**Chapter 7**

**Results**

The results of this research are attained through the use of a decision tree style model present in the Sci-kit Learn module for Python. The reason for choosing to use a decision tree is based off of the benefits that are naturally present when using them. First, decision trees are easy to understand and the model itself can be fully graphically visualized such as in the sample section of our decision tree in Fig ## decision tree sample.



Fig ## decision tree sample

Second, decision trees are able to handle blank data very well, this is extremely important when considering our data, this is because even if a line does not contain a symbol from the portion of selected ASCII characters that we are analyzing such as in this sample:

*[0.020833333333333332, 0.020833333333333332, 0.0, 0.0, 0.041666666666666664, 0.020833333333333332, 0.0, 0.0, 0.020833333333333332, 0.0, 0.020833333333333332, 0.020833333333333332, 0.0, 0.0, 0.0625, 0.041666666666666664, 0.0, 0.020833333333333332, 0.041666666666666664, 0.0625, 0.020833333333333332, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5625, 0.0, 0.0, 0.0, 0.020833333333333332, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]*

It is still very relevant to our decision-making process. Third, because decision trees are a white box model, we can verify all of the decisions through either Boolean or mathematical approaches. Finally, when considering the immense size of source code in this day and age and the rapid rate at which source code is growing the fact that the prediction process is logarithmic is extremely important.

In order to ensure that we are receiving high quality and statistically sound results the decision tree was trained using stratified K-fold cross validation utilizing five folds. Initial tests were then preformed on smaller samples of 250 lines of true comments and 250 lines of commented out code to prevent a training bias based on a dominate class and then mathematically verified to ensure that the results were holding true. Afterwards the questions generated by the tree were checked against the initial findings of the research that they were all mathematically sound questions. For example, the first question at the root of the tree asks whether or not a line is composed of less than or equal to 26.7% spaces and if the statement is true then the sample is likely a comment. This is a highly reliable question as initial research indicated that on average comments are constructed of approximately 24% spaces while commented out code is constructed of approximately 33% spaces.

Once the integrity of both the initial results and the decision tree model have been verified it is time to move on to larger test runs to pin down the overall quality of this identification approach, and the results are very promising. Over a series of 10 tests randomly selecting 1000 lines of comments from the 20 different projects we show an accuracy of 96.5%, a precision of 97.6%, a recall of 94.3% and a F1 score of 96.6%. All of these results were calculated automatically using metrics from Sci-kit Learn.