Michael Hollins - 02553236

1. Exploratory Data Analysis

1.1. Feature correlations and patterns

Fare and passenger class are strongly negatively correlated; this makes intuitive sense as higher class tickets would be more expensive. This comes to bear on survival, as passengers who had a higher class of ticket were more likely to survive.[1] Equivalently, passengers who paid more for their ticket were more likely to survive. Furthermore, given how many females survived out of all female passengers, this could imply that women tended to have a higher class of ticket.

This intuitive relationship between fares and passenger class is complicated slightly when we consider the role of families. There is a strong positive associated between passengers who had parents/children on board and those who had siblings and spouses. Both variables also have a positive relationships with fares so families paid more to travel. However, they have no relationship with passenger class, meaning that families' tickets cost more for a given class. This could explain the low correlation between the family variables and survival.

The other strong relationship is the negative correlation between age and travelling class, showing that younger people tended to travel on lower class tickets.

Where there are weak correlations, this could be due to conflicting forces. For example, age is uncorrelated with survival probability. However, the causal reasons behind this aren't clear. It could be that younger children were more likely to get aboard a lifeboat, but also more likely to perish in the cold. The point is that this correlation analysis doesn't necessarily explain why the outcomes occurred, only that certain outcomes are related.

1.2. Expected most and least important features

Given that the neural network will try to predict survival, I think that the most important features would be sex and passenger class. This is due to the aforementioned large correlation coefficients with survival, and the clear difference between proportional female and male survival rates. By extension, the least important features look to be age and the familial variables. However, given the correlations between features and with no formal predictive analysis yet undertaken, I cannot be too certain in this assessment.

1.3. Further inspection of the data

A key ingredient missing from the exploratory data analysis so far is a look at the distributions of each continuous variable. This would be important for identifying outliers that may skew the results, which is an important consideration for any preprocessing feature normalisation. It would also be desirable to check the raw data quality, such as the number of missing values, so we can anticipate the consequence of filling in missing age and fare with the median values as we do in the data preprocessing.

[1] As higher passenger class is represented by lower numbers, this explains the negative sign of correlation.

2. Feature Attribution Explanations

Table 1 shows the average results of the ten feature attribution experiments per implementation method.

Attribution method	pclass	age	sibsp	parch	fare	sex (f)	sex (m)
My Shapley	-0.0578	0.0184	-0.0208	-0.0199	0.0196	-0.0039	-0.0039
Captum SVS	0.2395	-0.0331	-0.0015	0.0305	0.1054	-0.3446	-0.3446
DeepLIFT	0.4646	0.0305	0.0038	0.1007	-0.3371	0.0272	-0.2935

Table 1. Feature attribution scores for the Titanic Dataset per implementation method.

2.1. Comparing the different implementations

2.1.1 Most/least important features per explanation

Importance is shown by the absolute value of the attribution coefficient. Accordingly, in my implementation of SHAP, the most important feature for the explained model is passenger class, and the least important is sex. For Captum's implementation of Shapley Value Sampling (SVS), the most important features are passenger class and sex; age, siblings/spouses and parents/children are all relatively unimportant. Finally, for DeepLIFT, passenger class and fare are most important, while age, siblings/spouse and (interestingly) being female are all unimportant.

2.1.2 Possible reasons for differences

There are substantial differences between the different attribution methods. Before reviewing them, it is worth exploring the different methodologies underpinning each method. My Shapley implementation replaces the value of each deleted feature by a newly sampled value from the background dataset. This is done for each deleted feature in each sample. By contrast, SVS simply replaces deleted features with zero.

DeepLIFT backpropagates the contributions of all neurons in the network to every feature in the input. It then compares this against a "neutral" or "baseline" input and then attributes the difference in output to the differences in input. The baseline I use in the code is zero. However, unlike in SVS/Shapley, DeepLIFT doesn't evaluate all possible combinations of features, and this could be especially

important in complex models which have sizable interactions between features.

Furthermore, this is likely to explain a large part of why the results differ. Both Captum SVS and DeepLIFT place importance on sex, whereas my Shapley implementation doesn't. Rather, my Shapley attribution for sex implies that in the context of other features, sex is not that important, and this underlines the necessity of considering how the features interact.

Regarding passenger class, my Shapley attribution value is negative. As I'm comparing against a broader context of data variability, in this case the negative Shapley value means that within the data variability, the specific passenger class contributes negatively to survival predictions, highlighting a detrimental impact on the model's learned patterns for predicting.

2.1.3 Do attributions match *a priori* expectations?

Both Captum's SVS and DeepLIFT are close to my pre-modelling expectations regarding the strength and sign of attributions. Broadly speaking, passenger class and sex are very important for predicting survivability whereas the other features such as age and familial connections are unimportant. My model still shows that passenger class is important but, in addition, that sex is not. This could be due to sex interacting with many other features (such as passenger class); this could help guard against a simplistic understanding of the role of sex on survivability in which being female made one more likely to survive. It could be rather than being female interacted with other features that were important determinants of survivorship.

2.1.4 Potential advantages/disadvantages of each approach

I would argue that by resampling a different value from the parent dataset for each feature in each sample, this most faithfully ensures a sensible baseline which is appropriate for our features. However, this clearly has computational costs as we must resample at least once for every feature we remove.

Introducing a baseline of zero could lead to undesirable counterfactual observations. For example, passenger class takes values $\{1, 2, 3\}$ it is unclear what a zero value replacing it would mean. It could also prove challenging for continuous features. For example, by deleting the fare paid and replacing it with zero, this creates an instance where the ticket is free. By assuming a baseline of zero, the method assumes that this won't introduce bias or misrepresentation of the feature's absence, and this is unlikely to be the case in categorical variables or when we have a feature that is non-zero-centred.

Unlike in SVS/Shapley, DeepLIFT doesn't evaluate all possible combinations of features, and this could be especially important in complex models which have sizable interactions between features. However, its interpretations is intuitive and tractable.

2.2. Infidelity on the whole dataset

Table 2 shows the total infidelity for each feature attribution method in the presence of varying Gaussian noise for continuous features and resampling probabilities for categorical features. The encouraging finding is that regardless of the combination of noise and resampling probabilities, my Shapley implementation has lower infidelity than the Captum implementation of SVS. This is likely due to the resampling process which gives more realistic baselines for masking features. However, as is shown in the next section, this resampling comes at significant computational cost.

DeepLIFT appears to perform better than the Shapley methods in the presence of low noise and resampling probabilities. However, as the perturbations get stronger in the continuous features, my implementation of Shapley using resampling performs the best. This could be because it is derived using resampling which, in a sense, mimics the impact of perturbation.

Noise	Resampling prob	Method	Total infidelity
0.1	0.1	My Shapley	0.6881
		Captum SVS	0.8717
		DeepLIFT	0.6136
0.5	0.1	My Shapley	1.8447
		Captum SVS	1.9723
		DeepLIFT	1.7156
0.1	0.5	My Shapley	0.8429
		Captum SVS	0.9073
		DeepLIFT	0.9480

Table 2. Sum of infidelity across all feature attributions per feature attribution method over varying Gaussian noise on continuous features and resampling probabilities on categorical features

2.3. Computational efficiency

While the titanic dataset has seven features, the dry bean dataset has sixteen. As Table 3 shows, this has significant computational implications.

Method	Titanic (s)	Dry beans (s)	Factor increase
My Shapley	0.527	1,434	2,721
Captum SVS	0.071	0.139	2
DeepLIFT	0.002	0.002	0

Table 3. Time taken in seconds to calculate attributions for one sample per dataset.

To process a single sample, DeepLIFT is quickest and computation time doesn't meaningfully increase with features. This is likely because it uses a pre-computed model so there is little further computation required. Captum SVS takes slightly longer but appears to have computation time increase linearly in the number of features. By contrast, my Shapley implementation takes significantly longer than the other methods for each dataset, and even more so with increased features in the case of the dry beans data. The jump between the datasets appears to be exponential in features. This is likely due to a combination of more coalitions to loop through and more resampling. Due to time constraints, I was unable to compare computational cost for 200 samples, but by extrapolating it would take my Shapley method around three and a half days to run on the dry beans dataset.

3. Counterfactual Explanations

3.1. Designing a distance metric

3.1.1 L1 and normalised L1

With an original, unprocessed dataset, the L1 norm is invariant to the range that the different features can take, and so might not be best-suited for counterfactuals. Moreover, even in the case of normalised L1, outliers might mean that the weights don't affect the features evenly. However, in our case of the preprocessed titanic dataset, each continuous feature has been transformed in sklearn by robust scaling. This technique removes the median and scales the data according to the quantile range between the 10th and 90th quantile, precisely in order to control for outliers.

3.1.2 Treating the features equally

While in the original weighted average counterfactuals (WAC) implementation the authors use the median absolute deviation to normalise the distance metric and control for outliers , we have done something very similar by robust scaling [1]. Therefore, a simple L1 norm will do. The caveat is that for the sex feature, which is one-hot encoded, we must multiply each column by one half in order to treat each feature equally in the original unprocessed dataset.

3.2. Proximity, validity and plausibility

Table 4 summarises how the WAC and nearest-neighbour counterfactual explanations (NNCE) perform against three measures of performance: validity, proximity and plausibility. This figures are derived from counterfactual explanations from twenty randomly selected test instances of the titanic dataset. This process is repeated five times for each method and with the performance metrics representing averages over these five times. Higher values of validity imply better performance, whereas both for cost and plausibility, lower values are preferred.

method	validity	cost	plausibility
NNCE	1.0 ± 0.0	0.544 ± 0.036	0.147 ± 0.04
WAC	0.86 ± 0.086	0.654 ± 0.096	0.704 ± 0.061

Table 4. Performance differences between counterfactual methods, standard deviations given following $\pm$ sign.

3.3. Evaluating WAC and NNCE performance

Using these three metrics, it is clear that NNCE performs better. Beginning with validity, this ensures that our counterfactuals lead to a different prediction to the original input. In the case of NNCE, this is guaranteed because only valid counterfactuals are considered by construction; this explains the zero standard deviation. By contrast, WAC is an approximate approach that uses gradient descent. Consequently, even the basic property of validity is not guaranteed, hence the non-unity measure of validity. While most counterfactuals are valid, not all are in WAC.

Next we consider cost which measures how close the original input is to the counterfactual. The measure of proximity is given by our distance function, which is the L1 norm calculated over the input features after robust quantile scaling (with the sex columns further weighted by a half to account for one-hot encoding). Again, NNCE performs better here too, and with a lower standard deviation is more consistent. This means that the counterfactuals chosen by NNCE are closer on average than WAC. This shouldn't be a surprise because NNCE, as the name suggests, puts emphasis on considering the nearest neighbours closest to the original input.

Related to this is the largest gap in performance: plausibility. Plausibility examines how close the counterfactual is to other points in the dataset. The idea is that it might not be desirable to have a counterfactual which itself is an outlier, as this suggests it is a somewhat implausible alternative to the original input. Part of the reason why NNCE's plausibility is better could be because it relies on actual data points (the nearest neighbours of the input) and so its counterfactuals are inherently plausible and realistic. By contrast, WAC generates new data points by optimisation which might lead to counterfactuals which, while theoretically possible, are perhaps more uncommon in the empirical distribution. Furthermore, the optimisation of WAC can be influenced by factors such as the choice of optimisation function, learning rate and so on, which might not adequately capture all realistic constraints in the data-generating process.

References

[1] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr, 2018. 3