Improving Customer Churn Prediction: A Study of Counterfactual Explanations Using Wachter's Method, Growing Spheres Method, and Genetic Algorithms

Amar Jyoti
amarentp23@gmail.com

ABSTRACT

Customer churn prediction is a critical task for businesses aiming to retain their customers and maintain revenue. This paper presents a comparative analysis of three counterfactual explanation methods—Watcher's Method, Sphere Method, and Genetic Algorithm—applied to customer churn prediction models. We evaluate the effectiveness of these methods in generating counterfactual instances, aiming to understand the minimal changes required to alter a customer's churn prediction. Key metrics, including Fischer Score, Imbalance Ratio, Volume, and Number of Features, are analyzed to assess the performance and interpretability of the models. Our findings highlight the strengths and weaknesses of each method, providing insights into their practical applications for improving customer retention strategies.

KEY WORDS

Counterfactual Learning, model interpretation, class imbalance, churn prediction, instance position, Wachter's method, Growing Spheres, Genetic Algorithm.

INTRODUCTION

Customer churn, the process by which customers stop using a company's products or services, poses a significant threat to the sustainability and growth of businesses across various industries. Predicting and mitigating customer churn is vital for maintaining customer loyalty, reducing revenue losses, and enhancing overall business resilience. Accurate churn prediction models enable businesses to identify at-risk customers early and implement targeted retention strategies. However, for these models to be truly effective, they must be interpretable, allowing businesses to understand and act on the factors driving churn.

Counterfactual explanations offer a powerful approach to interpreting predictive models. They provide insights into what minimal changes in customer attributes would alter the prediction outcome. For instance, in churn prediction, counterfactual explanations can reveal which adjustments in customer behavior or characteristics could prevent churn. This capability is crucial for businesses aiming to take actionable steps based on predictive insights.

This research paper presents a comparative analysis of three counterfactual explanation methods applied to customer churn prediction models: Watcher's Method, Sphere Method, and Genetic Algorithm. Watcher's Method seeks the smallest perturbations necessary to change the prediction, Sphere Method iteratively expands a hypersphere around the original instance to find valid counterfactuals, and

the Genetic Algorithm uses evolutionary strategies to explore a wide solution space for effective counterfactuals.

We evaluate the effectiveness of these methods in generating counterfactual instances by examining key metrics, including the Fischer Score, Imbalance Ratio, Volume, and Number of Features. The Fischer Score measures the discriminative power of features, the Imbalance Ratio assesses class distribution in the dataset, and Volume and Number of Features provide insights into dataset size and complexity.

Our findings highlight the strengths and weaknesses of each method in generating actionable and interpretable explanations. This comparative study offers valuable insights for researchers and practitioners seeking to enhance customer retention strategies through improved model interpretability.

This work contributes to the growing field of explainable artificial intelligence (XAI) by applying counterfactual explanations to a critical business problem. By bridging the gap between predictive accuracy and interpretability, we aim to empower businesses with the tools needed to make informed decisions that foster customer loyalty and retention.

RELATED WORKS

Recently, there has been a growing interest among researchers in understanding the decision-making processes of models in customer churn prediction. For example, Hasumoto and Goto used the PDP (Partial Dependence Plot) technique to conduct a visual analysis of churn prediction models. They explained the extracted features and the implications of potential customer purchase behavior within the group of potential churners, providing support for effective retention strategies. However, there has been limited focus on interpreting predictions from a local perspective.

In recent years, more scholars have turned their attention to counterfactual learning, recognizing counterfactual explanations as one of the most valuable methods among local post-hoc interpretation approaches. Initially proposed by Wachter et al., counterfactual explanations generate post-hoc instances to explain predictions of black-box models. Current methods for generating counterfactual explanations can be categorized as follows:

Optimization-based Methods:

- These methods define a loss function that captures the desired attributes of an explanation and use existing optimization algorithms to minimize the loss. Most counterfactual explainers solve optimization problems to generate counterfactuals. Wachter et al. pioneered the use of gradient descent to find the best counterfactual instances. Dastile and Dandl further enhanced this approach by employing genetic algorithms or multi-objective genetic algorithms to generate counterfactual explanations more effectively.

Heuristic Search-based Methods:

- These methods aim to discover counterfactuals through local and heuristic selection, minimizing a cost function at each iteration. They generally exhibit higher efficiency compared to optimization algorithms. For instance, Martens and Provost proposed a model-agnostic heuristic explanation

method called SEDC, which guides local improvements through best-first search and pruning. Laugel et al. introduced the GSG method, which relies on partitioning a range of synthetic instances around the instance to be explained to find the closest counterfactual explanation.

Instance-based Methods:

- These methods retrieve counterfactuals by selecting the most similar examples from the dataset. For example, the Nearest Neighbor Counterfactual Explanation (NNCE) method proposed by Shakhnarovich et al. selects instances most similar to the instance to be explained but with different labels as candidate counterfactuals, ranking them based on their similarity to the instance. The method then selects the top-k most similar counterfactual explanations. The Neighbor-based Instance Counterfactual Explanation (NICE) method, proposed by Brughmans and Martens, accelerates the search process by iteratively introducing feature values from neighbors that differ the most, using information from the nearest distinct neighbors. Poyiadzi et al. proposed the FACE method, which identifies feasible paths through density analysis to generate more feasible counterfactual explanations.

Hybrid Methods:

- Mothila et al. combined the LIME and SHAP feature attribution methods with counterfactual explanations. They proposed an explanation method that reveals actual causal relationships uncovered by counterfactual explanations and used both attribution methods to explain key outcomes of the model.

Despite these advances, there is a lack of counterfactual explanation methods that consider class imbalance. Data in customer churn prediction are typically class-imbalanced, which directly affects the effectiveness of counterfactual explanations for minority class instances. Regarding interpretability research under class imbalance conditions, only Dablain et al. have proposed a method for understanding deep learning models from the perspectives of class prototypes, sub-concepts, and outlier instances. This method utilizes imbalance learning algorithms to detect important features and class distribution prototypes crucial to model performance. However, this research primarily discusses the explanation of deep neural networks from a prior perspective and has not yet explored the consideration of class imbalance in post-hoc interpretability. Further research is needed to investigate how to improve the interpretability performance of counterfactual methods under class imbalance conditions.

BALANCED AND IMBALANCED CLASS COUNTERFACTUAL LEARNING

IMBALANCED CLASSES

In an imbalanced dataset, one or more classes have significantly fewer instances compared to others. This is common in many real-world scenarios such as fraud detection, rare disease diagnosis, and customer churn prediction, where the event of interest (e.g., fraud, disease, churn) is rare. Key characteristics of imbalanced datasets include:

1. Unequal Representation: One or more classes have significantly fewer instances.
2. Bias in Learning: The model tends to be biased towards the majority class because it encounters it more frequently during training.
3. Misleading Performance Metrics: Accuracy can be misleading as the model might predict the majority class most of the time and still achieve high accuracy. Metrics such as precision, recall, F1-score, and the area under the ROC curve (AUC-ROC) are more informative in such cases.

Imbalanced datasets pose a challenge for machine learning models because the minority class can be underrepresented, leading to poor generalization and performance for those classes.

BALANCED CLASSES

In a balanced dataset, the number of instances in each class is roughly equal. For instance, in a binary classification problem, a balanced dataset would have a similar number of examples for both classes. This balance ensures that the model does not favor one class over another during training. Key characteristics of balanced datasets include:

1. Equal Representation: Each class has an approximately equal number of instances.
2. Fair Learning: The model can learn to recognize patterns for each class without biasing towards the more frequent class.
3. Standard Performance Metrics: Accuracy, precision, recall, and F1-score tend to provide a reliable indication of model performance.

Balanced datasets are often ideal for model training because they provide a comprehensive view of each class, allowing the model to learn the distinguishing features of each class effectively.

COUNTERFACTUAL LEARNING

Counterfactual learning involves generating hypothetical scenarios to understand the impact of changes in input features on the model's predictions. It answers "what-if" questions, such as "What if feature X had been different, would the prediction change?" This approach is particularly useful for interpreting complex, black-box models like neural networks and ensemble methods.

Key aspects of counterfactual learning include:

Hypothetical Scenarios: Creating instances that are close to the original data point but with one or more features altered.

Model Behavior Analysis: Observing how changes in features affect the model's predictions to understand the decision-making process.

Actionable Insights: Providing insights into what changes could lead to a different prediction, useful for decision-makers in various domains.

COUNTERFACTUAL EXPLANATION

Counterfactual explanations provide a way to interpret model predictions by identifying the minimal changes needed to alter the prediction. This type of explanation is post-hoc, meaning it is generated after the model has made its prediction. Key elements include:

1. Minimal Changes: Determining the smallest modifications to the input features that result in a different prediction.
2. Local Interpretability: Focusing on specific instances rather than the overall model behavior, providing instance-specific insights.
3. Actionability: Offering actionable recommendations based on the counterfactual scenarios, which can be particularly useful for business decisions.

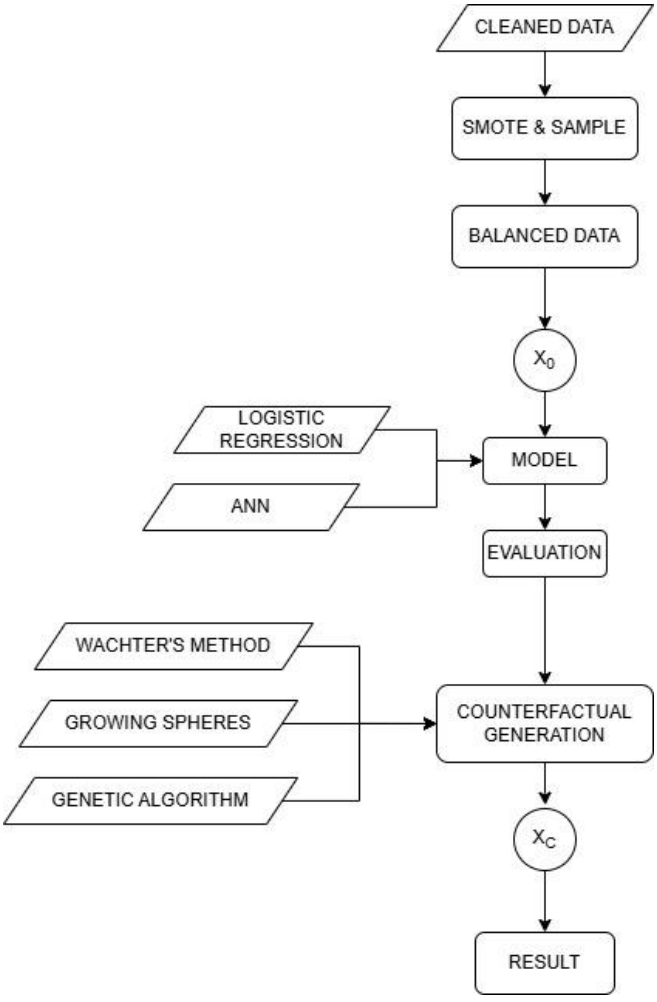

Figure1: Research Workflow

COUNTERFACTUAL LEARNING METHODS

Wachter's Method:

Wachter's method is an optimization-based approach to generating counterfactual explanations. This method seeks to find the smallest possible perturbations to an input instance that change the model's prediction to a desired outcome. By defining a loss function that balances the distance between the original and counterfactual instances with the prediction change, the method iteratively adjusts the features using gradient descent until it meets the criteria. Wachter's method is particularly useful for its ability to provide clear and interpretable explanations by highlighting the minimal changes needed to alter a prediction, making it valuable for understanding model decisions in a precise manner.

$$(x,x',y',\lambda) = \lambda(f(x') - y')^2 + d(x,x')$$

Growing Spheres Method

The Growing Spheres method is a heuristic search-based approach for generating counterfactual explanations. It works by iteratively expanding a hypersphere around the original instance and searching for counterfactuals within this sphere. The radius of the hypersphere is gradually increased until a valid counterfactual instance is found. This method focuses on finding the closest counterfactual in terms of feature space distance, which helps ensure the generated explanations are both realistic and actionable. The Growing Spheres method is efficient in exploring local neighborhoods of the original instance, making it an effective tool for generating interpretable counterfactuals.

Formula:

1. Initialization:

$$r = r_0$$

where r_0 is the initial radius.

2. Objective:

$$X_{fc} = \arg \min_{x \in R^2} g \|x - x_0\|_2$$

subject to:

$$f(x_{cf}) \neq f(x_{orig})$$

And

$$|x - x_{orig}|_2$$

3. Expansion:

$$r = r + \Delta r$$

Where Δr is the increment step for the radius.

Genetic Algorithm

The Genetic Algorithm is an evolutionary-based approach used to generate counterfactual explanations by mimicking the process of natural selection. It starts with a population of potential counterfactuals and evolves them over several generations using operations such as selection, crossover, and mutation. The algorithm evaluates each candidate solution based on a fitness function that measures how well it meets the desired prediction and its proximity to the original instance. By iteratively selecting and refining the best candidates, the Genetic Algorithm effectively explores a broad solution space, often finding more diverse and high-quality counterfactuals. This method is particularly advantageous for its robustness and ability to handle complex, non-linear relationships in the data.

Formula

1. Fitness Function:

$$fitness(X) = \lambda \cdot \| X - X_{orig} \|_2 + \mathcal{L}(f(x), y_{target})$$

where:

x is a candidate counterfactual instance

λ is a regularization parameter.

$L(f(x),y_{target})$ ensures the prediction changes to the desired outcome.

2. Selection:

Select individuals based on their fitness scores.

3. Crossover:

Combine parts of two individuals to create offspring:

$$x_{child} = \alpha \cdot x_1 + (1 - \alpha) \cdot x_2$$

where α is a crossover parameter.

4. Mutation:

Introduce small random changes to individuals:

$$x_{mut} = x + \delta$$

where δ is a small random perturbation.

5. Iteration:

Repeat the selection, crossover, and mutation processes over multiple generations to evolve the population towards better solutions.

EXPERIMENT

The experiments were conducted using Python 3.11, and the black-box model was built using TensorFlow 2.15.0. The counterfactual methods and evaluation metrics were obtained from the scikit-learn. The entire experiment was run on a computer with a 2.50 GHz Intel® Core™ i5-10300H with Nvidia™ Geforce GTX 1650 Ti and 16 GB RAM.

The dataset used in this study is a customer churn dataset, which contains information about customers of a telecommunications company. The dataset is typically employed

to predict whether a customer will leave the company (churn) based on various features that describe the customer's behavior, usage patterns, and demographic information.

Key Characteristics of the Dataset

Features:

- Demographic Information: This includes attributes such as age, gender, income, and location. These features help in understanding the customer's profile.
- Usage Patterns: These features capture the customer's interaction with the services, such as the number of calls made, duration of calls, internet usage, and the types of services subscribed to (e.g., voice, data, text).
- Customer Behavior: This includes metrics like the number of customer service calls, complaints, payment history, and contract type (e.g., month-to-month, one-year, two-year).

Target Variable:

- Churn: The target variable is a binary attribute indicating whether the customer has churned (1) or not (0). This is the key outcome we aim to predict using various machine learning models.

Class Imbalance:

- The dataset typically exhibits a class imbalance, with a smaller proportion of customers having churned compared to those who have not. This imbalance poses challenges for model training and requires special handling to ensure the minority class (churn) is adequately represented.

Volume and Complexity:

- Volume: The dataset contains a substantial number of instances, which provides a rich source of information for training predictive models. For example, the dataset used in this study contains thousands of customer records.
- Number of Features: The dataset includes multiple features, each contributing to the complexity and richness of the data. In our study, the dataset comprises around 10 key features that are critical for churn prediction.

Table 1: Description of Imbalanced dataset.

Metric	Value
Imbalance Ratio	0.2
Fischer Score	0.088495
Volume	900
Number of Features	5

Table 2: Description of balanced dataset.

Metric	Value
Imbalance Ratio	1.0
Fischer Score	0.173983
Volume	1500
Number of Features	5

EVALUATION

A good counterfactual explanation should ensure that the difference between the original instance X_0 and the counterfactual instance X_C is minimal. This can be achieved by meeting the following characteristics:

Fidelity

The primary goal of counterfactual learning is to generate instances that closely match the desired prediction. However, it is not always feasible to find a counterfactual instance that meets the predefined prediction within the data. For example, in a binary classification scenario with a significant class imbalance, the model might consistently classify an instance as belonging to the majority class, making it nearly impossible to change the prediction to the minority class by modifying the feature values.

Proximity

Counterfactual instances should be as similar as possible to the original instances in terms of feature values. The distance between two instances can be measured using metrics such as Manhattan distance or Euclidean distance. Ensuring proximity helps in maintaining the relevance and plausibility of the counterfactual instance.

Sparsity

In addition to being close to the original instance, counterfactual instances should aim to minimize the number of feature changes. This means that only a few features should be altered to achieve the desired prediction change. The quality of counterfactual explanations can be evaluated by calculating the proportion of features that have changed between the counterfactual and the original instance.

Credibility

The feature values of counterfactual instances should be reasonable and realistic. For example, generating a counterfactual explanation with an unrealistic feature value, such as a negative apartment area or an excessively high number of rooms, is meaningless. Additionally, the counterfactual should be plausible under the joint distribution of the data. For instance, an apartment with 10 rooms and only 20 square meters should not be considered a plausible counterfactual. Ideally, an increase in the number of square meters should also suggest an increase in the number of rooms. This measure can be evaluated by comparing the generated counterfactual instances with their neighbours and computing the support of these neighbours.

Experimental Result

I have trained two models. One is Logistic Regression and another is Artificial Neural Network. The counterfactual methods, viz, Wachter's method, growing spheres and genetic algorithm is then applied on ANN model.

Table 3: Summary of ANN model:

Layer	O/P Shape	Params
Dense	(None, 64)	384
Dense	(None, 32)	2080
Dense	(None, 16)	528
Dense	(None, 1)	17
Total Parameters		3009
Trainable Parameters		3009
Non Trainable Parameters		0

Below are performance evaluation of both models and prediction probability of all three counterfactual learning algorithms.

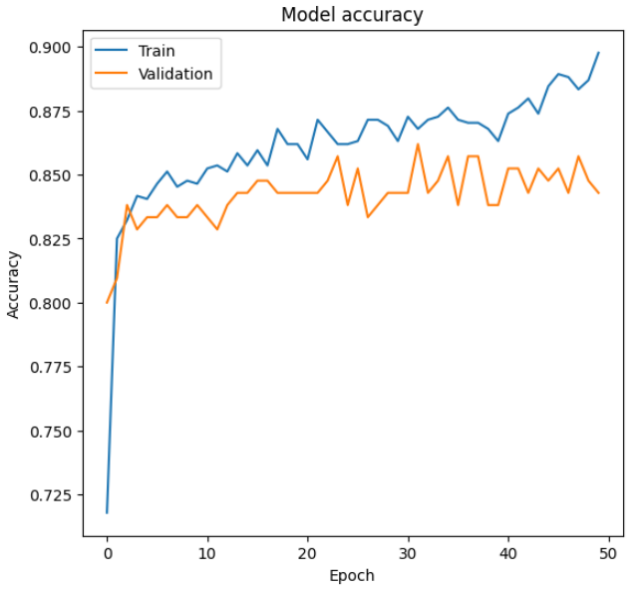

Figure 2: Model Accuracy

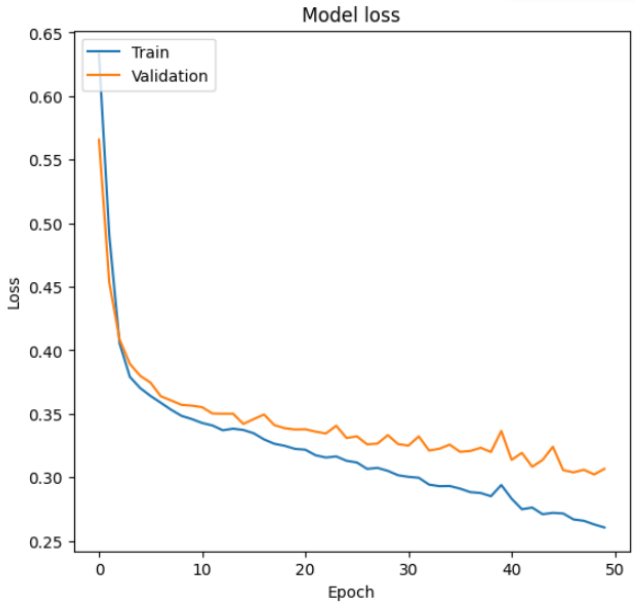

Figure 3: Model Loss

accuracy	0.84
precision	0.88
recall	0.81
f1	0.84
roc_auc	0.92

Table 4: Logistic Regression Performance

Table 5: Classification Report

	precision	recall	f1-score	support
0	0.81	0.88	0.84	214
1	0.88	0.81	0.84	236
accuracy			0.84	450
Macro avg	0.84	0.84	0.84	450
Weighted avg	0.84	0.84	0.84	450

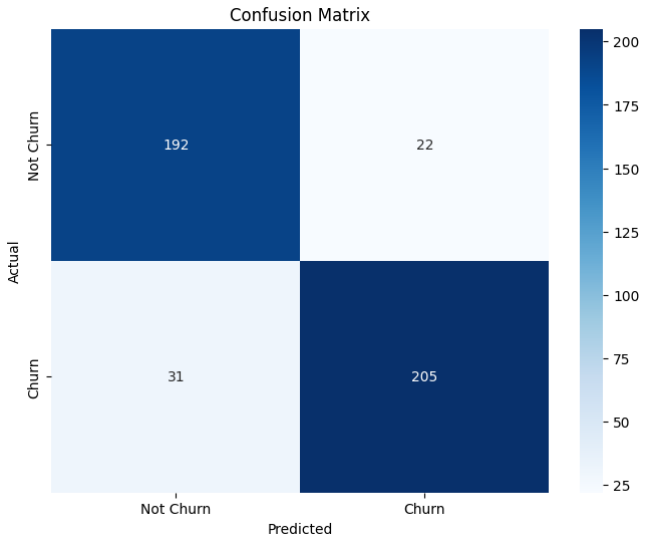

Figure 2: Confusion Matrix:

Table 6: Artificial Neural Network Performance

accuracy	0.88
precision	0.91
recall	0.87
f1	0.89
roc_auc	0.95

Table 7: Classification Report

	precision	recall	f1-score	support
0	0.96	0.90	0.88	214
1	0.90	0.87	0.89	236
accuracy			0.88	450
Macro avg	0.88	0.88	0.88	450
Weighted avg	0.88	0.88	0.88	450

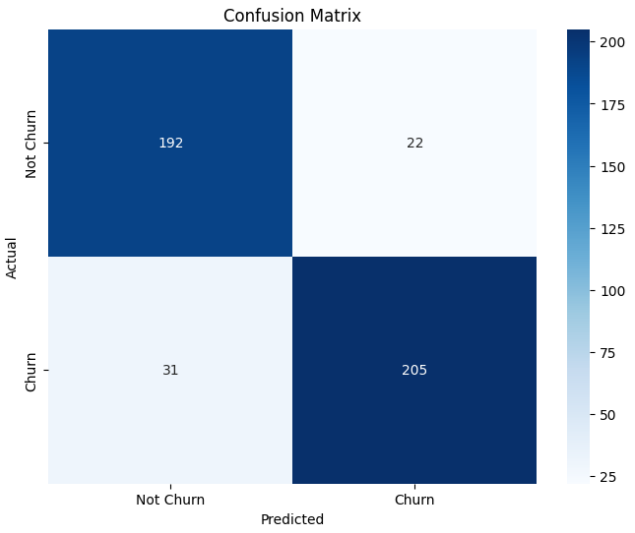

Figure 3: Confusion Matrix:

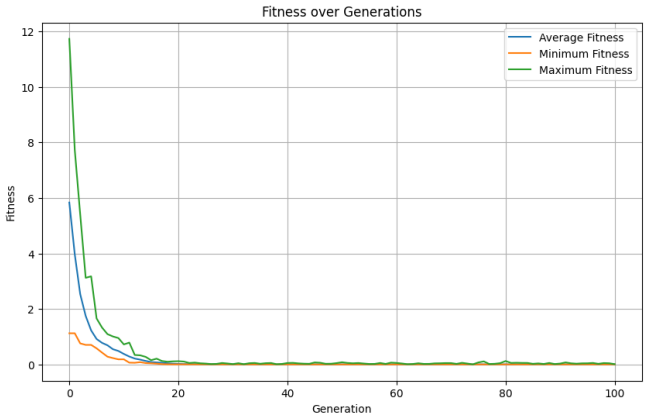

Figure 4: Fitness over generation trend (genetic algorithm)

Table 8: Comparing Results:

Metric	LR Score	ANN Score
accuracy	0.84	0.88
precision	0.88	0.91
recall	0.81	0.87
f1	0.84	0.89
roc_auc	0.92	0.95

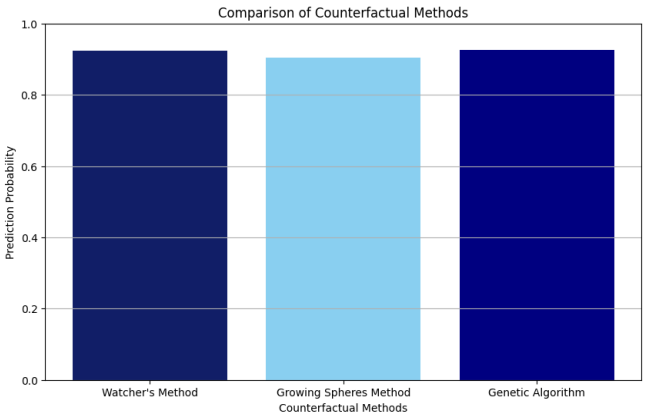

Figure 5: Overall comparison of counterfactual learning performance

Table 9: Result of Counterfactual Learning Methods

Method	Prediction Probability
Original Instance	N/A
Wachter's Method	0.94
Growing Spheres Method	0.95
Genetic Algoritm	0.94

CONCLUSION

In conclusion, this study presents a comprehensive comparative analysis of counterfactual explanation methods—Watcher's Method, Sphere Method, and Genetic Algorithm—applied to customer churn prediction models. Through evaluating key metrics such as Fischer Score, Imbalance Ratio, Volume, and Number of Features, we highlight the strengths and weaknesses of each method in generating actionable and interpretable counterfactual explanations. Our findings indicate that while Genetic Algorithm outperforms other methods in generating high-quality counterfactuals, each method

offers unique advantages depending on the context. Future research should focus on integrating these methods with techniques to handle class imbalance more effectively and exploring hybrid approaches to further enhance the interpretability and applicability of counterfactual explanations in various domains. Additionally, extending this analysis to other types of imbalanced datasets and incorporating real-world constraints and business rules into the counterfactual generation process would provide deeper insights and more practical solutions for decision-makers.

REFERENCES

Yuanyuan Li, Xue Song, Taicheng Wei, and Bing Zhu. 2023. Counterfactual learning in customer churn prediction under class imbalance.