The best model from the result test found that the best parameters given into gridsearch in order to optimize a DecisionTree on the data we are experimenting with are the parameters class\_weight: None, max\_depth: 4, and max\_leaf\_nodes: 10. These parameters mean that when we create a DecisionTreeClassifier, we should create a model that has a maximum depth on the tree of 4, weight one on classes, and 10 max leaf nodes in best-first fashion. This model on the testing data led to results of ROC AUC: 0.8846, PRC AUC: 0.4972, PSS 0.5368, and F1 Score 0.6162. This score is not great but could almost certainly be improved through the use of RandomForest or another classification model like SVM. A decision tree is meant to be light-weight, so the trade off for its efficiency is a weaker accuracy. If we stacked up our decision tree with RandomForest, then the efficiency to create a best fit model would decrease, but its accuracy would likely increase.

The scores returned to my test model can be interpreted as follows. The ROC AUC of 0.8846 shows that the ratio between true positive rate and false positive rate is good. We are not making false positives at a rate high enough to decrease my ROC AUC any lower. A score of one here would lead to all true positives. This is not the case, though, because my model does not get every positive correct. My PRC AUC is not great because the recall is so high. At 0.4971, this number could be higher. But because my model is having to recall at such a high percent, my recall perfect is so middling. This could against be improved by a model that better classifies than does my DecisionTreeClassifier. My PSS score is 0.5368. The PSS is a precision measurement score that is calculated by examining the false positive to true positive ratio. My F1 Score is 0.6162. This F1 Score is harmonic mean that helps to consider the tradeoff in importance between recall and precision.

The model probabilities for negative and positive examples shows how probability impacts the negative and positive scoringg. We can see that a low probability is where our Nonfraudulent charges usually find themselves, whereas a higher probability is where positive for Fraudulent charges often are. More interesting insights can also be seen when examining this distribution in a boxplot. The negative values are polarized towards in polar ends of 0 and 1 such that they are largely concentrated towards those ends. We see, though, that positives fall largely in-between the negative distribution. These two graphs help us visually gain insight into how the classifier is able to determine the differences between positives and negatives on classification.

The best model from grid search is an improvement over the exploration model I built. For exploration I just randomly guessed at parameters to best fit my model. I came to choose these parameters based off of reading the documentation for the DecisionTreeClassifier. So I chose max\_depth = 200 and max\_leaf\_nodes = 40. This gave me the following precision scores. ROC AUC: 0.6634, PRC AUC: 0.4517, PSS: 0.5173, and F1 Score: 0.6279. This was worse than my optimized gridsearch parameters which gave me best fit parameters of class\_weight: None, max\_depth = 4, and max\_leaf\_nodes = 10. These parameters gave me precision scores of ROC AUC: 0.8846, PRC AUC: 0.4971, PSS: 0.5368, and F1 Score: 0.6162. My optimized F1 score is lower than the exploration F1 score, but all other metrics are improved on my optimized model.

My best model tree diagram is actually more simple to view than my exploration model because of the parameters passed into the initializer. The model is less deep because max\_depth is smaller. And we can see the frequency at which test data flows through the diagram through coloration. DecisionTrees work by literally creating a tree by which independent data flows to reach its dependent conclusion. We can see how DecisionTreeClassifier has concluded by examining the tree.