Error Analysis on Mortality Prediction using Random Forest

Team 4: Julio Real Rojas, Aqib Nisar, Shwetha Vedavinayagam, Pham Thien Phuc Nguyen

Introduction

In the domain of healthcare analytics, predictive modelling plays a crucial role in forecasting patient outcomes and aiding clinical decision-making. The focus of this report is on conducting error analysis for a random forest predictive model designed to predict mortality using the MIMIC-III dataset.

The MIMIC-III (Medical Information Mart for Intensive Care III) dataset is a comprehensive collection of de-identified electronic health records (EHRs) from critical care units, offering a rich source of clinical data for research and analysis. Leveraging machine learning techniques, specifically ***a random forest model***, this project aims to assess the model's performance in predicting patient mortality.

Error analysis is a critical step in evaluating the effectiveness and reliability of predictive models. By scrutinizing the model's predictions and identifying cases of misclassification (false positives and false negatives), we can gain insights into the model's strengths, limitations, and areas for improvement.

Through this report, we seek to:

- Conduct a detailed error analysis to identify patterns and factors contributing to incorrect predictions of mortality by the random forest model.
- Investigate common characteristics among incorrectly predicted outputs, including both false positives (*patients predicted to die but survived*) and false negatives (*patients predicted to survive but died*).
- Utilize statistical tests and visualizations to uncover error patterns, assess model biases, and evaluate the predictive performance of the random forest model.
- Generate actionable insights and recommendations to enhance the accuracy and reliability of mortality predictions, thereby contributing to improved patient care and clinical decision support systems.

Methodology

Preprocessing steps (Phase 1)

In the phase 1 of the project, we focused on analysing the MIMIC-3 dataset to identify survivors and non-survivors among mechanically ventilated patients within a 24-hour timeframe. We extracted relevant data, conducted statistical tests, and created visualizations to understand the patterns and factors associated with patient outcomes.

At the end of the phase 1, we had our cleaned dataset with the size 16,078. We will be using this dataset to predict mortality using the Random Forest Algorithm in this report.

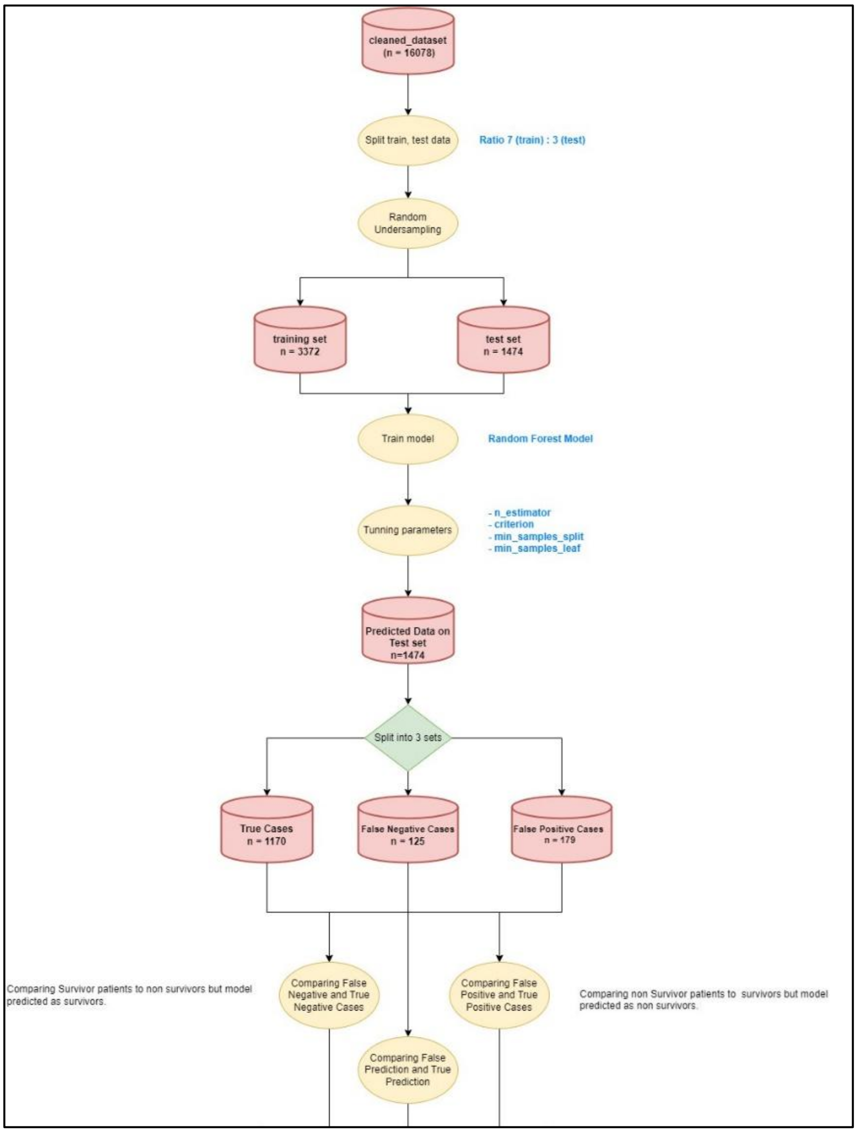
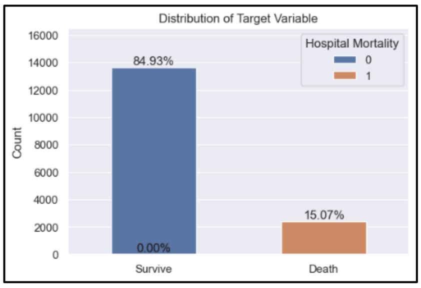

cleaned_dataset
(n = 16078)
Split train, test data
Ratio 7 (train) : 3 (test)
Random Undersampling
training set
n = 3372
test set
n = 1474
Train model
Random Forest Model
Tunning parameters
- n_estimator
- criterion
- min_samples_split
- min_samples_leaf
Predicted Data on Test set
n=1474
Split into 3 sets
True Cases
n = 1170
False Negative Cases
n = 125
False Positive Cases
n = 179
Comparing Survivor patients to non survivors but model predicted as survivors.
Comparing False Negative and True Negative Cases
Comparing False Positive and True Positive Cases
Comparing non Survivor patients to survivors but model predicted as non survivors.
Comparing False Prediction and True Prediction

Figure 1| Data Workflow

Data preparation

To prepare data for Random Forest model, we split the dataset from phase 1 into training dataset and test dataset with the ratio of 7:3. During the training, we let the model see the target variables (Hospital Mortality value), so it can learn how to predict. We also use random state to keep the consistency in our results.

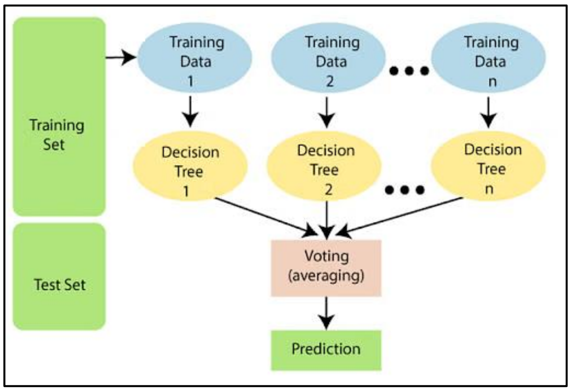
Additional, in the dataset from phase 1, we found that there is an imbalance between the number of patients who survive and non-survive. The imbalance can lead to bias and wrong prediction in our model. To address this imbalance, we use Under Sampling for both the training dataset and the test dataset. This will help us have the same number of patients who survive and death in our dataset.



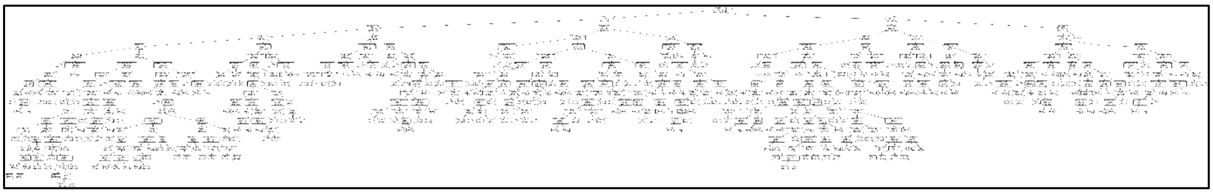
At the end, the size of the training data is 3,372 records and the test dataset has 1,474 records.

Random Forest model

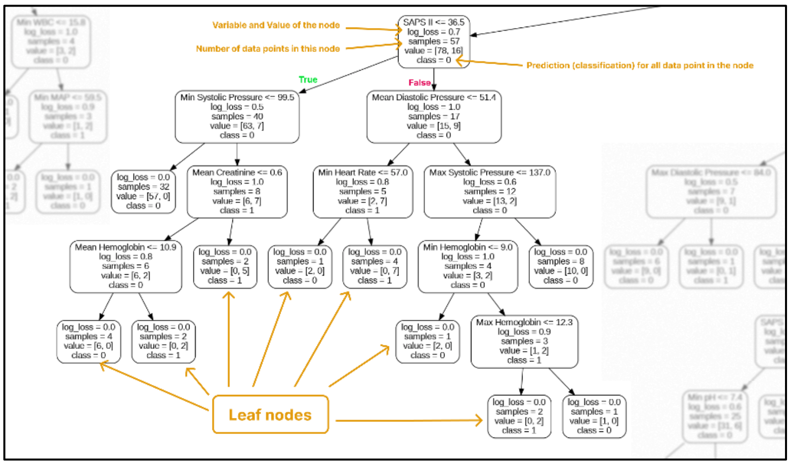
A Random Forest Algorithm is a supervised machine learning algorithm and can diagnose patients in health care application (*Random Forest Algorithm*, n.d.). In the Random Forest Algorithm, the model creates multiple decision trees from subsets of training data, so that we avoid the overfitting when the sample size is small.



With the goal of predict whether the patient survive or not based on their Demographic, Medical History, Disease Severity, Diagnosis, Vital Signs, Laboratory Results and the sample size of dataset is 4,846 records (after balancing survive and non-survive). In our case, we have both the features (patient medical information) and the targets (mortality), and it matches with supervised prediction. From that, the Randon Forest Model is suitable for our purpose.



For more understanding how the decision tree looks like, we extract one decision tree from our model (you can download the sample tree at this link for full resolution). As looking at the detail of a branch from the tree, we can see each node containing variable and condition, number of data points in this node, and the predict value. The left branch for true condition, and the right branch for the false condition.


Variable and Value of the node
Number of data points in this node
Prediction (classification) for all data point in the node
True
False
Leaf nodes
SAPS II <= 36.5 log_loss = 0.7 samples = 57 value = [78, 16] class = 0
Min Systolic Pressure <= 99.5 log_loss = 0.5 samples = 40 value = [63, 7] class = 0
Mean Diastolic Pressure <= 51.4 log_loss = 1.0 samples = 17 value = [15, 9] class = 0
log_loss = 0.0 samples = 32 value = [57, 0] class = 0
Mean Creatinine <= 0.6 log_loss = 1.0 samples = 8 value = [6, 7] class = 1
Min Heart Rate <= 57.0 log_loss = 0.8 samples = 5 value = [2, 7] class = 1
Max Systolic Pressure <= 137.0 log_loss = 0.6 samples = 12 value = [13, 2] class = 0
Mean Hemoglobin <= 10.9 log_loss = 0.8 samples = 6 value = [6, 2] class = 0
log_loss = 0.0 samples = 2 value = [0, 5] class = 1
log_loss = 0.0 samples = 1 value = [2, 0] class = 0
log_loss = 0.0 samples = 4 value = [0, 7] class = 1
Min Hemoglobin <= 9.0 log_loss = 1.0 samples = 4 value = [3, 2] class = 0
log_loss = 0.0 samples = 8 value = [10, 0] class = 0
log_loss = 0.0 samples = 4 value = [6, 0] class = 0
log_loss = 0.0 samples = 2 value = [0, 2] class = 1
log_loss = 0.0 samples = 1 value = [2, 0] class = 0
Max Hemoglobin <= 12.3 log_loss = 0.9 samples = 3 value = [1, 2] class = 1
log_loss = 0.0 samples = 2 value = [0, 2] class = 1
log_loss = 0.0 samples = 1 value = [1, 0] class = 0

Train model

In this project, we used the RandomForestClassifier from Scikit-learn library (*Sklearn.Ensemble.RandomForestClassifier*, n.d.). There are multiple hyper parameters that we can tun to get a better performance of the mode, and the list of hyper parameters that we considered is: criterion, n_estimators, min_samples_split, min_samples_leaf (Tillo, 2021)

To check if the number of features affect to the performance of the model, we also ran the model with 27 selected features[1]

For improve the accuracy of the model, we used GridSearch to find the optimal hyperparameters of a model which results in the most 'accurate' predictions (Joseph, 2022). If we had to select the values for two or more parameters, we would **evaluate all combinations of the sets of values** thus forming a grid of values (*Python Machine Learning - Grid Search*, n.d.).

After training the model with un-tuning and tunning parameters on all features and 27 features, we got the value of parameter as below:

Parameter	Untuned model (default)	Tuned model with all features	Tuned model with 29 features
n_estimator	100	100	250
criterion	gini	log_loss	entropy
min_samples_split	2	3	4
min_samples_leaf	1	1	2

Model Performance

Since we are working on the classification, we can use the Confusion Matrix to evaluate the performance of our models. We explain more about the Confusion Matrix in the next section.

We then use Accuracy, Precision, Recall, F1-score and ROC (Receiver Operating Characteristics), AUC (Area Under the Curve) metric which are the most popular metrics to compare the performance of between our models (Kumar, 2022)

Accuracy, Precision, Recall, F1-score

To get the value of Accuracy, Precision, Recall, F1-score, we use the ClassificationReport from the ScikitLearn. They Accuracy tells us how many predictions from the model are correct. The Precision tells us with all the positive prediction, how many are actually positive. The Recall tell us within the positive cases, how many predictions are correct. The F1-score help us to measure the Precision and Recall at the same time, it's useful in case the difference between Precision and Recall is large (Narkhede, 2021b).

[1] 27 selected features: 'SAPS II', 'SOFA', 'OASIS', 'Age', 'Min Heart Rate', 'Mean Heart Rate', 'Min Diastolic Pressure', 'Max Lactate', 'Min Lactate', 'Min pH', 'Mean pH', 'Min Glucose', 'Min WBC', 'Min BUN', 'Max Creatinine', 'Max Hemoglobin', 'Min Hemoglobin', 'Uncomplicated Hypertension', 'Uncomplicated Diabetes', 'Complicated Diabetes', 'Metastasis', 'Stroke', 'Sepsis', 'Any Organ Failure', 'Severe Cardiovascular Failure', 'Severe Renal Failure', 'Respiratory Dysfunction', 'Cardiovascular Dysfunction', 'Hematologic Dysfunction'

Since we use the under-sampling method, therefore our data is already balanced. And the different between the Precision and Recall is not much. To simplify the result, we use the macro average result. For detail of each classification of the report, please visit Classification Report

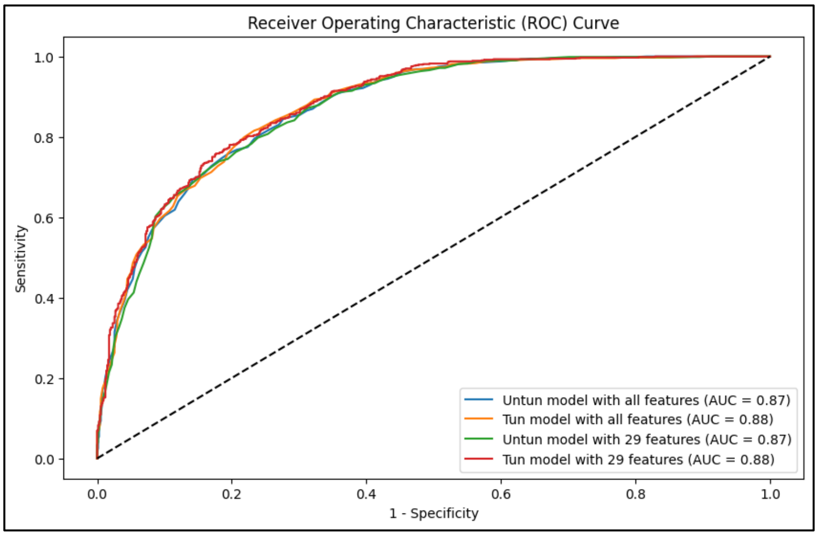
Metrics	Un-tuned model with all features	Tuned model with all features	Un-tuned model with 27 features	Tuned model with 27 features
Accuracy	0.7809	0.7938	0.7781	0.7894
Precision	0.78	0.79	0.78	0.79
Recall	0.78	0.79	0.78	0.79
F1-Score	0.78	0.79	0.78	0.79

In overall, the result from Random Forest is well, with the average accuracy of 0.78. By tunning the hyper parameter of the model we can get a better performance, in our case, the performance increased from 0.7809 to 0.7938 with all features and from 0.7781 to 0.7894 with 27 features.

We also notice that the performance is almost the same in model with all features and model with 27 features. This means we can use the model with 27 features to decrease the resource usage and get faster prediction.

ROC, AUC Curve

To visualize the performance, we use the ROC curve and AUC. The ROC and AUC tells us how much the model can distinguish between classes. The higher the AUC, the better the model is at predicting class 0 as 0 and class 1 as 1 (Narkhede, 2021a). In our case, the higher the AUC, the better model is at distinguishing between patients who die and survive.



With the ROC graph, all 4 models have almost the same AUC. The result from model with tuned parameters is 0.88, slightly better than one without tuned parameters is 0.87. This means that the prediction of our models can distinguish between patients who die and survive significantly.

In conclusion about the performance of our models, we can say that with our dataset, the Random Forest Model can perform well. And the best model, is the Tuned Model with all the features.

Error Analysis

The performance of our models is not really different, therefore we choose the prediction of the Tuned model with all features to analyse the error.

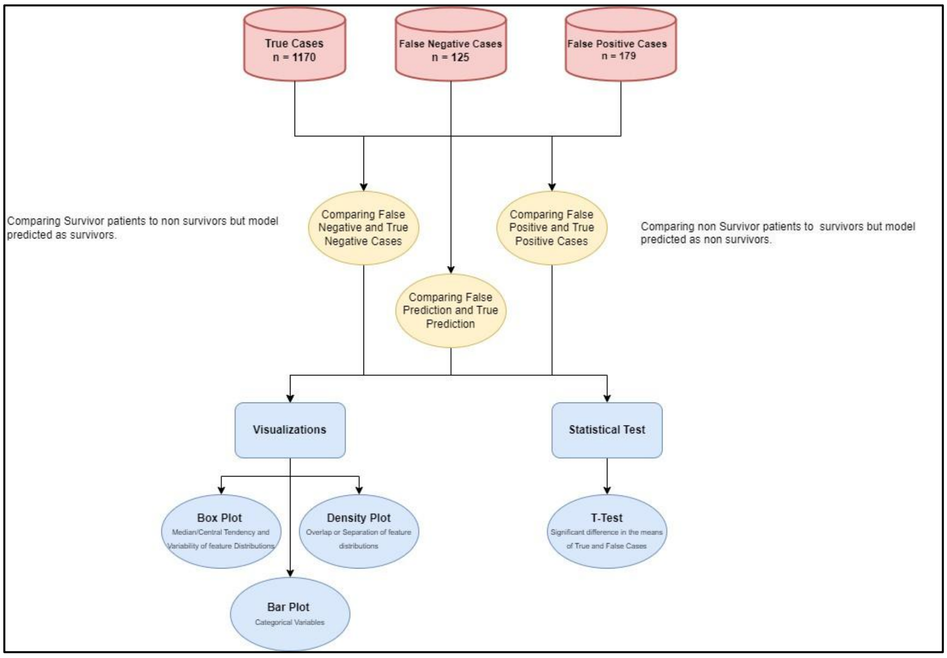

True Cases
n = 1170
False Negative Cases
n = 125
False Positive Cases
n = 179
Comparing Survivor patients to non survivors but model predicted as survivors.
Comparing False Negative and True Negative Cases
Comparing False Positive and True Positive Cases
Comparing non Survivor patients to survivors but model predicted as non survivors.
Comparing False Prediction and True Prediction
Visualizations
Statistical Test
Box Plot
Median/Central Tendency and Variability of feature Distributions
Density Plot
Overlap or Separation of feature distributions
T-Test
Significant difference in the means of True and False Cases
Bar Plot
Categorical Variables

Figure 2| Error Analysis Workflow

Confusion matrix

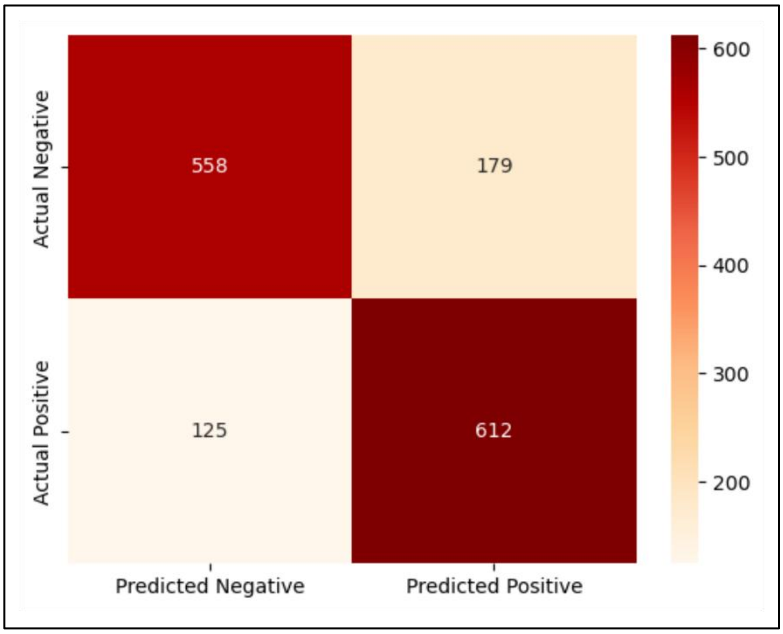


Figure 3| Confusion Matrix of best model

Defining false positives and false negatives:

False Positives (FP)

False positives occur when the model predicts that a patient will not survive (non-survived), but the patient survives (survived). In other words, the model incorrectly predicts a positive outcome (non-survival) when the actual outcome is negative (survival).

False Negatives (FN)

False negatives occur when the model predicts that a patient will survive (survived), but the patient does not survive (non-survived). This means the model incorrectly predicts a negative outcome (survival) when the actual outcome is positive (non-survival).

General Characteristics of misclassification

From the Confusion Matrix, we can see that ***False Positive cases (179) are higher than the False Negative cases (125)***. When there are more false positives than false negatives, it indicates that the model tends to ***overpredict mortality*** (non-survived cases) more frequently than underpredicting mortality.

This suggests a ***bias towards false positives***, where the model is more likely to predict non-survival when the actual outcome is survival.

Feature Importance

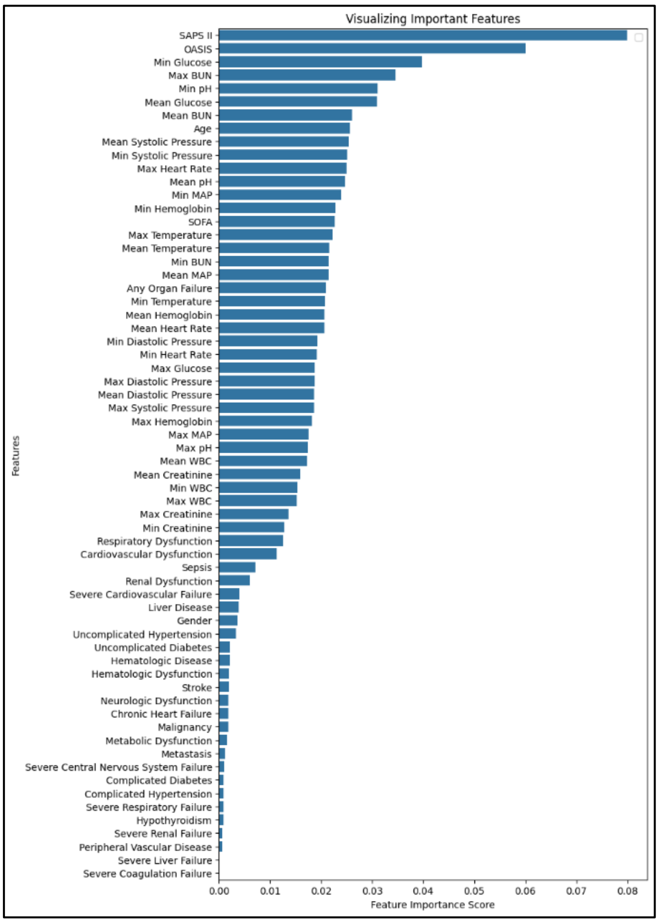


Figure 4 | Important Features

One of the main factors contributing to model errors are the ***important features*** of the training model.

Our model takes ***SAPS II as the most important feature*** for predicting mortality.

The high importance of SAPS II suggests that it is a strong predictor of mortality outcomes in the context of the model. Patients with higher SAPS II scores are likely to have a higher risk of non-survival, making it a crucial feature for mortality prediction.

It also suggests that while doing error analysis, if the highly important feature is misinterpreted or missing in certain instances, it can lead to errors in predictions.

Also, since our project does not do feature engineering therefore it might be contributing to the model errors.

Factors contributing to misclassification / Error Interpretation

For error interpretation, we are going to use visualization and statistical methods to find the patterns.

We are going to analyse ***False Negative vs True Negative*** to get the characteristics of False Negative cases. Similarly, comparing ***False Positive vs True Positive*** to get the characteristics of False positive.

The analysis of True predictions and False predictions to get the overall error patterns of the model.

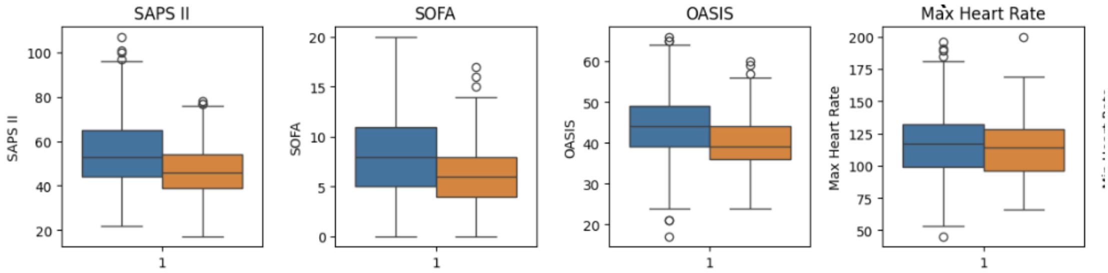
We will be analysing the most important features from the Feature Importance part, since it contributes most to the model.

False Positive analysis

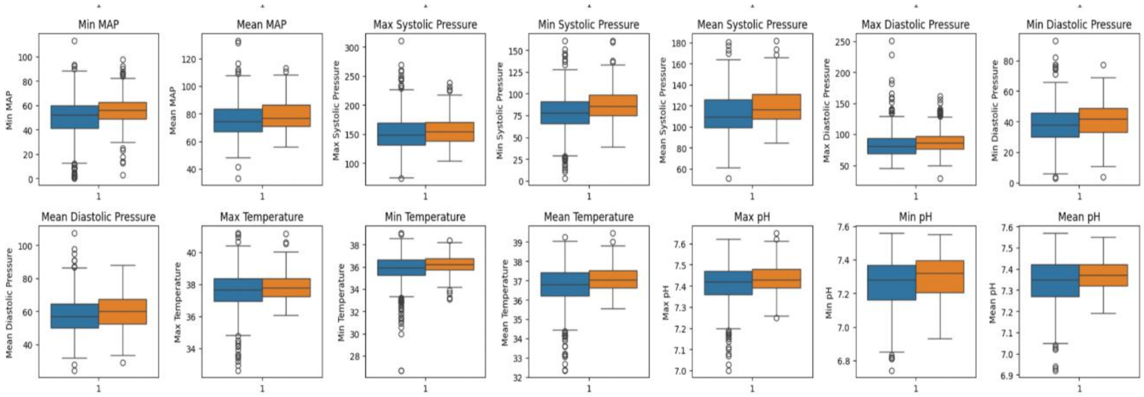
Box Plot

The following interpretation is done on the box plot visualization of FP and TP cases.

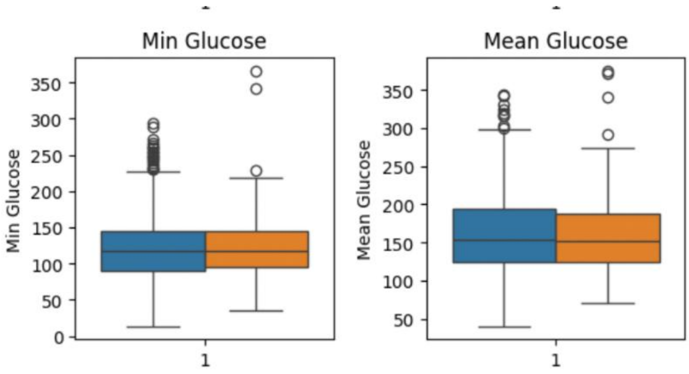
- A ***higher median of a feature in TP cases*** may suggest that patients with elevated values of the feature are at ***higher risk of mortality*** and the model has correctly predicted high risk patients.
- ***Lower Median in FP Cases***, clinically, suggests that the model is incorrectly identifying some ***low-risk patients as high-risk***.
  The following box-plots (blue as true positive cases and orange as false positive cases) of the features have higher median in TP cases and lower median in FP cases:



- A higher median of a feature in FP cases may suggest that the model is overestimating mortality risk for some patients.



- Conversely, if there's ***overlap between the boxes*** or ***similar median*** of the feature, the model's performance may be ***less reliable based*** solely on that feature**.**



For example, Min Glucose and Mean Glucose feature being the 3rd and 6th most important feature respectively have the same/overlap in their median. Which means that the model does not predict mortality correctly if these the patient has Min Glucose within the range 100 – 150.

- ***Outliers*** in FP cases might suggest that the feature's score is exceptionally high (or low) compared to most FP cases. In context of health care, Clinicians should investigate why certain patients with extreme values scores have survived. Further domain analysis is needed for these cases.

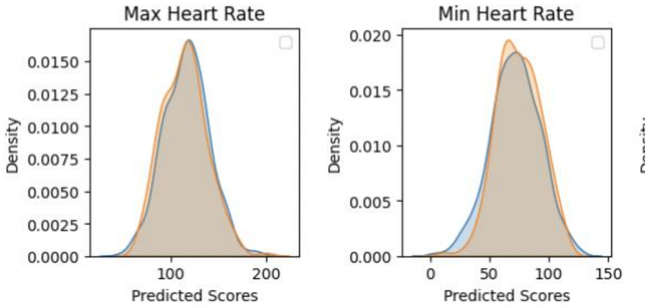
  From the graphs, we can see that Min Glucose, Mean Glucose have high scores compared to the majority. Also in Min Heart Rate, there is one case that is an outlier. Despite of the extremely high, the patient has survived but the model predicts that they have not survived.

  These FP cases may have subtle or atypical features that the model fails to recognize.

Density Plot

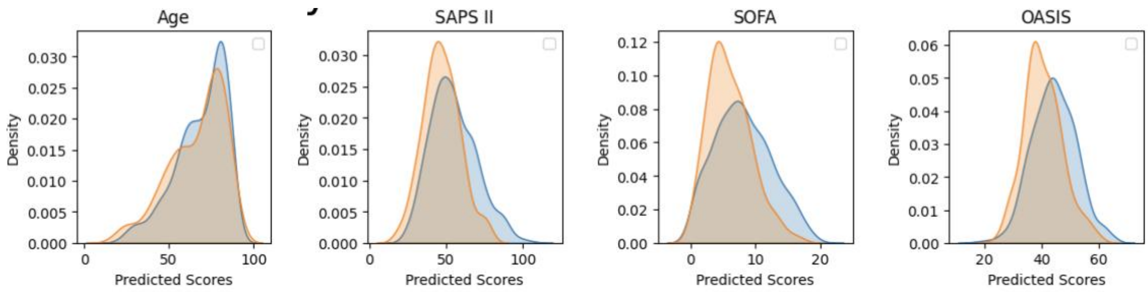
The following interpretation is done on the density plot visualization of FP and TP cases.

- **Overlap**: The area where both density plots overlap represents cases where it's challenging to distinguish between TP and FP based solely on that feature. These are ambiguous cases where the model struggles to differentiate survivors from non-survivors.



  Max Heart Rate and Min Heart Rate being in the top 10 of the important features, shows complete overlap in the cases of FP and TP. This suggests that the model struggles to differentiate solely on Max Heart Rate and Min Heart Rate.

- **Separation:** The regions where one density plot dominates over the other indicate confident predictions.



For example, there is a clear distinction between the FP (orange) and TP (blue) cases, therefore we can say that the model predicts confidently in that region. While the overlap tells us that there are ambiguous cases. In SAPS II, 50 has the most wrongly predicted cases. While in SOFA, it is between 0 – 10.

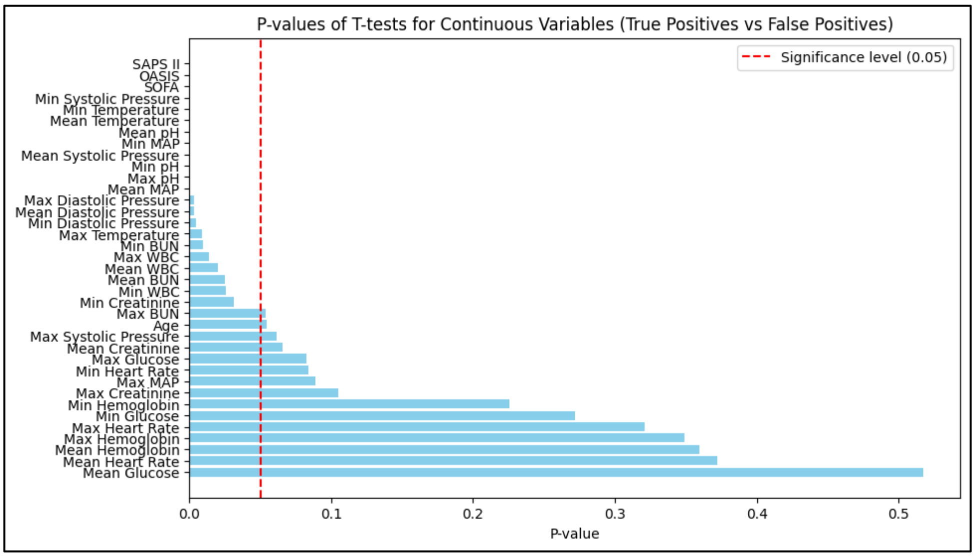
T-Test

This test tells us whether there is a significant difference between the means of two groups assuming the data is normally distributed.

In our case, we have two groups - False Positive and True Positive.

Null Hypothesis (H0): There is no significant difference in the distribution of features between false positive and true positive.

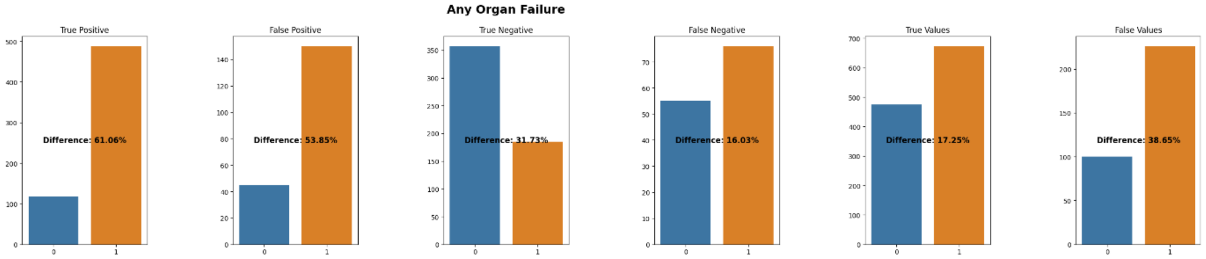
Alternative Hypothesis (H1): There is a significant difference in the distribution of features between false positive and true positive.



If the p_value is less than the significant value (0.05) we reject the Null Hypothesis. From the graph, we can see that from SAPS II to Min Creatinine it is less than 0.05 therefore for these features we can reject the null hypothesis which means that the alternative hypothesis is true therefore there is *a significant difference in the distribution of these features between false positive and true positive.*

Categorical Variables in False Positive Cases

The analysis revealed that categorical variables in the model did not exhibit significant differences in their distributions between true predicted and falsely predicted cases, as evidenced by similar modes, however, in variable **Any Organ Failure**, we can see that our model predicts Positive cases (patient dies) when there is organ failure, which, as can be seen in the comparison charts below, this also ***contributes to false positive cases.***



Overview of the False Positive Cases

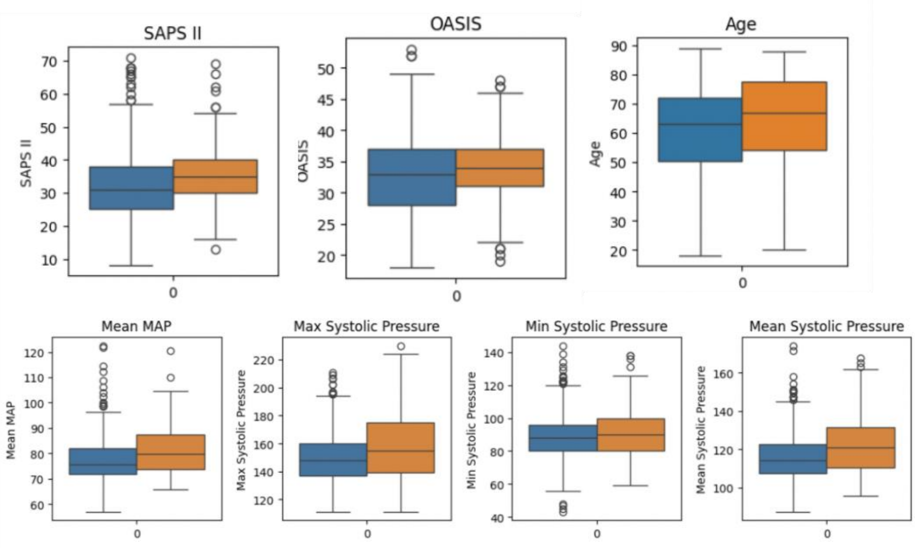
Overall, Box plots revealed that among the top 10 most important features, Min Systolic Pressure, Mean Systolic Pressure, Min PH, exhibited higher median values in FP cases, indicating their potential influence in incorrect predictions. Density plots of Mean Glucose and Min Glucose showed overlapping distributions, highlighting the complexity of distinguishing between FP and TP instances based solely on these features. But from the t-test we can see that the significance of difference between the groups for Mean and Min Glucose cannot be determined, therefore we can say that ***Min Systolic, Mean Systolic Pressure, Min PH mostly*** contribute to ***false positive cases*** in our random forest model.

False Negative Analysis

Box Plot

If the ***median of the feature for FN cases*** (patients who experienced mortality but were predicted as survivors) is s***ignificantly higher than the median for TN cases*** (correctly identified survivors):

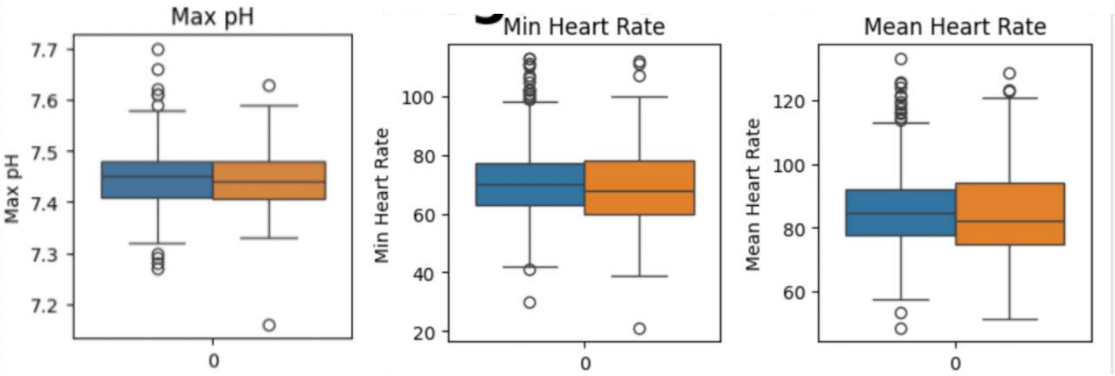
- It implies that the model is ***missing high-risk patients*** (false negatives).
- These FN cases have severe scores in the feature, but the model fails to predict their mortality.
- Clinically, this is concerning because these patients need close monitoring and timely intervention.



From the graphs (Blue is True Predictions and Orange is False Prediction), we can see that the most important feature ***SAPS II*** has a higher median in FN cases compared to TN cases along with other features which are on the top importance as well. This means that the model is missing these high-risk patients and stating that they have survived.

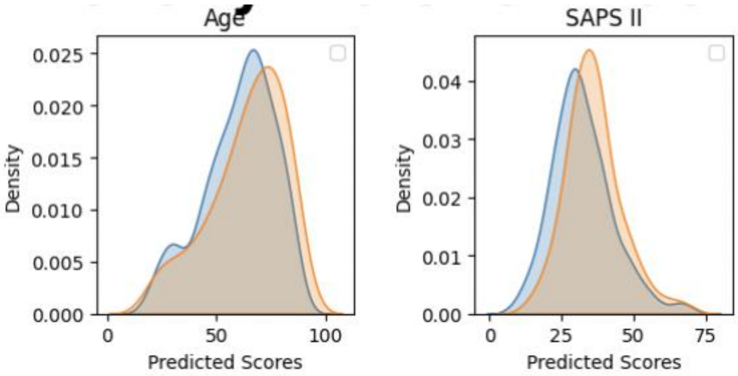
If the median of the feature for FN cases is lower than median for TN cases:

- Clinically, this suggests that the model is ***underestimating mortality risk*** for some patients***.***
- These FN cases may have ***subtle or atypical features*** that the model fails to recognize.



Density Plot

- **Overlap**: The area where both density plots overlap represents cases where it's challenging to distinguish between TN and FN based solely on that feature. These are ambiguous cases where the model struggles to differentiate survivors from non-survivors.



The density plots (Blue is True Predictions and Orange is False Prediction) for Age and SAPS II show that they almost have an overlap, meaning that the model struggles to differentiate for the values when the age is between 50 – 100, and the when the value of SAPS is between 25 – 50.

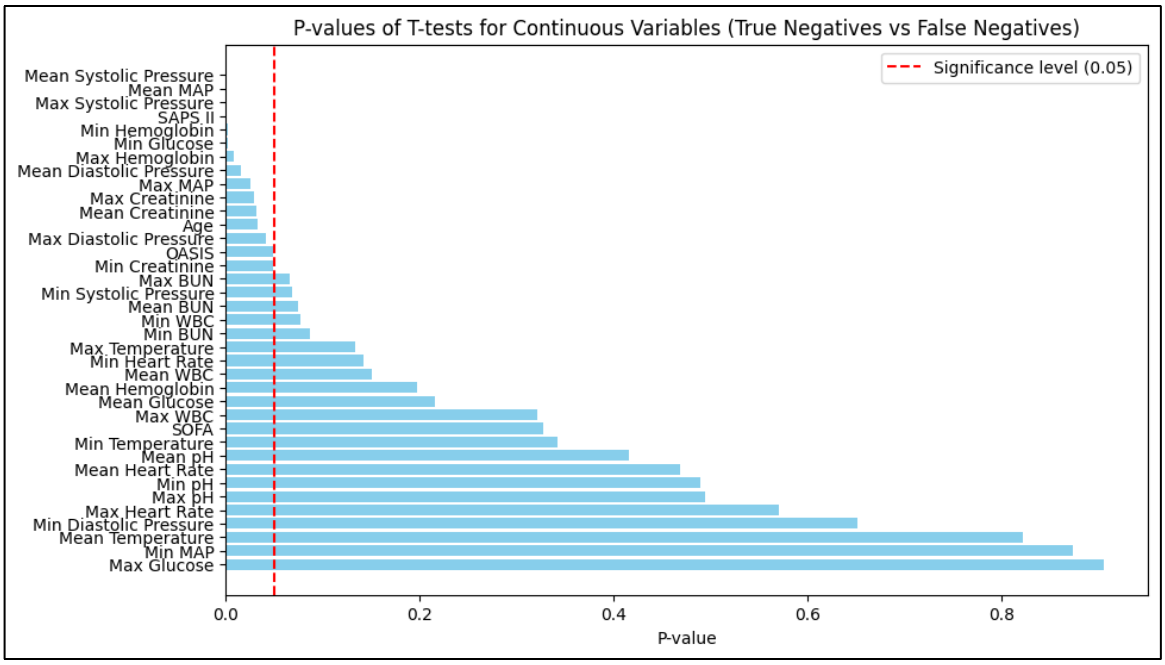
- **Separation:** The regions where one density plot dominates over the other indicate confident predictions.

T-Test

We also conducted T-test between two groups of True negatives and False negatives. Here also we have Null hypothesis and Alternate hypothesis like the previous one.

Null Hypothesis (H0): There is no significant difference in the distribution of features between false positive and true positive.

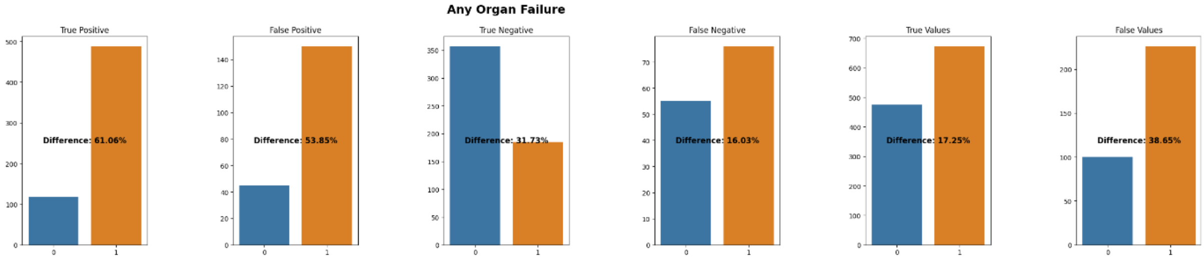
Alternative Hypothesis (H1): There is a significant difference in the distribution of features between false positive and true positive.



If the p-value is less than 0.05, we reject the null hypothesis. From the graph we can see that the variables from Mean Systolic Pressure to OASIS have p-values less than 0.05, therefore the null hypothesis does not stand for these variables and can be rejected, which means that there is a significant difference between the means of these variables in two groups.

Categorical Variables in False Negative Cases

Overall, Bar Charts for Categorical variables indicate the same tendency with the same mode for each of them, however in variable **Any Organ Failure**, we can see that our model predicts Negative cases (patient survives) when there is no organ failure, which, as can be seen in the comparison charts below, this also ***contributes to false negative cases.***

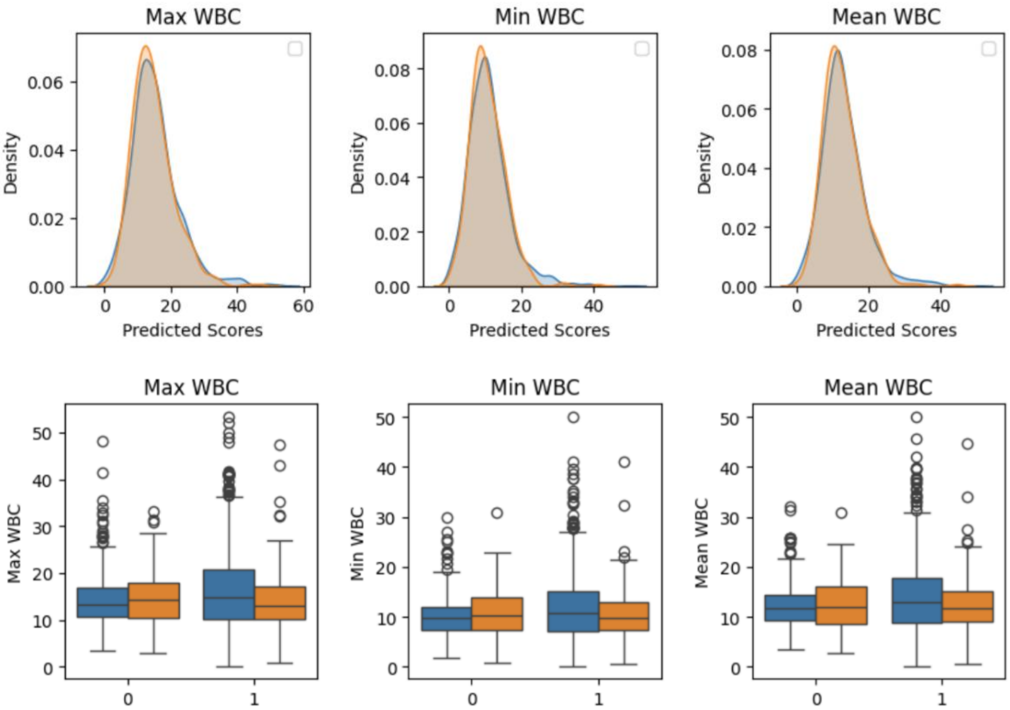


Overview of the False Negative Cases

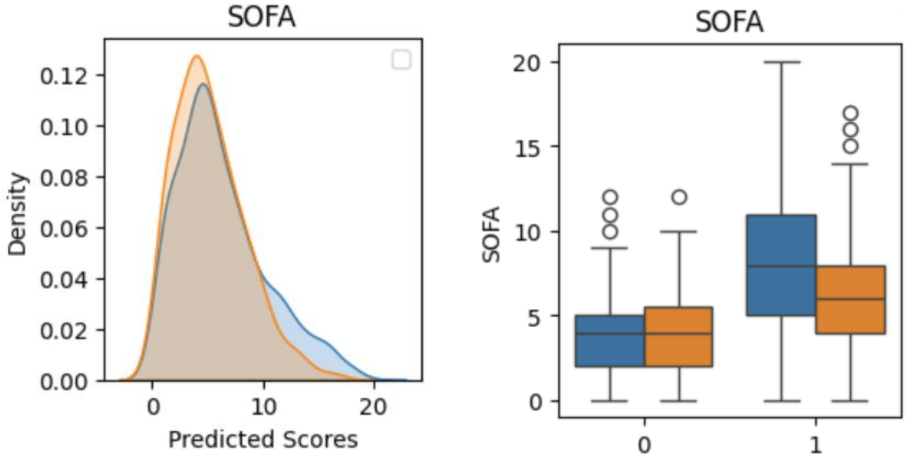
Overall, Box plots indicated that the most important features like SAPS II and OASIS had higher median values in FN cases, indicating their relevance in missed predictions. Density plots showed some overlap in Age and SAPS II but with noticeable differences, stating that the model finds it challenging in accurately classifying FN and TN instances based on these features alone. Also, from the T-Test, we found that these features have significant differences in their groups. Therefore, we can say ***SAPS II, Age, Mean MAP, Max BUN and Min Systolic Pressure*** (from the top 10 most important features) ***contribute to false negative cases.***

True values vs False values Analysis

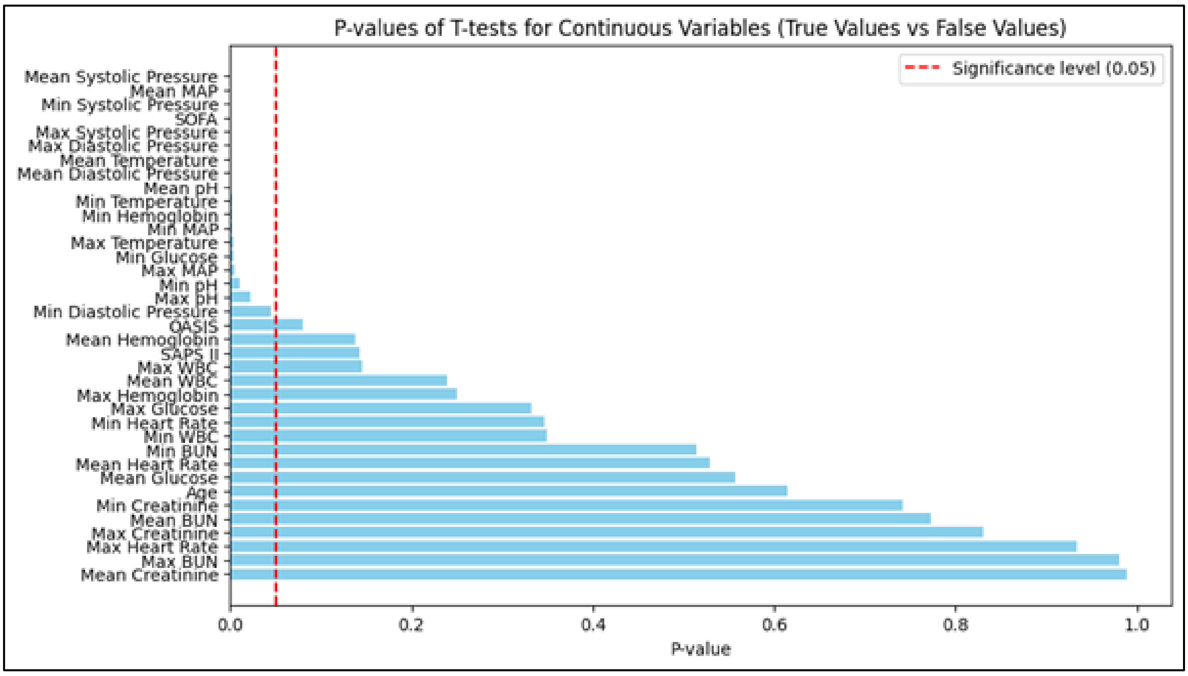
There are many cases in our Random Forest model where the model miss-classifies or predicts falsely. Looking at the density/box plots of Min BUN, Max BUN, WBCs, Creatinine, the area where both density plots overlap represents cases where it’s challenging to distinguish between true values and false values based solely on these features. These are ambiguous cases where the model struggles to differentiate survivors from non-survivors (Blue is True Predictions and Orange is False Prediction)



However, there are some variables that show a noticeable difference, for example, the disease severity variable SOFA, has a higher density from the range of 0-10 for falsely predicted values as compared to Truly predicted values.



Similarly, variables like Mean MAP, SOFA, SAPS II, Systolic, and diastolic variables also show some difference between true predictions and false predictions density/box plots. To confirm these findings, we performed a Welsh T-test for all the continuous variables between our two groups of all truly predicted values vs False predicted values and sorted them from lowest p-value to the highest. The Null Hypothesis here is that there is no significant difference between the means of our two groups. So, for the variables that have p-values less than 0.05, we conclude that there is a significant difference between the means of the two groups for that variable and reject the null hypothesis, which means that the differences in these variables are the common cases that our model miss classifies as false positives or false negatives. The graph below shows the p-values of variables that contribute to falsely predicted mortalities from most to least.



Recommendation

Based on the performance between models, we don’t have much difference between model with all features and model with 27 features. This could lead to a decrease in the number of features used, to decrease the resource usage and increase speed in making predictions. We could follow the variable important graph to find the variables that still give acceptance result.

In this project, we handle the imbalance in number of patients who die and survive by using Random Under-Sampling, however, this reduces a lot of our data. We can try with the origin dataset to see how the imbalance data affect to the model’s performance.

Suggestions for model enhancements

There are different ways by which we can try to enhance this model. Below are various strategies, that can be used to tailor this algorithm:

1. ***Changing the Tree Numbers(n_estimators):*** We can try to experiment with different number of trees i.e. n_estimators. Increasing the number of trees can sometimes improve performance, but there's a point where adding more trees may not significantly enhance performance while increasing computational cost.
2. ***Max features:*** Adjusting the max_features parameter to control the number of features for splitting at each node. By default, it's set to 'auto', but we can try to experiment with other values like 'sqrt', 'log2', or specific integers which can sometimes show better results.
3. ***Tree Depth and Complexity (max_depth, min_samples_split, min_samples_leaf):*** Modifying the parameters that are related to depth and complexity of individual tree, such as max_depth, min_samples_split, and min_samples_leaf. These parameters control the size of the trees and can help prevent overfitting.

Conclusion

With the mimic-iii dataset and through the data preparation from the phase 1, we have run the Random Forest model. We used the Random Under-Sampling method to handle the imbalance between patient who death and survives. We also get understand evaluation metrics of the classification model and compare the performance between models with all features and 27 features. The GridSearch is also used to tunning the parameter to get the optimal result. Then the we do the error analysis by using the box plot, density plot, bar plot and t-test to find the characteristics of the wrong prediction, false positive, and false negative.

The Random Forest machine learning model achieved a notable accuracy of 78%, demonstrating its ability to make correct predictions on a significant portion of the dataset. This level of performance underscores the model's effectiveness in capturing underlying patterns and relationships within the data. However, the error analysis revealed specific areas of improvement, particularly in addressing false positives and false negatives, which can further enhance the model's overall predictive capabilities and reliability in real-world applications.

In conclusion, the structured error analysis report sheds light on common characteristics among incorrectly predicted outputs, revealing distinct patterns contributing to false positives and false negatives in machine learning models. ***SAPS II, Age, Mean MAP, Max BUN and Min Systolic Pressure contribute to false negative cases***. ***Min Systolic, Mean Systolic Pressure, Min PH mostly contribute to false positive cases***. Statistical tests conducted uncover error patterns and biases, highlighting the feature characteristics and predictive outcomes. By investigating factors influencing errors, such as feature importance and distribution differences, this analysis provides insights for model refinement and enhancement of predictive accuracy using Random Forest Model.

Appendix

Name	Link
Flow chart	https://drive.google.com/file/d/1zmLbAhtNJeZ3Svtg4Xmr9L1wHvzHHYGB/view?usp=sharing
Error Analysis	https://drive.google.com/file/d/1LH4w7NDZRKG-YG279FGJyGjQl-kz1Q_s/view?usp=sharing
Decision tree	https://drive.google.com/file/d/1DknK3_Ux34loVm3gVyxR-XkxtWck5icK/view?usp=sharing
Classification Report	https://drive.google.com/file/d/1Qal8IRqSYLBMbygjNvpVDklHn_zJkau_/view?usp=sharing
Box plot of True Positive and False Positive	https://drive.google.com/file/d/1SMpP4wnSvzEdxH0UvLS_5--9d3zy4wmR/view?usp=drive_link
Box plot of True Negative and False Negative	https://drive.google.com/file/d/1WGZItstNdLjknNnoWSPm9b09ZrClqWUU/view?usp=drive_link
Box plot of True Prediction and False Prediction	https://drive.google.com/file/d/1L2dQirroXHzKUB0yDJUocqtZIWUtbCWd/view?usp=drive_link
Density plot of True Positive and False Positive	https://drive.google.com/file/d/1RAdWfA4yO4Zb4UzZi0urwTEyPtFgejIC/view?usp=drive_link
Density plot of True Negative and False Negative	https://drive.google.com/file/d/1-xf5g0OT80dqL0-A5RhTyOxdjHhCPjE3/view?usp=drive_link
Density plot of True Prediction and False Prediction	https://drive.google.com/file/d/1aD38N3L4396nG9KiXZ8BuRUUUEGswdCX/view?usp=drive_link
Bar plot of categorical variables	https://drive.google.com/file/d/1tPmnrW6hloPTJr_-bvNPd0gLoehfXV2K/view?usp=drive_link

Reference

Joseph, R. (2022, October 18). *Grid Search for model tuning*. Medium. https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e

Kumar, R. (2022, February 8). Methods to Check the Performance of the Classification Models. *EnjoyAlgorithms*. https://medium.com/enjoy-algorithm/methods-to-check-the-performance-of-the-classification-models-55ec50e0a914

Narkhede, S. (2021a, June 15). *Understanding AUC - ROC Curve*. Medium. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Narkhede, S. (2021b, June 15). *Understanding Confusion Matrix*. Medium. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Python Machine Learning—Grid Search. (n.d.). Retrieved March 31, 2024, from https://www.w3schools.com/python/python_ml_grid_search.asp

Random Forest Algorithm. (n.d.). Simplilearn.Com. Retrieved March 31, 2024, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

Sklearn.ensemble.RandomForestClassifier. (n.d.). Scikit-Learn. Retrieved March 31, 2024, from https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Tillo, T. (2021, May 29). Understanding the Random Forest Function Parameters in scikit-learn. *The Startup*. https://medium.com/swlh/understanding-the-random-forest-function-parameters-in-scikit-learn-9f42fde0101