Final Project Documentation

**The “.py” file named XGBTest contains:**

The dataset named heart\_failure\_clinical\_records\_dataset saved in a dataframe that was called heart\_failure, downloaded from Kaggle.

**The use of several methods and functions to clean and treat the data:**

‘.dtypes’ which led to the conclusion that the data types are all numerical which saves a lot of trouble changing data.

‘.value\_counts()’ which led to the conclusion that this is an imbalanced dataset and this needs to be handled.

‘.describe()’ and a custom piece of code that counts the number of datapoints outside the IQR, which led to the conclusion that the dataset doesn’t seem to have relevant outliers because the data is evenly distributed, and the outliers can be relevant to show anormal situations that can lead to heart failure.

‘.isna()’ which led to the conclusion that the dataset doesn’t have any missing data points.

‘.corr()’ and analyzing the correlations between the features, which led to the conclusion that there isn't the need to drop any columns from the dataset because there is no collinearity.

**The creation of an SQL table using alchemy, and the creation of a connector to use the data stored on MySQL to create the Machine Learning model.**

**The procedure to find out the best XGBoost model to use in order to get the best possible result:**

‘train\_test\_split()’ to separate the data that was imported from MySQL.

‘resample()’ to oversample the training data in order to deal with class imbalance.

‘GridSearchCV()’ with 3 fold cross validation and the use of XGBClassifier() and the hyper parameters specified in the grid to search for the best parameters to use on XGBoost.

‘DMatrix()’ to convert the train and test data into a matrix that can be read by the algorithm.

‘train()’ and ‘predict()’ to train the algorithm and predict the labels of each data point.

‘roc\_auc\_score()’ and ‘confusion\_matrix()’ to test the accuracy of the algorithm and to see the number of False Negatives.

There is also a piece of code commented that doesn’t need to be ran in order for the code to serve its purpose but was used to the determine a parameter (num\_boost\_round) to train the xgb model.

**The creation of a KNN model to compare a simple algorithm with the complex algorithm that is XGBoost:**

The training data was fit to the model using ‘fit()’, then it was standardized with the ‘StandardScaler()’ and the model was used to predict the labels using ‘predict’. Later the accuracy was checked using the ‘roc\_auc\_score()’ and the number of false negatives was checked using ‘confusion\_matrix()’.

**The use of pickle library to create a file with the XGBoost model created so it can be used in different occasions.**

**The “.py” file named Heart\_failure\_APP contains:**

The use of several tkinter imports to create an app with Labels, Buttons and Entry boxes that can be used to input various values to insert in the XGBoost model that was loaded into this file.

The data that was inserted into the app will be transformed to the appropriate data type and entered into a dataframe. This dataframe will be converted into a DMatrix and inserted into the XGBoost model. The model will predict the label and depending on the outcome the app will display the result that was predicted.