Amit Vikrant Dhavale

1. Introduction

The report begins with a simple breakdown of the complex implementation details of model selection, as well as statements about methods adopted in empirical evaluations. The report begins with a simple breakdown of the complex implementation details of model selection, as well as statements about methods adopted in empirical evaluations. The report is divided into 5 sections: **Introduction** presents an Introduction to the topic. **Task 1** performs model selection based on exploratory data analysis results provided in the notebook, **Task 2** focuses on implementing and evaluating different feature attribution methods, including SHAP, Shapley Value Sampling, and DeepLIFT, to analyze their effectiveness and computational efficiency. **Task 3** focuses on implementing and evaluating two counterfactual explanation methods: Nearest-Neighbour Counterfactual Explainer (NNCE) and Wachter et al. (WAC).

2. Task 1

After performing exploratory data analysis, several insights were inferred from the dataset.

Considering the feature correlations and the distribution of the 'sex' feature, certain trends and patterns were observed. The feature 'pclass' showed a strong negative correlation (-0.3) with the label "Survived," indicating that passengers in higher classes had a better chance of survival. Additionally, the feature 'fare' demonstrated a strong positive correlation (0.25) with survival, suggesting that passengers who paid more had a higher likelihood of survival, potentially due to access to better accommodations and services. Furthermore, a strong negative correlation (-0.56) was observed between 'pclass' and 'fare', indicating that passengers in higher classes paid more for their tickets than those in lower classes. Examining the distribution of the 'sex' feature revealed that 25.2% of female passengers survived, compared to only 11.3% of male passengers. This major difference in survival rates between males and females strongly suggests that male passengers had lower survival rates. **(Note: Higher classes correspond to lower numerical values of 'pclass'. For example, 'pclass' = 1 represents a higher class.)**

Without access to any particular model or its explanations, we can infer the most and least important features for a neural network (NN) trained on this dataset based on their correlations with the target variable. Features with strong positive or negative correlations with the label are likely to be the most important. Thus, 'fare' (correlation = 0.25) and 'pclass' (correlation = -0.3) emerge as significant predictors of survival. Additionally, the distribution of the 'sex' feature suggests that it is also an important factor. Conversely, features with weak correlations with the label can be considered less important. Attributes such as 'age' (correlation = -0.049), 'sibsp' (number of siblings/spouses, correlation = -0.015), and 'parch' (number of parents/children, correlation = -0.084) appear to have minimal impact. However, it is crucial to consider that machine learning models, particularly neural networks, can learn complex non-linear relationships that simple correlation analysis might not capture. A feature that seems unimportant based on correlation could still be influential within the model due to interactions with other variables. Therefore, this analysis is purely data-driven and does not guarantee insight into any specific model.

Beyond examining correlation plots, additional exploratory analysis techniques could further enhance our understanding of the dataset and model behaviour. Creating histograms for continuous features such as 'Age' and 'Fare' could help assess their distributions, whether they are normally distributed, skewed, or contain outliers. Additionally, scatter plots could be used to explore relationships between continuous variables such as 'Age' and 'Fare', potentially revealing correlations or clusters that may not be evident from individual distributions. These additional visualizations would provide a more comprehensive understanding of the dataset.

3. Task 2

In this task, three different methods: SHAP, Shapley Value Sampling and DeepLIFT are implemented. Further, these three methods are then used to compute the feature attribution scores for 10 randomly sampled points from the Titanic test set. Subsequently, the same process is applied to the 'Dry Bean Dataset'. Finally, the computational runtimes required to generate the attribution scores for each method are compared, providing insights into their efficiency respectively.

3.1. SHAP (My Implementation)

SHAP is a method that assigns each feature an importance score based on its contribution to the model's predictions by computing the average marginal contribution of the feature across all possible feature subsets. Table 1 shows the absolute mean feature importance values for the features calculated using SHAP in the Titanic dataset.

3.2. Shapley Value Sampling (Using Captum)

It is an approximation method for computing SHAP values. It estimates the Shapley values by averaging the marginal contributions of a feature over multiple sampled

Ranked Features	Absolute Mean SHAP Value
pclass *	0.2094
fare *	0.1878
sex_m *	0.1714
sex_f *	0.1714
age	0.1702
parch	0.1403
sibsp	0.1158

Table 1. Absolute mean feature importance values for the Titanic dataset using SHAP.

permutations of the input features, making it scalable for large models and datasets. Table 2 shows the absolute mean feature importance values for the features calculated using Shapley Value Sampling in the Titanic dataset.

3.3. DeepLIFT (Using Captum)

It is an attribution method used to explain the predictions of deep neural networks. Further, it works by assigning importance scores to input features based on their contribution to the output prediction, relative to a reference or baseline input. Table 3 shows the absolute mean feature importance values for the features calculated using DeepLift in the Titanic dataset.

Note: The features marked with an asterisk (*) have the highest attributions, highlighting their significant importance in the model's predictions.

Roughly, all three methods assign high importance to the 'sex', 'pclass', and 'fare' features, while 'age', 'sibsp' and 'parch' receive lower importance. These attributions align with the insights from the exploratory data analysis in **Task 1**. However, unlike Shapley Value Sampling and DeepLIFT, SHAP assigns the highest importance to 'pclass', whereas 'fare' ranks highest in the others. This discrepancy may arise from Shapley Value Sampling's inherent randomness and approximation. Also, SHAP assigns relatively high importance to 'age', indicating that most of the sample from the chosen 10 data points were children, where 'age' plays a significant role. Additionally, unlike SHAP and Shapley Value Sampling, where 'sex_m' and 'sex_f' have identical attributions, DeepLIFT gives significantly lower importance to 'sex_m'. This is likely due to its different mechanism, which uses reference points and backpropagation to compute the importance of features.

Thus, **Exploratory Data Analysis (EDA)** provides global insights, offering a broad understanding of the dataset and the relationships between features. It is also model-agnostic and the insights gained are generalizable. However, it is not model-specific making it unable to capture unusual or complex behaviors that a specific model might exhibit. Whereas, **Feature Attribution Methods (SHAP, Shapley Value Sampling, DeepLIFT)** are model-specific. They also provide instance-specific insights, which can be useful for understanding edge cases. However, they are computationally expensive, with SHAP being slow, especially for large datasets or complex models. Moreover, Shapley Value Sampling and DeepLIFT involve approximations, which can lead to inaccuracies. Therefore, according to me, Feature Attribution Methods are best for understanding specific model behaviours and predictions. However, they should be used in conjunction with EDA to ensure that the model's behaviour aligns with the data distribution.

3.4. Infidelity

The infidelity metric evaluates the robustness of attribution methods by measuring how sensitive the explanations are to perturbations in the input. The hyperparameters used for this evaluation include the noise scale, which controls the magnitude of Gaussian noise added to continuous features, with values tested at 0.1, 0.5, and 0.9. Another hyperparameter is the categorical resampling probability, which controls the probability of resampling categorical features, with values tested at 0.1, 0.5, and 0.9.

As observed in Table 4, the infidelity scores for all methods increase with the noise scale, as larger perturbations make it harder for the attribution methods to provide stable explanations. Similarly, the infidelity scores rise with the categorical resampling probability, though the effect is minimal.

DeepLIFT generally exhibits the lowest infidelity across most hyperparameter settings, particularly at higher noise scales (e.g., 0.9), indicating that it is more robust to input perturbations. Shapley Value Sampling performs slightly better than SHAP in most cases, especially at lower noise scales (e.g., 0.1 and 0.5). SHAP, on the other hand, tends to have the highest infidelity, suggesting it is less robust compared to the other methods.

Moreover, the performance gap between methods narrows as the noise scale increases, with all methods showing similar infidelity at a noise scale of 0.9. Therefore, DeepLIFT is the most robust attribution method for the Titanic dataset, particularly under high levels of input perturbation. Shapley Value Sampling performs slightly better than SHAP, which is the least robust, especially under high noise levels.

3.5. Evaluation of the computational efficiency

In this part, a Neural Network model is trained and evaluated on a preprocessed Dry Bean Dataset. The observed performance of the model is shown in Table 5

Each of the feature attribution methods was applied to both the Titanic and Dry Bean datasets, and the respective runtimes were calculated. Table 6 shows the runtimes for each method on the two datasets.

From Table 6, it can be observed that SHAP has a relatively high runtime as compared to Shapley Value Sampling and DeepLift, indicating that SHAP is much less efficient than the other methods. Further, as the number of features increases, the computational time for SHAP grows exponentially. Specifically, for n features, the total number of feature combinations is 2^n, leading to a substantial increase in computational cost. **(Note: To reduce the runtime for computing feature attributions for Dry Beans Dataset, the number of features in this dataset is reduced to 11)**

4. Task 3

In this task, two different Counterfactual Explainers, Nearest-Neighbour Counterfactual Explainer (NNCE) and Wacther et al (WAC) are implemented to generate counterfactual explanations for 20 randomly selected test instances of the Titanic Dataset. Further, the results are evaluated using the following metrics: proximity, validity and plausibility, to get some useful insights.

4.1. Distance Metric

Equation (1) depicts the formula for the Standard L1 distance, where k is the number of features:

$$d_{L1}(x, x') = \sum_{i=1}^{k} |x_i - x'_i| \tag{1}$$

In Standard L1, the implicit weightings of each feature are different, where the features with larger value ranges are implicitly given higher importance while calculating the distance. This issue is significant in the preprocessed Titanic dataset, where some features have different ranges compared to others due to the use of the Robust Scaler. Table 7 presents the range of each feature in the dataset.

To address the aforementioned problem, we can normalize the L1 distance using the maximum and minimum values of each feature (indexed as i) in the training dataset, denoted as $\max_i$ and $\min_i$, respectively. The normalized L1 distance is given by Equation (2):

$$d_{L1,\text{normalized}}(x, x') = \sum_{i=1}^{k} \left| \frac{x_i - x'_i}{\max_i - \min_i} \right| \tag{2}$$

However, the normalized L1 distance does not accurately capture the effect of categorical features, particularly '**sex**' in the dataset. For example, in the Titanic dataset, after applying One-Hot Encoding, the sex feature is transformed into two separate features, namely 'sex_m' and 'sex_f'. As a result, both distance metrics discussed earlier assign twice the importance to the sex feature compared to what it should have received.

To solve this issue, I introduced customized weighting factors $\mathbf{w} = \{w_1, ..., w_k\}$ to capture the importance of each feature. The weighted normalized L1 distance is given by Equation (3):

$$d_{L1,\text{normalized, weighted}}(x, x') = \sum_{i=1}^{k} w_i \left| \frac{x_i - x'_i}{\max_i - \min_i} \right| \tag{3}$$

where the weights for categorical features are set to $\frac{1}{n}$, where n is the number of unique values in the given categorical feature while for all other features, the weight is set to 1.

4.2. Implementation of Counterfactual Explanations

Table 8 presents the mean and standard deviation of the evaluated metrics: validity, cost, and plausibility for different methods. The validity of NNCE is 100% with 0 standard deviation. This is expected because NNCE explicitly searches for counterfactuals in the training dataset with the desired label. Thus, considering only valid counterfactuals. However, The validity of WAC is 85% ± 9.5%. This lower validity can be attributed to the Relaxed Loss Function where WAC optimizes a loss function that balances validity and proximity. This trade-off may result in counterfactuals that do not completely satisfy the validity condition. Further, the gradient-based optimization used in WAC may get stuck in local optima, leading to suboptimal counterfactuals that do not have the desired label.

Now, the observed proximity (cost) for NNCE is 0.137 ± 0.02. This indicates that NNCE finds counterfactuals that are relatively close to the input. However, the proximity of WAC is 0.14 ± 0.037, which is slightly worse than NNCE. This is not an expected behaviour as in theory, WAC should have better proximity than NNCE because it directly optimizes the distance to the input. But in practice, the gradient-based optimization may converge to suboptimal solutions, resulting in counterfactuals that are farther from the input. Furthermore, its performance is sensitive to the choice of hyperparameters (e.g., trade-off parameter λ).

Now, the plausibility of NNCE is 0.035 ± 0.004, which is significantly better than WAC (0.17 ± 0.032). One of the reasons could be that NNCE selects counterfactuals directly from the training dataset, ensuring that they are realistic and plausible. In fact, by definition, NNCE finds the nearest neighbour with the desired label, which is inherently plausible.

Finally, it is worth noting that WAC may generate counterfactuals with floating-point values for categorical features (e.g., sex_f = 0.9213, and sex_m = 0.0783), which are unrealistic. This is a direct consequence of the gradient-based optimization, which functions in continuous space and does not enforce discrete constraints.

5. Appendix

Ranked Features	Absolute Mean Shapley Value
fare *	0.2793
sex_m *	0.2462
sex_f *	0.2462
pclass *	0.1847
age	0.1201
sibsp	0.0673
parch	0.0265

Table 2. Absolute mean feature importance values for the Titanic dataset using Shapley Value Sampling.

Ranked Features	Absolute Mean DeepLIFT Value
fare *	0.3790
sex_f *	0.2640
pclass *	0.2388
age	0.1196
sibsp	0.0747
sex_m	0.0548
parch	0.0404

Table 3. Absolute mean feature importance values for the Titanic dataset using DeepLIFT.

Noise Scale	cat_resample_prob	SHAP Infidelity	Shapley Value Sampling Infidelity	DeepLIFT Infidelity
0.1	0.1	0.0898	0.0897	0.0874
0.1	0.5	0.0935	0.0907	0.0930
0.1	0.9	0.0958	0.0902	0.0990
0.5	0.1	0.2124	0.1993	0.2026
0.5	0.5	0.2178	0.2080	0.2101
0.5	0.9	0.2246	0.2215	0.2183
0.9	0.1	0.2598	0.2599	0.2609
0.9	0.5	0.2684	0.2684	0.2672
0.9	0.9	0.2852	0.2788	0.2730

Table 4. Infidelity metrics for different noise scale and categorical resampling probability values.

Metric	Value
Test Loss	0.2210
Test Accuracy	92.25%
Test F1 Score	0.9225
Test AUC Score	0.9946

Table 5. Model performance on the preprocessed Dry Bean Dataset.

Method	Titanic Runtime (s)	Dry Bean Runtime (s)
SHAP	122.3034	916.7
SVS	0.1520	0.1553
DeepLift	0.0150	0.0075

Table 6. Runtimes of feature attribution methods on Titanic and Dry Bean datasets.

Feature	Min Value	Max Value
pclass	-1.0000	0.0000
age	-0.8978	1.6774
sibsp	0.0000	8.0000
parch	0.0000	4.5000
fare	-0.2064	7.0881
sex_m	0.0000	1.0000
sex_f	0.0000	1.0000

Table 7. Feature Ranges in the Preprocessed Titanic Dataset

Method	Validity	Cost	Plausibility
NNCE	1.0 ± 0.0	0.137 ± 0.02	0.035 ± 0.004
WAC	0.85 ± 0.095	0.14 ± 0.037	0.17 ± 0.032

Table 8. Mean and standard deviation of the evaluated metrics: validity, cost, and plausibility.