Individual Assessment 1

COSC2753 Machine Learning
RMIT University Vietnam, Semester 2023A

Vo Tuong Minh
S-3877562
April 12th 2023

I. Definition

1. Project overview

Intensive Care Units (ICUs) are constantly challenged with monitoring their patients for the risk of developing sepsis. According to the Centers of Disease Control and Prevention [2]: "Sepsis is the body's extreme response to an infection. It is a life-threatening medical emergency. Without timely treatment, sepsis can rapidly lead to tissue damage, organ failure, and death."

Thus, the ability to predict in advance if a patient will develop sepsis is crucial for ICUs to manage their resources (beds, staff, medical kits, etc.) and ensure their patients' safety.

The goal of this project is to develop a Machine Learning model capable of predicting if a patient will develop sepsis during their ICU stay, based on their vital metrics, test results and age.

2. Problem Statement

Goal: ***Machine learning model capable of accurately predicting*** if a patient will develop sepsis (***Sepsis Positive / class 1***) or will not develop sepsis (***Sepsis Negative / class 0***) during their ICU stay, ***based on their vital metrics, test results and age***.

Strategy:

1. Exploratory data analysis.
2. PreprocessProcess training and testing data for machine learning algorithms.
3. Train classifier(s) on training data using appropriate Machine Learningmachine learning algorithms.
4. Evaluate and pick out the best classifier/algorithm.Evaluate and pick out the best algorithm classifier/algorithm.
5. Model optimization.Optimization.
6. Predict testing data.

3. Metrics

1. **F1 Score**: This is a common metric for binary classification. It is the harmonic mean of precision and recall, calculated using this formula[3]:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}$$

   I am using this as the main metric to evaluate our model's performance. It makes sure that both classes are prioritized equally, so that our model is effective at both predicting Sepsis Positive and Sepsis Negative.

   Reasoning: While every potential sepsis case needs immediate attention, false positive predictions might take medical resources away from patients that might need them. A balanced classifier would help medical staff better judge their situation and manage their resources. Furthermore, F1 score is robust to imbalanced data (having more of one class comparing to the other).

2. **ROC AUC**: Area under ROC (Receiver Operating Characteristic) curve – another popular metric for binary classification. When comparing different models for our problem, we will use this metric in combination with F1 to judge on which one is the best.

   Reasoning: While ROC AUC is more sensitive to imbalanced data, it is better than F1 at visualizing which model is better, as demonstrated by Fig. 1.

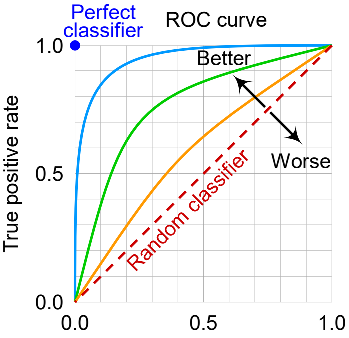

Fig. 1. ROC Curve[1]

II. ANALYSIS

1. Data Exploration

Our dataset is relatively small compared to the usual machine learning dataset. It contains two files:

Paitients_Files_Train.csv	Training dataset, contains all feature columns and a target "Sepssis" column.	11 columns by 599 samples
Paitients_Files_Test.csv	Test dataset, only contains features columns but missing the target column.	10 columns by 169 samples

- The target column is "Sepssis" (misspelled). It contains either the string Positive or the string Negative. We will convert these into numerical values as our models can only process numbers: 0 – Negative, 1 – Positive.
- Features: We have 8:

Column name	Data type	Description	Acceptable range [4-6]
PRG	int	Plasma Glucose (mmol/L)	- Normal: 3.9 < PRG < 5.6 - At risk of diabetes: 5.6 < PRG < 6.9 - Diabetes 7.0 < PRG
PL	int	Blood Work Result-1 (μU/ml)	Information not found. Common sense: Larger or equal to 0.
PR	int	Blood Pressure (mmHg)	Normal: 90/60 < PR < 140/90
SK	int	Blood Work Result-2 (mm)	Information not found. Common sense: Larger or equal to 0.
TS	int	Blood Work Result-3 (μU/ml)	Information not found. Common sense: Larger or equal to 0.
M11	float	Body Mass Index (kg/m^2)	- Normal: 18.5 < M11 < 24.9 - Overweight: 25 < M11 < 29.9 30.0 < M11
BD2	float	Blood Work Result-4 (μU/ml)	Information not found. Common sense: Larger or equal to 0.
Age	int	Patient's age (years)	Common sense: Larger or equal to 0.

- I could not find any information regarding the acceptable range for Blood Work Result-1 though 4. However, by looking at their unit (μU/ml – micro units per milliliters), we can reasonably assume that they are measurement of some kind of particle or cells commonly found in blood. Thus, they cannot be 0.
- Similarly, plasma glucose, blood pressure, and body mass index cannot be 0.
- Both datasets contain no duplication, and no null values. However, they do contain invalid values (0 – as explained above). We imputed these during data preprocessing.
- We have two unused columns: ID and Insurance. These columns have no connection to the patient's health and should have no correlation to our problem. We will drop them before training our models.
- **Distribution** (Fig. 2): Both test data and the train data are very equally distributed: We can confidently assume that our model trained on the train dataset will be able to perform well on the test dataset.
  - Some columns are not normally distributed. We will need to transform them to reduce adverse effects on regression models.
  - All features have different min and max. Feature scaling will be required when we process the data for our models to digest.
  - Both train and test data, there are many outliers affecting the distribution of some features. We will need to remove these outliers.

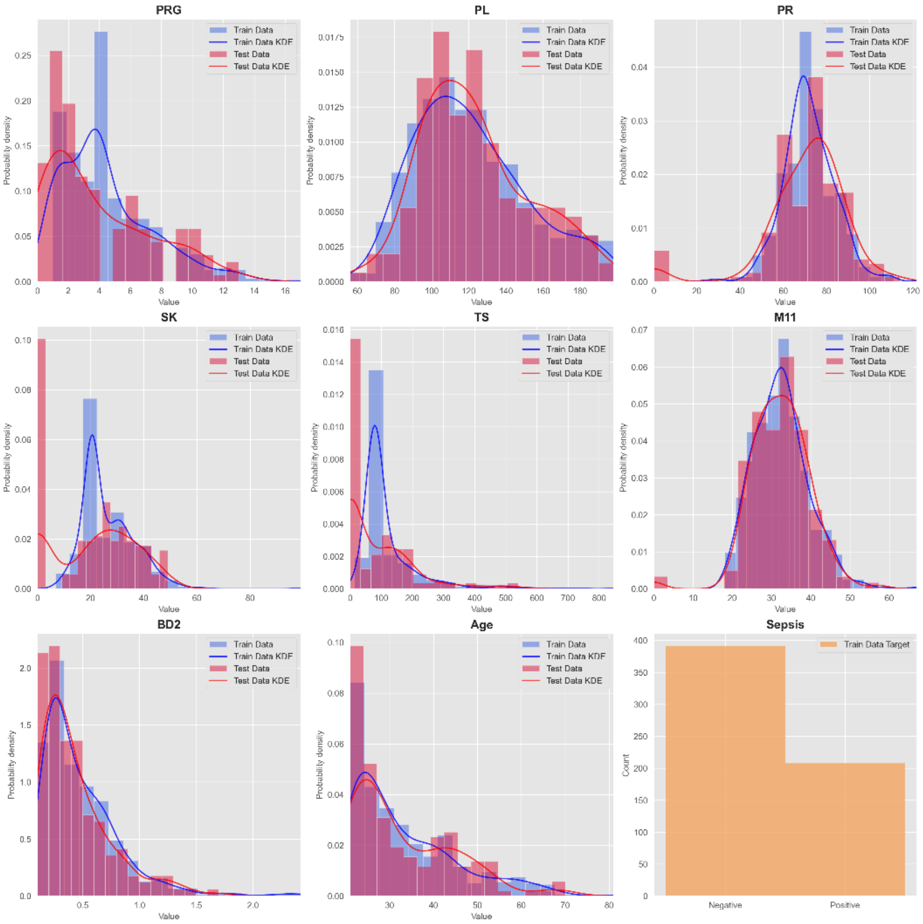

Fig. 2. Data Distribution

- **Feature correlation** (Fig. 3): There is no strong correlation between features. Additionally, both the train data and test data have very similar correlation between features. This further re-reinforces our confidence that our model trained on the train dataset will perform well on the test dataset.

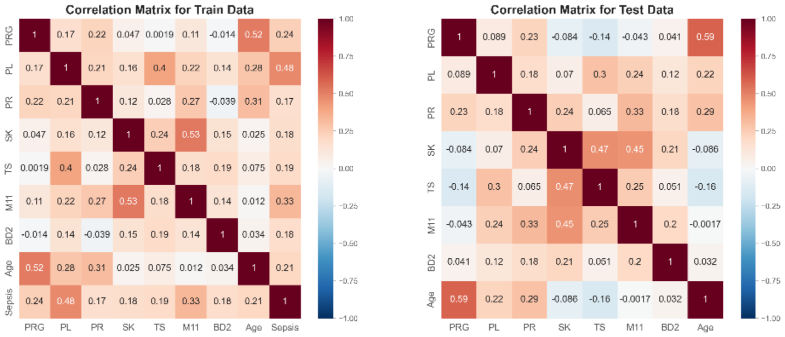

Fig. 3. Feature Correlation

2. Algorithms and Techniques

Considering the small number of samples in our dataset, the number of features, the binary nature of our problem, and the assignment constraints, I have developed models based on these algorithms:

Algorithm	Hyper-parameters
Logistic Regression	- Regularization strength λ (lambda) - Regularization method: Either L_1 (lasso) or L_2 (ridge)
Decision Tree	- Maximum tree depth - Minimum number of samples required to split an internal node
Bagged Trees	- Number of trees - Minimum number of samples required to split an internal node
Random Forest	- Number of trees - Maximum tree depth

We are using k-Fold cross validation and Grid Search method to select and fine tune our hyper-parameters to avoid under/overfitting and optimize our models.

For Logistic Regression, I experimented with both Linear Features and Polynomial Features of order up to 4. Any higher than that has negligible improvement in performance (F_1) at an extremely high computational cost (determined by experimentation on my personal computer).

3. Benchmark

The initial benchmark for the classifiers above is a simple Logistic Regression model with Linear Features and no regularization, trained using processed data and hold-out cross validation. Its F_1 validation score was **0.79**.

My goal is to develop a robust classifier with an F_1 validation score of **at least 0.80**.

III. METHODOLOGY

1. Data Processing

Data processing for our problem can be summarized in the following steps:

1. Data cleaning (remove duplicates, rename misspelled column "Sepssis" to "Sepsis," drop unused columns "ID" and "Insurance," impute invalid values with column mean).
2. Impute outliers using k-Nearest Neighbor algorithm to calculate new value, then cap the remaining outliers using IQR (Interquartile Range) method.
3. Using natural log transformation ($x_{new} = \ln(x_{old})$) to transform data into a more normal distribution.
4. Standardize (and in effect, scale) features: 0 mean, unit variance.
5. Feature sampling (because our training data is imbalanced) to synthesize more Positive cases so that its number is equal to the number of Negative cases.

The benchmark classifier mentioned in II.3. Benchmark was used to evaluate the effectiveness of my data processing procedure. I compared its validation F_1 score when it was trained on raw data versus when it was trained with processed data. The results are as follow:

- F_1 score with raw dataset: 0.73.
- F_1 score with processed dataset: **0.79.**
- Improvement: 0.06.

This confirms that this data processing procedure is appropriate and effective.

2. Implementation

2.1. Model development and algorithm selection

As outlined in II.2.Algorithms and Techniques, I have experimented with these 7 different classifiers:

- Logistic Regression – Linear Features (code: **logistic1**)
- Logistic Regression – 2nd Order Polynomial Features (code: **logistic2**)
- Logistic Regression – 3rd Order Polynomial Features (code: **logistic3**)
- Logistic Regression – 4th Order Polynomial Features (code: **logistic4**)
- Decision Tree (code: **tree**)
- Bagged Trees (code: **bagged_trees**)
- Random Forest (code: **forest**)

All of them are trained using pre-processed data with any additional polynomial features transformation if needed.

Since we are using k-Fold cross validation and Grid Search for hyperparameter tuning, for each of the above 7, many more models were generated and trained. We only pick the model with the best performance. This is automatically done by scikit-learn GridSearchCV and KFold Python classes.

We then compare all of them by F_1 score and ROC AUC calculated from predicting and validating the holdout validation set. Here are the results:

Model code	ROC AUC	F1
logistic1	0.85431	0.79012
logistic2	0.85724	0.77922
logistic3	0.79610	0.79290
logistic4	0.85659	0.85366
tree	0.82163	0.82500
bagged_trees	0.93789	0.86420
forest	0.94846	0.86076

Note: The base estimator for **bagged_tree** is **tree**.

We see that all models have high validation F_1 scores. However, a closer look into their training score reveals that all tree-based models are overfitting. As for the logistic regression model, the highest F_1 they can achieve is 0.85 with the model **logistic4**.

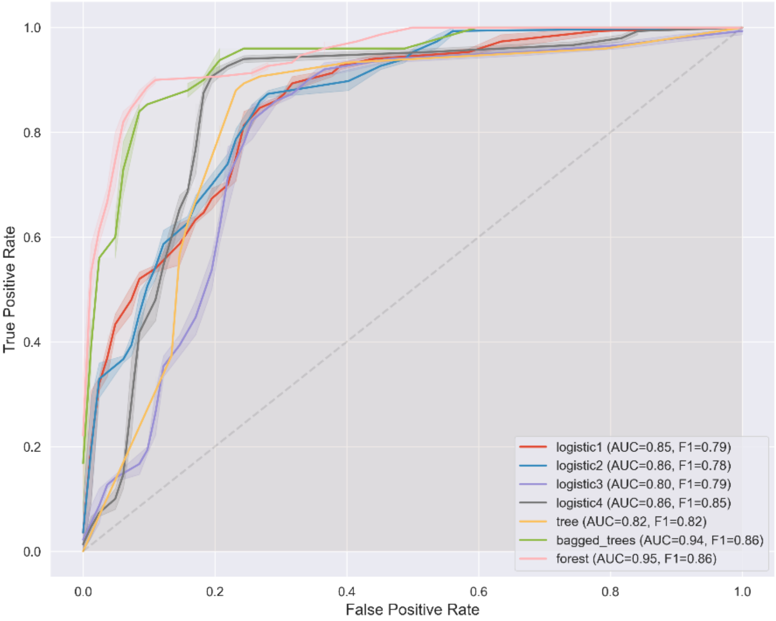

Fig. 4. Models Comparison - ROC Curve

However, when taking ROC AUC into account, we can clearly see that the tree-based models outperform the logistic regression models. I have decided Bagged Trees to be our model of choice for the following reasons:

- It has both high F_1 and ROC AUC.
- It has the most options to further optimize it to reduce the current overfit: Optimizing the base Decision Tree estimator by post-pruning, pre-pruning (hyperparameter tuning) for either the base Decision Tree or the Bagged Trees model itself.

2.2. Optimization

After a lot of experimentation, I was able to lessen the overfit of the Decision Tree model by adjusting by pre-pruning procedure to not use k-Fold cross validation. Further experimentation with post-pruning did not reduce the overfitting but instead just reduced my model's overall performance.

This Decision Tree model is then used as the base estimator for my Bagged Trees, which also has increased performance after I stop using k-Fold cross validation.

This suggests that I have reached the peak capacity for my model.

Conclusion

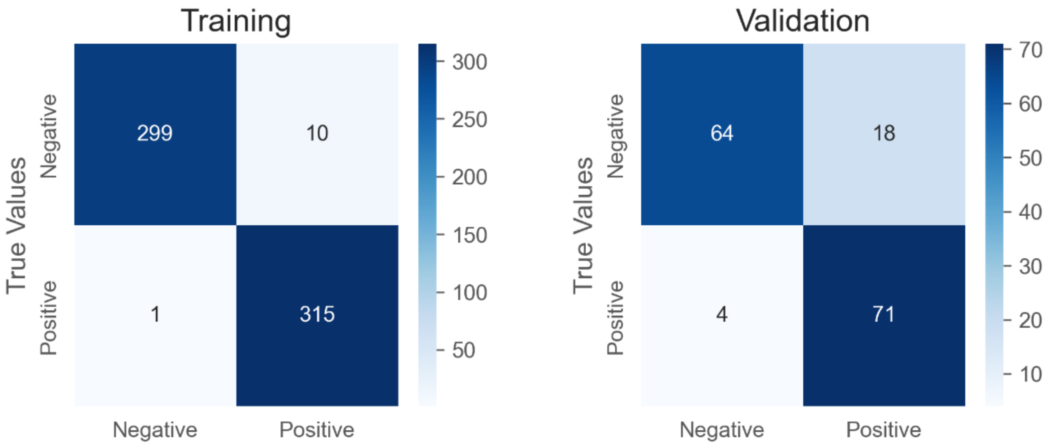

Fig. 5. Confusion Matrix - Final Model

Although there is still overfitting, The final model has achieved a high F_1 score of **0.86** for validation set. This is the peak capacity for this model. We can now conclude model development and use this model to predict the test dataset.

References

[1] cmglee and MartinThoma, "Receiver Operating Characteristic (ROC) curve with False Positive Rate and True Positive Rate.," ed, 2018.

[2] Centers of Disease Control and Prevention. "What is Sepsis." https://www.cdc.gov/sepsis/what-is-sepsis.html#:~:text=Sepsis%20is%20the%20body%27s%20extreme,%2C%20skin%2C%20or%20gastrointestinal%20tract (accessed 12 April 2023).

[3] F. Pedregosa *et al.* "Metrics and scoring: quantifying the quality of predictions." https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics (accessed 12 April 2023).

[4] L. Riley. *Mean fasting blood glucose*, World Health Organization. [Online]. Available: https://www.who.int/data/gho/indicator-metadata-registry/imr-details/2380#:~:text=The%20expected%20values%20for%20normal,(5.6%20mmol%2FL)

[5] Centers of Disease Control and Prevention. "Assessing Your Weight." https://www.cdc.gov/healthyweight/assessing/index.html#:~:text=If%20your%20BMI%20is%20less,falls%20within%20the%20obese%20range (accessed 12 April 2023).

[6] National Health Service. "What is blood pressure?" https://www.nhs.uk/common-health-questions/lifestyle/what-is-blood-pressure/ (accessed 12 April 2023).