Assignment 2: Cardiovascular Risk

Scott Ratchford, Ben Fioresi, Alex Reese

# Introduction

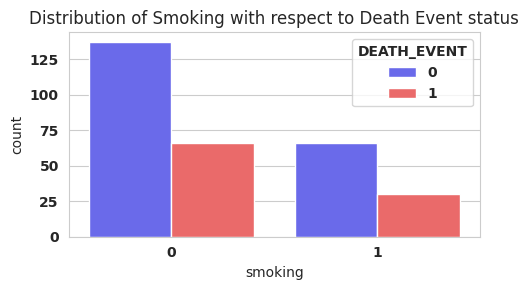
In this lab, we worked with comprehensive clinical records from 299 patients. This data encompasses several key health indicators that have been recorded to assess the severity and outcome of heart failure in each patient. Our goal was to apply and compare three machine learning models – Random Forest Classifier, K-Nearest Neighbor Classifier, and Decision Tree Classifier – to predict the mortality outcome of heart failure.

# Preprocessing Steps

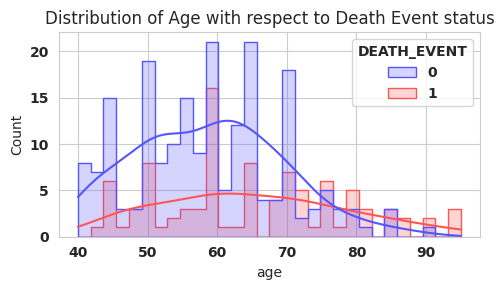
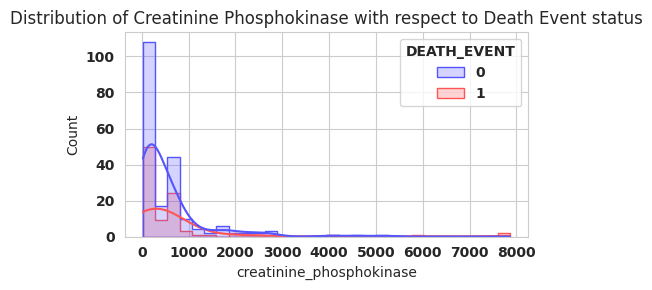
An interesting trend we noticed is that some of the platelet counts are set to 263358.03, which is the average of the entire dataset. This also occurs in the age column. We decided to drop these values because we anticipated it would skew the models’ results. We used an 80/20 test/train data set split.

We did not train our models on the “time” value for any data point because it’s a measurement of when the patient was tested, this is irrelevant to the patient's health. Accuracy of the optimized models greatly improves without this feature, as expected.

As part of our preprocessing, we visualized the distributions of each column with respect to the Death Event status. These visualizations are shown below.

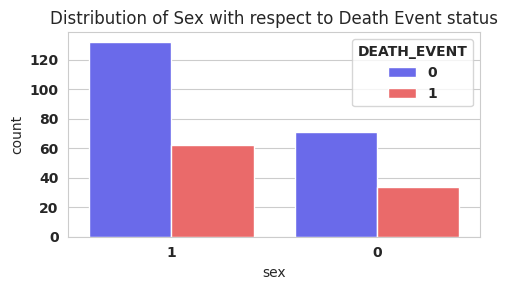


This graph visualizes the distribution of whether the patient smokes with respect to the Death Event status. This shows that there is no noticeable correlation between smoking and patient mortality as, similar to the distribution of sex, the ratio is about 2 survived to 1 deceased for both smokers and non-smokers.

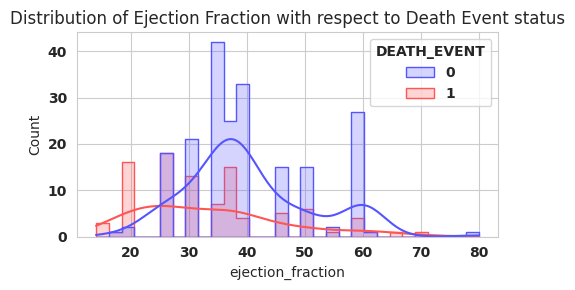


This graph visualizes the distribution of Creatinine Phosphokinase (CPK) level in the blood with respect to the Death Event status. This shows that the level of CPK in the blood has a slight. There is one outlier case, in which the patient had a CPK level of almost 8000, and it resulted in patient mortality.

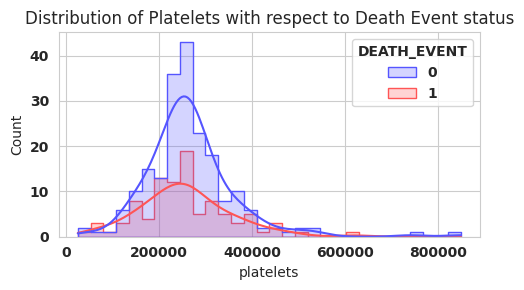
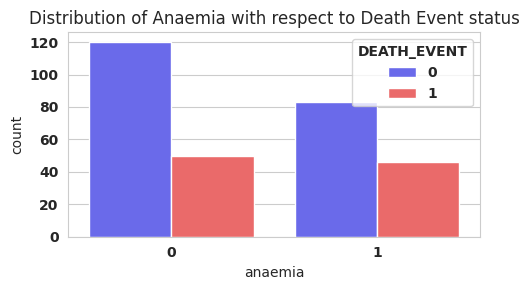
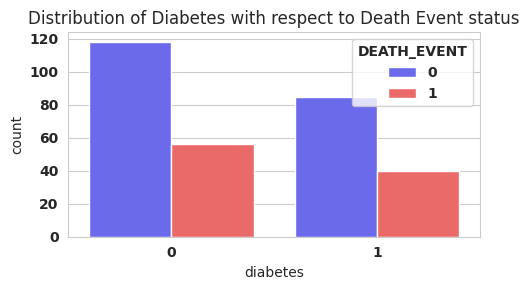
This graph visualizes the distribution of age with respect to the Death Event status. This shows that the outcome of mortality is more common among the older patients, while around age 60 there is a significantly larger amount of mortality outcomes than the ages around it. The number of patients for each age seems to spike every 5 or so years, which may imply that some rounding may have occurred.



This graph visualizes the distribution of sex with respect to the Death Event status. This shows that there is no correlation between sex and patient mortality. The ratio is about 2 survived to 1 deceased for both sexes, which is the trend across the whole dataset.



This graph visualizes the distribution of the ejection fraction (percentage of blood leaving the heart at each contraction) with respect to the Death Event status. This shows that patient mortality is heavily correlated to low ejection fraction values, while the survived status generally follows a bell curve that peaks at 40%, and peaks again slightly at 60%.

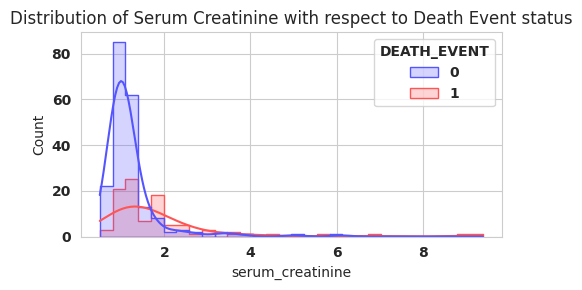
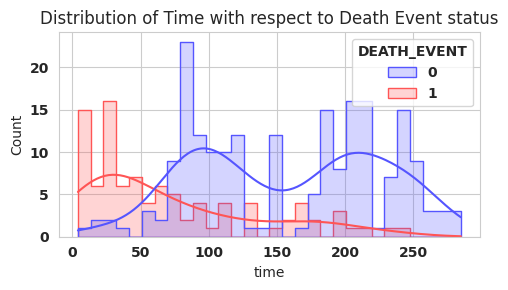
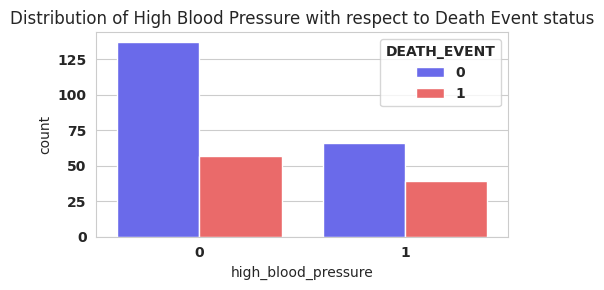


This graph visualizes the distribution of the presence of diabetes with respect to the Death Event status. Similar to the anemia graph, this shows that the presence of diabetes has a slight correlation to patient mortality.

This graph visualizes the distribution of the presence of anemia with respect to the Death Event status. This shows that the presence of anemia has a slight correlation to patient mortality, as the ratio of survived to deceased is more heavily weighted towards deceased when anemia is present.

This graph visualizes the distribution of the numbers of platelets in the blood with respect to the Death Event status. This shows that there is little to no correlation with patient mortality. The graph is shaped like a bell curve, with an average around 250000. The ratio of Death Event status is 2 survived to 1 deceased, which matches the dataset as a whole.

A graph with red and blue lines

Description automatically generated

This graph visualizes the distribution of the level of serum sodium in the blood with respect to the Death Event status. This shows that there is little to no correlation between serum sodium levels and patient mortality, as the ratio mostly stays at a consistent 1 deceased to 2 survived. However, there is a significant number of deceased statuses around 135 mEq/L. As well, the ratio closes in as the values increase or decrease from the average, but that may be due to not having enough samples.

This graph visualizes the distribution of high blood pressure with respect to the Death Event status. This shows that high blood pressure has a medium correlation with patient mortality, as the ratio of survived to deceased is much closer when the patient has high blood pressure.

This graph visualizes the distribution of the follow-up period with respect to the Death Event status. This shows that patient mortality is highest when the follow-up period is shorter than about 60 days. After that, the deceased result trends slightly lower as time increases.

This graph visualizes the distribution of the level of serum creatinine in the blood with respect to the Death Event status. This shows that there is no noticeable correlation between serum creatinine levels and patient mortality. There are some outliers, but they are a mix of outcomes that do not provide any useful information.

# Feature Engineering

According to the National Institute of Diabetes and Digestive Kidney Diseases, Kidney failure is a common sign of cardiovascular disease. They have and equation called Glomerular Filtration Rate(eGFR) that uses Race, Sex, Serum Creatinine, and Age to determine kidney risk. Since we are only missing Race in our data set, we tried a simplified version of the equation using the remaining 3 categories. Unfortunately, this did not seem to improve our models and were left out in the end. We also tried to make a column with the ratio between creatinine phosphokinase and serum creatinine but that didn’t help any of the models either. Normalizing any of the columns also had no affect.

# Model Performance Metrics

We evaluated the following metrics:

* Accuracy: number of correct predictions
* Precision ratio of correctly predicted positives to total predicted positives
* Recall: ratio of correctly predicted positives to all actual positives
* F1 Score: weighted average of Precision and Recall
* ROC AUC: Receiver Operating Characteristic - Area Under Curve, evaluates the ability of the model to differentiate between positives and negatives
* Confusion Matrix: table used to evaluate the performance of the model, showing the actual vs. predicted classifications

# Results Analysis

## Default Hyperparameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Random Forest** | | **Decision Tree** | | **K-Nearest Neighbor** | |
| Accuracy | 73.3% | Accuracy | 60.0% | Accuracy | 63.3% |
| Precision | 76.9% | Precision | 48.3% | Precision | 54.5% |
| Recall | 43.5% | Recall | 60.9% | Recall | 26.1% |
| F1 Score | 55.6% | F1 Score | 53.9% | F1 Score | 35.3% |
| ROC AUC | 85.3% | ROC AUC | 60.2% | ROC AUC | 56.3% |
| Confusion Matrix | |  |  | | --- | --- | | 34 | 3 | | 13 | 10 | | Confusion Matrix | |  |  | | --- | --- | | 22 | 15 | | 9 | 14 | | Confusion Matrix | |  |  | | --- | --- | | 32 | 5 | | 17 | 6 | |

It seems that TensorFlow's default Random Forest configuration would perform better with additional features, given the relatively small number of features in this data set. The default number of trees, 300, is excessive for a data set with a mere 13 features (counting dropped fields and DEATH\_EVENT, which were not included in the training set). There are fewer patients in the data set than the default number of trees.

A Decision Tree with the default parameters performs the worst of any attempted model, due to its unbounded depth. The natural tendency of the algorithm is to make itself as complex as possible, increasing its depth for little to no gain. This often harms the generality of the model. Needlessly high depth is an indication of overfitting.

The K-Nearest Neighbors algorithm lacks the number of data points it needs to perform well. At 299 data points, the KNN algorithm struggles to effectively classify any data points. It barely performs better than random chance, and fine-tuning the model does not improve its accuracy at all.

## Optimized Hyperparameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Random Forest** | | **Decision Tree** | | **K-Nearest Neighbor** | |
| Accuracy | 83.3% | Accuracy | 88.3% | Accuracy | 63.3% |
| Precision | 93.3% | Precision | 86.4% | Precision | 55.6% |
| Recall | 60.9% | Recall | 82.6% | Recall | 21.7% |
| F1 Score | 73.7% | F1 Score | 84.4% | F1 Score | 31.3% |
| ROC AUC | 85.2% | ROC AUC | 87.3% | ROC AUC | 55.5% |
| Confusion Matrix | |  |  | | --- | --- | | 36 | 1 | | 9 | 14 | | Confusion Matrix | |  |  | | --- | --- | | 34 | 3 | | 4 | 19 | | Confusion Matrix | |  |  | | --- | --- | | 33 | 4 | | 18 | 5 | |

Other models were tuned with particular hyperparameters. The libraries for the models provide functions and classes to search the space of particular hyperparameter combinations and return the best combinations, based on accuracy and similar metrics.

Random Forests are tuned with Tensorflow's tuner.RandomSearch class. This searches many possible combinations of hyperparameters, but not all possible combinations. The space is too large to try all combinations in a reasonable timeframe. The accuracy of the Random Forest model improves by 10% when its parameters are optimized. This significant improvement can be attributed to the reasonable number of trees and their depth. The random search of the space of hyperparameters found improved performance with low values for both.

Decision Trees and K-Nearest Neighbors models are tuned with scikit-learn's GridSearchCV class. This performs exhaustive evaluation of all combinations of the provided hyperparameters. Exhaustive evaluation is more feasible with Decision Trees and KNN than with Random Forests, due to their low runtimes.

The accuracy of the optimized Decision Tree improved by an incredible 28% with its new hyperparameters. With the capped depth, the model was forced to optimize all its decisions like high-level discriminants, instead of focusing on splitting hairs with arbitrary cut-offs for features containing low correlation.

KNN barely improved in any way from improved hyperparameters. As mentioned in the previous section, the KNN algorithm struggles to effectively classify any data points with such a small training set. Its performance metrics move less than a single percentage point in either direction.

# Real-World Applications

While these models were not accurate enough in the predictions to instill great confidence in their results, the optimized models performed well enough to point medical professionals in the right direction. Doctors could use an optimized decision tree to help them consider a course of action, given more training data. The random forest model may also be suitable for this, but it seems unlikely that KNN would be useful for diagnoses.

Given that the dataset comprises only 299 rows, practically applying its insights to a real-world application should be approached with caution. While the dataset may provide valuable preliminary insights to point medical professionals in the right direction, the findings would benefit greatly from validation with larger, more diverse data. Hence, any conclusions or predictive models derived from this dataset should be considered experimental until validated by further research with more extensive data.