DSC 550 – Final Project Case Study

For my case study I sought to analyze a dataset survival after a heart attack using Echocardiogram features.  There are several variables and echocardiogram characteristics that are a part of this dataset and one of the primary outcomes is survived 1 year or not.  This can be found at the following URL:

<https://www.kaggle.com/loganalive/echocardiogram-uci>

There are several variables in this dataset including:

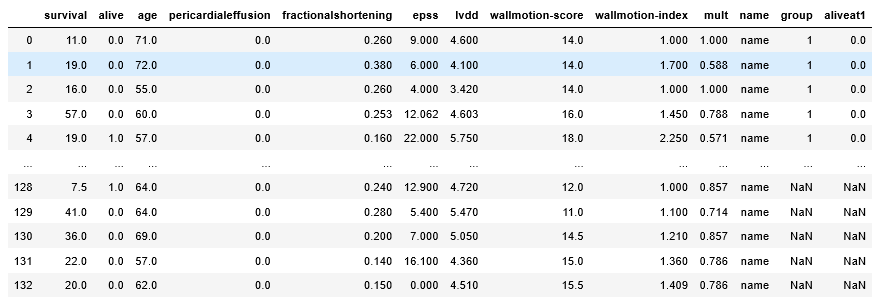
* Survival
* Alive (Binary Variable)
* Age
* Pericardial Effusion
* Fractional Shortening Measurement
* EPSS
* Left Ventricular Diastolic Dimension
* Wall Motion Score
* Wall Motion Index (calculated by Wall Motion Score divided by Number of Segments)
* Aliveat1 (Whether or Not Person Was Alive After One Year)

**I determined that the variables Alive and Survival helped derive our final categorical target variable of Alive At 1 Year so I dropped those variables from the data-set in that they are redundant. Also, I dropped the wall motion score as well as this was redundant since wall motion index was derived from this.**

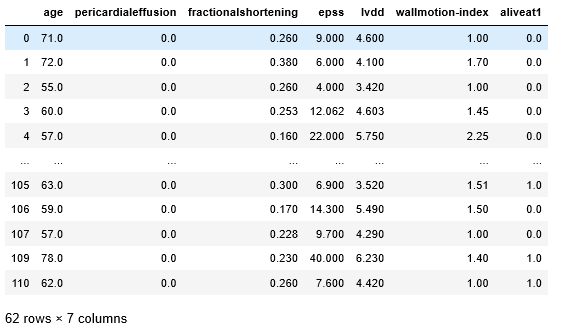
Step By Step of The Assignment:

1. Data was loaded into Python with Pandas.

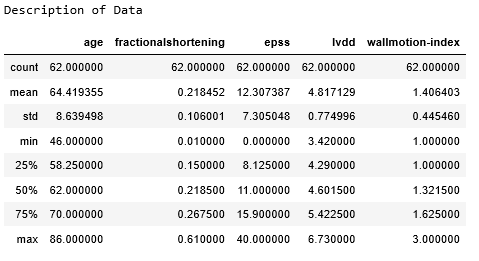
1. Display the dimensions of the file using df.shape and calling the dataframe



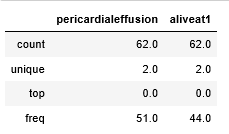
1. Using .dropna, I dropped all rows with null values from the dataframe and dropped the redundant variables and case identifiers (wallmotion score, mult, name, group, survival, and alive).



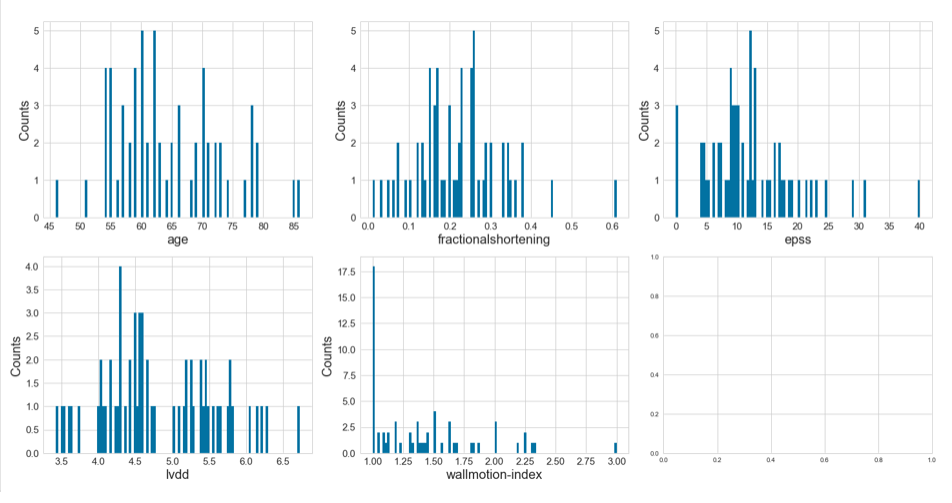
1. Using the .describe() function, I looked at basic statistics (mean, std, min, max, etc).

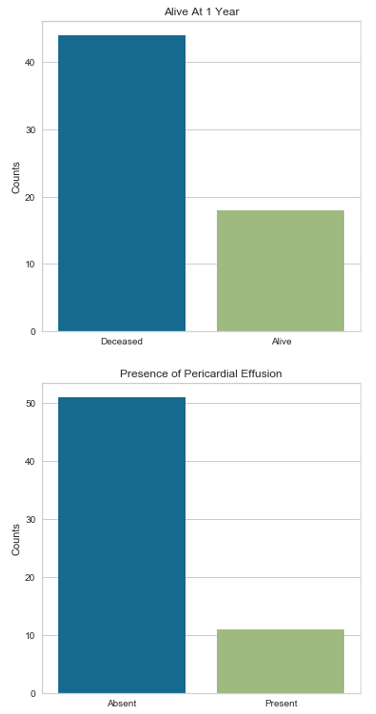


1. I encoded the two categorical variables as objects for work further down the road using the astype(‘object’) function and viewed the descriptive statistics using command of .describe(include=[‘O’]).

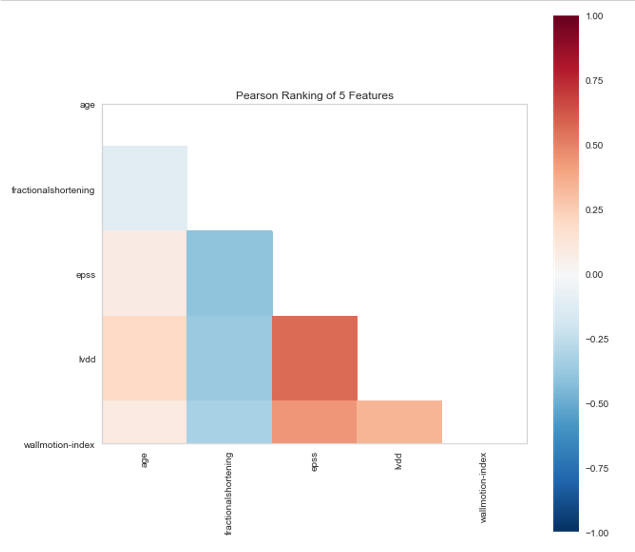


1. Using matplotlib, I will generate histograms of the numerical variables such as age, fractional shortening, EPSS, LVDD, Wall Motion Score, and Wall Motion Index.  I will also look at the binary variables of pericardial effusion presence and ‘Alive at 1 Year’ to determine the distribution.

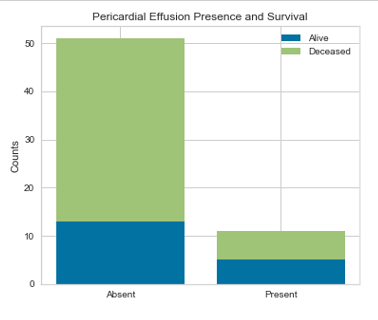




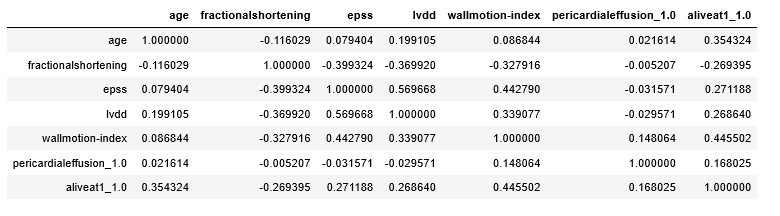
1. I checked for correlation using a Pearson Ranking chart using the Yellowbrick package and Rank2D command.



1. Finally, using the variables that appear to be correlated on the Pearson Ranking Chart, I created a stacked bar chart of my categorical variable to see the differences between the different variables and their relationship to survival.

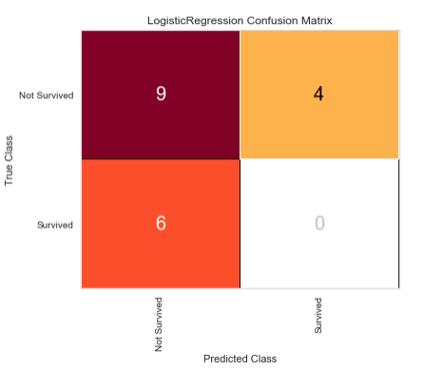


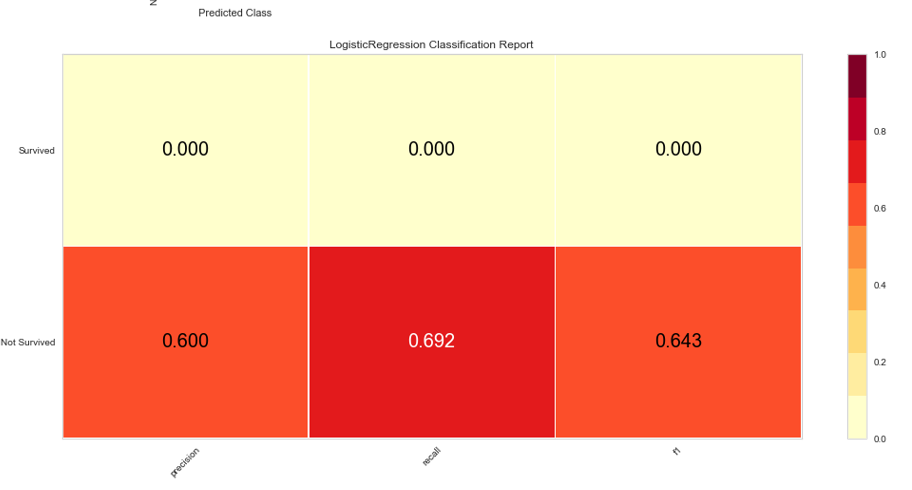
1. I encoded my two categorical variables (aliveat1 and pericardial effusion presence) with dummy variables using pd.get\_dummies.
2. I also checked a correlation matrix again for my newly refined numerical and dummy variables using.corr() to check for any high correlations.  The highest was 0.56 so I did not think any other highly correlated variables would be found.



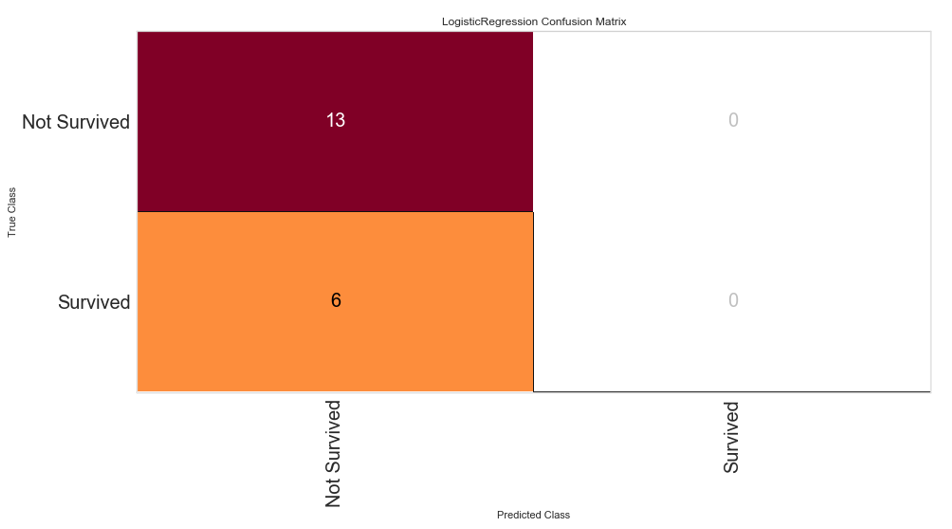
1. I set my target variable as Alive at 1 year, and then set all of the other variables as my features.
2. I standardized the numerical values to the same scale using sklearn’s StandardScaler.
3. Then I used select percentile and used the ANOVA F-classifier with percentile of 75% for feature selection.
4. I then used a Recursive Eliminating Features algorithm to determine the difference that it saw.  The regression method I used was a logistic regression since my target variable is categorical.  I set it using a StratifiedKFold cross-validation of 10 and scoring using neg mean squared error.  Even if I use accuracy, it still comes out with the same result.
5. I then used a Recursive Eliminating Features algorithm, but this time used a Random Forest Classifier as my regression algorithm.  There was no difference in negative mean squared error or accuracy used as a scoring metric.
6. Finally, I did the same Recursive Eliminating Features algorithm using an SVM model.

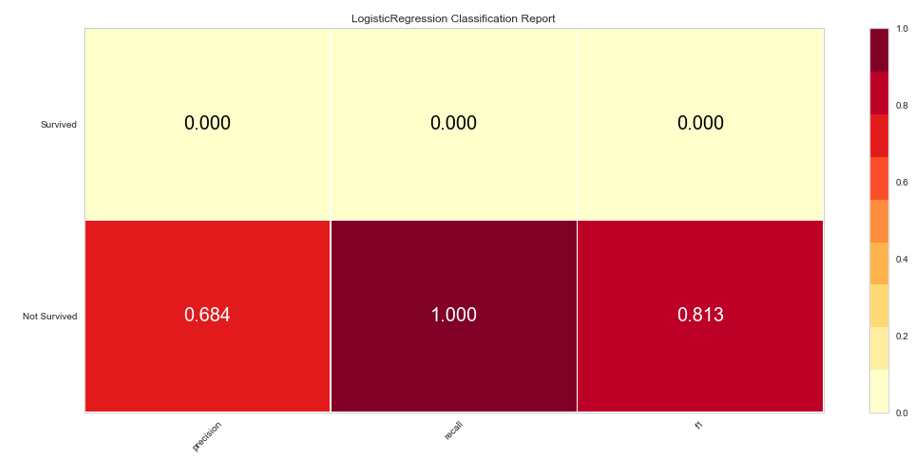
1. Using all features of my model I ran both a logistic regression and random forest classifier model on my data. I did this with K-fold stratified cross validation with # of splits of 10 and shuffle equal to True. I standardized the features, put them into my pipeline to get my cv\_results score and then using yellowbricks confusion matrix, classification report, and ROC AUC modules, evaluated the model’s performance. In this first set, the Random Forest Classifier model outperformed the logistic regression model.
2. Using All Features This Was the Result of the Logistic Regression Classifier:



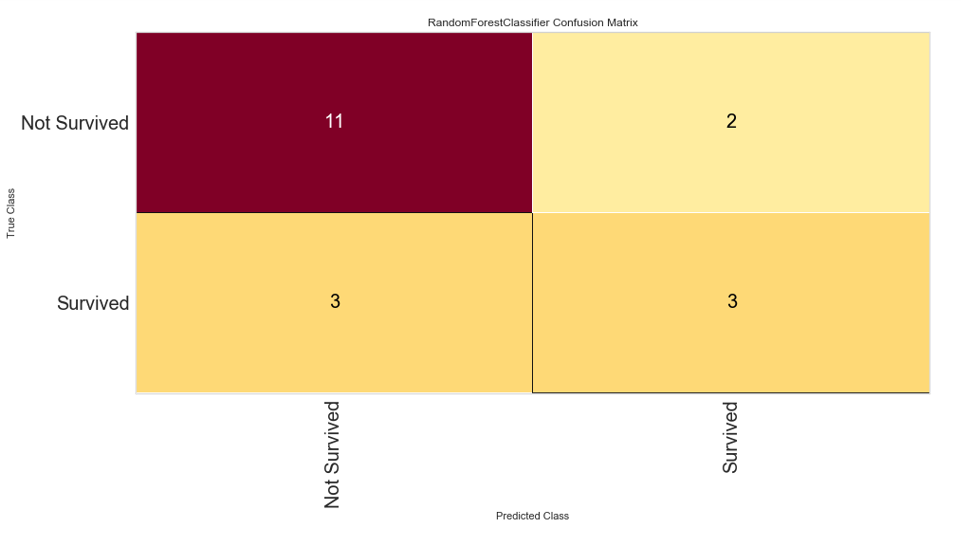


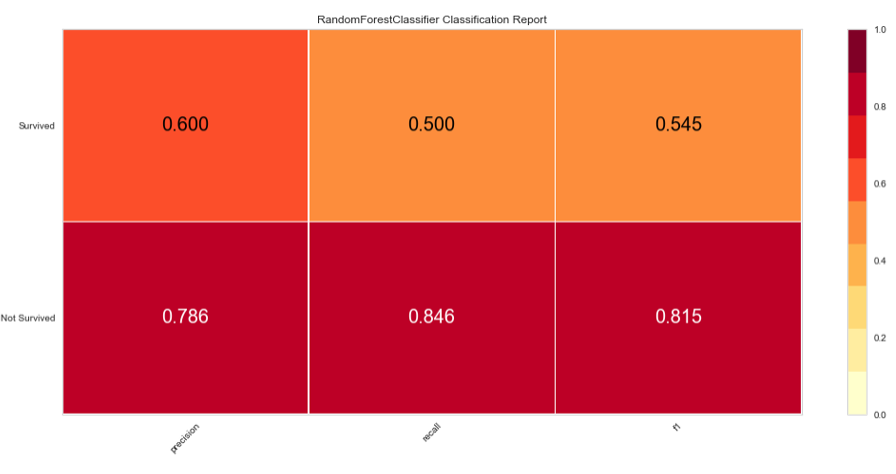
1. Below are the Logistic Regression Classifiers Using Feature Selection:



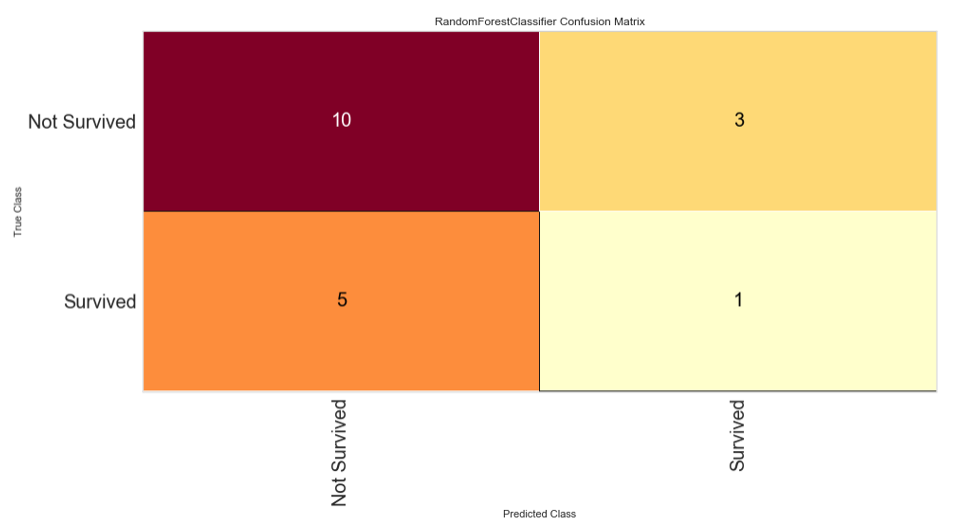


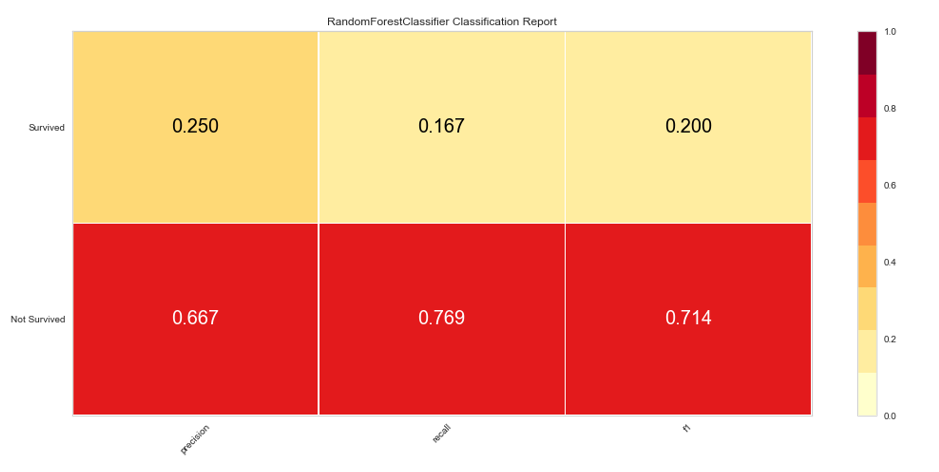
1. Using All Features, this was the Random Forest Classifier’s Results:



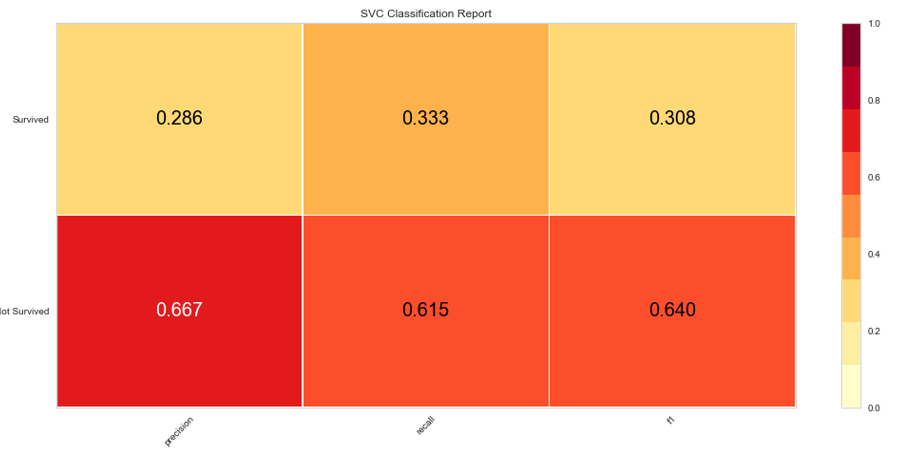
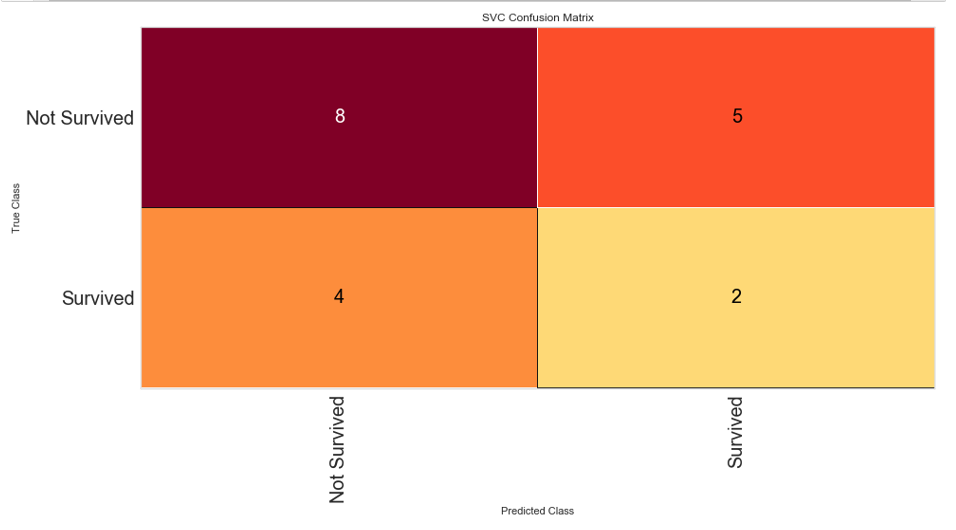


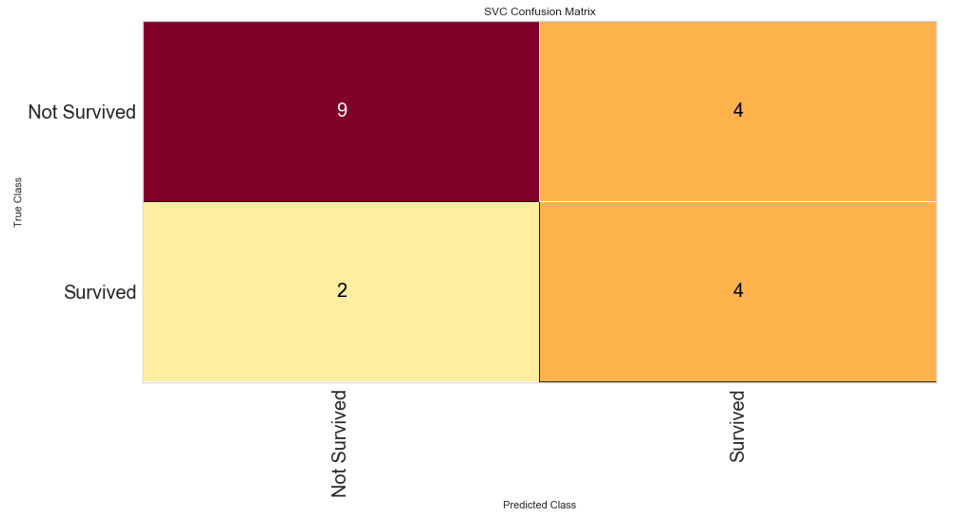
1. Using the Random Forest Model Using the Feature Selected Variables (excluding the pericardial effusion variable)

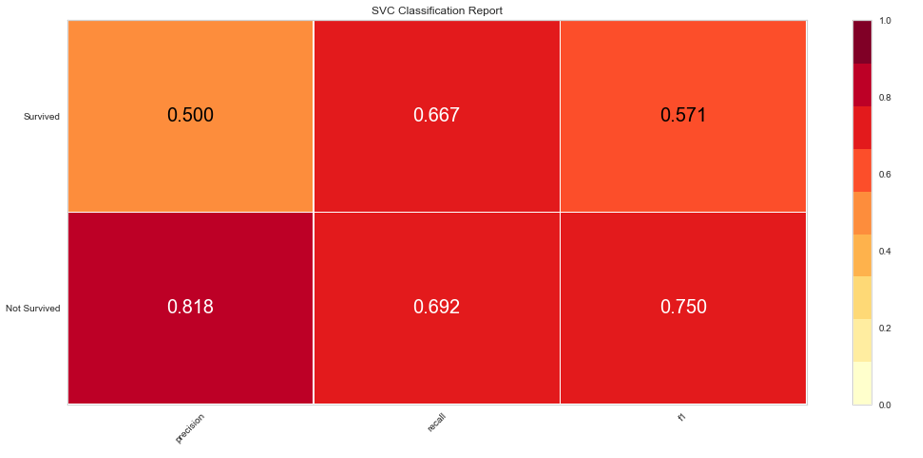




1. I also ran an SVM model as well to see its predictive accuracy. First is with all variables and the second is with feature selected variables (age, pericardial effusion, lvdd, and wall motion index).







23. One limitation of this dataset is the small sample size after we removed the null values. Another limitation of the dataset is that the deceased proportion of individuals was nearly two times those who survived in the dataset which will skew the results. The models tend to be better here at predicting not survived vs. survived. This is likely explained by the fact that our dataset had 2 times the amount of people deceased vs. survived so there was a smaller amount of data to help predict the survived categories.

I used three different methods, Logistic Regression, Random Forest Classification, and Support Vector Machine Classification. The best performing model was the Random Forest Classifier using all features. The worst performing model was the Logistic Regression classification model, even when using feature selection. It did not do well at predicting survival at all. The SVM model was better than the logistic regression model though not as good as the Random Forest Classifier model. Both the RFC and SVM models predicted not-survival fairly well but had approximately the same degree of success with predicting survival with the RFC slightly outperforming the SVM model.