specific patterns or anomalies not as apparent in the mixed dataset.

Overall, the integration of counterfactual explanations into model training presents nuanced outcomes that vary significantly across different datasets and model types. While enhancements or maintenance of model performance can sometimes be achieved by introducing counterfactuals, it can be concluded that, their efficacy largely depends on the proportion of the dataset they constitute and their alignment with the underlying data distribution. It is indicated by the research that counterfactuals hold promise for improving model insights, but their application should be carefully calibrated to the specific characteristics of each dataset to avoid detrimental effects on model accuracy and reliability.

4.1.2. Decision Tree Distillation

Distillation of decision tree models utilizes a complex random forest to boost the performance of the simpler model, which is a decision tree model in this case. In this section, all the results are meticulously presented in tabular form for clarity and precision. However, for a visual representation of the outcomes, particularly concerning the AUC score plots, please see Appendix D.

Census Adult Income Dataset

Specifically for the Adult Income dataset, different sample sizes, representing 10, 20, 30, and 38 percent of the dataset, equivalent to 3166, 6331, 9495, and 12000 samples, were utilized. At each sample size, one or two counterfactual explanations were produced, with the target variable altered to the opposing class.

Table 7:

Decision Tree Trained on data with CFs - Average of 5 iterations – Adult Income Dataset

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31655	0,6677	0,0009	0,7697	0,0007
DT default on original data	5	0	0	31655	0,6228	0,0020	0,7465	0,0013
Models Trained on CFs + Original data								
DT sample: 10% CF:1	5	3166	1	34821	0,6189	0,0052	0,7431	0,0038
DT sample: 10% CF:2	5	3166	2	37987	0,6184	0,0039	0,7420	0,0025
DT sample: 20% CF:1	5	6331	1	37986	0,6188	0,0025	0,7423	0,0016
DT sample: 20% CF:2	5	6331	2	44317	0,6165	0,0035	0,7395	0,0024
DT sample: 30% CF:1	5	9495	1	41150	0,6173	0,0026	0,7403	0,0018
DT sample: 30% CF:2	5	9495	2	50645	0,6180	0,0032	0,7394	0,0022
DT sample: 38% CF:1	5	12000	1	43655	0,6270	0,0051	0,7489	0,0032
DT sample: 38% CF:2	5	12000	2	55655	0,6302	0,0021	0,7508	0,0014
Models Trained on Only CFs								
DT sample: 10% CF:1	5	3166	1	3166	0,3836	0,0295	0,6152	0,0115
DT sample: 10% CF:2	5	3166	2	6332	0,4070	0,0282	0,6251	0,0115
DT sample: 20% CF:1	5	6331	1	6331	0,4099	0,0241	0,6258	0,0104
DT sample: 20% CF:2	5	6331	2	12662	0,4090	0,0233	0,6256	0,0099
DT sample: 30% CF:1	5	9495	1	9495	0,4154	0,0304	0,6287	0,0129
DT sample: 30% CF:2	5	9495	2	18990	0,4142	0,0216	0,6273	0,0093
DT sample: 38% CF:1	5	12000	1	12000	0,5890	0,0140	0,7219	0,0092
DT sample: 38% CF:2	5	12000	2	24000	0,5936	0,0159	0,7247	0,0111

The evaluation of decision tree model distillation using counterfactuals on the Adult Income Dataset provides a detailed look at how the addition of synthetic data points impacts the Area Under the Curve (AUC) score, a key metric for assessing the performance of classification models. The results span various configurations, ranging from mixing counterfactuals with original data to training exclusively on counterfactuals. Baseline models where only the original data was used for training Random Forest achieved an AUC of 76.97% and Decision Tree recorded an AUC of 74.65%.

When counterfactuals were added to the original dataset for training decision trees, the AUC scores generally hovered slightly below the baseline decision tree model. The highest AUC score observed among models with added counterfactuals was 75.08% (DT sample: 38% CF:2), which slightly surpasses the baseline DT model's AUC. This suggests that under certain configurations, particularly at higher counterfactual proportions, the models can either closely match or slightly exceed the baseline DT AUC score. Other configurations with varying percentages of counterfactuals (10% to 30%) resulted in AUC scores ranging from 74.20% to 74.89%. These scores are consistently close to but generally below the baseline DT performance.

Decision trees trained exclusively on counterfactuals showed significantly lower AUC scores compared to those trained on mixed datasets or the original data. The AUC scores for models trained only on counterfactuals were substantially lower across all configurations compared to the baseline DT, with scores increasing with the proportion of counterfactuals used. However, at the highest counterfactual proportion (38%), the AUC scores for models with one and two CFs (72.19% for CF:1 and 72.47% for CF:2) approached but did not quite reach the baseline DT AUC score. This indicates that while purely counterfactual-based training significantly reduces model performance, a sufficiently large volume of counterfactuals can bring the decision tree's discriminative ability closer to that of the baseline model.

The findings suggest that counterfactuals can be beneficial for decision tree training when used in conjunction with original data, especially at higher proportions where the synthetic data points might better challenge and refine the model's decision boundaries. The ability of purely counterfactual-trained models to approach baseline AUC scores at high counterfactual volumes indicates that while counterfactuals alone are less effective, they contain valuable information that can enhance model understanding and decision-making under specific conditions.

South German Credit Dataset

This research specifically employed the South German Credit dataset, choosing sample sizes that made up 10, 20, and 30% of the dataset, equating to 100, 200, and 300 samples, respectively. At each selected sample size, either one or two counterfactual explanations were generated, each altering the target variable to its opposite class.

Table 8

Decision Tree Trained on data with CFs - Average of 5 iterations – South German Credit dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	700	0,8500	0,0100	0,6743	0,0205
DT default on original data	5	0	0	700	0,7821	0,0092	0,6747	0,0068
Models Trained on CFs + Original data								
DT sample: 10% CF:1	5	100	1	800	0,7706	0,0143	0,6400	0,0273
DT sample: 10% CF:2	5	100	2	900	0,7653	0,0133	0,6357	0,0329
DT sample: 20% CF:1	5	200	1	900	0,7798	0,0168	0,6436	0,0211
DT sample: 20% CF:2	5	200	2	1100	0,7902	0,0105	0,6639	0,0213
DT sample: 30% CF:1	5	296	1	996	0,7809	0,0065	0,6452	0,0156
DT sample: 30% CF:2	5	296	2	1292	0,8142	0,0070	0,6776	0,0214
Model Trained on Only CFs								
DT sample: 10% CF:1	5	100	1	100	0,7452	0,0232	0,5717	0,0203
DT sample: 10% CF:2	5	100	2	200	0,7342	0,0340	0,5763	0,0327
DT sample: 20% CF:1	5	200	1	200	0,7591	0,0294	0,5953	0,0245

DT sample: 20% CF:2	5	200	2	400	0,7479	0,0284	0,6025	0,0217
DT sample: 30% CF:1	5	296	1	296	0,7636	0,0260	0,5859	0,0350
DT sample: 30% CF:2	5	296	2	592	0,7649	0,0213	0,5996	0,0242

The results from training decision tree models on the South German Credit dataset, utilizing both original data mixed with counterfactuals and training exclusively on counterfactuals, offer insightful perspectives on the utility and limitations of counterfactuals in enhancing model performance. Random Forest (RF) on original data achieved an AUC of 67.43% and Decision Tree (DT) on original data recorded a nearly identical AUC of 67.47%.

For decision trees trained on a combination of counterfactuals and original data, the results varied. In configurations where 10% and 20% of the data comprised counterfactuals, the AUC scores generally decreased compared to the baseline decision tree, ranging from 63.57% to 66.39%. This indicates that a lower proportion of counterfactuals may not be sufficient to significantly challenge or improve the decision boundaries of the model. However, at 30% counterfactual inclusion, one configuration (DT sample: 30% CF:2) notably improved, achieving an AUC of 67.76%, slightly surpassing the baseline DT model. This suggests that a higher proportion of counterfactuals can potentially enhance model discrimination capabilities, aligning better with decision-making boundaries.

Decision trees trained exclusively on counterfactuals demonstrated significantly lower AUC scores across all configurations. The AUC scores for these models were markedly lower, ranging from 57.17% to 60.25%. Such scores are substantially below those achieved by the baseline models, indicating that training solely on counterfactuals significantly impairs the model's ability to discriminate effectively between classes. The highest AUC score achieved exclusively with counterfactuals was 60.25% (DT sample: 20% CF:2), which is still considerably lower than the baseline DT's AUC score. This reinforces the notion that while counterfactuals contain useful information for challenging model assumptions, they do not suffice as standalone training data.

The integration of counterfactuals with original data shows potential under certain conditions, particularly at higher proportions, to slightly improve or at least maintain decision tree performance in terms of AUC. In contrast, models trained solely on counterfactuals struggle to perform effectively, lacking the broader context and variability.

Banking Dataset -Marketing Targets

The analysis of the Banking dataset, where samples constituting 10, 20, and 30 percent of the total dataset were selected, corresponding to 3164, 6328, and 9492 samples, respectively. It is important to highlight that for the 30% sample, the distribution of the target value could not be preserved due to the limited availability of instances in the minority class. To reach the necessary sample size, all instances from the minority class were utilized, supplemented by additional instances from the majority class. For each sample size, one or two counterfactual explanations were produced, with the target variable in each modified to the opposite class.

Table 9:

Decision Tree Results - Average of 5 iterations – Banking dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31647	0,4993	0,0064	0,6885	0,0032
DT default on original data	5	0	0	31647	0,4818	0,0045	0,7139	0,0035
Models Trained on CFs + Original data								
DT sample: 10% CF:1	5	3164	1	34811	0,4651	0,0094	0,6956	0,0060
DT sample: 10% CF:2	5	3164	2	37975	0,4588	0,0076	0,6891	0,0046
DT sample: 20% CF:1	5	6328	1	37975	0,4833	0,0154	0,7071	0,0106
DT sample: 20% CF:2	5	6328	2	44303	0,4781	0,0093	0,7023	0,0057

DT sample: 30% CF:1	5	9492	1	41139	0,4916	0,0063	0,7153	0,0034
DT sample: 30% CF:2	5	9492	2	50631	0,4937	0,0047	0,7132	0,0031
Model Trained on Only CFs								
DT sample: 10% CF:1	5	3164	1	3164	0,3008	0,0232	0,5922	0,0095
DT sample: 10% CF:2	5	3164	2	6328	0,2994	0,0130	0,5914	0,0049
DT sample: 20% CF:1	5	6328	1	6328	0,4321	0,0119	0,6577	0,0050
DT sample: 20% CF:2	5	6328	2	12656	0,4585	0,0095	0,6692	0,0048
DT sample: 30% CF:1	5	9492	1	9492	0,4767	0,0148	0,6898	0,0062
DT sample: 30% CF:2	5	9492	2	18984	0,4911	0,0070	0,6956	0,0046

In the table presented, the performance of Decision Tree models trained with different fitting datasets is documented. The original dataset served as a baseline for comparison, with both the Random Forest and Decision Tree models trained without counterfactuals. It was observed that the Decision Tree model, trained on the original data, achieved an AUC score of 71.39%, slightly outperforming the Random Forest model, which scored 68.85%, which can be due to overfitting.

The addition of counterfactual explanations to the dataset impacted AUC scores differently across the configurations of fitting sets. Initially, when 10% of the dataset included counterfactuals, AUC scores increased to 69.56% for one counterfactual and 68.91% for two. Further improvements were observed with 20% counterfactuals, where scores rose to 70.71% for one counterfactual and 70.23% for two. The most substantial gains were noted when 30% of the dataset comprised counterfactuals, achieving AUC scores of 71.53% for one counterfactual and 71.32% for two.

Conversely, when models were trained exclusively on counterfactuals, performance notably declined. For instance, AUC scores were only 59.22% and 59.14% for models trained with 10% counterfactuals (one and two CFs, respectively). With increased counterfactual percentages, a slight improvement in AUC was observed; however, these scores remained significantly lower than those of models trained on combined datasets. Specifically, AUC scores were 65.77% and 66.92% for 20% counterfactuals (one and two CFs), and 68.98% and 69.56% for 30% counterfactuals (one and two CFs).

These findings indicate that while the inclusion of counterfactual explanations does not consistently enhance the performance of the Decision Tree model, an optimal balance of counterfactual volume can lead to improvements in the model's ability to discriminate between classes. This suggests that strategic integration of counterfactual data into the training set can be beneficial, provided the volume and integration strategy are carefully managed.

Summary of Decision Tree Distillation

The distillation of decision tree models through the utilization of a complex random forest aims to transfer deep insights from the advanced model to the simpler decision tree. This approach seeks to align the simpler model's performance more closely with that of the complex model while minimizing computational demands and enhancing clarity. A synthetic dataset created by the random forest is used to refine the decision tree model, illustrating the practical advantages of model distillation in boosting both effectiveness and efficiency.

In the specific case of the Adult Income Dataset, decision trees trained with added counterfactuals generally achieved AUC scores slightly below the baseline decision tree model. However, the highest AUC score, slightly surpassing the baseline model, was observed at the highest counterfactual proportion (38%). This indicates that under certain configurations, particularly with a higher proportion of counterfactuals, decision trees can match or slightly exceed the baseline performance. Conversely, decision trees trained exclusively on counterfactuals displayed significantly lower AUC scores across all configurations, suggesting that while purely counterfactual-based training reduces performance, a sufficiently large volume of counterfactuals can slightly improve the decision tree's discriminative ability.

Similarly, the South German Credit Dataset showed that decision trees trained on a mix of counterfactuals and original data at higher proportions (30% counterfactual inclusion) could slightly surpass the baseline decision tree model. This finding suggests that a higher volume of counterfactuals can potentially enhance model discrimination capabilities, aligning better with decision-making boundaries.

For the Banking Dataset, while the inclusion of counterfactuals did not consistently enhance performance, an optimal balance of counterfactual volume led to improvements in the model's ability to discriminate between classes, especially when a significant portion of the dataset comprised counterfactuals. Most interesting finding here was, using only the counterfactual explanations, which are generated from 30% of the dataset and had outperformed the RF baseline model.

In conclusion, the integration of counterfactual explanations with original data shows potential under certain conditions to improve or at least maintain decision tree performance. However, models trained solely on counterfactuals generally struggle, lacking the broader context and variability necessary for effective performance. These findings underscore the nuanced role of counterfactuals in decision tree training, suggesting that while they are less effective alone, they contain valuable information that can enhance model understanding and decision-making under specific conditions, for specific datasets.

4.1.3. Simple Random Forest Distillation

Simple Random Forest distillation involves a technique where a more complex model, specifically a default random forest in this context, is employed to enhance the performance of a simpler model, such as random forest with restricted tree depth and lower n estimators. In the scope of this research, the intention is to utilize the random forest to generate a synthetic dataset. This newly created dataset is then used to refine and improve the simpler random forest models, effectively demonstrating the practical application of model distillation in enhancing model efficacy and efficiency.

Default Random Forest has configuration of 100 n estimators and no restriction on the tree depth. Here in this research, 2 different Simple RF models were trained with the new datasets. First Simple Random Forest, which will be referenced as RF Simp 1, has configuration of 50 n estimators and 5 as the maximum depth of the trees. Second Simple Random Forest which will be references as RF Simp 2, has configuration of 30 n estimators and 5 as the maximum depth of the trees.

In this section, all the results are meticulously presented in tabular form for clarity and precision. However, for a visual representation of the outcomes, particularly concerning the AUC score plots, please see Appendix D.

Census Adult Income Dataset

In the experiments with Adult Income dataset, sample sizes of 10, 20, 30, and 38 percent were selected, which correspond to 3166, 6331, 9495, and 12000 samples, in that order. One or two counterfactual explanations with the target variable changed to the opposite class were created for each sample size. When these counterfactuals were being created, the DICE model was configured to allow for changes in every feature.

Table 10:

2 Simpler Random Forest Models Trained on dataset with CFs - Average of 5 iterations – Adult Income Dataset

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31655	0,6719	0,0016	0,7722	0,0010
Simple RF 1 on original data – n:50 max_depth: 5	5	0	0	31655	0,6101	0,0096	0,7244	0,0060
Simple RF 2 on original data – n:30 max_depth: 3	5	0	0	31655	0,4190	0,1033	0,6350	0,0416
Models Trained on CFs + Original data								
RF simp 1 sample: 10% CF:1	5	3166	1	34821	0,4632	0,0582	0,6515	0,0245
RF simp 1 sample: 10% CF:2	5	3166	2	37987	0,3563	0,0092	0,6082	0,0034
RF simp 1 sample: 20% CF:1	5	6331	1	37986	0,3662	0,0037	0,6118	0,0014
RF simp 1 sample: 20% CF:2	5	6331	2	44317	0,3133	0,0216	0,5927	0,0077
RF simp 1 sample: 30% CF:1	5	9495	1	41150	0,3326	0,0229	0,5996	0,0082
RF simp 1 sample: 30% CF:2	5	9495	2	50645	0,2946	0,0035	0,5861	0,0012

RF simp 1 sample: 40% CF:1	5	12000	1	43655	0,5279	0,0724	0,6824	0,0353
RF simp 1 sample: 40% CF:2	5	12000	2	55655	0,4813	0,0321	0,6585	0,0140
RF simp 2 sample: 10% CF:1	5	3166	1	34821	0,2927	0,0273	0,5856	0,0091
RF simp 2 sample: 10% CF:2	5	3166	2	37987	0,2480	0,0408	0,5709	0,0130
RF simp 2 sample: 20% CF:1	5	6331	1	37986	0,2276	0,0454	0,5644	0,0143
RF simp 2 sample: 20% CF:2	5	6331	2	44317	0,2356	0,0375	0,5669	0,0119
RF simp 2 sample: 30% CF:1	5	9495	1	41150	0,2353	0,0569	0,5671	0,0178
RF simp 2 sample: 30% CF:2	5	9495	2	50645	0,2239	0,0398	0,5631	0,0122
RF simp 2 sample: 40% CF:1	5	12000	1	43655	0,3982	0,0441	0,6243	0,0174
RF simp 2 sample: 40% CF:2	5	12000	2	55655	0,3860	0,0165	0,6191	0,0062
Models Trained on Only CFs								
RF simp 1 sample: 10% CF:1	5	3166	1	3166	0,1930	0,0004	0,5531	0,0001
RF simp 1 sample: 10% CF:2	5	3166	2	6332	0,1931	0,0008	0,5531	0,0002
RF simp 1 sample: 20% CF:1	5	6331	1	6331	0,1930	0,0011	0,5527	0,0009
RF simp 1 sample: 20% CF:2	5	6331	2	12662	0,1903	0,0056	0,5522	0,0017
RF simp 1 sample: 30% CF:1	5	9495	1	9495	0,1941	0,0002	0,5534	0,0001
RF simp 1 sample: 30% CF:2	5	9495	2	18990	0,1927	0,0023	0,5529	0,0007
RF simp 1 sample: 40% CF:1	5	12000	1	12000	0,1869	0,0155	0,3143	0,0068
RF simp 1 sample: 40% CF:2	5	12000	2	24000	0,1826	0,0322	0,3195	0,0167
RF simp 2 sample: 10% CF:1	5	3166	1	3166	0,1083	0,0683	0,5254	0,0243
RF simp 2 sample: 10% CF:2	5	3166	2	6332	0,1032	0,0451	0,5272	0,0130
RF simp 2 sample: 20% CF:1	5	6331	1	6331	0,1155	0,0441	0,5307	0,0123
RF simp 2 sample: 20% CF:2	5	6331	2	12662	0,0757	0,0432	0,5105	0,0255
RF simp 2 sample: 30% CF:1	5	9495	1	9495	0,1057	0,0403	0,5205	0,0121
RF simp 2 sample: 30% CF:2	5	9495	2	18990	0,0761	0,0311	0,5094	0,0260
RF simp 2 sample: 40% CF:1	5	12000	1	12000	0,2701	0,0705	0,3670	0,0633
RF simp 2 sample: 40% CF:2	5	12000	2	24000	0,3305	0,0660	0,4170	0,0654

The examination of two simplified Random Forest (RF) models trained on the Adult Income dataset, incorporating various levels of counterfactuals (CFs), yields a nuanced view of how model simplification and synthetic data integration impact performance.

The standard RF model achieves an AUC of 77.22%, setting a high benchmark. When simplified, the models exhibit different levels of efficacy: Simple RF 1, configured with 50 trees and a maximum depth of 5, achieves an AUC of 72.44%. Simple RF 2, with 30 trees and a maximum depth of 3, records a significantly lower AUC of 63.50%. This indicates a considerable drop due to reduced complexity and capacity, showing the effect of simplifying the model.

When counterfactuals are introduced alongside original data, the impact varies. Simple RF 1 displays moderate AUC scores with 10% and 20% counterfactual inclusion, ranging from 60.82% to 61.18%, all below the baseline of the model trained solely on original data. However, an increase in counterfactual proportion to 40% results in improved AUC scores of 68.24% and 65.85% for CF:1 and CF:2, respectively. Although these scores do not surpass the original baseline, they suggest that a higher volume of CFs can elevate model performance closer to standard levels. Simple RF 2 model generally underperforms in comparison to Simple RF 1, with AUC scores consistently below those achieved by the baseline. The AUCs are particularly low, even declining as the

proportion of counterfactuals increases, except at the highest levels where there is a slight increase to 62.43%. This trend underlines the model's difficulty in effectively leveraging synthetic data for learning, especially when simplified to a greater extent.

Training exclusively on counterfactuals significantly impairs both models. Simple RF 1 sees its AUC plummet, especially at 38% CF inclusion, dropping as low as 31.43%. Simple RF 2 also records low AUC scores, with the highest at just 41.70%. These outcomes harshly illustrate the limitations of using counterfactuals as the sole data source, underscoring their inadequacy in mimicking the complexity and variance necessary for robust model performance.

The study reveals that while counterfactuals can enhance the discrimination power of simpler models at higher proportions, the effectiveness heavily depends on the specific model configuration and the counterfactual proportion. Moreover, simpler models, particularly those with significantly reduced tree counts and depth, struggle to maintain efficacy with an increasing share of synthetic data.

South German Credit Dataset

The South German Credit dataset was employed with sample sizes chosen to represent 10, 20, and 30% of the dataset, or 100, 200, and 300 samples, in that order. One or two counterfactual explanations were created for each sample size, each of which moved the target variable to the other class.

Table 11

Simple Random Forest models Trained on dataset with CFs - Average of 5 iterations – South German Credit dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	700	0,8577	0,0067	0,7039	0,0084
Simple RF 1 on original data – n:50 max_depth: 5	5	0	0	700	0,8342	0,0049	0,5687	0,0122
Simple RF 2 on original data – n:30 max_depth: 3	5	0	0	700	0,8237	0,0030	0,5094	0,0087
Models Trained on CFs + Original data								
RF simp 1 sample: 10% CF:1	5	100	1	800	0,8423	0,0051	0,6010	0,0113
RF simp 1 sample: 10% CF:2	5	100	2	900	0,8421	0,0034	0,6158	0,0079
RF simp 1 sample: 20% CF:1	5	200	1	900	0,8382	0,0023	0,6014	0,0172
RF simp 1 sample: 20% CF:2	5	200	2	1100	0,8402	0,0060	0,6057	0,0246
RF simp 1 sample: 30% CF:1	5	296	1	996	0,8378	0,0042	0,6003	0,0100
RF simp 1 sample: 30% CF:2	5	296	2	1292	0,8418	0,0073	0,6147	0,0171
RF simp 2 sample: 10% CF:1	5	100	1	800	0,8249	0,0038	0,5244	0,0130
RF simp 2 sample: 10% CF:2	5	100	2	900	0,8265	0,0042	0,5329	0,0157
RF simp 2 sample: 20% CF:1	5	200	1	900	0,8296	0,0055	0,5451	0,0278
RF simp 2 sample: 20% CF:2	5	200	2	1100	0,8293	0,0046	0,5487	0,0137
RF simp 2 sample: 30% CF:1	5	296	1	996	0,8275	0,0057	0,5438	0,0160
RF simp 2 sample: 30% CF:2	5	296	2	1292	0,8254	0,0048	0,5425	0,0096
Models Trained on Only CFs								
RF simp 1 sample: 10% CF:1	5	100	1	100	0,8169	0,0112	0,6002	0,0255
RF simp 1 sample: 10% CF:2	5	100	2	200	0,8185	0,0097	0,6209	0,0316

RF simp 1 sample: 20% CF:1	5	200	1	200	0,8293	0,0042	0,6087	0,0316
RF simp 1 sample: 20% CF:2	5	200	2	400	0,8327	0,0021	0,5968	0,0188
RF simp 1 sample: 30% CF:1	5	296	1	296	0,8338	0,0026	0,6183	0,0237
RF simp 1 sample: 30% CF:2	5	296	2	592	0,8342	0,0097	0,6058	0,0191
RF simp 2 sample: 10% CF:1	5	100	1	100	0,8200	0,0142	0,5812	0,0232
RF simp 2 sample: 10% CF:2	5	100	2	200	0,8177	0,0289	0,6074	0,0242
RF simp 2 sample: 20% CF:1	5	200	1	200	0,8268	0,0084	0,5704	0,0446
RF simp 2 sample: 20% CF:2	5	200	2	400	0,8290	0,0053	0,5811	0,0211
RF simp 2 sample: 30% CF:1	5	296	1	296	0,8261	0,0046	0,5782	0,0166
RF simp 2 sample: 30% CF:2	5	296	2	592	0,8230	0,0044	0,5683	0,0102

The evaluation of two simpler Random Forest (RF) models trained on the South German Credit dataset, including varying levels of counterfactuals (CFs), provides a detailed perspective on how model performance is affected. The standard RF model achieves an AUC of 70.39%, setting a strong benchmark. When simplified, Simple RF 1 (n=50, max_depth=5) on original data achieves an AUC of 56.87% and Simple RF 2 (n=30, max_depth=3) on original data records an even lower AUC of 50.94%. These results illustrate a significant drop due to reduced complexity, which impacts the model's ability to generalize effectively. When counterfactuals are introduced alongside original data, the AUC scores of the simpler models exhibit varied changes.

For Simple RF 1, AUC scores with 10% and 20% CF inclusion improve slightly over this model's baseline but remain substantially below the standard RF's performance. Notably, with 30% CF inclusion, AUC scores show minor improvements (e.g., 61.58% for 10% CF:2), suggesting some recovery in model performance with higher CF proportions. The best performance within this configuration reaches an AUC of 61.58%, indicating that while higher CF proportions can enhance performance, they do not fully compensate for the loss from simplification. The Simple RF 2 model, being more simplified than Simple RF 1, shows generally lower AUC scores across configurations, starting from 52.44% and marginally increasing to 54.87% with CF inclusion. These scores underline the limitations in the model's capacity to leverage additional synthetic data effectively. The performance of Simple RF 2 remains significantly lower than that of the more complex standard RF model, highlighting the challenges faced by overly simplified models in complex data environments.

When these models trained on solely counterfactuals, Simple RF 1 shows an interesting trend where AUC scores actually see a relative increase in some configurations compared to those mixed with original data. For instance, an AUC of 62.09% is achieved with 10% CF:2, indicating that under certain conditions, counterfactuals can provide meaningful insights even in the absence of original data. Despite these gains, the scores generally remain below the standard RF model trained on original data, affirming the supplementary role of counterfactuals rather than a standalone solution. Simple Random Forest 2 continues to show poor performance when trained exclusively on counterfactuals, with AUC scores not surpassing 60.74%. This pattern reaffirms that overly simplified models struggle more markedly with synthetic data, possibly due to a lack of ability in capturing complex patterns and relationships inherent in counterfactuals.

The analysis suggests that while counterfactuals can enhance the performance of simpler models, their effectiveness is significantly bound by the intrinsic capacity of the model. Higher proportions of counterfactuals tend to improve performance but do not fully resolve the deficits introduced by model simplification. This underscores the need for a balanced approach in model design where the complexity is sufficient to exploit the nuanced information offered by counterfactuals. Additionally, these findings highlight the importance of model and data congruence, suggesting that for simpler models, a careful consideration of counterfactual quality and integration strategy is crucial to maximize utility and maintain model performance.

Banking Dataset -Marketing Targets

For the Banking dataset, samples constituting 10, 20, and 30 percent of the total dataset were selected, corresponding to 3164, 6328, and 9492 samples. It is important to highlight that for the 30% sample, the distribution of the target value could not be preserved due to the limited availability of instances in the minority class. To reach the necessary sample size, all instances from the minority class were utilized, supplemented by additional instances from the majority class. One or two counterfactual explanations were

created for each sample size, each of which moved the target variable to the other class. The DICE model was set up to support changes to every feature while producing the counterfactuals.

Table 12:

Simple Random Forest models Trained on dataset with CFs - Average of 5 iterations – Banking dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	5	0	0	31647	0,5089	0,0041	0,6939	0,0024
Simple RF 1 on original data – n:50 max_depth: 5	5	0	0	31647	0,2272	0,0062	0,5641	0,0020
Simple RF 2 on original data – n:30 max_depth: 3	5	0	0	31647	0,0112	0,0091	0,5028	0,0023
Models Trained on CFs + Original data								
RF simp 1 sample: 10% CF:1	5	3164	1	34811	0,1372	0,0475	0,5368	0,0139
RF simp 1 sample: 10% CF:2	5	3164	2	37975	0,0185	0,0081	0,5046	0,0020
RF simp 1 sample: 20% CF:1	5	6328	1	37975	0,1125	0,0506	0,5298	0,0144
RF simp 1 sample: 20% CF:2	5	6328	2	44303	0,0318	0,0047	0,5079	0,0012
RF simp 1 sample: 30% CF:1	5	9492	1	41139	0,1932	0,0676	0,5542	0,0217
RF simp 1 sample: 30% CF:2	5	9492	2	50631	0,1795	0,0874	0,5508	0,0281
RF simp 2 sample: 10% CF:1	5	3164	1	34811	0,0000	0,0000	0,5000	0,0000
RF simp 2 sample: 10% CF:2	5	3164	2	37975	0,0000	0,0000	0,5000	0,0000
RF simp 2 sample: 20% CF:1	5	6328	1	37975	0,0042	0,0065	0,5011	0,0016
RF simp 2 sample: 20% CF:2	5	6328	2	44303	0,0011	0,0015	0,5003	0,0004
RF simp 2 sample: 30% CF:1	5	9492	1	41139	0,0231	0,0121	0,5057	0,0030
RF simp 2 sample: 30% CF:2	5	9492	2	50631	0,0203	0,0083	0,5050	0,0021
Models Trained on Only CFs								
RF simp 1 sample: 10% CF:1	5	3164	1	3164	0,0000	0,0000	0,5000	0,0000
RF simp 1 sample: 10% CF:2	5	3164	2	6328	0,0000	0,0000	0,5000	0,0000
RF simp 1 sample: 20% CF:1	5	6328	1	6328	0,2701	0,0072	0,5803	0,0029
RF simp 1 sample: 20% CF:2	5	6328	2	12656	0,2676	0,0083	0,5793	0,0032
RF simp 1 sample: 30% CF:1	5	9492	1	9492	0,3370	0,0078	0,6102	0,0043
RF simp 1 sample: 30% CF:2	5	9492	2	18984	0,2908	0,0592	0,5885	0,0332
RF simp 2 sample: 10% CF:1	5	3164	1	3164	0,0000	0,0000	0,5000	0,0000
RF simp 2 sample: 10% CF:2	5	3164	2	6328	0,0000	0,0000	0,5000	0,0000
RF simp 2 sample: 20% CF:1	5	6328	1	6328	0,2098	0,0412	0,5587	0,0135
RF simp 2 sample: 20% CF:2	5	6328	2	12656	0,1972	0,0498	0,5547	0,0164
RF simp 2 sample: 30% CF:1	5	9492	1	9492	0,2012	0,0719	0,5188	0,0560
RF simp 2 sample: 30% CF:2	5	9492	2	18984	0,2097	0,0738	0,5154	0,0763

RF default on original data achieves an AUC of 69.39%, establishing a benchmark for subsequent models. Simple RF 1 (n=50, max_depth=5) and Simple RF 2 (n=30, max_depth=3) trained on the original dataset exhibit significantly lower AUCs of 56.41% and 50.28%, respectively. This marked decrease highlights the limitations of

simplifying model parameters, such as reducing the number of trees and the depth, which impairs the model's ability to capture complex patterns in the data.

Simple RF 1 models trained with 10% and 20% CFs produce suboptimal AUCs ranging from 50.46% to 53.68%. This suggests that the addition of a small number of CFs is insufficient to enhance model performance and may be indicative of the CFs introducing noise or misrepresentations that the simplified model cannot adequately handle. When 30% CFs are added, there is a slight improvement with AUCs reaching up to 55.42%. However, these scores still fall short of the benchmark set by the default RF model, implying that while adding more CFs provides a richer training set, the simple model configurations are not fully equipped to leverage this added complexity for better predictive accuracy. Simple RF 2 models consistently report AUC scores around 50%, even with increased CF percentages. These outcomes point to the model's inability to distinguish between classes better than random chance, reflecting the profound impact of overly simplistic model configurations in a complex data environment.

Training exclusively on CFs demonstrates notably poor performance in models with very simple configurations (Simple RF 2), where the AUC remains close to 50%, indicating no predictive power. Simple RF 1 shows some improvement with AUCs reaching 61.02% at 30% CFs, yet these are still below the performance of the more robust default RF model. It suggests that while CFs can enhance the model's understanding of edge cases or underrepresented data, they cannot substitute for the diverse examples found in the original dataset.

The results underscore the critical balance between model complexity and training data characteristics. Over-simplified models fail to harness the nuanced information in the data, particularly when CFs are introduced. These models lack the necessary depth and breadth (in terms of tree numbers and depth) to process and learn effectively from complex or non-linear relationships introduced by CFs. CFs appear to have a dual effect; they can potentially enrich the model's training environment but also introduce complexities that simple models cannot manage. This leads to variable performance improvements and in some cases, performance degradation.

Summary of Simple Random Forest Distillation

In this section, Simple Random Forest distillation leverages a more complex random forest model to generate a synthetic dataset aimed at enhancing the performance of simpler Random Forest configurations, which have restricted tree depth and a reduced number of estimators. This process underscores the practical application of model distillation, demonstrating how it can effectively improve the efficacy and efficiency of simpler models.

In this study, two configurations of simpler Random Forest models (RF Simp 1 and RF Simp 2) were examined. RF Simp 1, with 50 trees and a maximum depth of 5, and RF Simp 2, with 30 trees and a maximum depth of 3, were trained using both original and synthetic datasets generated via counterfactuals.

For the Adult Income dataset, initial training on the original dataset set benchmarks for subsequent evaluations. RF Simp 1 demonstrated moderate performance with an AUC of 72.44%, while RF Simp 2 showed a considerably lower efficacy with an AUC of 63.50%. The integration of counterfactuals yielded varied impacts; while RF Simp 1 showed potential improvement in performance at higher counterfactual proportions (38% CF inclusion), RF Simp 2 generally underperformed, with limited enhancement even at higher counterfactual volumes. This trend highlights the sensitivity of model performance to the depth and complexity of the model when counterfactuals are used.

Similar patterns were observed with the South German Credit dataset. Simplified models trained on original data displayed significantly reduced performance compared to the default Random Forest. The integration of counterfactuals improved the performance of RF Simp 1 slightly at higher counterfactual proportions, but overall, both models remained below the standard Random Forest's efficacy. This underlines the challenge of leveraging synthetic data effectively within simpler model structures.

In the Banking dataset, both simplified models exhibited poor performance when trained exclusively on counterfactuals, and only marginal improvements were noted when counterfactuals were mixed with original data. The more complex default Random Forest model consistently outperformed the simpler models, underscoring the limitations of reduced complexity in handling enriched datasets, such as those including counterfactuals.

The study reveals that while counterfactuals can potentially enhance the discrimination power of simpler models, their effectiveness is heavily contingent on the specific configuration of the model and the proportion

of counterfactuals used. Models with greater simplification (lower tree counts and depth) often struggle to maintain efficacy as the proportion of synthetic data increases, suggesting a delicate balance is necessary between model complexity and the nature of the training dataset.

Furthermore, the findings suggest that counterfactuals, while useful for challenging and refining model boundaries, do not provide a standalone solution for training. Instead, they should be integrated thoughtfully with original data to optimize model performance, particularly in simpler models that may lack the robustness required to handle complex synthetic datasets alone.

4.1.4. Additional Experiments on Adult Income dataset using Simple Random Forest

Given that the initial experiments did not exactly achieve the anticipated enhancement of the simpler model's predictive power through the inclusion of a synthetic dataset with counterfactuals, further experiments were carried out on the Adult Income dataset. As detailed in the subsequent section, it was observed that the generated counterfactuals significantly altered the models' feature importance in many instances. Consequently, a decision was made to conduct an additional experiment focusing solely on the Adult Income dataset, where the variation was limited to one or two specific features.

This refined approach involved the DICE model with restrictions to modify only one or two features of the selected sample. In selecting the features for alteration in the counterfactual generation process, care was taken to choose features with a balanced distribution. This strategy was intended to ensure that the generated counterfactuals would remain representative of the original dataset and not skew its overall distribution.

In this section, counterfactual explanations were specifically generated by altering the 'sex' and 'education-num' features. This focused approach was intended to assess the impact of modifying these specific features on the model's performance, offering insights into how subtle changes can affect predictive outcomes while preserving the integrity of the dataset's distribution. During the generation of these counterfactuals, the DICE framework struggled to find suitable counterfactual explanations for all instances. As a result, despite the large sample sizes initially selected, the actual number of generated counterfactuals was significantly lower than anticipated. Additionally, due to the difficulty in finding counterfactuals with the specified feature restrictions, the generation process was more time-consuming compared to configurations that allowed changes to all features. Consequently, only three iterations were conducted for these experiments.

Table 13

Simple Random Forest models Trained on datasets with CFs that only altered 'sex' and 'education-num' features - Average of 3 iterations – Adult Income dataset.

Model	Iteration	Sample Size	CF Num	Fitting Data	F1	F1 std	AUC	AUC std
RF default on original data	3	0	0	31655	0,6713	0,0003	0,7719	0,0002
Simple RF 1 on original data – n:50 max_depth: 5	3	0	0	31655	0,6210	0,0030	0,7312	0,0018
Simple RF 2 on original data – n:30 max_depth: 3	3	0	0	31655	0,3811	0,0443	0,6176	0,0165
Models Trained on CFs + Original data								
RF simp 1 sample: 10% CF:1	3	3166	1	33807	0,5865	0,0342	0,7117	0,0191
RF simp 1 sample: 10% CF:2	3	3166	2	35930	0,5718	0,0416	0,7037	0,0223
RF simp 1 sample: 20% CF:1	3	6330	1	36110	0,5972	0,0030	0,7175	0,0020
RF simp 1 sample: 20% CF:2	3	6330	2	40369	0,5428	0,0532	0,6889	0,0263
RF simp 1 sample: 30% CF:1	3	9495	1	38073	0,5926	0,0135	0,7152	0,0079
RF simp 1 sample: 30% CF:2	3	9495	2	44602	0,5979	0,0117	0,7180	0,0070
RF simp 2 sample: 10% CF:1	3	3166	1	33807	0,3818	0,0665	0,6186	0,0260
RF simp 2 sample: 10% CF:2	3	3166	2	35930	0,3473	0,0658	0,6055	0,0243

RF simp 2 sample: 20% CF:1	3	6330	1	36110	0,2852	0,0326	0,5829	0,0110
RF simp 2 sample: 20% CF:2	3	6330	2	40369	0,2685	0,0100	0,5774	0,0033
RF simp 2 sample: 30% CF:1	3	9495	1	38073	0,2801	0,0255	0,5811	0,0088
RF simp 2 sample: 30% CF:2	3	9495	2	44602	0,2133	0,0359	0,5598	0,0111
Models Trained on Only CFs								
RF simp 1 sample: 10% CF:1	3	3166	1	2152	0,0902	0,0439	0,5151	0,0088
RF simp 1 sample: 10% CF:2	3	3166	2	4275	0,1225	0,0734	0,5291	0,0199
RF simp 1 sample: 20% CF:1	3	6330	1	4455	0,2573	0,1188	0,5705	0,0396
RF simp 1 sample: 20% CF:2	3	6330	2	8714	0,2176	0,1464	0,5604	0,0431
RF simp 1 sample: 30% CF:1	3	9495	1	6418	0,2608	0,1636	0,5760	0,0519
RF simp 1 sample: 30% CF:2	3	9495	2	12947	0,1306	0,0668	0,5326	0,0183
RF simp 2 sample: 10% CF:1	3	3166	1	2152	0,0037	0,0052	0,4991	0,0013
RF simp 2 sample: 10% CF:2	3	3166	2	4275	0,0027	0,0038	0,5005	0,0007
RF simp 2 sample: 20% CF:1	3	6330	1	4455	0,0118	0,0166	0,5023	0,0033
RF simp 2 sample: 20% CF:2	3	6330	2	8714	0,0000	0,0000	0,5000	0,0000
RF simp 2 sample: 30% CF:1	3	9495	1	6418	0,0004	0,0005	0,5001	0,0001
RF simp 2 sample: 30% CF:2	3	9495	2	12947	0,0059	0,0084	0,5015	0,0021

Disappointingly, counterfactual explanations generated by altering only two fairly distributed features, such as 'sex' and 'education-num,' did not enhance the predictive power of the simpler models as anticipated. This outcome suggests that while these features are represented evenly across the dataset, their modification alone may not provide sufficient variation or relevant information to significantly impact the models' ability to make accurate predictions.

4.2. Investigation of the Quality of Generated Counterfactual Explanations

The Diverse Counterfactual Explanations (DICE) framework is an innovative tool for generating counterfactual explanations, which are vital for shedding light on model behaviors and decision-making processes. In this section, the diversity and quality of the counterfactuals produced are assessed.

4.2.1. Euclidean Distance Analysis to Examine the Position of Counterfactual Explanations Relative to the Decision Boundary

Counterfactuals provide valuable insights into how minor alterations in input features can influence the predicted outcome. Thus, analyzing the distribution of counterfactuals around decision boundaries is crucial. Understanding their positioning relative to these boundaries could help evaluate their practical value in improving model performance.

In mathematics, the Euclidean distance between two points in Euclidean space is defined as the length of the line segment connecting them (Dokmanic, 2015). In this section, Euclidean distance analysis is used to examine the distribution of counterfactual explanations around decision boundaries, and same analysis is conducted for the original data points for comparison.

To compare counterfactuals and original data points, a sample from each sample size and counterfactual number pair is selected and combined into a data frame. From this data frame, a random sample matching the size of the original dataset is taken. Euclidean distances are then calculated for each data point in both the original and counterfactual datasets using the default random forest model, which was the model that used to generate the counterfactuals. Summary statistics are subsequently calculated for both datasets to facilitate comparison. It is important to note that slight variations in the summary statistics may occur each time the model is run; however, the overall pattern remains consistent.

Adult Income Dataset

Table 14:
Summary Statistics of the Original and Counterfactual Data Points' Distance to the Model Decision Boundary - Adult Income Dataset

	Original	Counterfactual
Count	45222	45222
Mean	9772	1957
Std	31555	3727
Min	0	0
25%	1282	64
50%	2977	673
75%	6047	3028
Max	1101146	247103

The count refers to the total number of computed distances from each data point to the decision boundary. When examining the mean, original data points are much farther from the decision boundary compared to counterfactual points. This indicates that, on average, the counterfactuals are significantly closer to the decision boundary than the original data points. The standard deviation reveals that the original distances exhibit much greater variability compared to the counterfactual distances, indicating a wider range of distances from the boundary.

The 25th, 50th (median), and 75th percentiles for counterfactuals are all lower than those for original data point distances, demonstrating that counterfactual points are, on average, closer to the decision boundary. Both datasets have some distances of 0, indicating that there are points on the decision boundary. However, the maximum distance for original data is significantly higher than for counterfactuals, reflecting the larger spread of distances in the original dataset.

South German Credit Dataset

Table 15:
Summary Statistics of the Original and Counterfactual Data Points' Distance to the Model Decision Boundary - South German Credit Dataset

	Original	Counterfactual
Count	1000	1000
Mean	4.57	0.93
Std	0.84	1.33
Min	0.00	0.00
25%	4.05	0.00
50%	4.49	0.00
75%	5.01	2.20
Max	8.24	4.43

The average distance from the decision boundary is significantly higher for original examples than for counterfactuals. The standard deviation indicates that distances for original examples are more consistent with less variability, while counterfactual distances vary more widely. This variability is likely due to the generation process, which places counterfactuals at various distances around the boundary. For 25% of the original data points, the distance from the boundary is 4.05 or less, while for 25% of the counterfactuals, the distance is 0.00. Additionally, the median distance for original points is 4.49, whereas for counterfactuals, it is 0.00. This suggests that half of the counterfactual points are exactly on the decision boundary, while original points tend to be further away. According to all percentile results, counterfactual points are generally much closer to the decision boundary compared to original points. The significantly lower median and quartile values for counterfactuals further support the idea that counterfactuals are specifically designed to be near the decision boundary.

Banking Dataset – Marketing Targets

Table 16:
Summary Statistics of the Original and Counterfactual Data Points' Distance to the Model Decision Boundary - Banking Dataset

	Original	Counterfactual
Count	45211.00	45211.00
Mean	857.49	38.75
Std	2479.00	65.12
Min	0.00	0.00
25%	154.24	0.00
50%	320.94	20.54
75%	733.55	44.46
Max	99456.00	1945.49

The average distance of the original data points from the decision boundary is significantly higher than that of the counterfactuals. The distances for counterfactual points are much less spread out compared to the original points, indicating less variability in how close counterfactual points are to the decision boundary. Both datasets have a minimum distance of 0.00, indicating that some points are exactly on the decision boundary for both datasets. For the lower 25% of the data, original points have a distance of at least 154.24 units from the boundary, while counterfactual points can be exactly on the boundary. The median distance for original points is 320.94 units, showing that half of the original data points are farther from the boundary compared to counterfactual points, where the median is 20.54 units. This indicates that half of the counterfactuals are within 20.54 units of the decision boundary.

For the top 25% of the data, original points are at least 733.55 units away from the boundary, while counterfactual points are, on average, 44.46 units away from the boundary. This consistent pattern across various percentiles indicates that original points are generally farther from the boundary compared to counterfactual points.

The counterfactual examples are generally much closer to the decision boundary compared to the original examples and show less variability in proximity. This outcome was anticipated, as counterfactuals are specifically designed to be near the decision boundary to illustrate how small changes in inputs can alter predictions. The larger standard deviation for original distances, observed in the Adult Income and Banking datasets, suggests more variability in how far original examples are from the decision boundary. This variability is reduced in the counterfactuals, indicating a more concentrated distribution of distances.

Only in the South German dataset, with a small difference, does the standard deviation show that counterfactual explanations are more widely distributed. This could be due to the low number of instances in this dataset. Having fewer instances can lead to less stable and less well-defined decision boundaries, as the model has less data to learn from. This can result in greater variability in the generated counterfactuals, as the model's understanding of the decision space is less precise. Nonetheless, all the other statistics of the South German dataset demonstrate that the counterfactuals are situated around the decision boundary with relatively higher variability.

In conclusion, for the Adult Income and Banking datasets, the generated counterfactuals are closer to the decision boundary with less variability compared to the original data points. In the South German Credit dataset, the generated counterfactuals are still closer to the decision boundary but exhibit higher variability.

4.2.2. Kolmogorov–Smirnov Test to Investigate the Feature Distribution Changes

To rigorously assess the quality of the counterfactuals generated by the DICE framework, several experiments were conducted across various datasets. The evaluation of the quality of the generated counterfactuals done by analyzing the differences in feature distribution between the original dataset and the dataset containing the generated counterfactuals. For this analysis, the Kolmogorov–Smirnov test (KS test) is employed. This test is commonly used in predictive models (Steinskog, 2007) to determine whether two samples originate from the same distribution. This analysis is critical because it identifies which features experienced significant changes

during the counterfactual generation process. Understanding these correlations is vital for interpreting the research findings and assessing the impact of counterfactuals on model performance within this study.

The scatter plots presented below illustrate the relationship between the Kolmogorov-Smirnov (KS) statistic results and the AUC scores of the baseline models (trained on only original data) and the models trained on augmented data with counterfactuals. These scores are calculated using the averages of five iterations. For each set of generated counterfactual explanations, KS scores were calculated, and then the mean of these five scores was taken to ensure the integrity of the analysis. In this section, for the simple random forest distillation, only the results of the first random forest (with 50 estimators and a maximum tree depth of 5) are presented, as the results of the second random forest (with 30 estimators and a maximum tree depth of 3) are remarkably aligned. However, the results of the second random forest can be found in the Appendix E.

Additionally, to gain more insight into the generated counterfactuals, an analysis that compares the feature importances of a Random Forest model trained on the original data versus the expanded data to observe how the addition of counterfactuals affects the relative importance of specific features. By analyzing differing instances, the results help understand how counterfactuals influence model predictions and which features' importance changes significantly with the expanded dataset. Given that all the generated counterfactuals showed almost indistinguishable feature importance shifts, only two plots (with 12,000 samples and either 1 or 2 counterfactuals) are included in Appendix D and all the other results can be found at the bottom of the Jupiter Notebook of Adult Income.

Adult Income Dataset

Figure 1: Scatter Plot of Logistic Regression AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

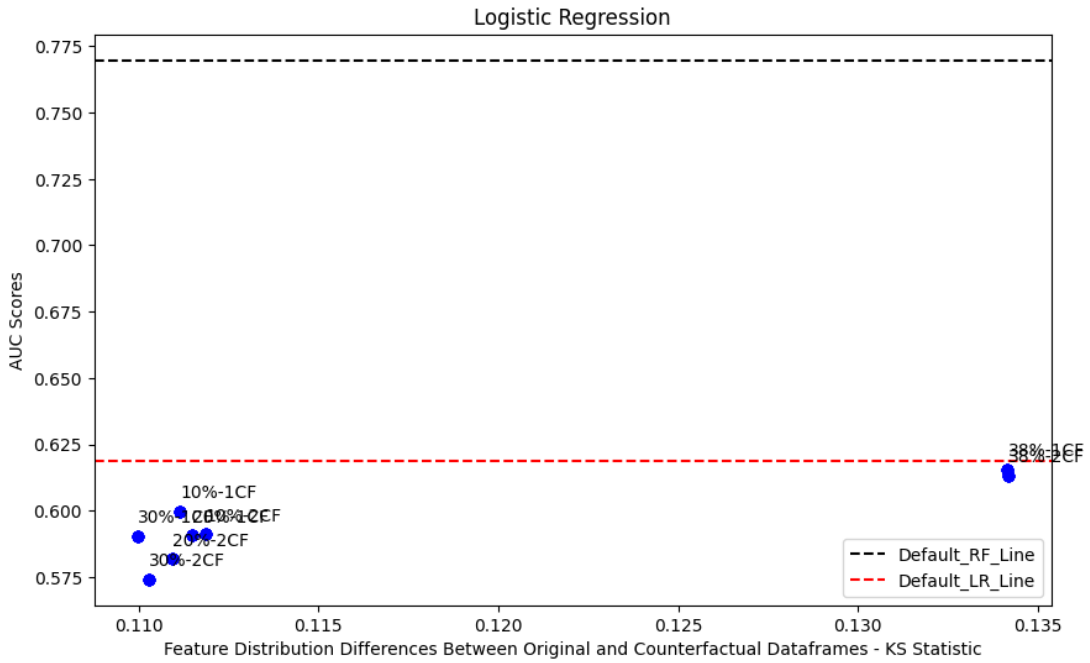


Figure 2: Scatter Plot of Decision Tree AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

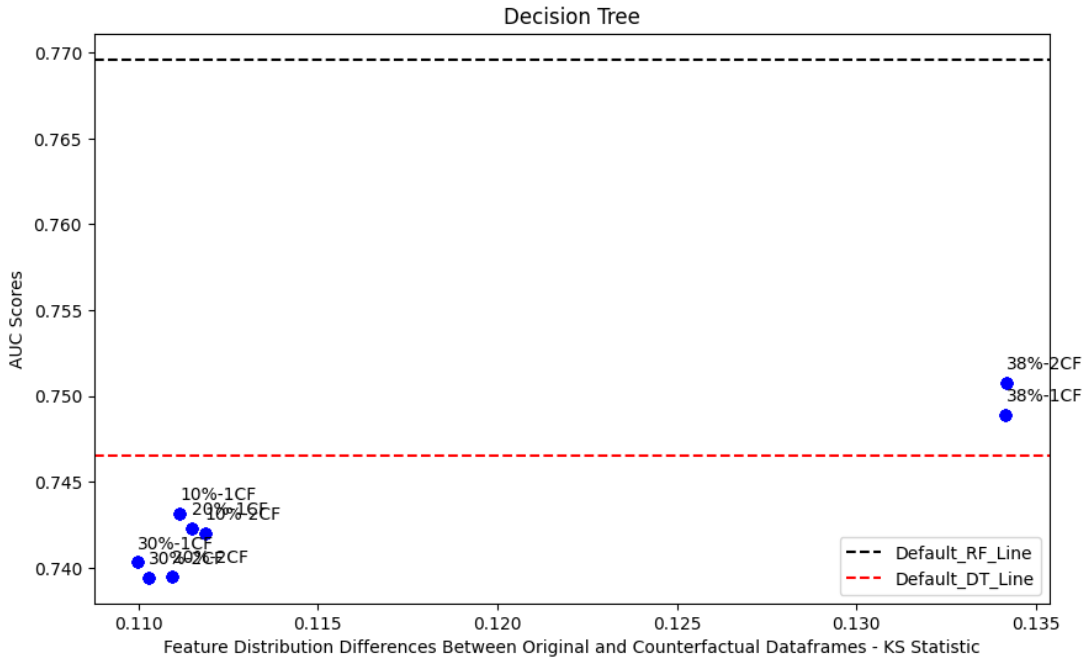
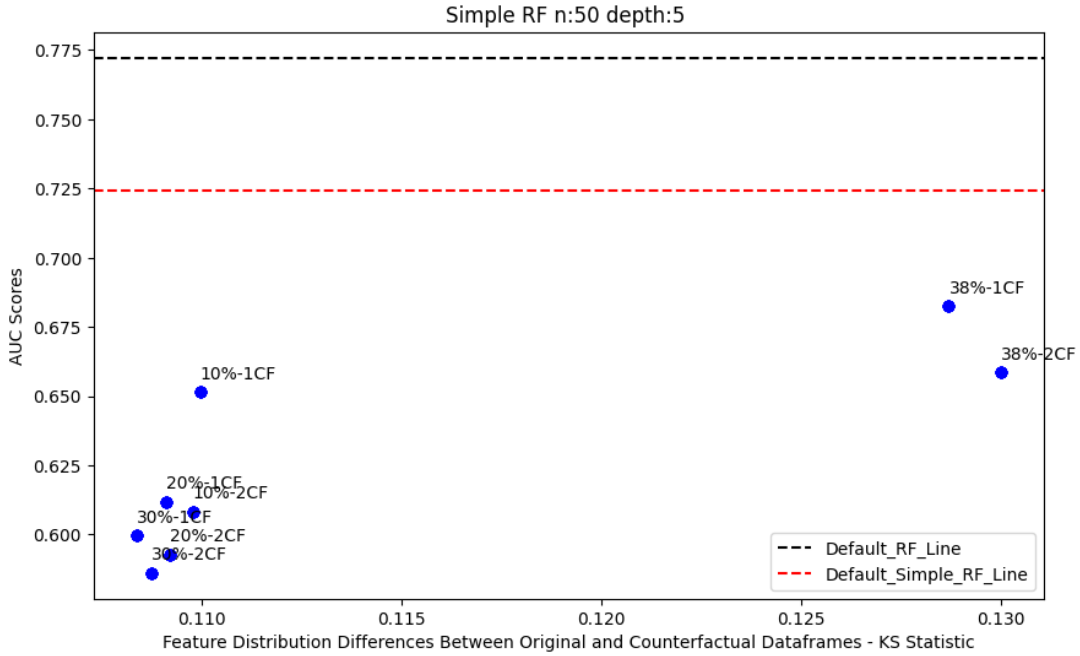


Figure 3: Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic



In the Adult Income dataset, augmented datasets generated from a sample of 38% of the original dataset exhibit greater differences in feature distribution compared to other counterfactuals. This discrepancy is attributed to the dataset's imbalanced target variable. Despite efforts to maintain the same balance in the target variable across all sets of generated counterfactuals, the dataset lacked sufficient cases from the minority class to both preserve the target variable distribution and generate counterfactuals with the opposite class. Unexpectedly, these datasets achieved higher AUC scores across all models. However, for the logistic regression and decision tree models, the AUC score differences between these datasets and others are less than 0.05, and for the simple random forest model, the differences are less than 0.1. These differences are minor, indicating that, despite the higher feature distribution differences leading to improved AUC scores, the overall improvements in model performance are not substantial.

South German Credit Dataset

Figure 4: Scatter Plot of Logistic Regression AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

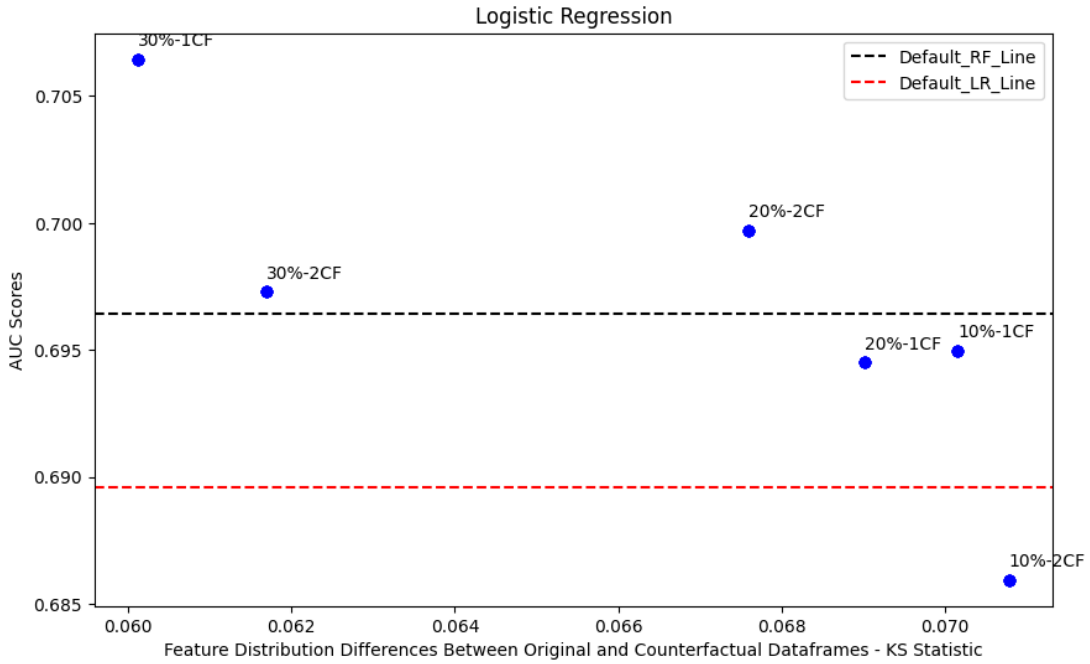


Figure 5: Scatter Plot of Decision Tree AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

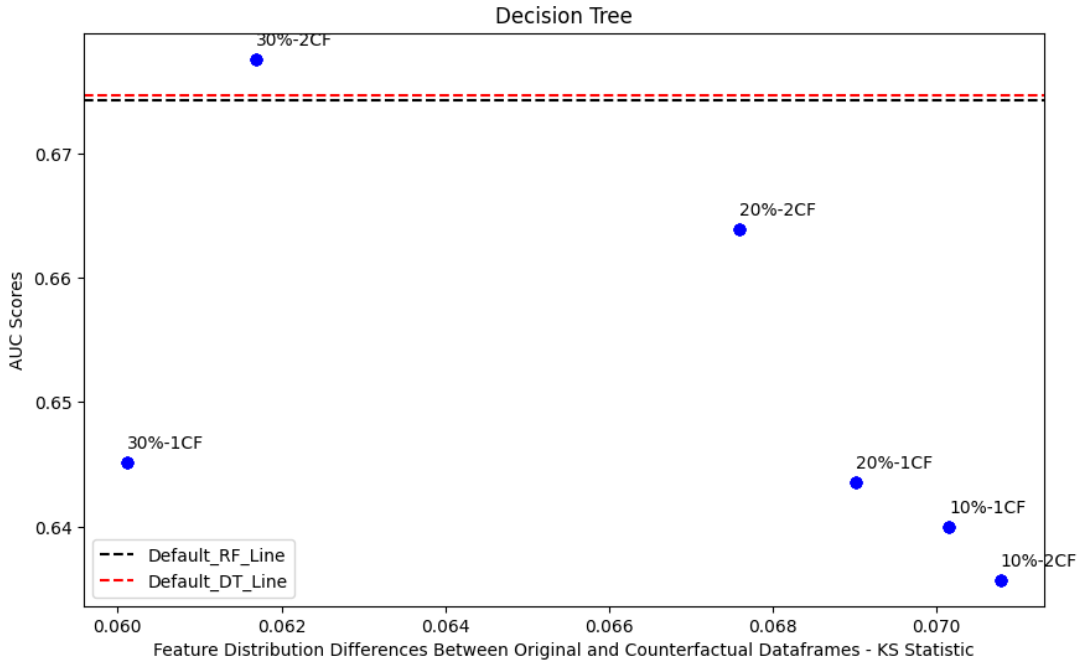
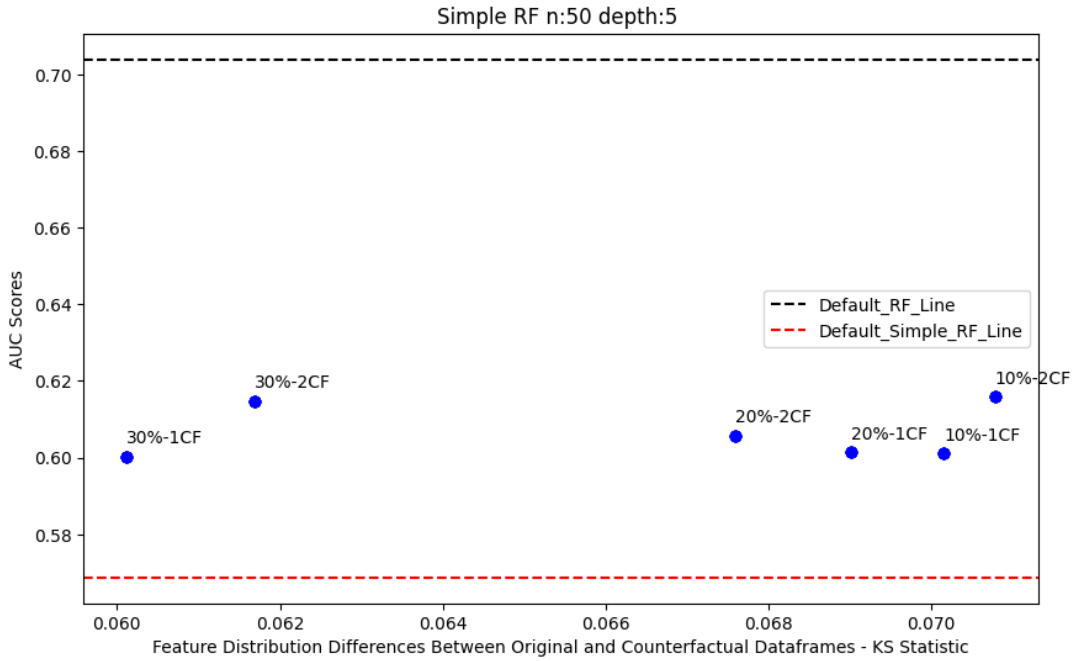


Figure 6: Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic



For the South German Credit dataset, smaller sample sizes resulted in higher feature distribution differences. This outcome was expected due to the dataset's relatively small size of 1,000 rows. For instance, using 10% of the training dataset meant having only 70 examples to generate the counterfactuals, which couldn't adequately capture the feature distribution of the entire dataset. Conversely, larger sample sizes included more examples, which better imitated the feature distribution and resulted in lower KS statistics.

In logistic regression distillation, cases with 30% sample sizes performed better than the other augmented datasets, but the AUC score differences remained below 0.02. Although the results may appear better, they do not display significant improvements. Only in the simple random forest case, all scenarios, regardless of feature distribution differences, did perform better. This improvement was expected since the simple model trained on the original dataset already had a very low AUC score of 0.56, making it more likely that additional examples would enhance model performance.

Banking Dataset

Figure 7: Scatter Plot of Logistic Regression AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

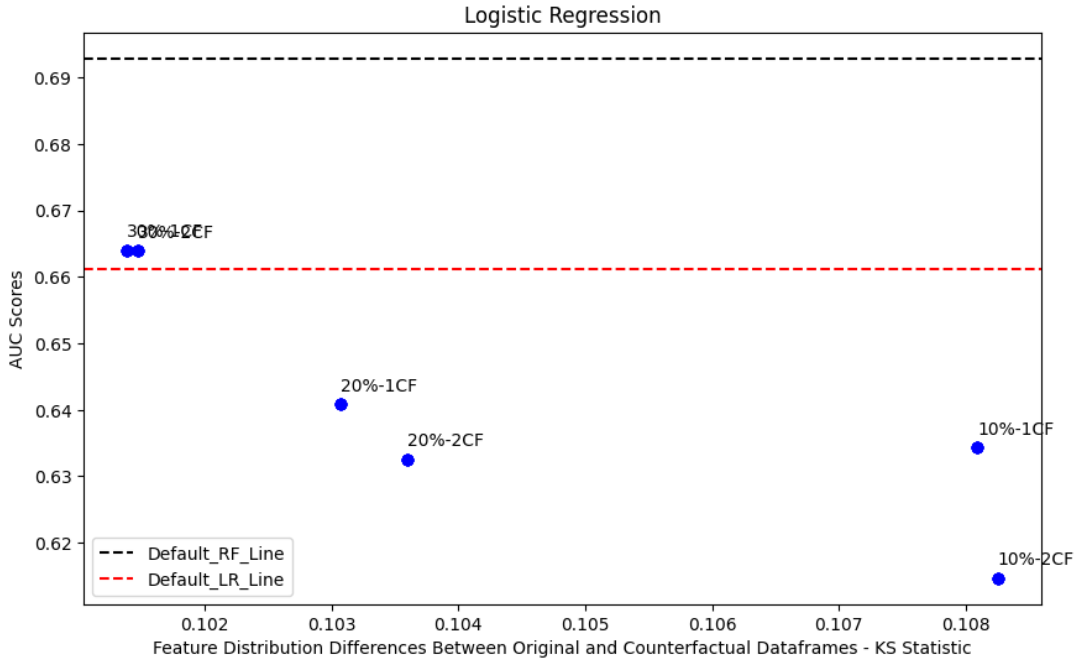


Figure 8: Scatter Plot of Decision Tree AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

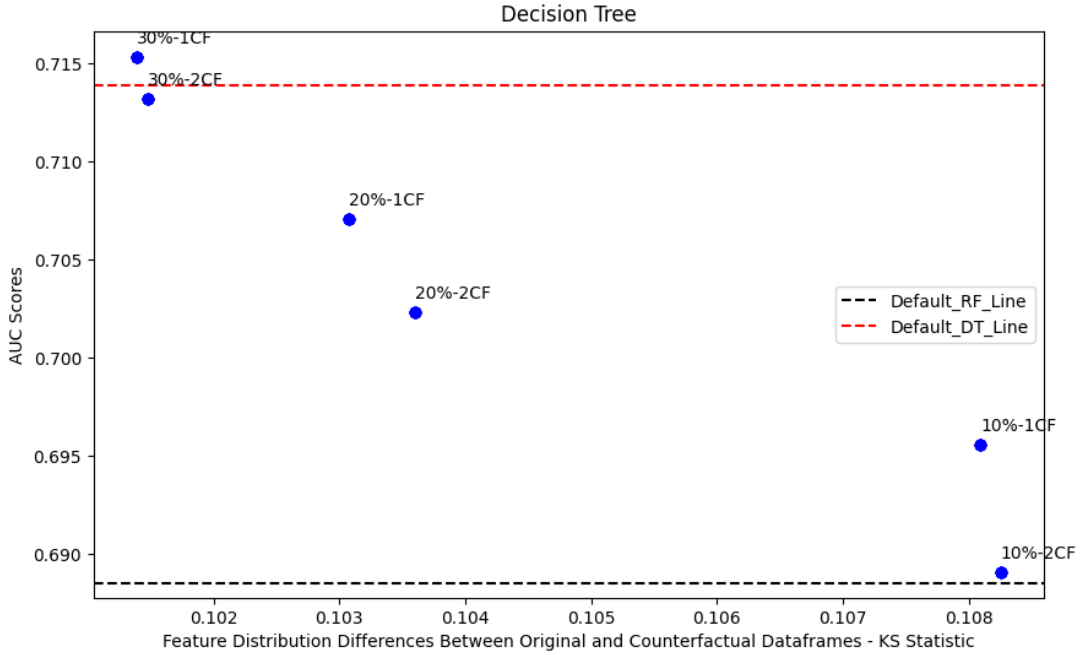
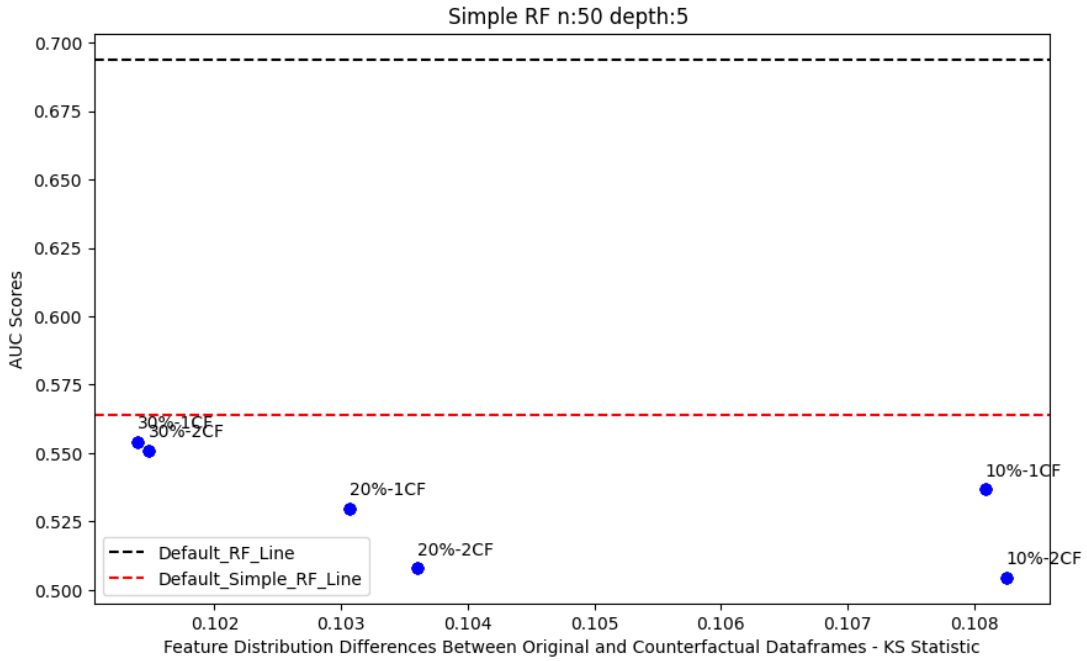


Figure 9: Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic



Similar to the South German Credit dataset, in the Banking dataset, higher percentages of samples resulted in lower feature distribution differences. This can be explained by the fact that larger samples better capture the feature distribution of the original dataset. As with the other datasets, cases with higher sample sizes performed better in terms of AUC scores. However, the AUC scores from models trained on the augmented datasets are still not significantly better than those of models trained on the original dataset.

For logistic regression, cases with lower sample sizes significantly decreased the AUC score. This decrease may be due to the addition of a high number of new instances generated from only a small part of the dataset, resulting in the model having less predictive power on more diverse examples.

5. Discussion

The impact of adding counterfactual explanations to the training datasets for decision trees, logistic regression, and simple random forests are investigated in this paper. Counterfactuals that switched the target variable to the opposite class were created using the DICE framework, and their impact on model performance was investigated on three different datasets.

The use of counterfactual explanations in re-training logistic regression, decision trees, and simpler random forest models demonstrated nuanced, dataset-specific effects on model performance. While the integration of counterfactuals occasionally led to performance enhancements, it more frequently resulted in outcomes that were inferior to baseline models trained on original datasets. This variability indicates that while counterfactuals have the potential to improve model predictive power, their effectiveness is critically dependent on the proportion of counterfactuals incorporated and their similarity with the original data distribution.

5.1. Answers to Sub Research Questions

What are the state-of-the-art computational frameworks for generating counterfactual explanations?

There are numerous frameworks for generating counterfactual explanations, distinguishable by their methodologies, characteristics, the nature of the counterfactuals they produce, and the types of data they accommodate (Guidotti R., 2022). Each framework represents a unique advancement in the field of counterfactual explanation generation, tackling distinct challenges and enhancing various aspects over prior methods. In selecting a framework for this research, factors such as the model's compatibility with categorical and tabular data were crucial. Additionally, the prominence of the framework's documentation in academic literature was also a consideration. In the end, DICE model chosen for this research. The table below presents

a comparison of state-of-the-art counterfactual generation models that were considered, outlining their key properties and capabilities.

Table 17: State-of-the-art Counterfactual Generation Models

Name	Strategy	Model Agnostic	Data Agnostic	Categorical	Causality	Paper Cited By	Multiple
DICE	OPT	DIF	TAB	+		925	+
C-CHAVE	OPT	+	TAB	+		166	+
SYNTH	OPT	DIF	TAB	+		33	+
CEML	OPT	+	TAB	+		18	
MACE	OPT	+	TAB	+		298	+
CADEX	HSS	DIF	TAB	+	+	39	
CERTIFAI	HSS	+	+	+		109	+
MOC	HSS	+	TAB	+		260	+
VICE	HSS	+	TAB	+		81	
NNCE	IB	+	TAB	+	+	98	+

Note: Strategy adopted: Optimization (OPT), Heuristic Search Strategy (HSS), Instance-Based (IB). If it is not model agnostic, then it is able to explain with Differentiable (DIF) classifier. If it is not data agnostic, it is able to process Tabular (TAB). A plus occurs if it can handle categorical features, causality, and if it returns multiple counterfactuals.

How do the generated counterfactual explanations distribute around the decision boundaries?

Analyzing the distribution of counterfactual explanations around decision boundaries provides valuable insights into model performance. This analysis employs Euclidean distance to compare the proximity of original data points and counterfactuals to the decision boundaries across different datasets.

In the Adult Income Dataset, counterfactuals are significantly closer to the decision boundary (mean distance: 1957) compared to original data points (mean distance: 9772). The standard deviation of distances is also lower for counterfactuals (3727) than for original points (31555), indicating less variability and a more concentrated distribution near the boundary. The percentiles further support that counterfactuals are generally nearer to the decision boundary.

For the South German Credit Dataset, counterfactuals remain closer to the boundary (mean distance: 0.93) but show higher variability (standard deviation: 1.33) compared to original points (mean distance: 4.57, standard deviation: 0.84). Notably, many counterfactuals are exactly on the decision boundary, as indicated by the 25th and 50th percentiles.

In the Banking Dataset, counterfactuals are much closer to the decision boundary (mean distance: 38.75) with less variability (standard deviation: 65.12) than the original data points (mean distance: 857.49, standard deviation: 2479). The percentile values consistently indicate that counterfactuals are situated nearer to the decision boundary.

Overall, counterfactuals are designed to be close to the decision boundary, illustrating how small changes in input features can alter predictions. This is evident across the datasets, with counterfactuals generally showing less variability in their distances to the boundary compared to original data points. The only exception is the South German Credit Dataset, where the higher variability of counterfactual distances is likely due to the smaller dataset size, resulting in less stable sample sets.

How does the inclusion of counterfactual explanations generated by a complex non-explainable model in the training dataset influence the accuracy of an intrinsically explainable model?

The inclusion of counterfactual explanations generated by a complex, non-explainable model (like a random forest) in the training dataset of an intrinsically explainable model (such as logistic regression or decision trees) can influence the accuracy of the simpler, more explainable models both positively and negatively.

Counterfactual explanations provide insights into the decision-making processes of complex models, helping to refine and sharpen the decision boundaries of simpler models. By highlighting scenarios where small changes to input features result in a different classification, counterfactuals can aid simpler models in capturing nuanced patterns that might not be evident from the original dataset alone. When effectively aligned with the underlying data distribution and integrated in a balanced manner, counterfactuals can help simpler models

generalize better to new, unseen data. This is because generated counterfactual explanations often lie very close to the decision boundary, resulting in a clearer and more defined decision boundary.

On the other hand, if counterfactuals do not accurately represent the true data distribution, there is a risk that they could lead the simpler model to overfit to these examples. This can reduce the model's ability to perform well unseen data, particularly if the counterfactuals are not properly capturing the characteristics of the original dataset. They might reflect the biases or specific decision-making patterns of these models rather than true, underlying data relationships. This can mislead simpler models, especially if the counterfactuals provide more examples of patterns that already exist in the original dataset. While simpler models are valued for their transparency and ease of interpretation, integrating counterfactuals often complicates the model performance or the model itself, resulting in lower AUC and F1 scores. This can be explained by counterfactual explanations' inability to represent edge cases or less common scenarios that may not be sufficiently sampled from the original training data.

In this research, it was observed that the inclusion of counterfactual explanations generated by complex models most often lowered the predictive power of the simpler, intrinsically explainable models. The overall impact on the accuracy of these simpler models depends critically on how well the counterfactuals are integrated and their representativeness of actual or potential data scenarios.

How can the proposed model distillation method using counterfactual explanations be demonstrated for high-stake use cases such as credit risk and insurance prediction?

To effectively demonstrate the proposed model distillation method using counterfactual explanations in high-stakes sectors like credit risk and insurance prediction, a streamlined approach is essential. Complex models known for their high accuracy in these areas are chosen to generate meaningful counterfactual explanations. These explanations are designed to identify minimal yet realistic changes to input features that would alter predictions, ensuring that the counterfactuals are both practical and compliant with ethical and legal standards.

Once generated, the counterfactuals are integrated into simpler, more transparent models such as logistic regression or decision trees. This integration aims to enhance the decision-making capabilities of the simpler models by imbuing them with deeper insights gleaned from the more complex models while maintaining their inherent explainability.

The effectiveness of these distilled models is then should rigorously evaluated. Improvements in predictive power should be measured, and fairness should be assessed to ensure no discriminatory biases are introduced, and robustness should be tested to validate the model's performance under various conditions.

5.2. Interpretation of the Results

This study on the integration of counterfactual explanations into logistic regression, decision trees, and simpler random forests training, highlights the complicated balance between model complexity, dataset characteristics, and the nature of training data. Counterfactuals have shown potential in enhancing model understanding and refining decision-making processes under specific conditions. However, their effectiveness is not uniform across different models and datasets, indicating that their use must be carefully calibrated.

Logistic Regression

The findings from the logistic regression models across the Census Adult Income, South German Credit, and Banking datasets illustrate a primarily negative impact when counterfactuals form a substantial part of the training data. This suggests that while counterfactuals can potentially align well with a model's decision boundaries in certain contexts, they often fail to provide a comprehensive representation of the underlying data distribution. This could be attributed to the possibility of overfitting or a misalignment with true data characteristics, especially when used exclusively.

Decision Trees

For decision trees, the results were somewhat more promising under specific conditions, particularly with higher proportions of counterfactuals. This indicates that counterfactuals can enhance the decision-making capabilities of simpler models like decision trees when they are blended judiciously with original data. However, the performance decline in scenarios where decision trees were trained solely on counterfactuals, which underscores the importance of a balanced training dataset that includes both original and synthetic data, or synthetic data set generated from high proportion of the original dataset.

Simpler Random Forest Models

The performance of simpler random forest models also reflected the challenges of utilizing counterfactuals effectively. While there were glimpses of slightly improved performance, particularly with certain configurations and higher proportions of counterfactuals, the overall efficacy was generally below that of more complex models. This highlights the sensitivity of simpler models to the depth and complexity of training data and suggests that the robustness required to handle enriched datasets is somewhat lacking in reduced complexity models.

5.3. Conclusion

The main research question: *How can counterfactual explanations be effectively utilized in model distillation for classification models to enhance both interpretability and accuracy?*

By strategically integrating counterfactual explanations into the training process, classification models can achieve enhanced interpretability without significant loss in accuracy. The key lies in balancing the amount of counterfactual data, selecting appropriate features for modification, and continuously evaluating and refining the approach based on performance metrics. Integrating counterfactuals with original data must be done judiciously. Counterfactuals should constitute a significant portion of the training dataset. Experimenting with different proportions, such as 10%, 20%, 30%, or higher percentages, can help determine the optimal balance. This balanced integration ensures that the synthetic data enhances the model's learning process without overshadowing the genuine data, capturing key patterns without overfitting, and maintaining the model's ability to generalize effectively.

The selection and modification of features for generating counterfactuals is another critical aspect. Focusing on features that are evenly distributed and have significant predictive power could help maintain the integrity of the original dataset while providing insightful variations. For instance, modifying only the features like 'sex' and 'education-num' in the Adult Income dataset can yield useful counterfactuals that offer valuable insights into model behaviour without distorting the data distribution. However, the DICE model did not generate a sufficient number of counterfactuals by altering only two features, as it could not find viable counterfactual scenarios. Even if it had generated counterfactuals by altering only two features, this approach might result in lower predictive power due to the generation of many similar data points and a lack of diversity in the cases predicted.

Regular evaluation of the impact of counterfactuals on model performance is essential. Using metrics like the F1 score and AUC and conducting tests like the Kolmogorov-Smirnov (KS) test and Euclidean distance analysis to compare data point and feature distributions, can reveal how counterfactuals influence the model and how it can get better.

Considering the complexity of the model being distilled is vital. Simpler models, such as logistic regression and decision trees, may benefit differently from counterfactuals compared to more complex models. Simplified random forests, for example, might need a more nuanced integration of counterfactuals to effectively leverage their predictive power. It's also crucial to be mindful of potential overfitting or the introduction of noise. Counterfactuals should be realistic and actionable, avoiding biases or misrepresentations that could degrade model performance. Regularization techniques can help mitigate these risks.

In conclusion, the findings suggest that counterfactual explanations can either enhance or diminish the predictive power of machine learning models, depending on the dataset, the generated counterfactuals, and the models used. In many cases, incorporating counterfactual explanations into the training dataset tended to reduce the predictive power of simpler models. This outcome can be attributed to the nature of counterfactual explanations, which often align closely with decision boundaries, leading to a higher number of similar instances and a diminished ability of the models to capture more diverse patterns. Achieving a balance in the amount of counterfactual data, selecting appropriate features for modification, reducing the proximity of counterfactuals to the decision boundary, and continuously evaluating and refining the process are essential for developing more reliable models.

5.4. Recommendations

Based on the findings of this thesis, several recommendations can be made to enhance the integration of counterfactual explanations into model training processes. These recommendations aim to optimize the balance between model complexity, dataset characteristics, and the proportion of counterfactuals to achieve improved performance and reliability:

Careful Calibration of Counterfactual Proportions: Different models and datasets respond uniquely to the inclusion of counterfactuals. It is crucial to experiment with various proportions to determine the optimal balance that enhances model performance without leading to overfitting or reduced generalizability. For instance, while logistic regression models might benefit from a lower proportion of counterfactuals, decision trees might require a higher proportion to see performance gains.

Hybrid Training Datasets: Models trained exclusively on counterfactuals tend to underperform compared to those trained on a mix of original and synthetic data, in many cases. Therefore, it is advisable to create hybrid datasets that include both original data and counterfactuals. This approach ensures that the model retains the ability to generalize across the broader data distribution while still benefiting from the refined decision boundaries provided by counterfactuals.

Incremental Integration Strategy: Gradually increasing the volume of counterfactuals in the training dataset can help in understanding their impact on the model's performance. This incremental approach allows for continuous monitoring and adjustment of the counterfactual ratio, minimizing potential overfitting risks and identifying the threshold at which counterfactuals begin to diminish returns.

Proximity Adjustments: Since the generated counterfactual explanations lie very close to the decision boundary, the model's predictive power on uncommon cases might be reduced. Conducting more experiments with counterfactuals generated at a lower proximity to the decision boundary, which can be adjusted in the DICE model, could potentially yield better results.

Enhanced Model Robustness**:** For simpler models, particularly those with limited complexity such as simpler random forests, strategies should be developed to enhance their robustness when handling enriched datasets. This might include techniques such as feature engineering, regularization, and depth adjustments to better accommodate the nuances introduced by counterfactuals.

Diverse Dataset Testing**:** Since the impact of counterfactuals varies significantly across different datasets, it is recommended to test these strategies across a wide range of datasets. This will help in developing more generalized strategies that are effective across various domains and data characteristics.

Tool Development for Counterfactual Generation and Integration**:** In the development of tools for generating and integrating counterfactuals, particularly in machine learning applications, the configuration of counterfactual generation plays a crucial role. Configuring the generation process involves adjusting various parameters that influence how counterfactuals are constructed. Adjusting the proximity, diversity, parameter, or properly managing constraints could be essential for ensuring that the generated counterfactuals are close to the original instances in the feature space. By fine-tuning these configurations, the tool for counterfactual generation can be optimized to produce more effective and applicable counterfactuals.

6. Limitations

This thesis, while providing valuable insights into the integration of counterfactual explanations into machine learning models, operates within several constraints that may affect the generalizability and applicability of its findings:

Randomness of Counterfactuals (CFs): The generation of counterfactuals is inherently subject to randomness, influenced by the initial conditions and configurations of the model used to create them. This randomness can lead to variations in the quality and relevance of the counterfactuals produced, potentially impacting their effectiveness for training subsequent models. Identical conditions across different runs often result in varying counterfactuals, contributing to inconsistencies in training outcomes across experiments. Furthermore, each new batch of counterfactual explanations generally generated different model performance metrics, suggesting that extensive experimentation is necessary to derive more reliable conclusions. This variability underscores the need for robust experimental designs to ensure that findings are both accurate and generalizable.

Complexity and Scope of Experiments**:** The scope of experiments conducted as part of this thesis was extensive, covering multiple models and datasets. However, the complexity of these experiments resulted in only a limited number of configurations testing within each model-dataset combination. There are potentially many more experiments that could be conducted to explore additional configurations of counterfactual proportions, model parameters, and hybrid data setups. Each additional experiment could provide further insights into the nuanced effects of counterfactuals on model performance.

Time Constraints**:** The entire project was constrained by a time limit of 2-3 months for completion. This time frame limited the depth of exploration into each model and dataset configuration, as well as the ability to iteratively refine the experimental design based on preliminary results. A longer timeline might allow for a more thorough investigation with different sets of counterfactuals and their impact on model stability and drift.

Hardware Limitations**:** Hardware constraints impacted the number of iterations and the extent of hyperparameter tuning that could be feasibly performed, potentially affecting the optimization of model performance.

Long Process of Counterfactual Generation: Generating counterfactuals is a computationally demanding and time-consuming task, particularly when aiming to produce high-quality counterfactuals that are diverse and accurately reflect real-world data distributions. For larger datasets, creating samples of 30% to 40% may involve generating up to 24,000 counterfactual explanations, a process that can take an entire day. This extensive time requirement can significantly impact the overall efficiency of preparing training datasets and the research project as a whole. Additionally, it is important to note that some sets of counterfactuals may randomly yield better or worse predictive performance, highlighting the necessity of conducting same counterfactual generation process multiple times to ensure robust results. In this research results of 5 iterations were presented.

Exploring Uncharted Territory: This research delves into the relatively unexplored territory of using counterfactuals for model distillation within machine learning. The novelty of the research presents certain limitations, also due to the sparse preliminary analyses available in the literature. While most existing studies on counterfactual explanations have concentrated on non-tabular data, such as in image and text processing applications, there remains a substantial void in systematic methodologies and proven best practices for tabular data.

These limitations underscore the challenges involved in conducting comprehensive research within a constrained environment and suggest areas for improvement in future studies. Addressing these limitations could involve expanding the scope of experiments, extending the project timeline, enhancing computational resources, and refining the methodologies for generating and utilizing counterfactuals. Such improvements could lead to more robust findings and a deeper understanding of the potential and limitations of using counterfactual explanations in machine learning.

7. Supplementary Documentation and Consistency Notes

In the documentation submitted for the algorithm, distinct notebooks are provided for each dataset, dedicated to different aspects of the model distillation process. These notebooks systematically organize and present the results, generated counterfactual explanations, and corresponding plots for clarity and ease of review. It is important to note that within the submitted documentation, the notebooks pertaining to the Adult Income Dataset are thoroughly commented, providing detailed explanations of the code. For the other two datasets, aside from the dataset cleaning portion, the remainder of the code is almost identical, ensuring consistency in the methodology applied across different datasets.

It is important to note that re-running the codes may yield different results from those presented in this paper due to variations in the counterfactual explanations generated anew. For all experiment results, please refer to the GitHub repository of this research: [https://github.com/dideuzun/Thesis_Project.git].

References

Ahsan, M. M. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*(9(3)), 52. Retrieved from https://www.mdpi.com/2227-7080/9/3/52

Alexandropoulos, S. A. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, 34.

Ayer, T. C. (2010). Comparison of logistic regression and artificial neural network models in breast cancer risk estimation. *Radiographics, 30(1)*, 13-22. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3709515/

Barbaglia, L. M. (2023). Forecasting loan default in Europe with machine learning. *Journal of Financial Econometrics*, 569-596.

Becker, B. a. (1996). *Adult.* Retrieved from UCI Machine Learning Repository: https://doi.org/10.24432/C5XW20

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 32. Retrieved from https://doi.org/10.1023/A:1010933404324

Charbuty, B. &. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends, 2(01)*, 20-28. Retrieved from https://www.jastt.org/index.php/jasttpath/article/view/65

Díaz-Uriarte, R. &. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics, 7*, 1-13. Retrieved from https://link.springer.com/content/pdf/10.1186/1471-2105-7-3.pdf

Dokmanic, I. P. (2015). Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine, 32(6)*, 12-30.

Ferrario, A. &. (2022). The Robustness of Counterfactual Explanations Over Time. *IEEE Access, 10*, 82736-82750.

Gislason, P. B. (2006). Random Forests for land cover classification. *Pattern Recognition Letters*(27), 294-300. Retrieved from https://doi.org/10.1016/J.PATREC.2005.08.011

Gou, J. Y. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 1789-1819.

Guidotti, R. (2022). Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 1-55.

Guidotti, R. M. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR), 51(5)*, 1-42.

Huang, J. L. (2015). An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Information and software Technology*(67), 108-127. Retrieved from https://hal.science/hal-01340341/file/IST_hal_submission.pdf

Huang, Y. C. (2019). Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*(32). Retrieved from

https://proceedings.neurips.cc/paper_files/paper/2019/file/093f65e080a295f8076b1c5722a46aa2-Paper.pdf

Jeanneret, G. S. (2024). Text-to-Image Models for Counterfactual Explanations: a Black-Box Approach. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, (pp. pp. 4757-4767).

Jordan, M. I. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349(6245)*, 255-260.

Karimi, A. H. (2022). A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys, 55(5)*, 1-29.

Kohavi, R. (1996). *Census Income.* Retrieved from UCI Machine Learning Repository: https://doi.org/10.24432/C5GP7S

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. *Entropy, 23(1)*, 18.

Liu, D. C. (2023). KDCRec: Knowledge distillation for counterfactual recommendation via uniform data. *IEEE Transactions on Knowledge and Data Engineering, 35(8)*, 8143-8156. Retrieved from https://drive.google.com/file/d/1h2KkYcLh4CIiptdlx5Ft15VaODacphhb/view

Moro, S. R. (2012). *Bank Marketing.* Retrieved from UCI Machine Learning Repository: https://doi.org/10.24432/C5K306

Mothilal, R. K. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. *In Proceedings of the 2020 conference on fairness, accountability, and transparency*, (pp. pp. 607-617).

Pawelczyk, M. B. (2021). Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint arXiv:2108.00783*.

Rodriguez-Galiano, V. G.-O.-S. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *Isprs Journal of Photogrammetry and Remote Sensing, 67*, 93-104. Retrieved from https://doi.org/10.1016/J.ISPRSJPRS.2011.11.002

Seger, C. (2018). *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.* STOCKHOLM. Retrieved from https://www.diva-portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf

South German Credit. (2019). Retrieved from UCI Machine Learning Repository: https://doi.org/10.24432/C5X89F

Steinskog, D. J. (2007). A cautionary note on the use of the Kolmogorov–Smirnov test for normality. *Monthly Weather Review, 135(3)*, 1151-1157. Retrieved from https://journals.ametsoc.org/view/journals/mwre/135/3/mwr3326.1.xml?tab_body=fulltext-display

Stepin, I. A.-F. (2021). A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access, 9*, 11974-12001.

Tahirovic, E. &. (2023). Interpretability and Explainability of Logistic Regression Model for Breast Cancer Detection. *In ICAART (3)*, 161-168.

Tan, S. C. (2018). Distill-and-compare: Auditing black-box models using transparent model distillation. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* , (pp. 303-310).

Verma, S. B. (2020). Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010*, 10596.

Wachter, S. M. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech., 31*, 841.

Watson, D. (2020). Conceptual challenges for interpretable machine learning. *Synthese, 200*, 1-33.

Appendix A – Adult Income Dataset – Feature Details

Table 18: Detailed Information about the features of Adult Income Dataset

Feature Name	Feature Type	Feature Description
Age	**Numeric**	Represents the age of the individual. The most frequent ages are 36, 33, 31, and 35, each occurring in more than 1270 instances. The least frequent age is 80, appearing in 11 instances.
Workclass	**Categorical**	Indicates the type of employment and includes 7 different classes such as 'Private', 'Never worked' etc. The most common category is 'Private,' with 33,307 instances, significantly outnumbering the second most frequent category, 'Self-employed-not-incorporated,' which has 3,796 instances. The least frequent category is 'Without pay,' with only 21 instances.
Fnlwgt	**Continuous**	Final weight, a census-computed variable used to adjust the data.
Education	**Categorical**	Indicates the highest level of education attained by the individual, encompassing 16 different levels, including categories such as Bachelors, Some-college, 11th, HS-grad etc. The most frequent category is 'HS-grad,' with 14,783 instances. The least frequent category is 'Preschool,' with only 72 instances.
Education-num	**Numerical**	This is an ordinal numeric representation of education, aligned identically with the education column, suggesting that both features capture the same information. The most frequent value is '9,', matching the 'HS-grad' count in the education category.
Marital-status	**Categorical**	Reflects the marital status of the individual, including 7 different categories such as Married-civ-spouse, Divorced, Never-married etc. The most frequent category is 'Married-civ-spouse' with 21,055 instances, and the least frequent is 'Married-AF-spouse' with only 32 instances.
Occupation	**Categorical**	Describes the type of occupation with 14 different categories, including Tech-support, Craft-repair, Exec-managerial, Prof-specialty, Handlers-cleaners, etc. The most common occupation is 'Craft-repair' with 6,020 instances, while the least frequent is 'Armed-forces' with only 14 instances.
Relationship	**Categorical**	Details the individual's relationship status within 6 different categories, such as Wife, Own-child, Husband, Not-in-family etc. The two most frequent categories are 'Husband' and 'Not-in-family,' each with over 11,700 instances, while the less frequent ones are 'Wife' and 'Other-relative,' each with less than 2,100 instances.
Race	**Categorical**	Indicates the individual's race, with categories including White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, and Black. 'White' is the most frequent with 38,903 instances.
Sex	**Categorical**	Represents the gender of the individual, with 30,527 instances identified as male and 14,695 as female.
Capital-gain	**Continuous**	Reflects income from capital gains across 121 different values. This feature predominantly centres around the value '0', with 41,432 instances recording no capital gain.
Capital-loss	**Continuous**	Shows income from capital losses, featuring 97 different values. Similarly to capital gain, this attribute is largely concentrated around the value '0', with 43,082 instances showing no capital loss.

Hours-per-week	**Numeric**	Indicates the number of hours worked per week by the individual, spanning 96 different values. The most frequent value is '40' hours, noted in 21,358 instances, which significantly differs from the second most frequent value of '50' hours, observed in 4,094 instances.
Native-country	**Categorical**	Represents the individual's country of origin, encompassing 41 different categories including the United States, Cambodia, England, Puerto Rico, Canada, Germany, and others. This feature is highly imbalanced, with 91.3% of the instances originating from the 'United-States'.

Table 19: Adult Income Dataset Descriptive Statistics of Numerical Columns

	Count	Mean	Standard Deviation	Min	Max
Age	45222.0	38.55	13.22	17.0	90.0
Fnlwgt	45222.0	189734.73	105639.20	13492.0	1490400.0
Education-Num	45222.0	10.12	2.55	1.0	16.0
Capital-gain	45222.0	1101.43	7506.43	0.0	99999.0
Capital-loss	45222.0	88.60	404.96	0.0	4356.0
Hours-per-week	45222.0	40.94	12.01	1.0	99.0
Income	45222.0	0.25	0.43	0.0	1.0

Appendix B – South German Credit Feature Details

Table 20: Detailed Information about the features of South German Credit Dataset

Feature Name	Feature Type	Feature Description
Status	**Categorical**	This feature is status of existing checking account. There are 4 different categories such as (< 0 DM), (0 <= ... < 200 DM), and (no checking account).
Duration	**Continuous**	This feature is the duration of the credit in months. There are 33 different values, 24 and 12 are being the most frequent ones.
Credit_history	**Categorical**	This feature is history of compliance with previous or concurrent credit. There are 5 different categories, most frequent ones are being (no credits taken/all credits paid back duly), and (all credits at this bank paid back).
Purpose	**Categorical**	This feature is the purpose for which the credit is needed. There are 10 different categories, furniture\equipment being the most frequent and vacation is the least frequent.
Amount	**Continuous**	Credit amount in discount margin (actual data and type of transformation unknown).
Savings	**Categorical**	Savings in account/bonds, categories such as (< 100 DM), (500 <= ... < 1000 DM), (unknown/no savings account) where (unknown/no savings account) being the most frequent with 603 instances.
Employment duration	**Categorical**	This feature represents the present employment duration since, including categories like (unemployed), (< 1 year) and (>= 7 years). Most frequent one being (1 <= ... < 4 years) with 339 instances.
Instalment rate	Numerical	The instalment rate as a percentage of disposable income. Most frequent category is (< 20) with 476 instances.
Personal status sex	**Categorical**	This feature is combining information about gender and marital status of the individuals with 4 different categories such as (male: divorced/separated), and (female: single) where male: married/widowed being the most frequent one with 548 instances.
Other debtors	**Categorical**	This feature has 3 different categories explaining whether another debtor or a guarantor for the credit is there or not. 'None' is the most frequent category with 907 instances.
Present residence	**Continuous**	This feature is the length of time (in years) the debtor lives in the present residence (ordinal; discretized quantitative).
Property	**Categorical**	This feature is the debtor's most valuable property, i.e. the highest possible code is used. Code 2 is used, if codes 3 or 4 are not applicable and there is a car or any other relevant property that does not fall under variable savings.
Age	Numerical	The age of the individual in years. The most frequently occurring age is 27 years, while the least frequent is 70 years.
Other instalment plans	**Categorical**	Instalment plans are typically from providers other than the credit-giving bank, with 814 instances having no plans and 47 involving store plans.
Housing	**Categorical**	The majority of debtors live in rented housing (714 instances), while a smaller portion owns their homes (107 instances).

Number of existing credits at this bank	Ordinal	Most debtors have only one credit at this bank (633 instances), whereas a very few have six or more credits (6 instances).
Job	Ordinal	The job quality distribution shows 630 skilled employees or officials and 22 unemployed or unskilled non-residents.
Number of Dependents	Binary	Number of people being liable to provide maintenance for. A large majority of debtors have between 0 to 2 dependents (845 instances), compared to 155 who have three or more.
Telephone	Binary	This feature answers the question 'Is there a telephone landline registered on the debtor's name?'. There are 596 debtors without a registered telephone landline under their name, while 404 have one registered under the customer's name.
Foreign worker	Binary	Answers the question 'Is the debtor a foreign worker?'. The vast majority of debtors are not foreign workers (963 instances), with only 37 being foreign workers.

Table 21: South German Credit Dataset Descriptive Statistics of Numerical Columns

	Count	Mean	Standard Deviation	Min	Max
Duration	1000	20.90	12.05	4	18
Amount	1000	3271.24	2822.75	250	18424
Age	1000	35.54	11.35	19	75

Appendix C – Banking dataset

Table 22: Detailed Information about the features of Banking Dataset

Feature Name	Feature Type	Feature description
Age	**Integer**	Age of the individual.
Job	**Categorical**	'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown'
Marital	**Categorical**	'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed
Education	**Categorical**	'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown'
Default	**Binary**	Has credit in default?
Balance	**Integer**	Average yearly balance
Housing	**Binary**	Has housing loan?
Loan	**Binary**	Has personal loan?
Contact	**Categorical**	Contact communication type (categorical: 'cellular', 'telephone')
Day	**Date**	Last contact day of the week
Month	**Date**	Last contact month of year (categorical: 'Jan', 'feb', 'mar', ..., 'nov', 'dec')
Duration	**Integer**	Last contact duration, in seconds.
Campaign	**Integer**	Number of contacts performed during this campaign and for this client (numeric, includes last contact)
Pdays	**Integer**	Number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)
Previous	**Integer**	Number of contacts performed before this campaign and for this client
Poutcome	**Categorical**	Outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Table 23: Banking Dataset Descriptive Statistics of Numerical Columns

	Count	Mean	Standard Deviation	Min	Max
Age	45211.00	40.94	10.62	18	95
Balance	45211.00	1362.27	3044.77	-8019.00	102127.00
Day	45211.00	15.81	8.32	1	31
Duration	45211.00	258.16	257.53	0	4918
Campaign	45211.00	2.76	3.10	1	63
Pdays	45211.00	40.20	100.13	-1	871
Previous	45211.00	0.58	2.30	0	275

Appendix D -Visual Results of the experiments

Logistic Regression

Adult Income Dataset

Figure 10

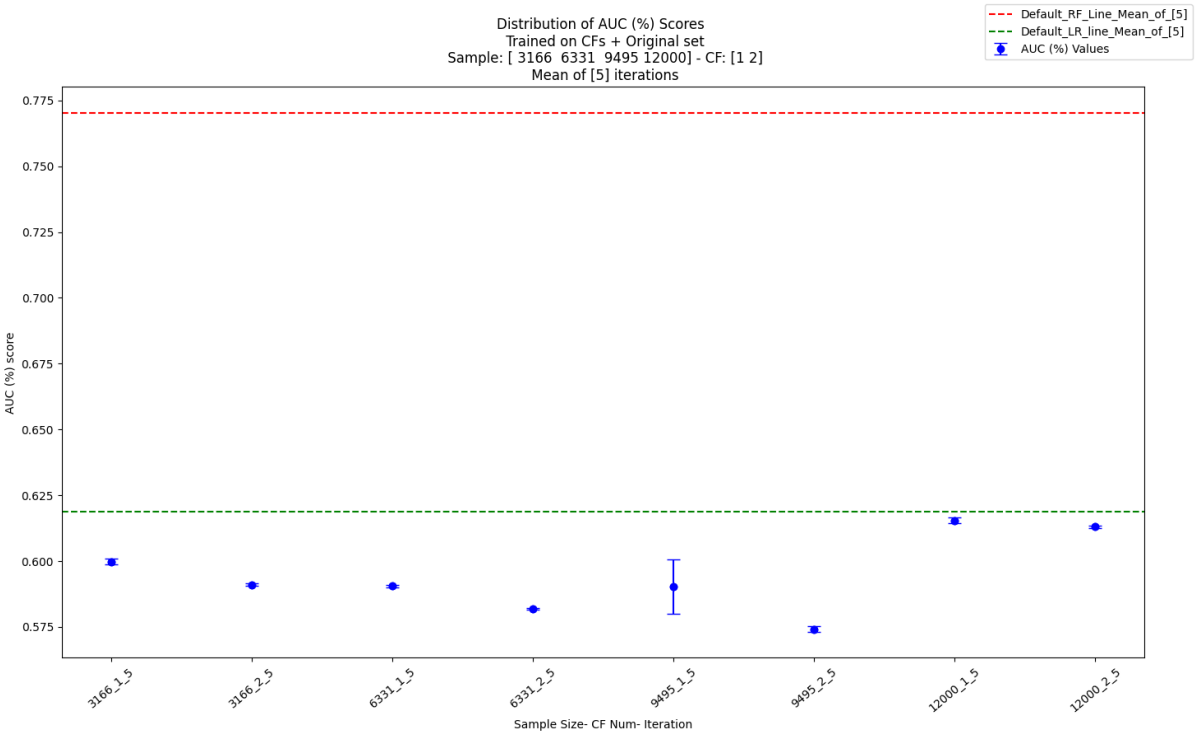
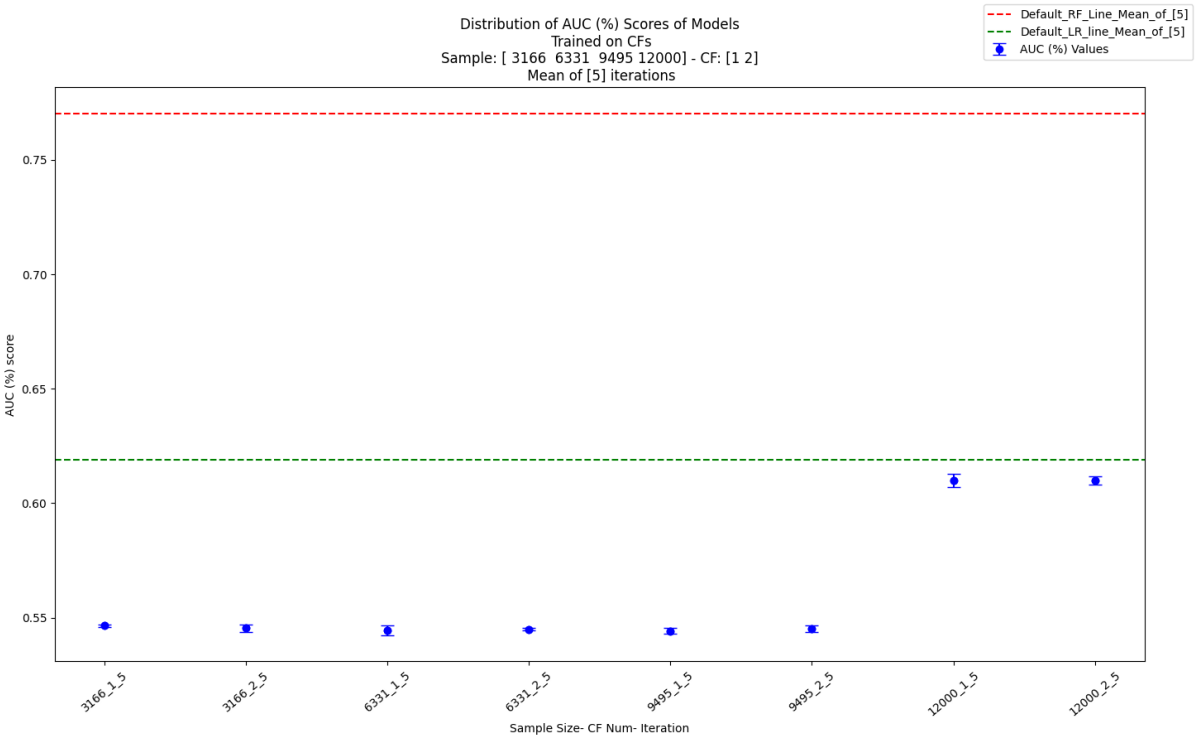


Figure 11



South German Credit Dataset

Figure 12

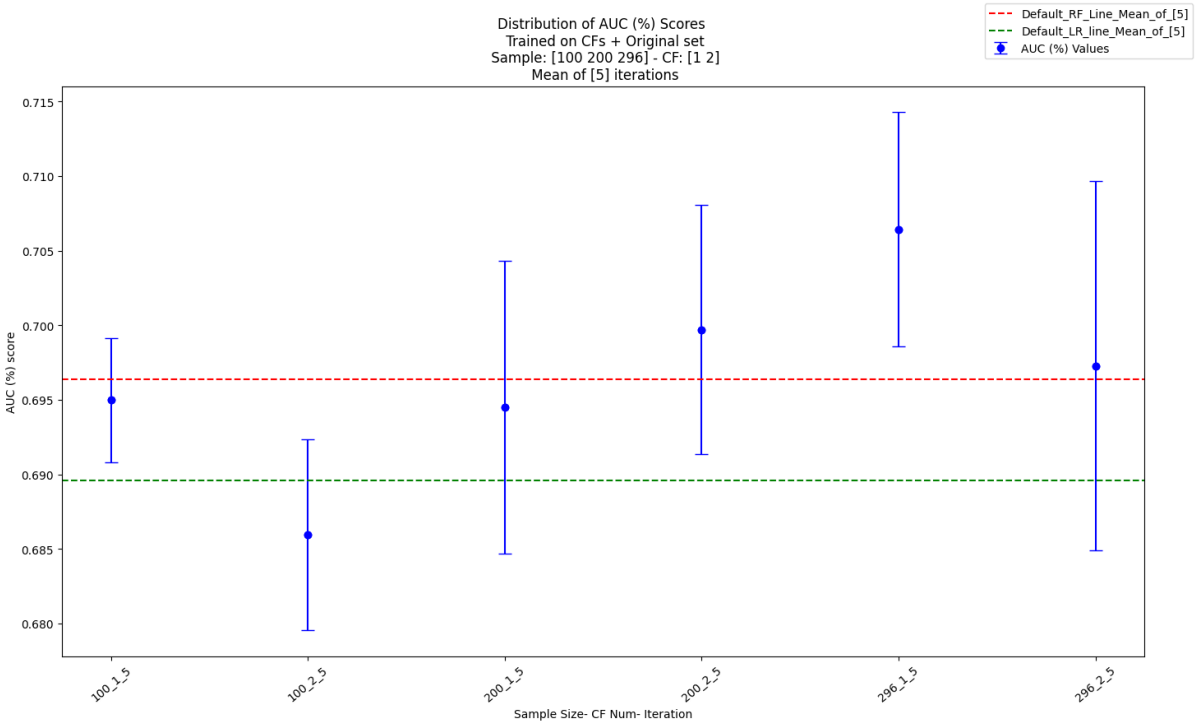
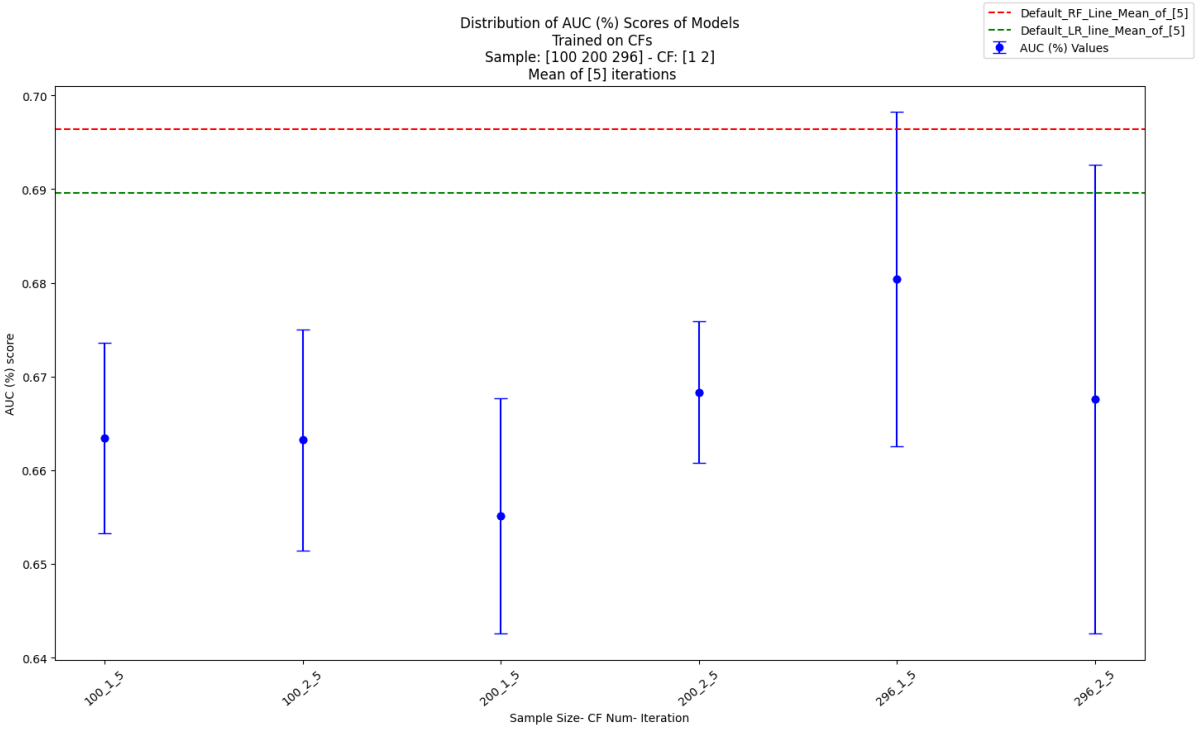


Figure 13



Banking Dataset

Figure 14

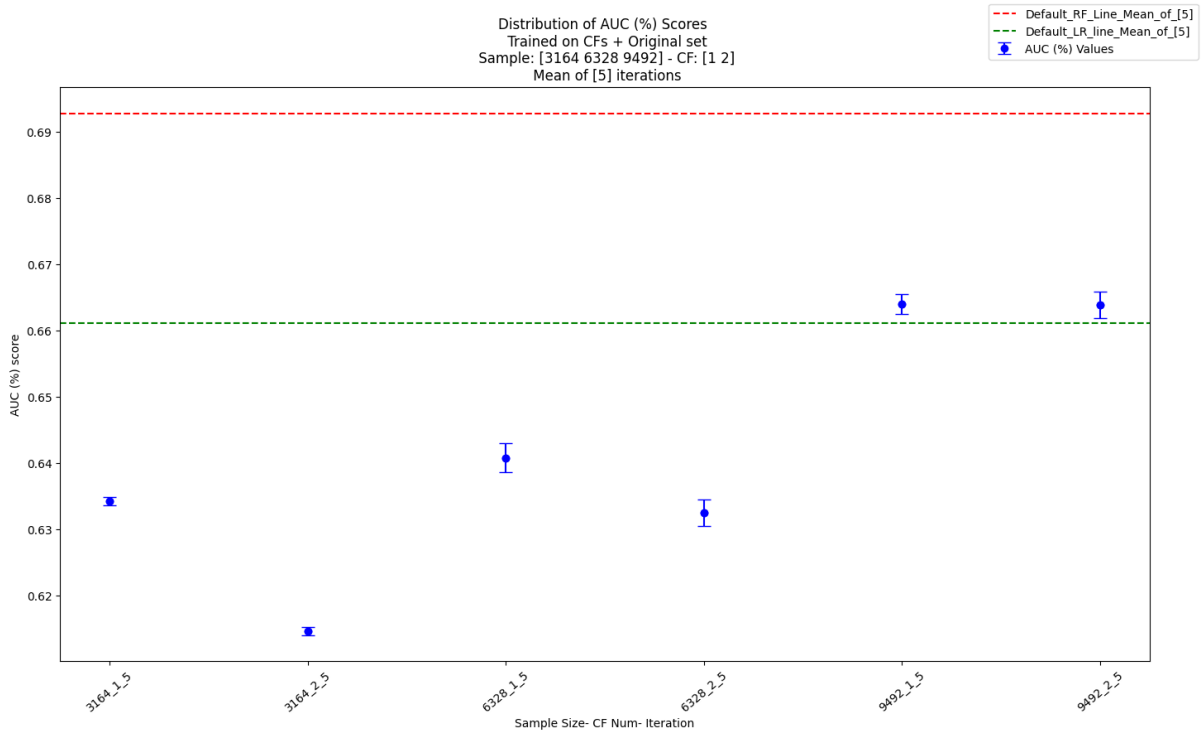
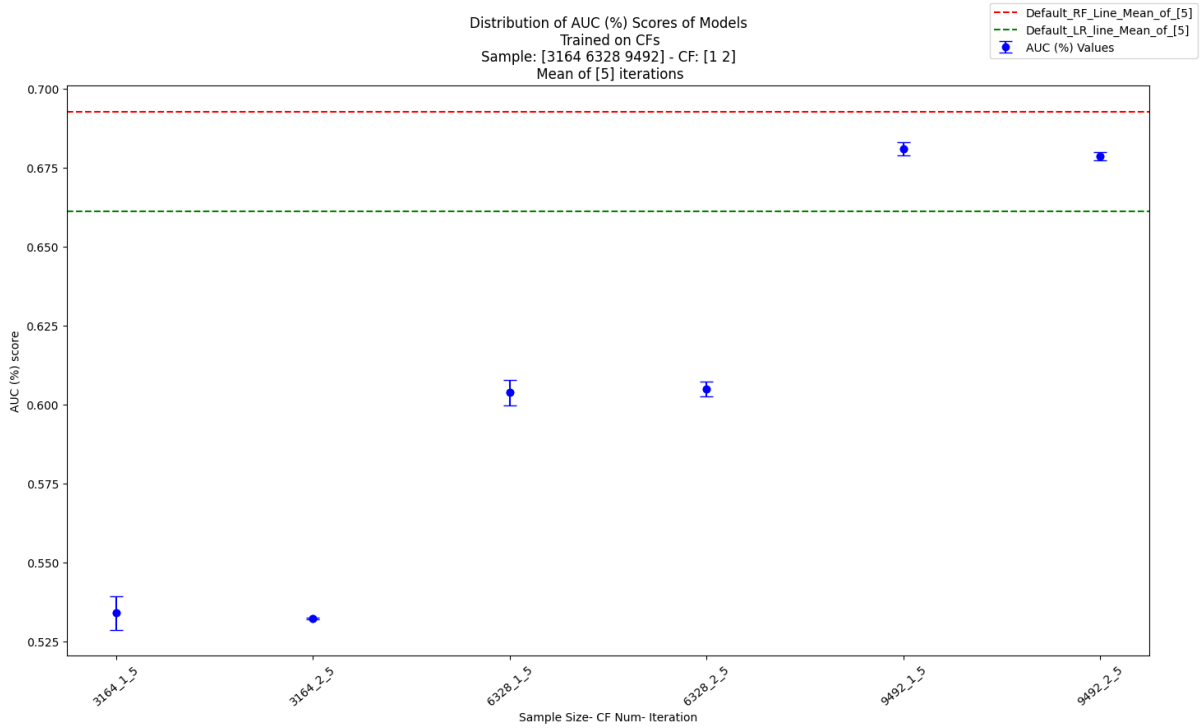


Figure 15



Decision Tree

Adult Income Dataset

Figure 16

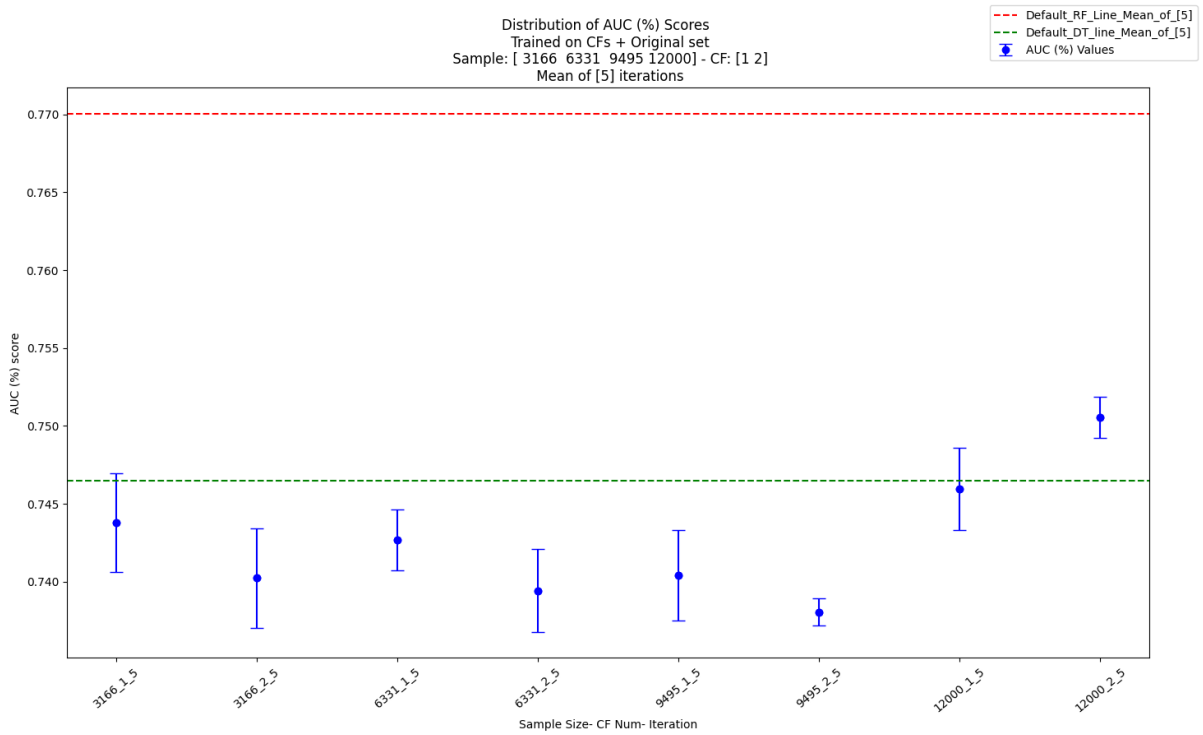
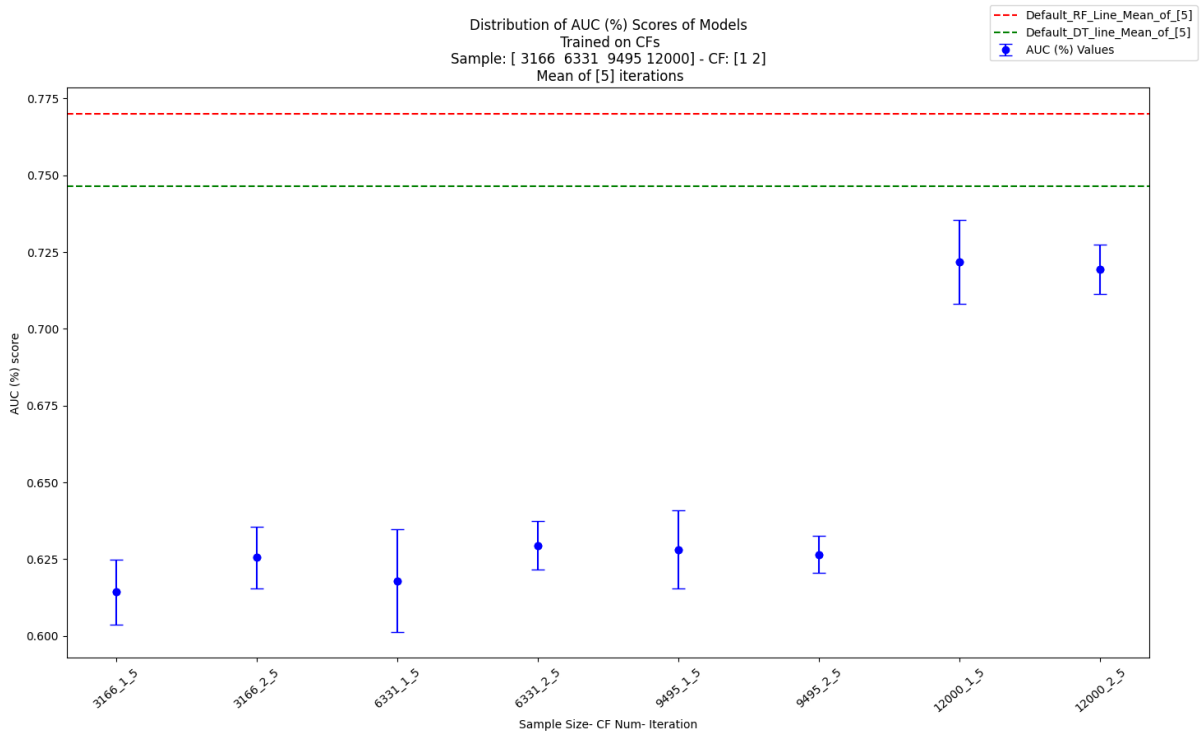


Figure 17



South German Credit

Figure 18

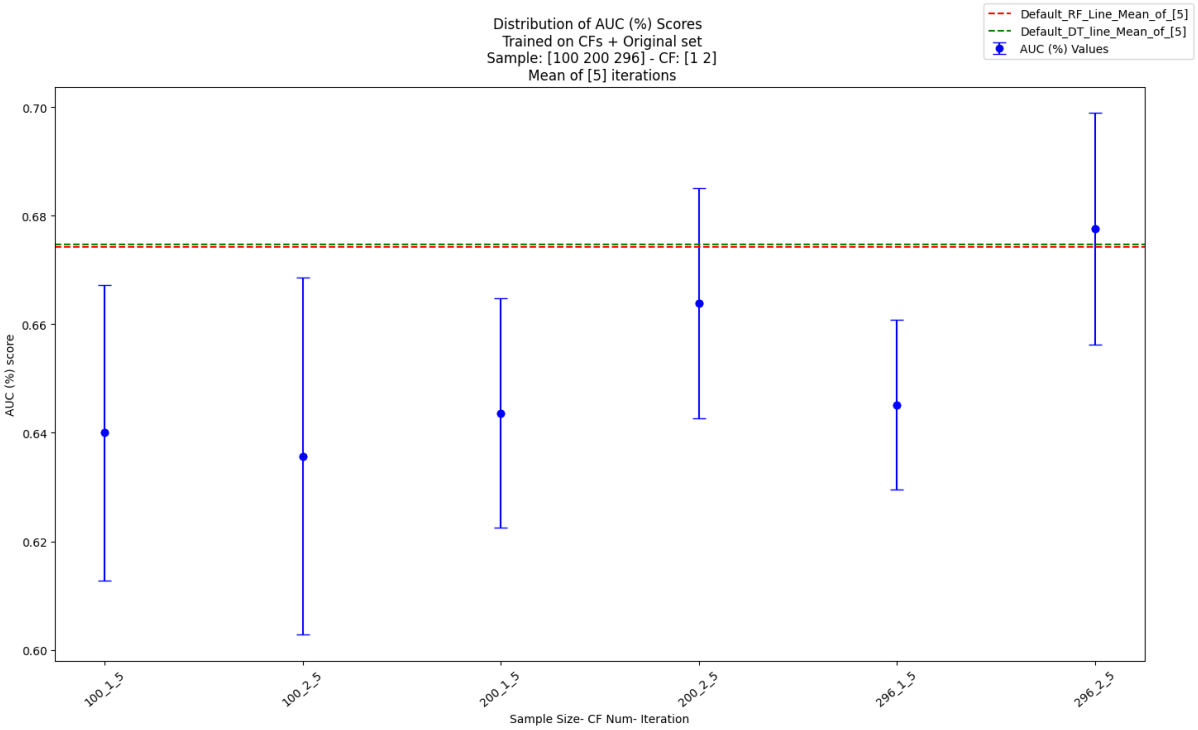
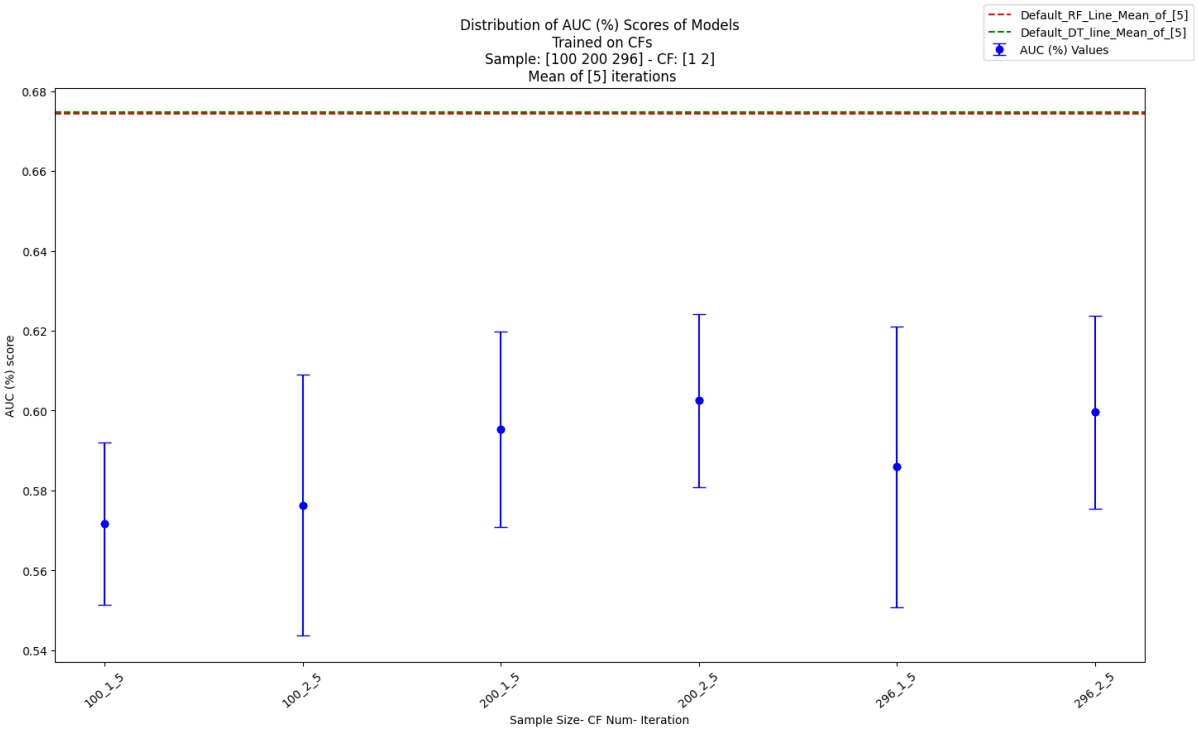


Figure 19



Banking Dataset

Figure 20

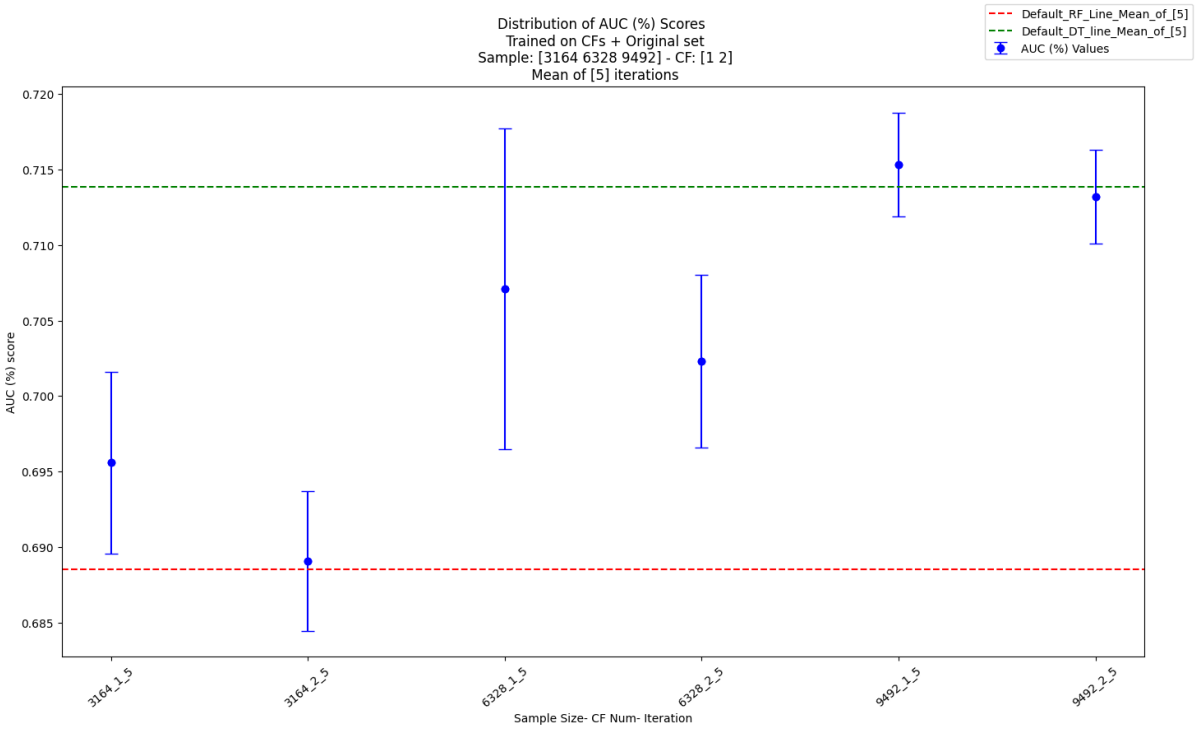
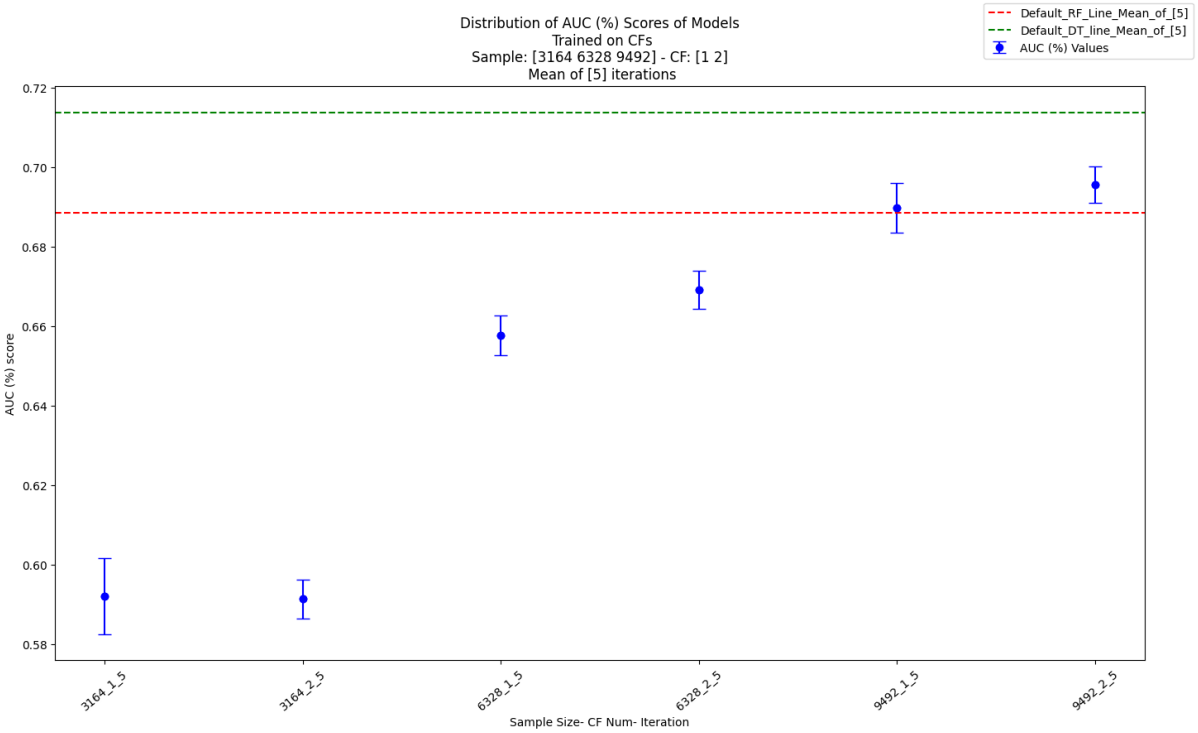


Figure 21



Simpler Random Forests

Simple Random Forest 1 is with 50 n estimators and 5 maximum tree depth.
Simple Random Forest 2 is with 30 n estimators and 3 maximum tree depth.

Adult Income Dataset

Figure 22

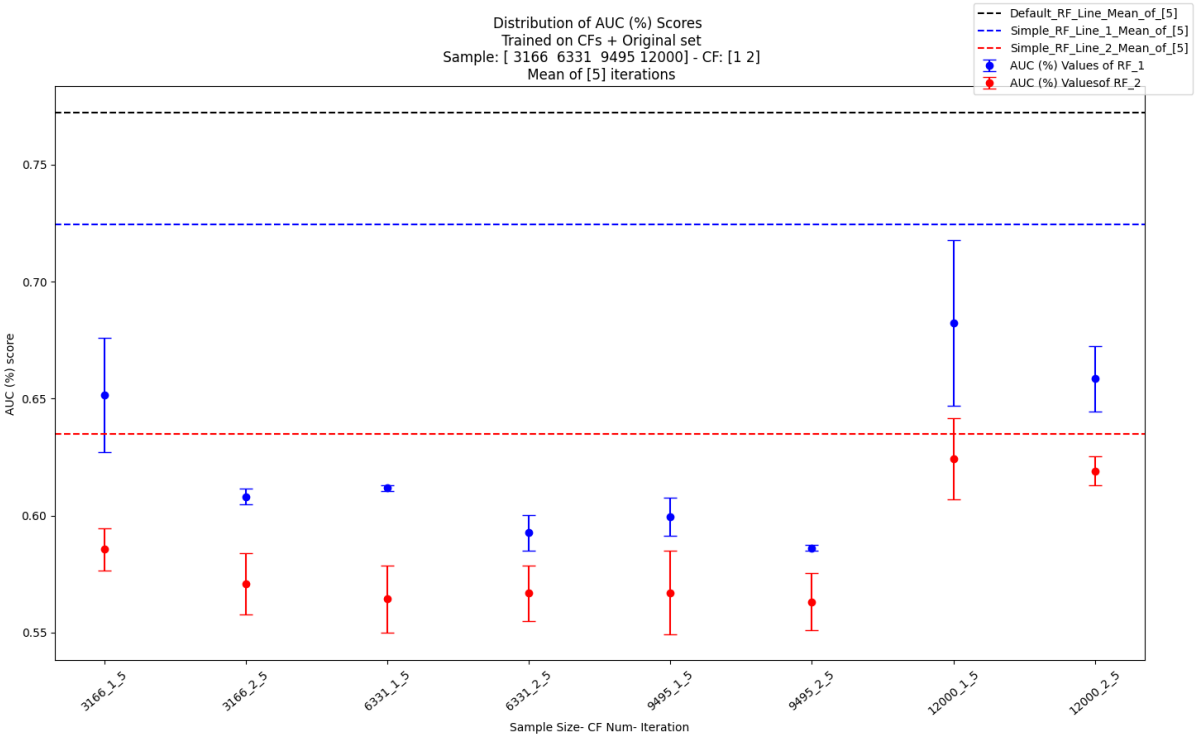
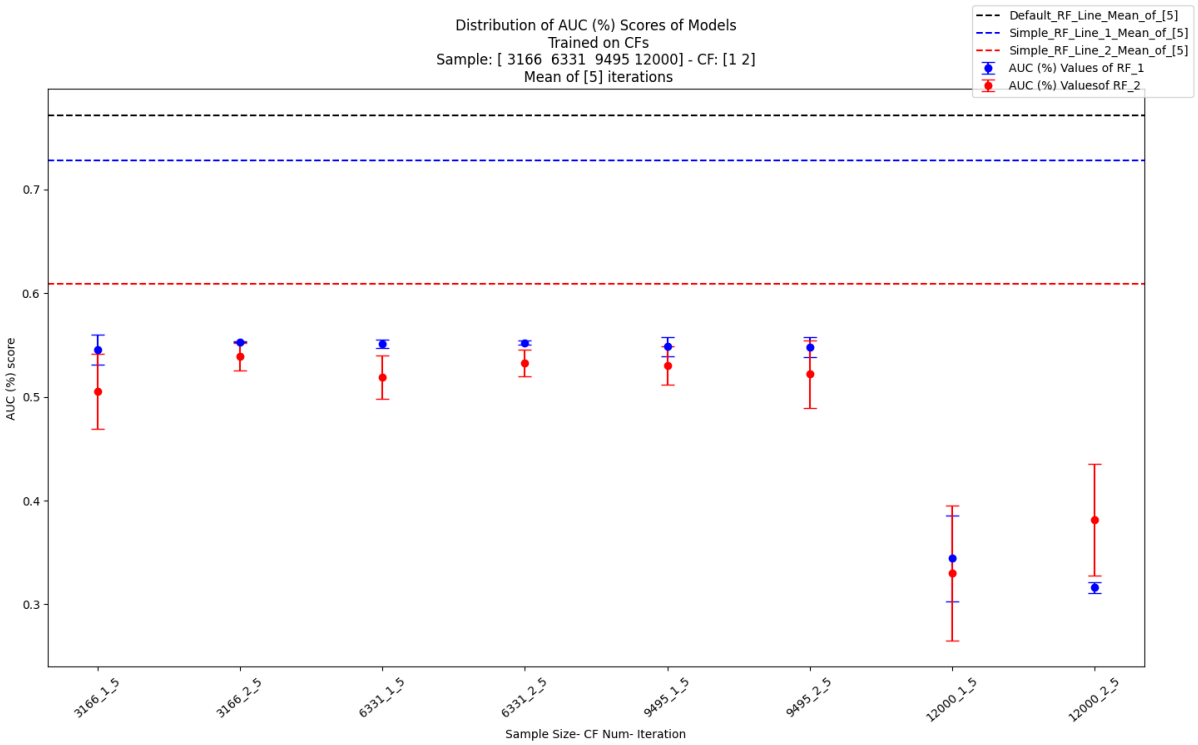


Figure 23



South German Credit Dataset

Figure 24

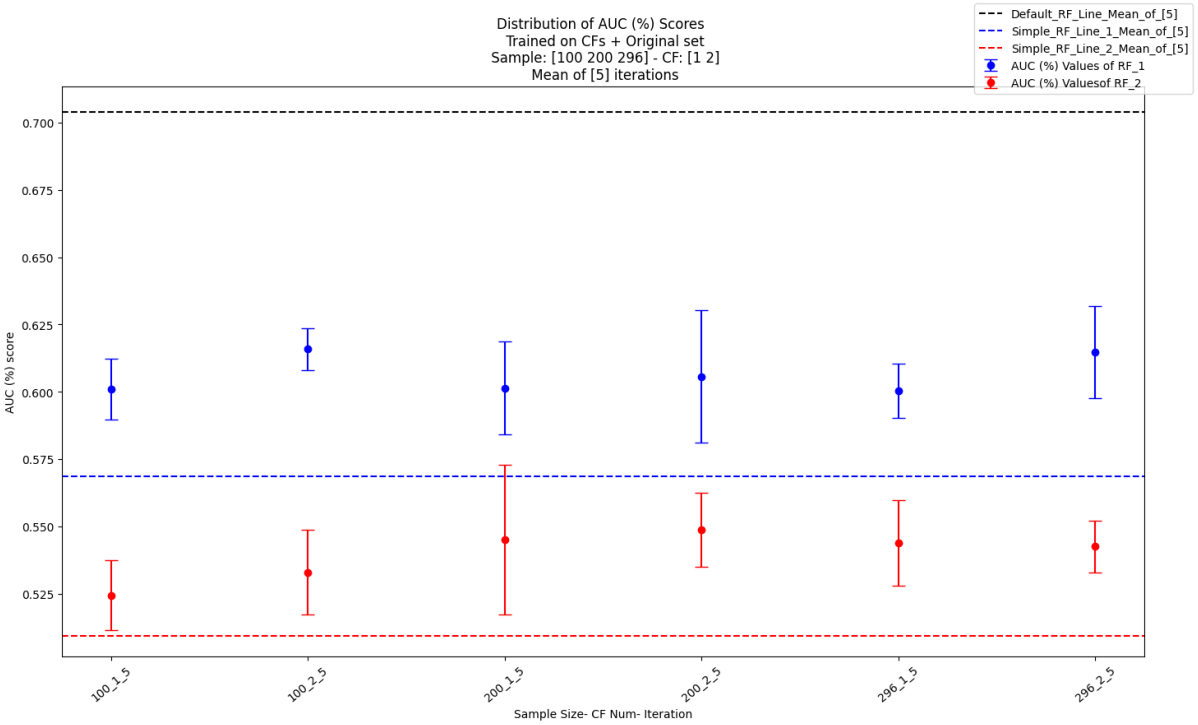
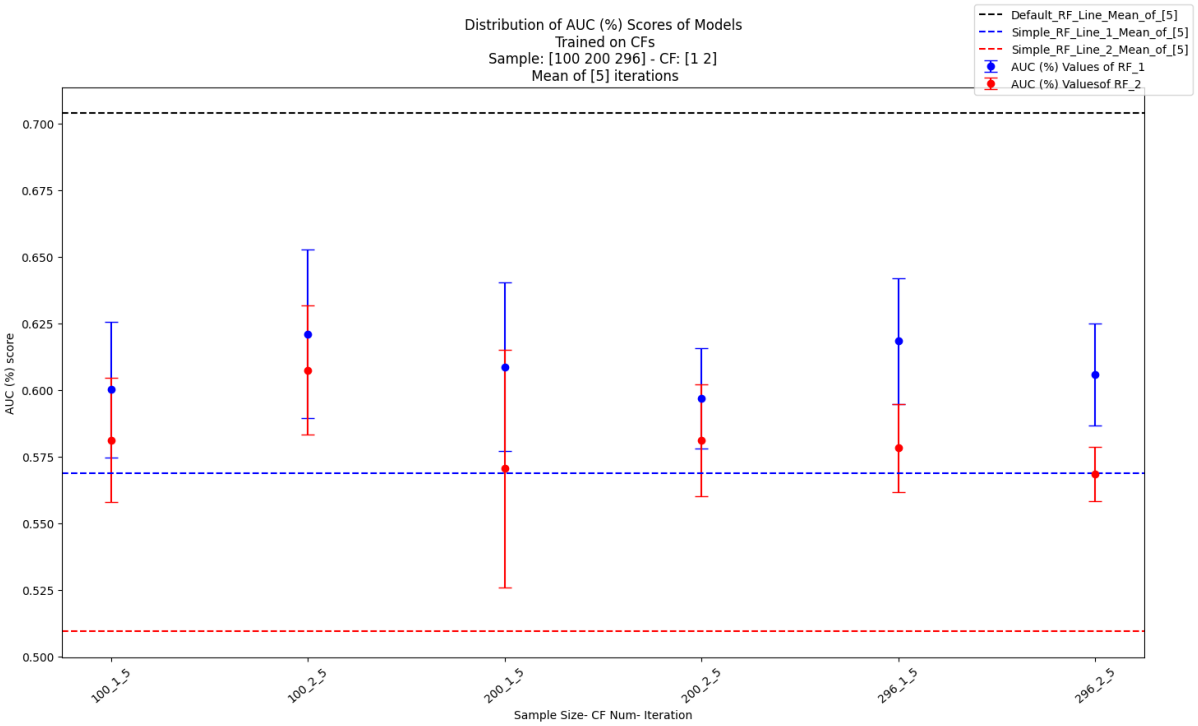


Figure 25



Banking Dataset

Figure 26

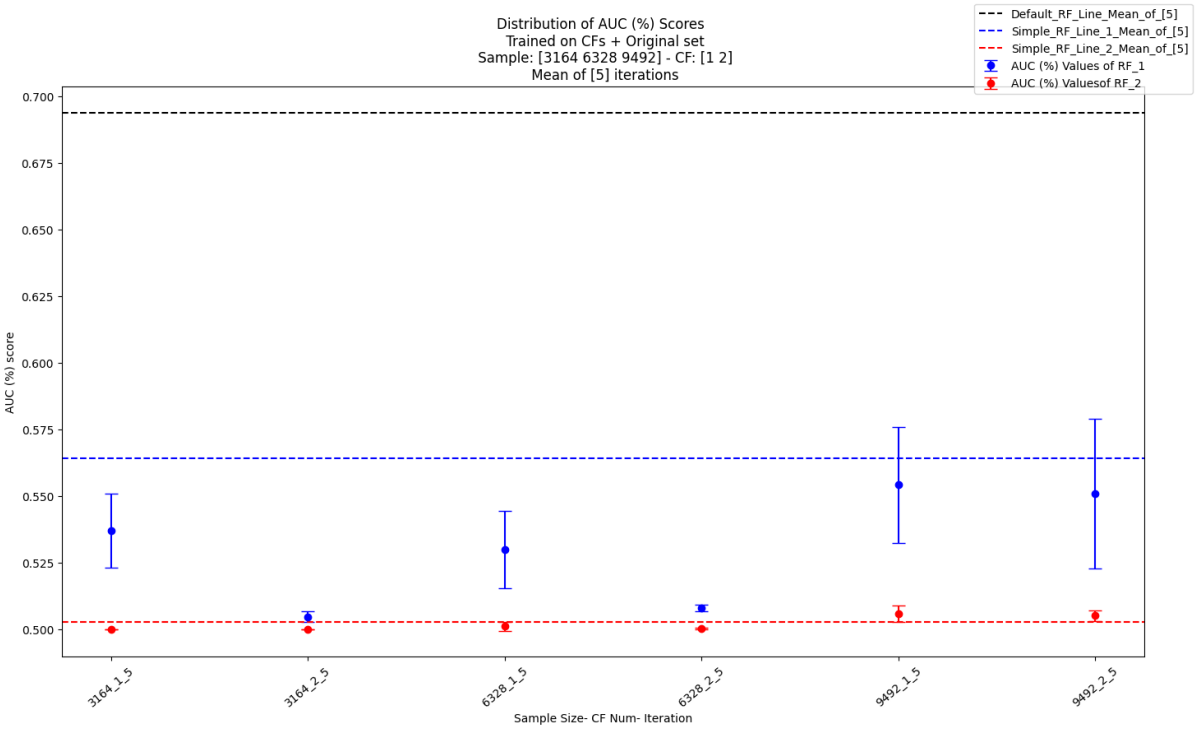
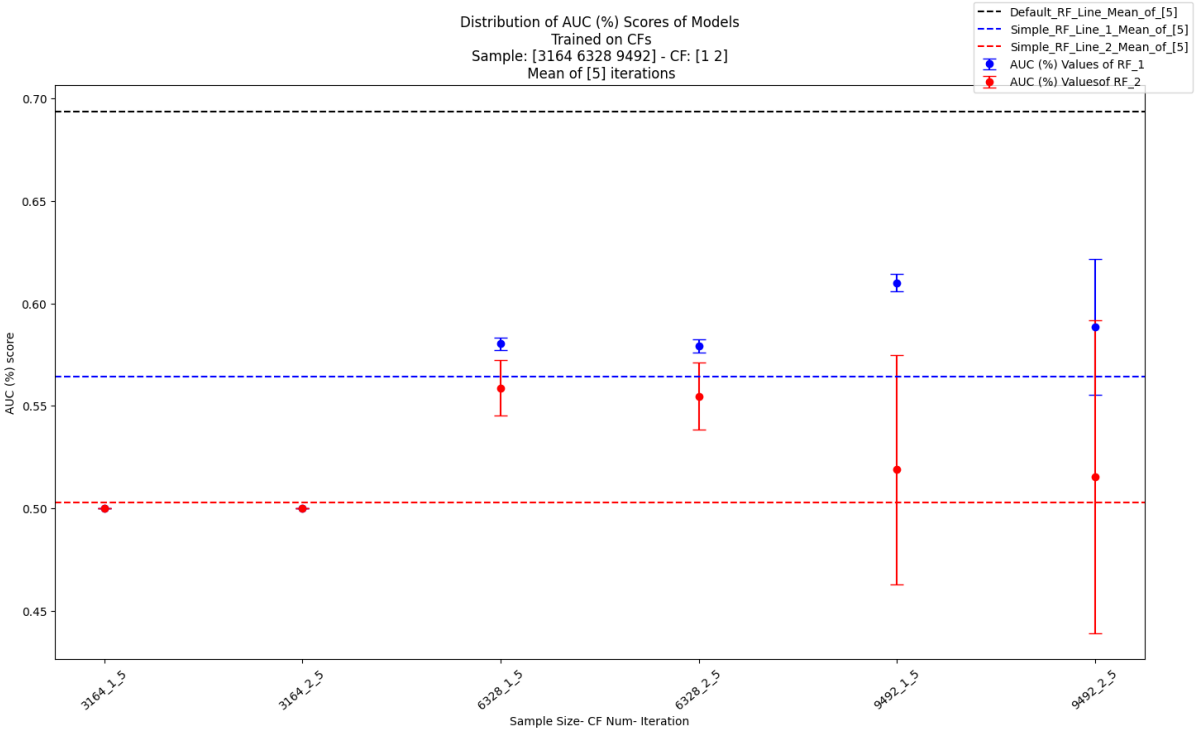


Figure 27



Appendix D: Feature Importance Shifts

Figure 28: Feature Importance Shifts in Simple RF 1 (n:50 max_depth:5)- Sample Size: 12000 CF:1

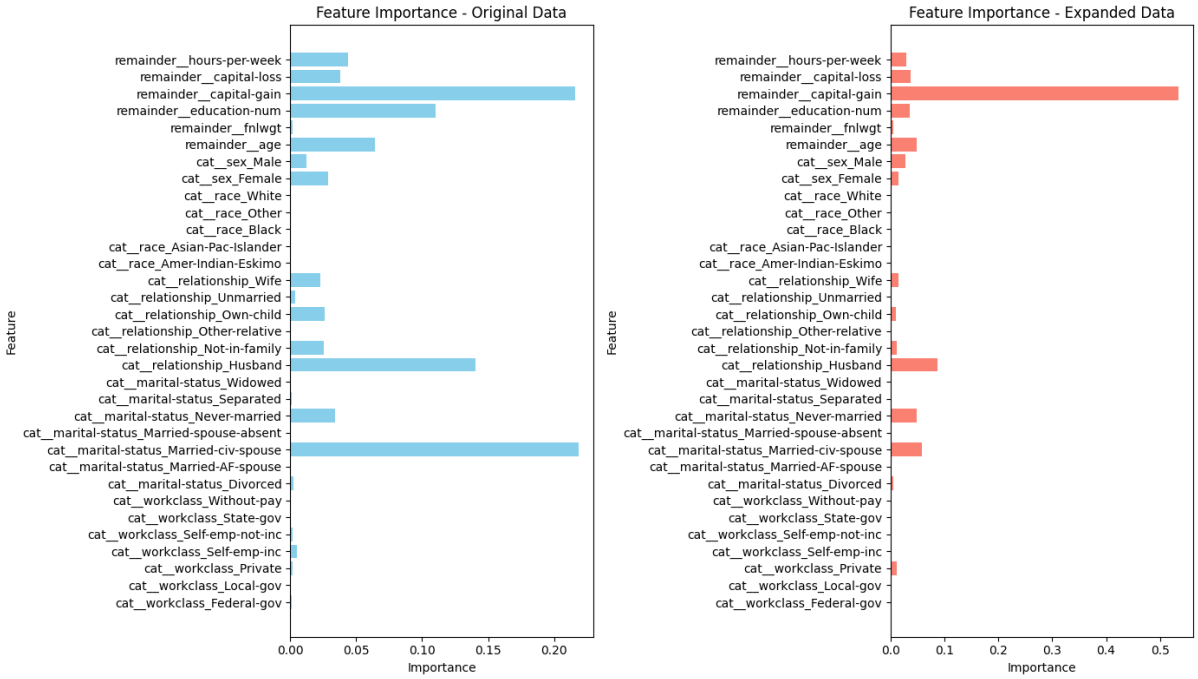
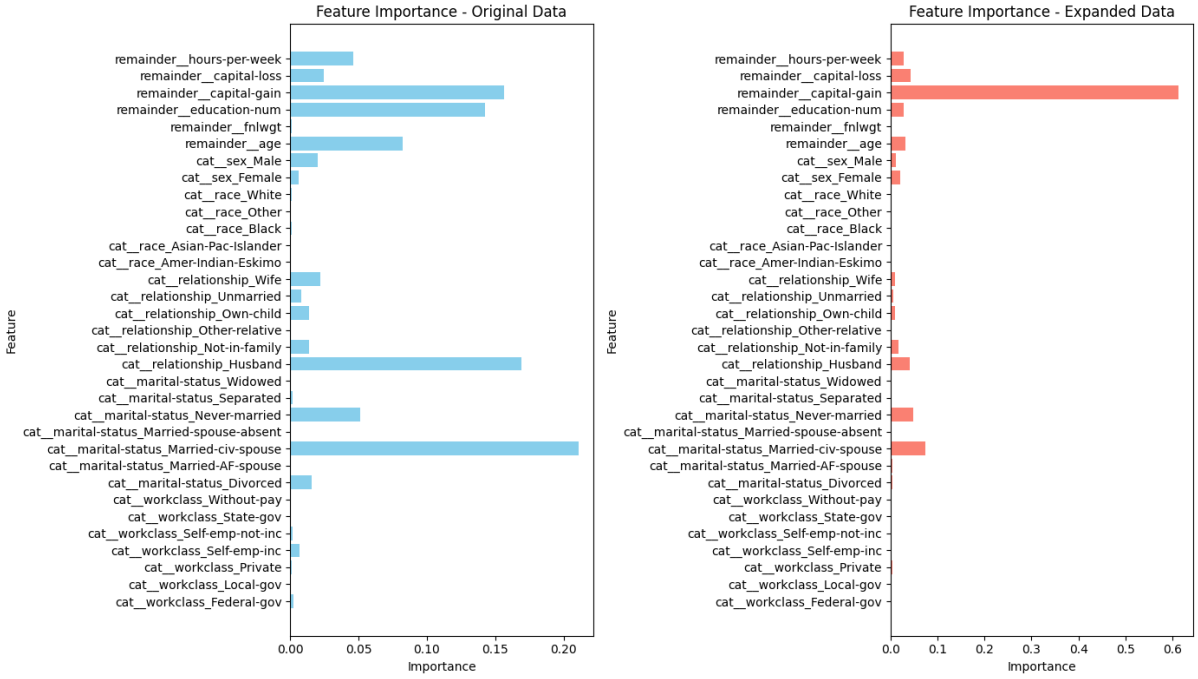


Figure 29: Feature Importance Shifts in Simple RF 1 (n:50 max_depth:5)- Sample Size: 12000 CF:2



Appendix E: Scatter Plot of Second Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using Kolmogorov–Smirnov Test

Figure 30: Adult Income Dataset - Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

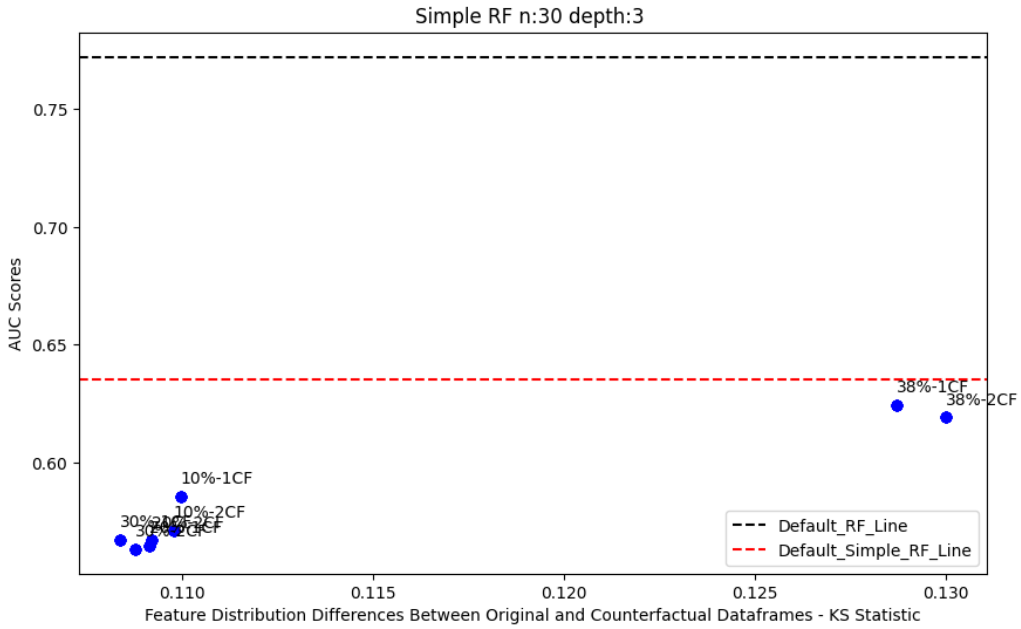


Figure 31: South German Credit Dataset - Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic

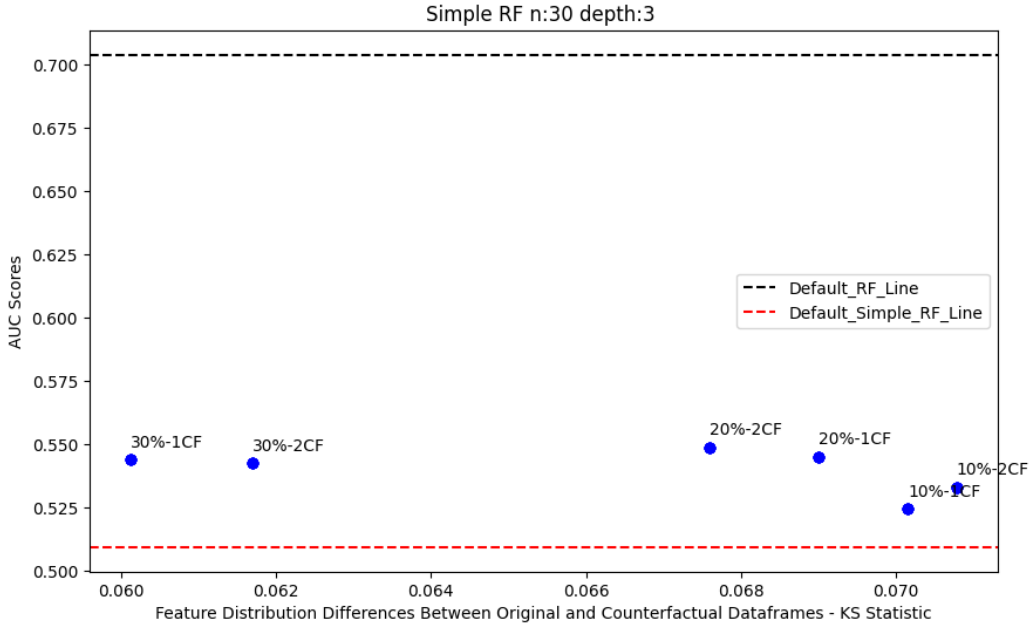


Figure 32: Banking Dataset - Scatter Plot of Simple Random Forest AUC Scores vs. Difference in the Distribution of Features Between the Original and Counterfactual Data Sets Using KS Statistic