1 Classification Modelling

1.1 Task Description

The ACE team hypothesise that the referral data they collect can be used to predict treatment outcomes. Defining this as a machine learning problem, they suspect a relationship exists between the input variables - the referral data, X - and the outcome variable - discharge with / without the need for hospital treatment, y. This relationship can be formalised as:

$$y = f(X) + \epsilon \tag{1.1}$$

where y is an unknown function of the input variables and ϵ is a random error term (independent of with mean zero). Defining the problem in this way, the aim of this experiment is to approximate f, and subsequently make predictions of y of the form:

$$\hat{y} = \hat{f}(x) \tag{1.2}$$

where $\hat{y}$ and $\hat{f}$ are approximations of the true underlying y and f. This process can be thought of more generally as training a predictive model.

The accuracy of such a predictive model depends on two terms, the reducible and irreducible error. The reducible error is the degree to which $\hat{f}(X)$ accurately approximates $f(X)$ - the more accurate the representation, the lower the reducible error. The irreducible error is the ϵ term - this is independent of X and can be thought of as the unavoidable error - the factors that affect outcomes and that aren't captured in the data. In testing our hypothesis we hope to establish:

1. the degree to which the referral data is able to explain the outcome - the relative sizes of $f(X)$ and ϵ as proportions of y
2. how accurately we might approximate f

No "one-size-fits-all" predictive model exists. The modern machine learning toolkit includes vast number of approaches to classification modelling - each has it's own prior assumptions of the form that $\hat{f}$ takes, and thus each has it's own associated benefits and drawbacks. This experiment will test a range of these approaches, with the expectation that one amongst these techniques will establish a reasonable baseline for $\hat{f}$ that minimises the reducible error as much as possible. Unfortunately, the complexity of these modelling techniques and their variety renders their discussion in this report impractical, though the general intuition established above and the following discussion is sufficient to understand the results.

1.1.1 Encoding Non Numeric Data

Machine learning models require data to be represented numerically. This is an issue when considering categorical data, such as the "referral from" or "allergy" features in the ACE dataset. There are a number of approaches that represent categorical data numerically, some of which cannot be used in this setting given the small size of the dataset. The following approaches will be used in this experiment:

- **One-hot encoding**: Each categorical feature is split into its respective categories, each with a simple 1/0 or "on"/"off" value. For example, the following data:

Patient	**Referral Time**
1	Morning
2	Afternoon
3	Morning
4	Evening

would be one-hot encoded as:

Patient	**Referral Time Morning**	**Referral Time Morning**	**Referral Time Morning**
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1

- **Target encoding**: Each category is given a numerical value based on the proportion of target variable it represents, in this case the proportion of patients that require hospital treatment. For example, if 15% of patients referred in the evening require hospital treatment, the "evening" category is replaced by the figure 0.15. Care must be taken to avoid "leakage" when using this approach - that is, encodings should not be calculated using the label for the example in question, or from the labels of examples that will be used to evaluate model performance.

The free-text fields, such as "medical history" and "examination summary" present a greater challenge, and will therefore be excluded from this experiment. Further analysis and modelling of these text features can be found in Chapter **??**.

1.1.2 Balancing Target Labels

As discussed in Chapter **??**, examples of children that required hospital treatment are far fewer in number than those successfully treated by ACE. This presents a significant challenge when attempting to train a classification model to accurately predict the probability of a hospital referral. Models trained on imbalanced data can achieve relatively high prediction accuracy by predicting the majority label only - so, any model that predicts every patient will be treated successfully by ACE will be approximately 86% accurate. Optimising for prediction accuracy alone is likely to result in many such models.

To mitigate these issues, the following data preparation techniques will be tested, each of which attempts to address the imbalance of labels in the dataset:

- **Weighted labels**: Models are "punished" during training for making incorrect predictions. This penalty can be weighted depending on the label, so a model can be more heavily "punished" for making incorrect predictions of the minority label. The size of weighting is usually determined by the proportion of majority/minority labels

in the dataset - so a label that is five times less common than another is weighted five times more heavily. Note: Label weighting is only available in modelling techniques that use certain optimisation approaches, and thus is not available for some of the modelling techniques tested in thie experiment.

- **Synthetic Minority Oversampling Technique (SMOTE)**: SMOTE generates new synthetic examples of the minority label to balance the proportion of labels in the dataset. New samples are generated by selecting a random minority example, and a small number of "neighbours" for that example - other examples that are the most similar to the selected example. One of the neighbours is then randomly chosen, and a synthetic data point is sampled by interpolating between the random example and the selected neighbour. This process is repeated until the number of examples of each label match.

- **Undersampling**: Similar in spirit to oversampling, undersampling is the removal of examples from the majority label until the number of examples with each label match. There are many approaches to systematically select examples to remove - in this experiment random undersampling will be used.

1.1.3 Evaluating Models

An imbalance of labels also makes model evaluation more challenging. Simple accuracy is not an effective measure of performance if the proportion of labels is skewed heavily in one direction. Given this, it is important to use metrics that measure the proportion of the imbalanced labels that are correctly classified:

- **Precision**: This is the proportion of examples that are classified correctly, among those that are predicted to have a positive outcome - in terms of the ACE task, this is the proportion of patients that actually require hospital treatment, out of those predicted to need hospital treatment

- **Recall**: This is the proportion of positive examples that are classified correctly (ignoring every negative example) - in terms of the ACE task, this is the proportion of the children that need hospital treatment that are correctly identified.

- **F1 Score**: F1 is a combination of precision and recall. F1 calculates the harmonic mean between the precision and recall, offering a balance between these metrics:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{1.3}$$

  In isolation, one can achieve a perfect precision score by predicting one positive example correctly and all the others negative, or a perfect recall score by simply predicting every example as positive - F1 avoids this by evaluating the two metrics together. As F1 is a harmonic mean of two proportions, it also takes on values between 0 and 1 and can be easily interpreted like precision and recall.

- **AUC/ROC**: This is a this represents the degree of "seperability" of the model predictions based on the true labels - it measures the degree to which a model is capable of separating between the two labels correctly. The theory is too complex to discuss here, but an interpretation of the metric can be easily explained. Perfect seperability will achieve an AUC of 1. 0.5 indicates no separation or that a model is choosing randomly. Values below 0.5 indicate that the model is skewed toward

making incorrect decisions. Values closer to zero are rarely seen, given that one could simply flip the predictions to achieve an accurate model, but occasionally models will stray slightly below 0.5 - these results should be interpreted in much the same way as those at 0.5 or slightly above.

1.2 Experimental set-up

1.2.1 General set-up

Predictive models were trained using a combination of each of the following modelling techniques, and approaches to categorical encoding and label balancing:

Predictive Modelling Technique*	Logistic Regression Support Vector Machines K-Nearest Neighbours Random Forest Classifier Gradient Boosted Decision Trees Ada-Boost Classifier Gaussian Naive Bayes Classifier Quadratic Discriminant Analysis
Categorical Encoding Technique	One-Hot Encoding Mean Target Encoding
Label Balancing Technique	Balanced (Weighted) Labels SMOTE Random Undersampling

**Amongst these models, a wide variety of hyperparameters specific to each technique were tested. These are too numerous to detail here, but details can be seen in [LINK TO SCRIPT]*

A grid search method was used to test each combination of model, hyperparameters, categorical encoding approach, and label balancing approach. Models were scored using a 3-fold cross validation method, given that dividing the training data any further would result in too few positive examples in each validation fold. Variability of outcomes was a significan issue in early experimentation - to establish a reliable estimate of the variance of cross validation results, the training data was shuffled and the 3-fold cross validation scoring was repeated 10 times.

The models, hyperparameters and data preparation methods that performed best in cross validation were then tested against the holdout test set, to estimate how well the cross-validation scores represent the prediction scores for data that wasn't used during training, and how well the models generalise. Only the best performing models in cross validation were scored against the holdout test set, to minimise the risk of biasing the results to those that perform best against the test set.

1.2.2 Precautions

Particular care was taken to write a custom cross validation loop that accounted for the following complexities of this experiment:

- Synthetic samples were added to the training folds only - care was taken to ensure models were validated against genuine training examples only, without any added synthetic data
- Target encoding was calculated using a "leave-one-out" method from the training folds only to avoid "data leakage" - no holdout validation examples were used to calculate target encodings
- Target/one-hot encoding was completed after generating synthetic examples - otherwise the SMOTE algorithm would treat the encoded categories as numeric, and interpolates between them creating erroneous "sub-categories"
- The random fold samples were kept identical when training each individual model and configuration to ensure an unbiased comparison of each model

1.3 Results

Results cross validation results from each of the classification models can be seen in [**APPENDIX**] and the test set results of the best performing models can be seen in [figure]. None of the classification models are able to make useful predictions of hospital outcomes from the ACE data during cross validation or against the holdout test set. Those models that achieve good overall accuracy (>70%) do so at the expense of identifying patients that required hospital treatment - recall (<30%) and precision (<25%). Conversely, models that are able to identify greater numbers of patients that require hospital treatment, do so at the expense of overall accuracy. None of the models achieve an F1 score above 0.3 or an AUC above 0.55, indicating the low degree to which the models are able to separate patients that were successfully treated by ACE from those that required hospital referral.

		F1	AUC	Acc	Rec	Prec	True +ve	True -ve	False +ve	False -ve
SMOTE	**Logistic Regression**	0.158	0.452	0.605	0.222	0.122	6	92	43	21
	Random Forest	0.067	0.422	0.654	0.074	0.061	2	104	31	25
Balanced	**Logistic Regression**	0.244	0.519	0.617	0.37	0.182	10	90	45	17
	Random Forest	0.208	0.533	0.765	0.185	0.238	5	119	16	22

The observed standard deviations between the cross validation folds also indicate that model's predictions vary significantly depending on the data they see during training. This indicates that the decisions, or heuristics, of the classification models are not robust to small changes in the training dataset. These results support the issues discussed in Section **??** - there are very few examples that exhibit the features that are most indicative of hospitalisation risk, and model results vary dramatically depending on the inclusion/exclusion of these examples during training.

Visualising the model predictions further emphasises the poor performance of the classification models. Figures [figures] show the test set predictions of the models that performed best in cross-validation. The distributions of predictions barely differ between patients that were successfully discharged from ACE and patients that were referred to

hospital. The logistic regression model also lacks confidence when making predictions - the vast majority of the predictions made fall within the mid range of probabilities, indicating that the model will rarely deviate from an approximate 50-50 chance of hospitalisation.

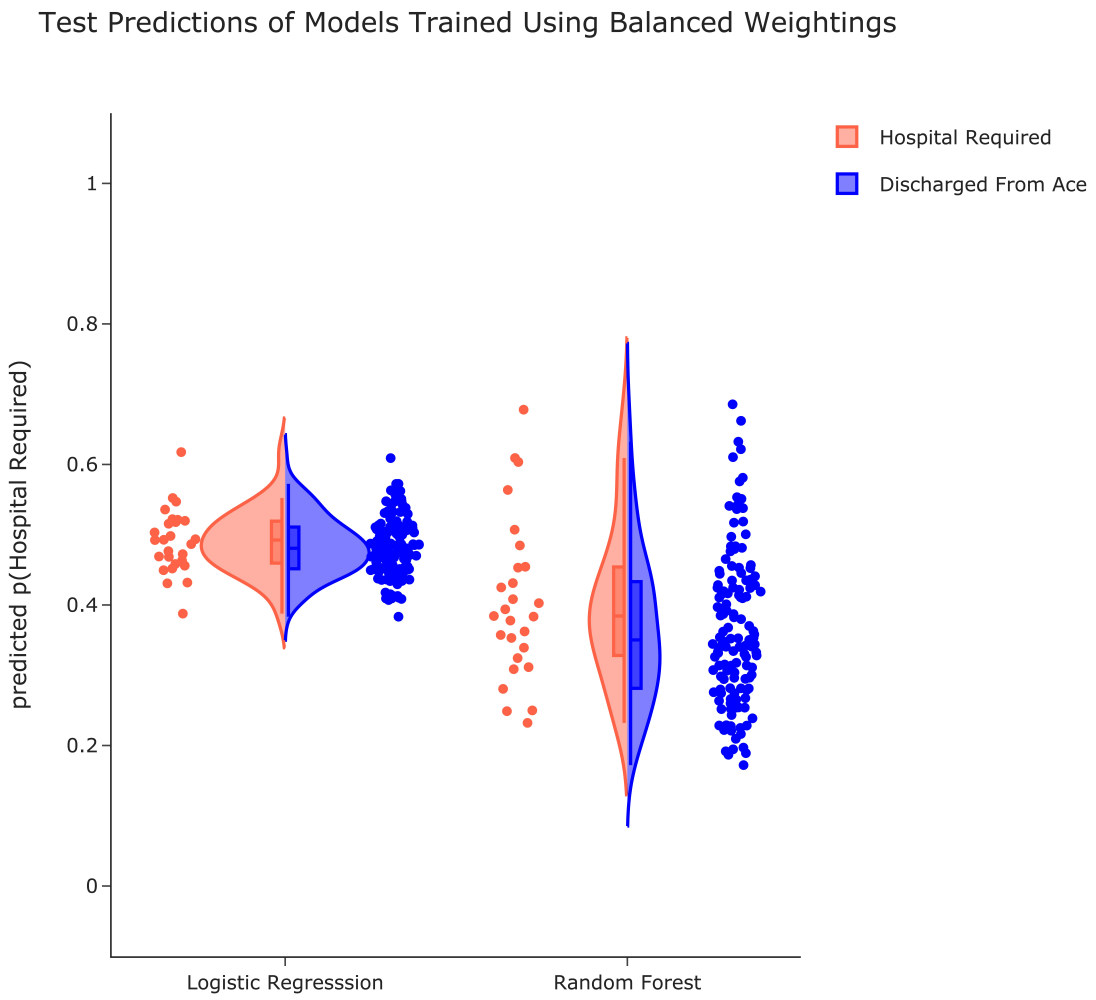

Figure 1.1: *** balanced model predictions ***

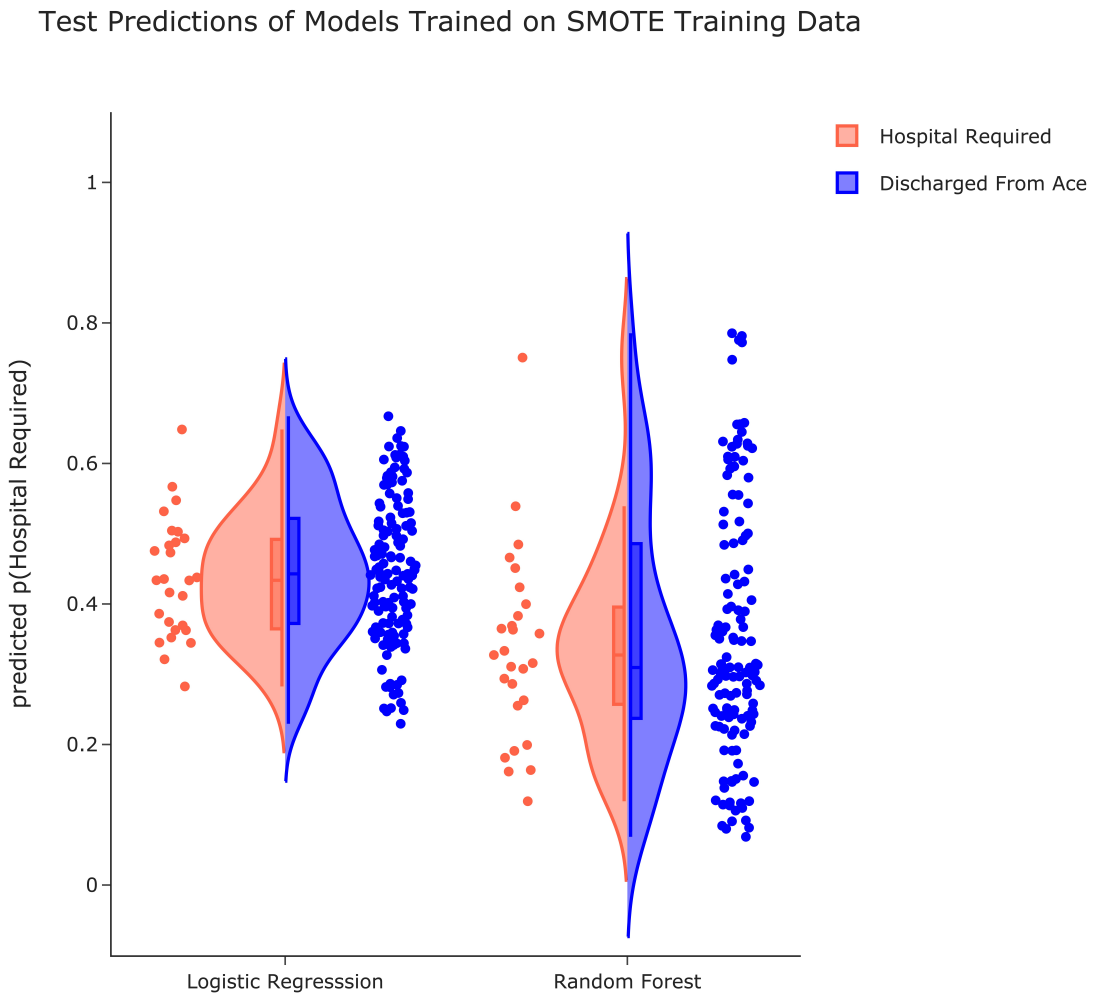

Figure 1.2: *** SMOTE model predictions ***

1.3.1 Reproducing results

Code to reproduce these results can be found in models/sklearn_models.py and models/plot_predictions.ipyn in the project repo

1.4 Conclusions

A broad range of data preparation methods and modelling techniques were used in this experiment. It is reasonable, therefore, to assume that we have established a good baseline for $\hat{f}$, our approximation of the true relationship between the referral data and hospitalisation outcomes. Given that, from this baseline we were unable to generate anything approximating accurate predictions of hospitalisation, we can reasonably reject the initial hypothesis of the ACE team - it is not possible to predict treatment outcomes using the referral data they have provided. The irreducible error presented by this problem appears to be such that little can be determined about the outcomes from the input data. (It should be noted

that the experiments thus far have excluded the free-text features, which are analysed in Chapter **??**).

These results are unsurprising when considered in the context of the findings of the initial data analysis. The lack of obvious predictors of hospitalisation found in the data analysis is reflected by the poor prediction performance of the models trained using the dataset. It bears repeating that the absence of predictors in the dataset serves to affirm the referral decisions made by the ACE team, given that the dataset comprises only patients that were accepted for treatment within ace. If instead we had trained extremely accurate classification models, this would indicate the presence of obvious indicators of hospitalisation in the referral data that ACE clinicians were oblivious to.

A Appendices

A.1 The Data

Feature	Values	P(Hospital Required)	Total Examples
Referral From	**CCDA**	0.067	45
	ED	0.172	87
	GP	0.177	203
Referral Profession	**ANP**	0.159	82
	Consultant	0.086	35
	Doctor	0.172	157
	Registar	0.177	62
Gender	**Female**	0.155	129
	Male	0.164	207
Referral Date	**Autumn**	0.125	112
	Spring	0.154	52
	Summer	0.169	59
	Winter	0.195	113
Referral Time	**Afternoon**	0.161	137
	Evening	0.4	15
	Morning	0.141	184
Illness Severity	**Mild**	0.155	290
	Moderate	0.205	44
Activity Level	**Lower**	0.188	112
	Usual	0.145	221
Gut Feeling	**Low Concern**	0.149	188
	Unwell	0.667	3
	Well	0.162	142
Sepsis	**Low Level**	0.158	19
	None Noted	0.161	317
Safeguarding	**No**	0.167	288
	Yes	0.125	48
Food Allergy	**No**	0.159	290
	Yes	0.174	46
Drug Allergy	**No**	0.163	301
	Yes	0.143	35
Other Allergy	**No**	0.167	305
	Yes	0.097	31

Feature	Values	P(Hospital Required)	Total Examples
Group Ethnicity	**Asian**	0.174	184
	European	0.158	120
	Other	0.094	32

Table A.1: Hospitalisation frequency and number of examples for each of the original categorical features in the ACE dataset

	Chi^2	p	dof
Referral Time	6.881	0.032	2
Gut Feeling	5.929	0.052	2
Referral From	3.446	0.179	2
Activity Level	0.719	0.397	1
Other Allergy	0.579	0.447	1
Group Ethnicity	1.307	0.52	2
Illness Severity	0.371	0.542	1
Referral Date	2.078	0.556	3
Safeguarding	0.266	0.606	1
Referral Profession	1.738	0.628	3
Sepsis	0.082	0.774	1
Gender	0.005	0.943	1
Drug Allergy	0.004	0.952	1
Food Allergy	0.002	0.963	1

Table A.2: Chi^2 statistics for the original categorical features from the ACE referral data

*** add distribution figs for continuous features - decide if it's worth creating plotly plot instead of the existing seaborn ones ***

	r	p
Ox Sat	-0.164	0.003
Age	0.081	0.141
Temp	0.048	0.409
Resp Rate	0.028	0.617
Heart Rate	0.017	0.752

Table A.3: Pearson's R statistics for the numeric/continuous features in the ACE referral data

*** ???? include interaction effects results ???? ****

Feature	Values	P(Hospital Required)	Total Examples
APLS Resp Rate	**High**	0.207	82
	Low	0.0	4
	Normal	0.148	250
APLS Heart Rate	**High**	0.242	33
	Low	0.5	2
	Normal	0.15	301
Ox Sat Low	**No**	0.162	333
	Yes	0.0	3
Age Range	**Pre School**	0.161	180
	Primary	0.141	142
	Secondary	0.357	14
ACE Heart Rate	**high**	0.154	65
	Low	0.286	7
	Normal	0.159	264
ACE Resp Rate	**high**	0.17	112
	Low	0.175	40
	Normal	0.152	184
Meets ACE Criteria	**No**	0.17	182
	Yes	0.149	154

Table A.4: Hospitalisation Frequency and number of examples for each of the features engineered using the ACE referral criteria and APLS observation guidelines

	chi2	p	dof
Age Range	4.421	0.11	2
APLS Heart Rate	3.621	0.164	2
APLS Resp Rate	2.386	0.303	2
ACE Heart Rate	0.839	0.657	2
Meets ACE Criteria	0.139	0.709	1
ACE Resp Rate	0.226	0.893	2
Ox Sat Low	0.001	0.978	1

Table A.5: Chi^2 statistics for the engineered features using the ACE referral criteria and APLS guidelines

A.1.1 Interaction Effects

A.2 Classification Modelling

	F1	AUC	Acc	Rec	Prec
One Hot - Weighted Labels					
K Nearest Neighbours	0.154 (0.071)	0.502 (0.038)	0.742 (0.027)	0.144 (0.073)	0.169 (0.074)
Support Vector Machines	0.269 (0.056)	0.535 (0.061)	0.576 (0.082)	0.474 (0.140)	0.191 (0.041)
Random Forest Classifier	0.209 (0.066)	0.530 (0.035)	0.745 (0.033)	0.209 (0.079)	0.218 (0.065)
Gradient Boosting Classifier	0.153 (0.092)	0.519 (0.039)	0.788 (0.029)	0.119 (0.073)	0.233 (0.162)
Ada Boost classifier	0.173 (0.081)	0.518 (0.041)	0.762 (0.034)	0.154 (0.078)	0.212 (0.109)
Gaussian Naieve Bayes	0.238 (0.055)	0.521 (0.047)	0.645 (0.066)	0.335 (0.096)	0.189 (0.048)
Logistic Regression	0.265 (0.049)	0.534 (0.050)	0.596 (0.060)	0.441 (0.097)	0.191 (0.037)
Quadratic Discriminant Analysis	0.036 (0.048)	0.497 (0.016)	0.815 (0.016)	0.022 (0.031)	0.106 (0.151)
One Hot - SMOTE					
K Nearest Neighbours	0.248 (0.052)	0.512 (0.057)	0.557 (0.047)	0.444 (0.111)	0.173 (0.035)
Support Vector Machines	0.292 (0.048)	0.552 (0.054)	0.555 (0.071)	0.546 (0.076)	0.200 (0.040)
Random Forest Classifier	0.267 (0.050)	0.555 (0.035)	0.717 (0.039)	0.313 (0.072)	0.238 (0.051)
Gradient Boosting Classifier	0.255 (0.063)	0.547 (0.043)	0.717 (0.041)	0.293 (0.076)	0.231 (0.064)
Ada Boost classifier	0.278 (0.059)	0.537 (0.061)	0.539 (0.093)	0.535 (0.130)	0.192 (0.049)
Gaussian Naieve Bayes	0.196 (0.060)	0.465 (0.051)	0.540 (0.059)	0.352 (0.141)	0.138 (0.038)
Logistic Regression	0.285 (0.066)	0.556 (0.062)	0.634 (0.055)	0.441 (0.108)	0.213 (0.051)
Quadratic Discriminant Analysis	0.256 (0.058)	0.519 (0.063)	0.561 (0.061)	0.456 (0.112)	0.179 (0.042)
One Hot - Undersampling					

	F1	AUC	Acc	Rec	Prec
K Nearest Neighbours	0.246 (0.063)	0.505 (0.071)	0.532 (0.059)	0.463 (0.126)	0.169 (0.043)
Support Vector Machines	0.276 (0.046)	0.535 (0.056)	0.528 (0.084)	0.546 (0.138)	0.187 (0.032)
Random Forest Classifier	0.275 (0.047)	0.536 (0.054)	0.544 (0.055)	0.524 (0.106)	0.187 (0.032)
Gradient Boosting Classifier	0.262 (0.045)	0.520 (0.050)	0.530 (0.066)	0.506 (0.111)	0.178 (0.033)
Ada Boost classifier	0.267 (0.050)	0.528 (0.055)	0.540 (0.054)	0.511 (0.126)	0.182 (0.034)
Gaussian Naieve Bayes	0.269 (0.048)	0.534 (0.053)	0.561 (0.069)	0.493 (0.121)	0.187 (0.031)
Logistic Regression	0.267 (0.051)	0.525 (0.062)	0.531 (0.062)	0.517 (0.106)	0.181 (0.035)
Quadratic Discriminant Analysis	0.269 (0.067)	0.529 (0.070)	0.557 (0.083)	0.489 (0.125)	0.189 (0.054)
Mean Target - Weighted Labels					
K Nearest Neighbours	0.108 (0.069)	0.475 (0.036)	0.723 (0.033)	0.104 (0.068)	0.115 (0.076)
Support Vector Machines	0.279 (0.039)	0.545 (0.039)	0.570 (0.042)	0.509 (0.107)	0.193 (0.024)
Random Forest Classifier	0.193 (0.075)	0.512 (0.045)	0.715 (0.041)	0.209 (0.088)	0.186 (0.081)
Gradient Boosting Classifier	0.135 (0.097)	0.510 (0.044)	0.780 (0.028)	0.106 (0.079)	0.195 (0.130)
Ada Boost classifier	0.161 (0.076)	0.512 (0.038)	0.761 (0.031)	0.141 (0.071)	0.197 (0.099)
Gaussian Naieve Bayes	0.205 (0.070)	0.515 (0.036)	0.684 (0.093)	0.263 (0.140)	0.181 (0.049)
Logistic Regression	0.285 (0.042)	0.549 (0.046)	0.569 (0.049)	0.519 (0.090)	0.197 (0.030)
Quadratic Discriminant Analysis	0.148 (0.086)	0.440 (0.058)	0.527 (0.172)	0.311 (0.226)	0.113 (0.066)
Mean Target - SMOTE					

	F1	AUC	Acc	Rec	Prec
K Nearest Neighbours	0.209 (0.062)	0.474 (0.064)	0.547 (0.051)	0.365 (0.124)	0.147 (0.043)
Support Vector Machines	0.279 (0.034)	0.539 (0.040)	0.529 (0.054)	0.554 (0.102)	0.188 (0.023)
Random Forest Classifier	0.235 (0.072)	0.534 (0.047)	0.708 (0.039)	0.274 (0.092)	0.209 (0.063)
Gradient Boosting Classifier	0.242 (0.072)	0.540 (0.047)	0.715 (0.042)	0.280 (0.100)	0.219 (0.063)
Ada Boost classifier	0.274 (0.059)	0.531 (0.065)	0.538 (0.093)	0.520 (0.125)	0.190 (0.050)
Gaussian Naieve Bayes	0.271 (0.041)	0.532 (0.046)	0.546 (0.050)	0.511 (0.089)	0.185 (0.029)
Logistic Regression	0.271 (0.046)	0.539 (0.044)	0.591 (0.049)	0.463 (0.094)	0.193 (0.032)
Quadratic Discriminant Analysis	0.275 (0.036)	0.536 (0.038)	0.535 (0.067)	0.537 (0.105)	0.187 (0.024)
Mean Target - Undersampling					
K Nearest Neighbours	0.240 (0.059)	0.504 (0.064)	0.561 (0.061)	0.420 (0.115)	0.169 (0.041)
Support Vector Machines	0.266 (0.054)	0.524 (0.061)	0.528 (0.065)	0.519 (0.120)	0.180 (0.038)
Random Forest Classifier	0.265 (0.045)	0.525 (0.049)	0.524 (0.068)	0.526 (0.132)	0.179 (0.029)
Gradient Boosting Classifier	0.258 (0.045)	0.515 (0.053)	0.525 (0.056)	0.500 (0.102)	0.175 (0.032)
Ada Boost classifier	0.264 (0.042)	0.522 (0.049)	0.531 (0.056)	0.509 (0.105)	0.179 (0.031)
Gaussian Naieve Bayes	0.254 (0.056)	0.520 (0.052)	0.562 (0.068)	0.457 (0.126)	0.178 (0.040)
Logistic Regression	0.258 (0.045)	0.512 (0.052)	0.504 (0.055)	0.524 (0.109)	0.172 (0.029)
Quadratic Discriminant Analysis	0.252 (0.073)	0.519 (0.073)	0.568 (0.069)	0.446 (0.153)	0.177 (0.050)

Table A.6: Cross-Validation results for each combination of data preparation/label weighting/modelling technique tested during the classification modelling. Figures in brackets are the standard deviations for the relevant statistic.

1.3 Free Text Analysis

1.4 Bayesian Analysis

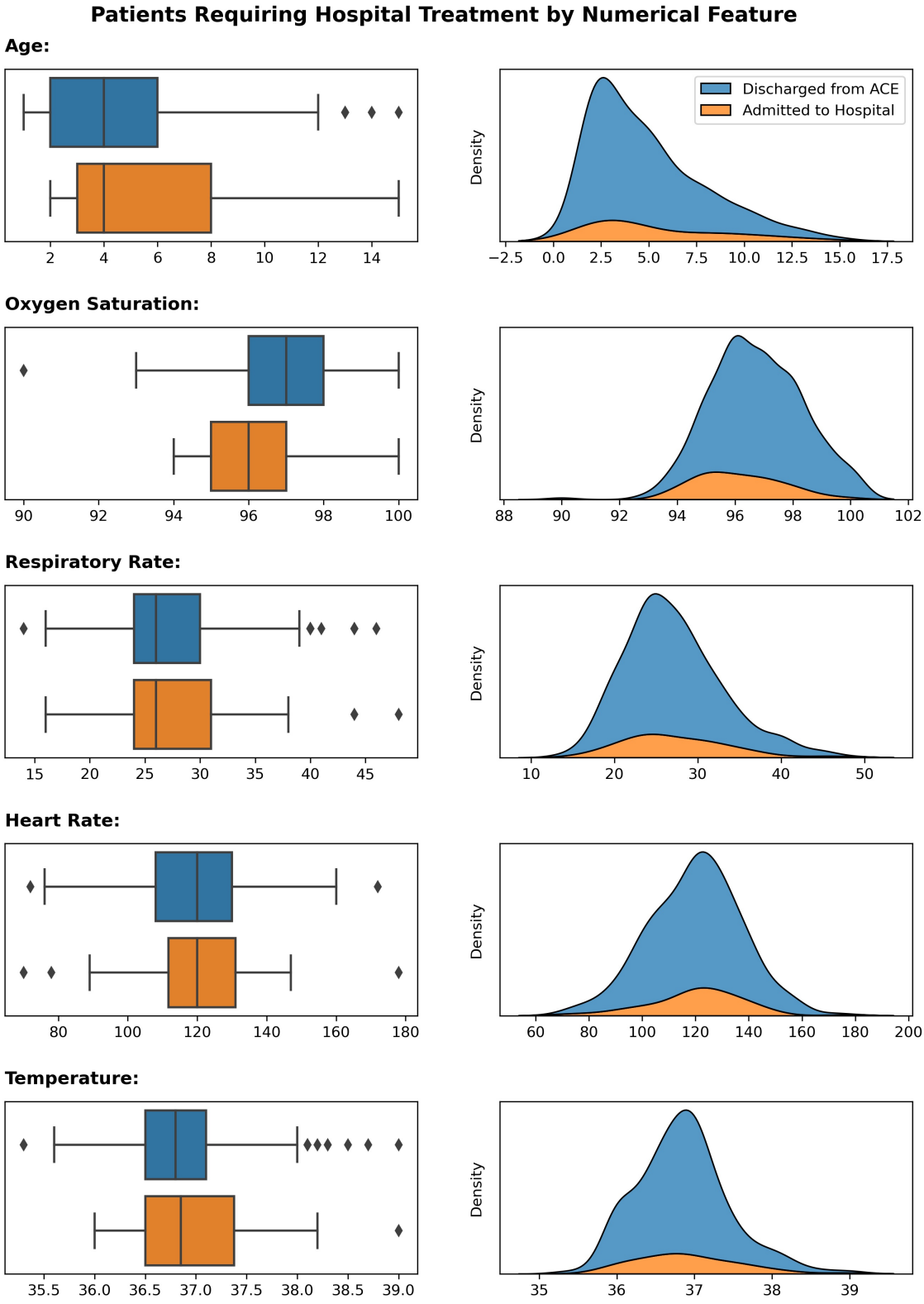

Figure A.1: Box plots and stacked KDE plots for each of the numerical features, grouped by examples that required hospital treatment and those that were successfully discharged from ACE

Bibliography