Counterfactual Explanations for Commonly Used Text Classifiers focus on Review Classification

Warnasooriya S.D
Faculty of Computing
Sri Lanka institute of information Technology Malabe
10115, Sri Lanka
sashiniwarnasooriya@gmail.com

Britto T.A
Faculty of Computing
Sri Lanka institute of information Technology
Malabe 10115, Sri Lanka
thilinianjalika76@gmail.com

Lakshani N.V.M
Faculty of Computing
Sri Lanka institute of information Technology
Malabe 10115, Sri Lanka
maneeshalakshani9@gmail.com

Srinidee Methmal
Faculty of Computing
Sri Lanka institute of information Technology
Malabe 10115, Sri Lanka
srinidimunasinghe@gmail.com

Prasanna S.Haddela
Faculty of Computing
Sri Lanka institute of information Technology
Malabe 10115, Sri Lanka
prasanna.s@sliit.lk

Jeewaka Perera
Faculty of Computing
Sri Lanka institute of information Technology
Malabe 10115, Sri Lanka
jeewaka.p@sliit.lk

***Abstract*— In recent years, the use of AI-driven decision-making systems-based applications has increased significantly, and many of these applications involve making critical decisions that have a significant impact on human lives. Even though AI made systems are automated and easier to use, but lack of transparency and interpretability has become a major drawback of it. That causes most of the machine learning models that are used for decision-making processes are black box, which means the internal behaviors of the models are more complex and difficult for humans to understand. With the implementation of the European Union's GDPR, the area of Model explainability is being raised as a hot topic and it is the next step of AI. Explainable Artificial intelligence provides an explanation for the complex machine learning processes into insights that are easily comprehensible and provide transparency into the decision-making processes of AI systems. This research is about the development of a novel counterfactual rule generation-based explainable AI solution for the text classification domain, focusing on K nearest neighbor, logistic regression, random forest, and support vector machine models. The novel explanations of the KNN and SVM models are based on a customized word-flipping generator. Further explanation of Logistic regression provides a prediction score-based method, and Random Forest provides a feature importance-based method.**

***Keywords*— artificial intelligence, XAI, model explainability, text classification, black box models**

I. INTRODUCTION

At the dawn of the fourth industrial revolution, different domains are witnessing a fast and widespread adoption of artificial intelligence (AI) in the decision-making process under domain experts. Healthcare, Legal, military, transportation, and finance are some domains where AI models are used for the decision-making process. Decisions made by the AI model are extremely critical in a situation like a medical diagnosis of a disease. It's crucial that both clinicians and patients can understand and validate the reasoning behind AI driven diagnoses. Here explainable AI (XAI) provides clear, understandable insights into how an AI model arrives at its conclusions. After an AI model makes a prediction (e.g., the likelihood of a patient having a certain disease), it can present the relative importance of each input feature (e.g., patient symptoms, test results) that contributed to the prediction. Also, through the explanation, doctors can get an idea of how the outcome would be different when input features slightly changed. Also, there are some situations in which the models can be biased and provide inaccurate results. In such situations model explainability is well needed to identify and avoid errors in decision making.

There is no exact definition for model explain ability/interpretability and different authors prefer to use the most sensible definitions. The most popular and widely used definitions are "Interpretability is the degree to which a human can understand the cause of a decision" [1] and "Interpretability is the degree to which a human can consistently predict the model's result" [2]. The main objective of XAI is to answer the "wh" questions related to AI-based decision-making. For example, XAI should be able to answer, "why a particular answer was obtained?", "how a particular answer was obtained" and "when a particular AI based system can fail?" [3].

This research mainly focuses on enhancing the model explainability of selected Black-Box type classification models, K Nearest Neighbor, Logistic Regression, Random Forest, and Support Vector Machine. It provides a novel explainable method which is related to the text classification domain. The black box models refer to a complex computational algorithms or systems that process inputs to produce outputs, but the internal workings or mechanisms that are responsible for the transformation are not readily understandable or explainable from an external perspective. KNearest Neighbors (KNN) can become a black box due to its inherent nature of relying on local similarity measures for prediction. As the dimensionality of data increases, it becomes harder to interpret which specific features are driving the predictions, obscuring the underlying relationships. Additionally, the choice of the number of neighbors (k) and the distance metric can greatly influence outcomes but understanding their impact might not be straightforward. When used on high dimensional data the logistic regression model also acts as a 'black box' model. Because in high dimensional spaces the number of features

exceeds the number of instances, leading to sparse data distribution. The random forest model can become a black box when it has a large number of decision trees with complex interactions between the features. It is difficult to -linearly separable data. When the input data is transformed into a higher-dimensional space using a non-linear kernel, the decision boundary can become highly complex and difficult to visualize. Additionally, the kernel function used by the SVM may not have a direct interpretation in terms of the original input features, making it hard to understand which features are driving the SVM's predictions.

There are many explainable techniques that can be used to provide explainable solutions for Black-Box models. This is where the idea of "counterfactual explanation" comes in. A counterfactual explanation provides insight into a machine learning model's prediction by illustrating a hypothetical scenario wherein specific feature values are altered to achieve a different, desired outcome. In essence, it answers the "what if" questions by demonstrating the minimal changes needed to reverse a prediction.

Counterfactual analysis assists to detect and mitigate bias in text classification models by generating hypothetical scenarios where a particular protected attribute (such as race or gender) is changed. Here we can observe whether the model's output is affected by that attribute. If the model's output changes significantly when the protected attribute is changed, it may be a sign that the model is biased. Also, this Counterfactual analysis can be used to understand why a particular text classification model is making certain errors. As the previous bias detection solution, it generates hypothetical scenarios to identify where the input is slightly changed to cause model misclassification. By observing what aspects of the inputs are causing the error we can identify areas for improvement in the model. Furthermore, this counterfactual analysis can use to evaluate the fairness of text classification models. The rules can identify where certain groups are overrepresented or underrepresented and if the model is affected by the group differences.

This research study focuses on the implementation of novel counterfactual explanations for above mentioned black box models. The literature review section describes the existing methods related to the explainable AI domain and the methodology section discusses the proposed novel counterfactual explanation solutions in detail. Further results and discussion section provide a comparison of the outcomes between the existing method and the proposed methods.

Finally, discuss the future works related to this research study.

II. LITERATURE REVIEW

Since the XAI research area is still in its early stages, most of the techniques are in the research phase, and researchers have built communities for contributions to the XAI domain. From the proposed techniques, some have become more popular among the research community, and new contributions are also made on top of these techniques. LIME (Local Interpretable Model-Agnostic Explanation) [4], SHAP (Shapely Additive Explanations) [5], NICE (Nearest Instance Counterfactual Explanations) [6], and DICE (Diverse Counterfactual Explanations) [7] are examples of that. When we consider LIME, it interprets the model and explains the classification of the model in a faithful manner. It provides local optimum explanations that compute the important features by generating samples of the feature vector. Those

understand the overall decision-making process because the final decision is based on the output of many decision trees. The black box behavior of the SVM becomes more

samples follow a normal distribution. After getting the pronounced, When SVM is used to classify non predictions from the samples, it assigns weights to each of the rows to get an idea of how close they are to the original sample. Then LIME uses a feature selection technique to identify the most significant features. SHAP is a unified approach that has been developed based on coalition game theory. In that theory, they assign a reward to the game players according to their contributions to the game. SHAP assigns a feature importance value for each feature that affects a particular output result. It maps the input features to the output results based on that Sharpley value. The key difference between LIME and SHAP is in the way they assign weights to the input features. LIME uses a cosine measurement, while SHAP uses the Shapley formula. In this review, rule-based explanation methods will be discussed.

NICE is a counterfactual explanation method that aims to find the smallest and most meaningful changes to an instance that would alter the model's prediction. Finding the nearest instance and ensuring feasibility can be computationally intensive, especially for high-dimensional and complex datasets. This could limit the scalability of the method, and because of that, it provides solutions only for binary classification problems. DICE generates a diverse set of counterfactual explanations by providing insights into how slight changes in the input can alter the model's prediction. DICE assumes that the features are independent, which might not be the case in many real-world datasets, and the process of generating diverse counterfactuals can be computationally demanding, especially for high-dimensional data or complex models.

The SEDC method identifies counterfactual explanations in textual data [8]. The method iteratively removes features to gauge shifts in the model's predictions, with features that significantly alter predictions being highly influential for initial classification. The SHAP-C method, combining the strengths of Shapley Additive Explanations and SEDC, determines feature importance and crafts counterfactual explanations by sequentially omitting crucial features.

Shubham Rathi has attempted to generate partial post hoc P-type contrastive explanations [9] and corresponding counterfactual data points by illustrating the specific changes required in the data to attain the desired output. This methodology addresses the classifier's prediction for a given datapoint, which serves as a reference. These P-contrast questions take the form "Why [predicted class] not [desired class]?" It allows for a focused exploration of a single alternative by specifying the desired class.

The author of [10] had come up with a model-specific, explainable approach to SVM that focuses on inducing logic programs. Inductive logic programming (ILP) is a subfield of machine learning, and here the models do the learning process in the form of logic programming rules (Horn Clauses) that are comprehensible to humans. They use SHAP to calculate the feature importance value of the input features. This paper makes a novel contribution to introducing a novel ILP algorithm called SHAP-FOIL that iteratively learns a single clause for the most influential support vector based on the global behavior of the SVM model.

The author of [11] introduces a novel approach for NLP model interpretability and performance enhancement using k Nearest Neighbors (kNN) representations. By employing kNN, it uncovers insights at both individual and dataset levels, identifying influential training examples and exposing spurious associations. This technique improves predictions in tasks like Natural Language Inference (NLI) and bolsters model robustness against adversarial inputs. By leveraging kNN as a backoff strategy, it offers a holistic solution to tackle opacity, artifact identification, and performance issues in deep NLP models. The method involves mapping sequences to normalized hidden representations using a neural network like BERT or RoBERTa, computing kNN based on L2 distances, and utilizing a weighted softmax function to integrate kNN scores with base model predictions. This comprehensive approach enhances both understanding and efficacy in NLP models.

III. METHODOLOGY

The main objective of the project is to develop a novel method for generating counterfactual rules for the text classification domain using XAI, with the purpose of making Black-Box models such as K Nearest Neighbor, Logistic Regression, Random Forest, and Support Vector Machine more comprehensible. Here, the process of constructing the model, coming across the innovative XAI solution, and creating the frontend for end users accomplishes the main goal. The system's operational workflow comprises several discrete steps. Initially, using the user interface, the user must supply the system with the instance that needs to be forecasted as well as the pertinent training dataset. The chosen machine learning model will next be applied to the supplied data to obtain the forecast. Subsequently, the system will employ the innovative method for generating counterfactual rules to analyze the machine learning model and derive the counterfactual rules. Ultimately, the results of this process—counterfactual rules—will be brought to the user interface.

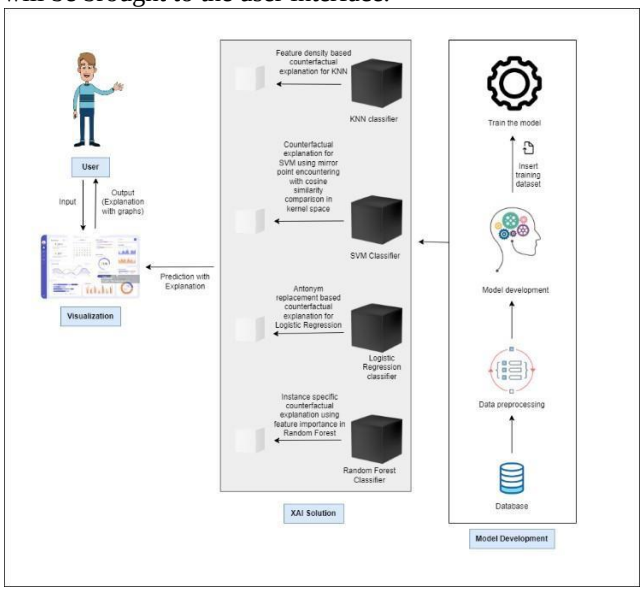

Fig. 1. Overall System Diagram of the proposed solution.

Dataset Description - The study used the IMDB dataset, consisting of 50,000 movie reviews. Each review, presented as a textual statement, encapsulates a user's opinion and sentiments regarding a particular film. Reviews are labeled either as 'positive' or 'negative', signifying the sentiment expressed in the review.

Data Preprocessing - Movie reviews often contain various textual elements like quotations, character names, or plot points. So, it is essential to follow the necessary preprocessing steps; In the model development process Initially, all characters in the text are converted to lowercase. Then remove any HTML content, any special characters, and stopwords present in the text. Finally did the words tokenization and lemmatization.

A. Feature Density Comparison based Counterfactual Explanation for K-Nearest Neighbor.

This research section contributes to the development of an explainable AI solution in the context of K-Nearest Neighbour (KNN). The approach is to develop counterfactuals for explaining the reviews. The following are the main steps followed in the system to generate counterfactuals for the KNN model: First, the system generates counterfactuals. To generate them, there is a variation count where the required number of counterfactuals is mentioned. According to the variation count, by using the word-flipping generator, the system will generate counterfactuals. Before generating the counterfactuals, the original review would be tokenized. After tokenizing, the system will flip words. The words that need to be flipped will be decided according to a token list. This token list will show what word category should be flipped (verbs, nouns, etc.). When flipping these words, to avoid getting the same opposite word again, it uses a random sampling method. Next, the system will vectorize the original review and the counterfactuals using TFID. This will generate separate vectors for the original review and the counterfactuals. Next, the system generates statistics for the given review. Here, the distance from the training data set vectors to the review would be calculated according to the given k value. By using the nearest k number of training data vectors, the system will classify the review. Here, a probability would be generated for the prediction based on how the training data set classifies. Finally, to decide the best counterfactual, the density of features is considered. Because there are many counterfactuals, deciding the best among them is important. So, in this step, the distribution of features of each counterfactual is considered. Here, the system generates the density using the following formula.

$$\text{density} = \frac{q}{average}\, d$$

In the above formula, q donates the total number of features that contributed to the prediction of the counterfactual / original review and daverage denotes the average distance from the new feature point to each feature. When calculating the density, the feature which contradicts the prediction was ignored. As in the above formula, the density of each prediction and the original review is calculated. The counterfactual with the nearest density to the original review's density is considered the best counterfactual. These steps will help to generate the best counterfactual for the given review with the probabilities for the predicted counterfactual.

B. *Antonym Replacement based Counterfactual Explanation for Logistic Regression Model.*

To shed light on the inner workings of the logistic regression model, we can conduct a feature analysis. Here the proposed method, uses score change to determine feature importance,

which is more efficient. So here we proposed a novel counterfactual opposite word replacement method using NLTK library. Here the steps involved in the proposed method are discussed further. First, we get the indices of features as an array. Then get the indices of the replacement features if such replacement exists. If an antonym exists, replace the feature with its antonym. If there is no antonym, remove the feature index from that array. After that use the logistic regression classifier to get the probability of the predicted class. If the probability is less than the predefined threshold value, consider it as a class change and add it to the list of counterfactual explanations. Here we generate counterfactuals by getting the change done to minimize the threshold value, if no class changes, add it to the array of combinations to expand. We should do it iteratively by removing or replacing each word. Then terminate the above process if the maximum number of iterations and time exceeds.

After that, we take the word combination to remove where changing the prediction score towards the reverse class is maximum. Then we expand the above word combination without the combinations in explanation. For each word combination above, replaces or removes the specified features. Then get the probability using logistic regression classifier. If the probability is less than the threshold value, we take it as a class change and add it to the explanation. If not, add it to the word combination to expand. Finally calculates shap values and iterates until termination. This process generates a novel counterfactual opposite word replacement method.

C. Instance-specific Counterfactual Explanation using Feature Importance in Random Forest Model.

In the context of the Random Forest model, feature importance plays a crucial role in decision-making. Random forest uses Gini impurity to calculate feature importance. However, a primary concern arises from the fact that these feature importance values are global, meaning they reflect the importance of features concerning the entire dataset, not for individual instances. This global nature makes it difficult to pinpoint which features are significant for a particular instance or prediction. Furthermore, they do not provide the direction of class change, which can be crucial for interpretability. Recognizing these challenges, our research aims to develop an instance-specific feature importance mechanism, accompanied by counterfactual explanations. Further discussion of the suggested method's steps follows. First, Extract feature importance from the trained Random Forest model. Then remove feature importance not related to the chosen movie review. Get the instance's prediction score from the model and label it as positive or negative. Remove common stop words from the specified instance. As the next step transform words to their base or root form by applying stemming and making an array. After that, every word in our processed array was considered a feature. Get the feature importance for each feature. Next, Compute the direction of change each feature offers to the given instance. If a feature pushes the prediction towards the positive class, assign a "+" sign, while those pushing towards negative received a "-". Then sort the features according to their importance. This helps us choose counterfactual explanation terms.

The main part of this method is removing features one by one. The proposed method starts removing the most impactful words and see how the model's prediction shifts. If the movie review is positive, remove the feature that pushes the most to the positive. Then proceed by iteratively removing the most impactful words from the text based on their feature importance. The idea is to see how the absence of these words affects the model's prediction. For the first three iterations, the guidance toward the counterfactual is based on the algebraic sum of the feature importance of the removed features. However, starting from the fourth iteration, guiding is done through the regular scoring method (according to the SEDC method). This is because when using only the algebraic sum of feature importance, it tends to neglect the combinational effect of features, which may lead to non-convergence to a result. If the instance is positive, continue this removal process, rechecking the prediction score each time, until the score drops below the predefined threshold value. This threshold is crucial because it helps us determine if the model's prediction changes from one class to another. This iterative removal helps us pinpoint which words, when absent, lead to significant changes in model predictions. The words removed in this process essentially form the counterfactual explanation. They represent the minimum changes required to the original text to change the model's decision.

D. Counterfactual Explanation for Support Vector Machine using mirror point encountering with cosine-similarity comparison in kernel space.

The novel SVM explainable method provides the counterfactual solution by going through five steps. First it generates contractionary prompts for a given prompt $\boldsymbol{prompt_0}$ using finetuned T5 model or custom WordFlippingGenerator. These new prompts will be [$\boldsymbol{Contradictionary\_prompt_i}$]. WordFlippingGenerator randomly flips the words with defines POS tags to their antonyms. Here the user should define the POS tags that are relevant to the words that must be flipped and invoke the functionality by specifying an original sentence with the number of variations needed. WordFlippingGenerator algorithm will tokenize the words and generate a mask list that corresponds to the tokens by referring to the POS tags previously defined by the user. The true value of the mask represents that the word must be flipped and false means otherwise. Finally, the algorithm generates a set of lists that contain antonyms that are ordered according to the descending order of occurrence probability for the flipped words referring to the mask. New sentences will be generated by merging the original words with the antonyms generated by the WordFlippingGenerator.

After the contractionary prompt generation, TFIDF vectorizer vectorizes all the prompts into vector space X. In there the $\boldsymbol{Prompt_0}$ will be mapped to $\boldsymbol{x_0}$ and the [$\boldsymbol{contradictionary\_prompt_i}$] will be mapped to $\boldsymbol{x_{c,i}}$ vector space using TFIDF vectorizer. As the third step of the method, algorithm project all the vectors ($\boldsymbol{x_0}$ and $\boldsymbol{x_{c,i}}$) into the SVM' s kernel space K. Here SVM use Radial Basis Function (RBF) as the kernel [$\emptyset(.)$]. Next, must find the mirror point of the given prompt's TFIDF vector on the hyperplane of the SVM ($\boldsymbol{C}$). Once have $\emptyset(\boldsymbol{x_0})$ for the given prompt, find its opposite projection on the hyperplane of the SVM characterized by $\boldsymbol{w}$ and $\boldsymbol{b}$. As the final step, algorithm find the closest point to mirror point ($\boldsymbol{C}$) and retrieve the most accurate contractionary

prompt as the output. Here the mirror points and contra dictionary points all are in the kernel space. Most accurate contradictory prompt return using the cosinesimilarity between the mirror point and the contradict nary prompts.

IV. RESULT AND DISCUSSION

The components outlined in the previous sections are instrumental in generating optimal counterfactual explanations for a given review. For the result discussion, we selected the novel explanation method of the Random Forest model from the implemented novel methods. To evaluate the performance of the novel Random Forest explanation method, we compared it to the SEDC method that initiated the implementation of the novel Random Forest explanation. To provide novel explanation solutions, we utilize the IMDB movie reviews dataset.

A. Negative Review Evaluation

For the negative review evaluation, we applied the same review for both the novel and the SEDC methods. The class change of the two methods related to the applied negative review is visualized using a multiple-line graph. The y-axis shows the prediction score, and the x-axis shows the iteration number. The blue and green lines represent the class changes of the novel method and SEDC method, respectively. The orange line represents a predefined value of 0.493, which is utilized in both SEDC and novel methods known as threshold.

"Disappointing movie. Weak storyline, wooden performances, and uninspired direction. A waste of time and money." is the review that we have considered in this section.

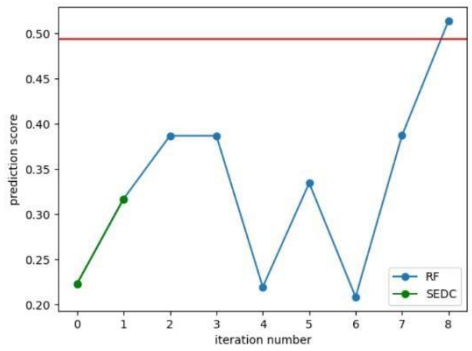

Fig. 2. Prediction score for negative to positive class change

According to Figure 2, the initial prediction score of the novel method of the above review is 0.223 indicating a negative review (threshold > 0.22). To achieve a class change, first, start the removal process according to the "instancespecific counterfactual explanation using feature importance in Random Forest model". Then continue this removal process by rechecking the prediction score, until the score reaches to the threshold value. Figure 2 shows that there is a class change to positive in the 8th iteration with a final score of 0.528. Conversely, in the SEDC method, even after the maximum number of iterations, there is no class change from negative to positive.

The instance-specific counterfactual explanation for random forest returns the most affecting words to provide a counterfactual explanation. For the above-mentioned review, the explanation returns "disappointing", "weak", "wooden", "uninspired" and "waste" words. Therefore, those features mostly affected to the label change from negative to positive.

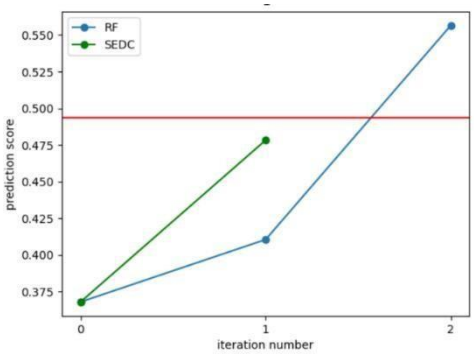

Fig. 3. Negative to positive review prediction score graph

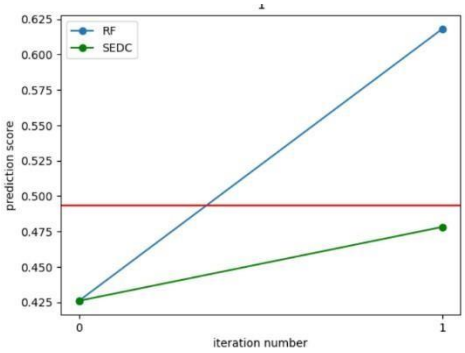

Fig. 4. Negative to positive review prediction score graph

Upon conducting this analysis with multiple negative reviews as shown in figure 3 and Figure 4, it became evident that the SEDC method consistently fails to provide a class label change from negative to positive. In contrast, the novel Random Forest explanation method consistently achieved this change.

B. *Positive Review Evaluation*

Here we applied the same positive review for both the novel and the SEDC method. Also used the same threshold value as 0.493. In both SEDC and novel methods, to change the class label to negative, the prediction score must be less than the threshold value for the positive reviews.

"Fantastic film! Gripping plot, superb acting, and breathtaking visuals. A must-see for all movie lovers." is the review that we have considered in this section.

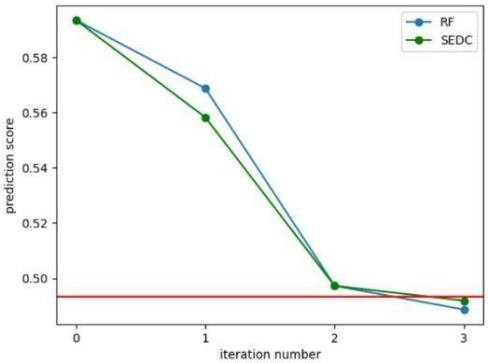

Fig. 5. Prediction score for novel method of RF model

According to Figure 5, the initial prediction score of the novel method for the above review is 0.593, indicating a positive review (threshold > 0.22). Then it starts the feature removal process as before and continues it until the score is less than the threshold value. There is a class change from positive to negative on the 3rd iteration and after the 3rd iteration, the final score is mentioned as 0.489. When we apply the SEDC method for the same positive review it also changes the class to negative after 3 iterations. Here, the

prediction score value is less than the threshold value after 3 iterations and it shows the score as 0. 493. For the abovementioned review, the explanation returns “fantastic”, “gripping”, “superb”, “breathtaking” and “must-see” as the most affected words for the counterfactual explanation.

“The cast’s performances were exceptional, each actor delivered with passion and authenticity.”

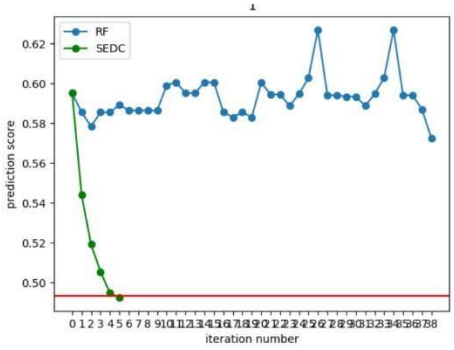

Fig. 6. Prediction scores of novel and SEDC methods

When we considered another positive review as above, and its prediction scores as shown in figure 6, the initial prediction score of the novel method for that review is 0.595, indicating a positive review (threshold > 0.22). After the feature removal process, the class did not change even after 38 iterations occurred. However, the SEDC method changed the class label to negative after the 5th iteration.

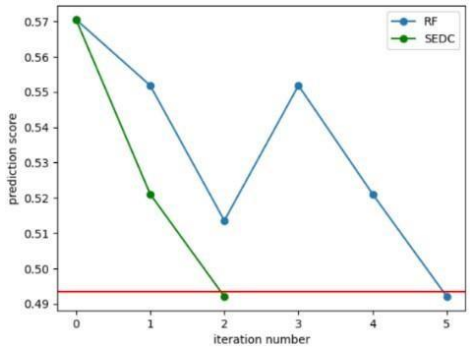

Fig. 7. Prediction scores from positive to negative class change

Figure 7 shows the results of another positive review when we applied both novel Random Forest and SEDC methods. Here both the methods change the class after a few iterations. In that, we can realize the novel counterfactual method of Random Forest does not provide class change to negative in some situations.

IV. CONCLUSION AND FUTURE WORKS

This research aims to improve model explainability in AI driven decision-making systems, focusing on black-box classification models like K Nearest Neighbor, Logistic Regression, Random Forest, and Support Vector Machine for text classification. The goal is to create counterfactual-based explainable AI solutions for each model, using innovative and model-specific approaches. The research produced generated counterfactuals, optimal counterfactuals, mirror point distances, and positive and negative prediction probabilities. Future research should focus on dynamic POS tag selection and contrast variations, and advanced visualization techniques to make XAI solutions more accessible and user-friendly. The results have established a foundation for improving AI-driven decision-making system transparency and trustworthiness.

Acknowledgement: The author/s would like to express heartfelt gratitude to the IEEE CSDE 2023 Organizing Committee for their generous conference registration fee scholarship, which made our participation in this event possible and published our research paper with IEEE Xplore.

REFERENCES

[1] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267. 2019. doi: 10.1016/j.artint.2018.07.007.

[2] B. Kim, R. Khanna, and O. Koyejo, “Examples are not enough, learn to criticize! Criticism for interpretability,” in *Advances in Neural Information Processing Systems*, 2016.

[3] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2870052.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘why should i trust you?’ explaining the predictions of any classifier,” in *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016. doi: 10.18653/v1/n16-3020.

[5] S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, 2017.

[6] D. Brughmans, P. Leyman, and D. Martens, “NICE: an algorithm for nearest instance counterfactual explanations,” *Data Min Knowl Discov*, 2023, doi: 10.1007/s10618-023-00930-y.

[7] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020. doi: 10.1145/3351095.3372850.

[8] Y. Ramon, D. Martens, F. Provost, and T. Evgeniou, “A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C,” *Adv Data Anal Classif*, vol. 14, no. 4, 2020, doi: 10.1007/s11634-02000418-3.

[9] S. Rathi, “Generating Counterfactual and Contrastive Explanations using SHAP,” *arXiv.org*, Jun. 21, 2019. https://arxiv.org/abs/1906.09293 (accessed Oct. 07, 2023).

[10] F. Shakerin and G. Gupta, “White-box Induction from SVM Models: Explainable AI with Logic Programming,” in *Theory and Practice of Logic Programming*, 2020. doi: 10.1017/S1471068420000356.

[11] N. F. Rajani, B. Krause, W. Yin, T. Niu, R. Socher, and C. Xiong, “Explaining and Improving Model Behavior with k Nearest Neighbor Representations,” *arXiv.org*, Oct. 18, 2020. https://arxiv.org/abs/2010.09030 (accessed Oct. 07, 2023)

[12] Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers*, *4*(8).

[13] Peng, J., Zou, K., Zhou, M., Teng, Y., Zhu, X., Zhang, F., & Xu, J. (2021). An explainable artificial intelligence framework for the deterioration risk prediction of hepatitis patients. Journal of medical systems, 45, 1-9.

[14] Ge, Q., Huang, X., Fang, S., Guo, S., Liu, Y., Lin, W., & Xiong, M. (2020). Conditional generative Adversarial networks for individualized treatment effect estimation and treatment selection. Frontiers in genetics, 11, 585804.

[15] Sharma, S., & Sharma, V. (2023). Comparison of machine learning techniques in the diagnosis of erythematous squamous disease. Journal of Scientific Research and Technology, 1-9.

[16] Rathi, S. (2019). Generating counterfactual and contrastive explanations using SHAP. arXiv preprint arXiv:1906.09293.

[17] Laugel, T., Lesot, M. J., Marsala, C., Renard, X., & Detyniecki, M.). Unjustified classification regions and counterfactual explanations in machine learning. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II (pp. 37-54). Springer International Publishing.