**Willy Esquivel-Lopez**

**(wesquive - G01127937)**

**(miner2 user-name): Gopher123 – Best Public Score: .67**

**HW2 Report**

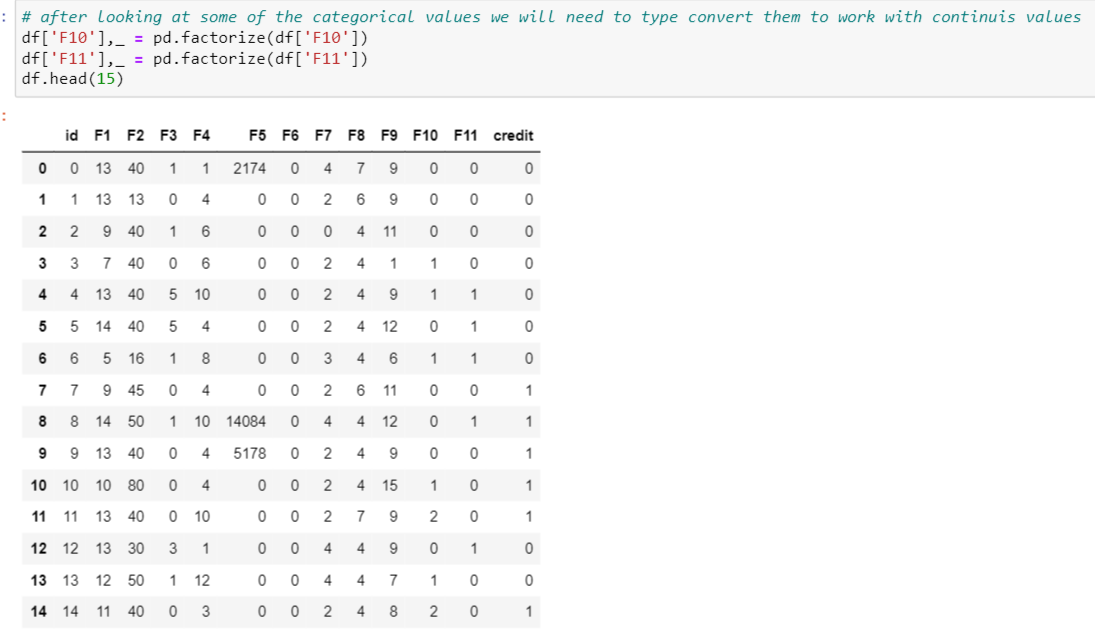
**My Approach:**

For This assignment I approached by looking at the data and seeing what I can do to decide the best model to use for classification. I saw a combination of categorical and continuous data, thus between SVMs and Decision trees, decision trees made more sense as it would be faster to compute, it would be less complex, and would give me more control over the classification.



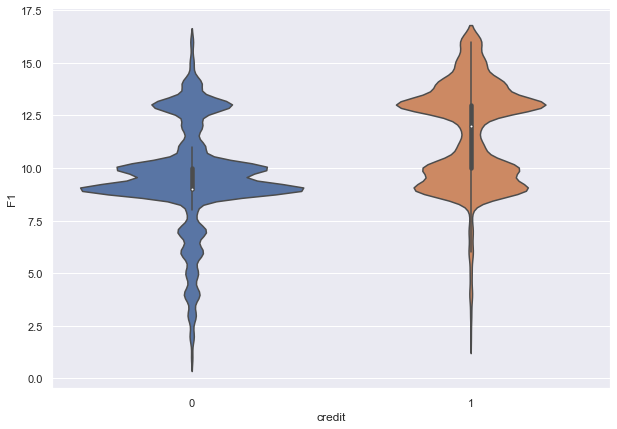
Figure - Train data unchanged

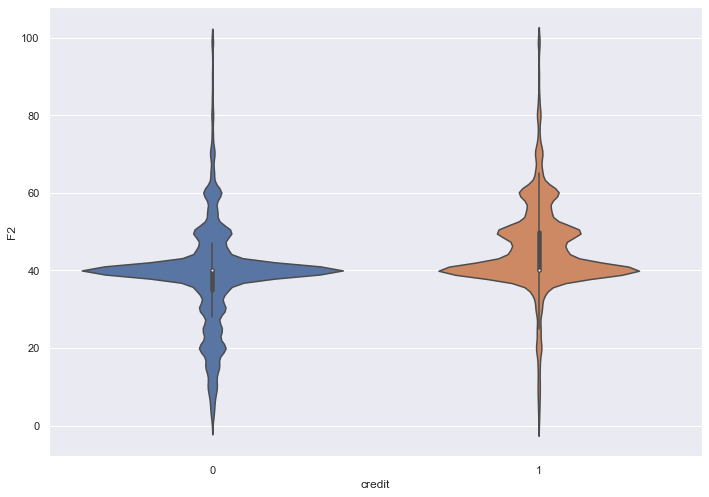
I had to modify two columns to continuous variable to simplify the tree

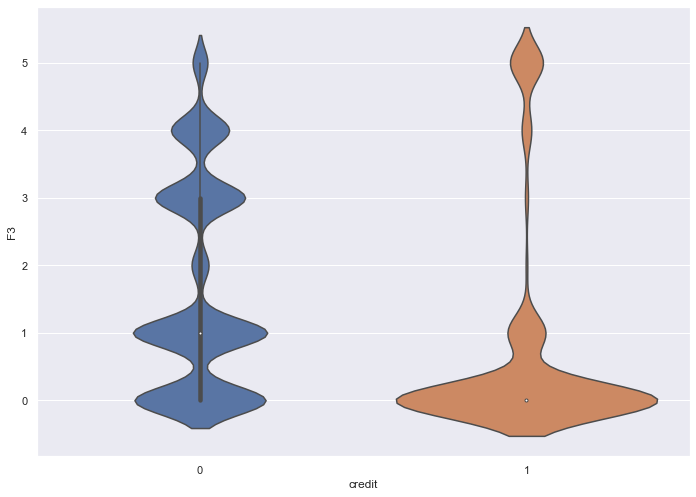


**Analysis:**

To Gather a clearer and broader picture of the distribution in the data I plotted violin plots for each feature against my target value (credit). There appeared to be some imbalances, l will explain how I solved that issue.



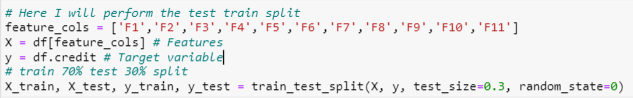


 etc. (The rest are on the notebook)

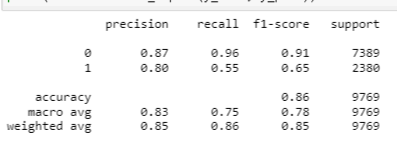
F3 had no label but appeared to be crucial to the outcome of the decision tree.

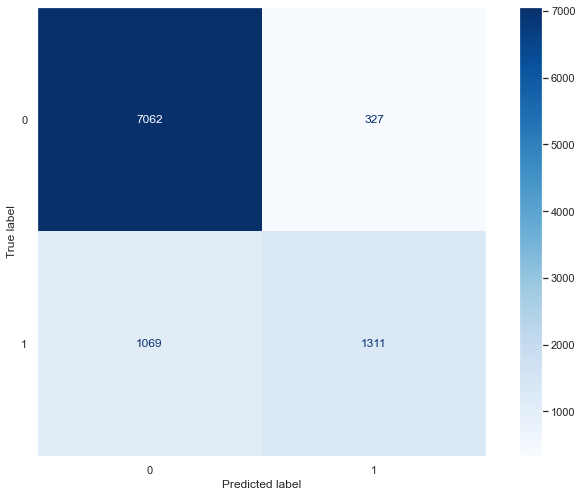
**The Training and Testing:**

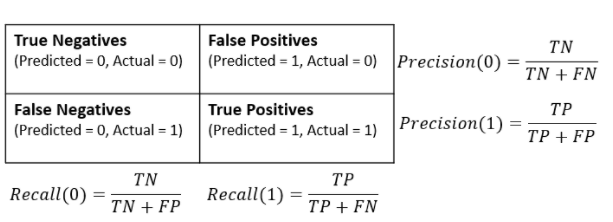
I Played it safe and used a 70-30 split for my models. Labeled the features and my target variables like so:



I decided to test the Gini Index on using a decision tree with max depth of 3, 5, and 7. All of which appeared to do fine but would not break on 85% accuracy. But the F1 score suffers regarding false negatives upon investigating the confusion matrix generated:



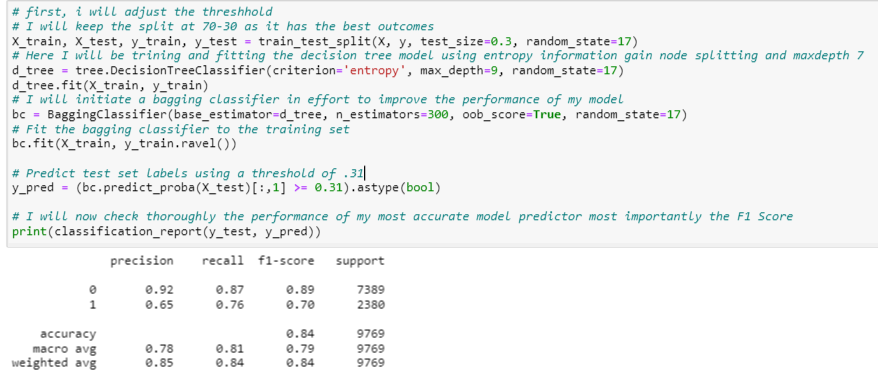




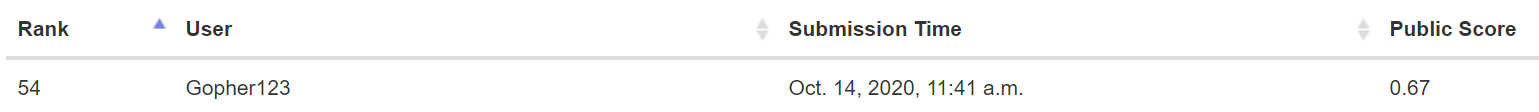
**Bagging and Predictive Threshold:**

I had to improve the decision tree model somehow, so I decided to use the Bagging technique to get a better performing model. I discovered with bagging the best performance results when using a decision tree of max\_depth=9 to create subsets of random replacement.

After doing some research it appears the best way to solve the misclassification errors resulting in false negatives from my model prediction was to limit the threshold on which the predictions are made. The major reason this was happening is due to the unbalance of the target classification. So, I tuned the threshold to .31 to compensate for the imbalances, like so:



Resulting in a best score of:



This approach increased my score form 62% to 67% by using bagging and a threshold delimiter to handle data imbalances and boosting the performance of my decision tree.