Aaron Templeton

CS 5350 Machine Learning

Final Project Report – Competitive Project

Models

Decision Tree

I spent a lot of hours working on my decision tree implementation. My decision tree from the first assignment did not function correctly so I eventually ended up "fixing" my tree more or less. I believe now that it functions better, but it did not score as well as I would have liked. I made some initial submissions on Kaggle with my decision tree model from the first assignment, however, the score was lower than expected, so I did not continue to use that implementation.

Once I had fixed the decision tree I decided to try and run it again with no cross-fold validation to see what it would score. I also tried to fix the depth of my tree to certain heights to see if that would affect the accuracy in any particular way. I submitted a prediction with a depth of 2. My best submission for the decision tree is 50.12. it is a tree of depth only 2.

Perceptron

I found the perceptron model a bit easier to build than the decision tree. I made several submissions to Kaggle with the perceptron model; however, I believe that there is an error in my predictor for perceptron because the scores are pretty low. I made several submissions using the same hyperparameters that were used in the homework, which are, .01, .1, 1, 5 and ran it for 20 epochs in one case and 30 epochs in another. My best submission for perceptron was 30.2 with 30 epochs.

SVM

The SVM model was very similar in theory to the perceptron model and was easy to implement. I made a few submissions with SVM model, after, fixing the negative number to 0 instead of -1 like it was in the homework, it scored similarly to perceptron. Based on class lectures I believe that this is what was supposed to be expected for SVM and perceptron. It was still lower than what I wanted it to score. My best submission for SVM was 35.5

Naïve Bayes

The Naïve Bayes classifier was probably the most difficult to understand for me. In the end I think that I implemented it correctly, but it always predicts positive and I couldn't figure out what the issue was with the prediction or the probabilities in the model. I tried to dedicate as much time as I could to the implementation on this model, but I needed more. I made a few submissions the Bayesian classifier on Kaggle using cross-validation techniques and data discretization. My best submission for Naïve Bayes was ~50.

Random Forest

Once I fixed my decision tree for homework 5 the Random Forest model was a simple implementation. Although I used cross-validation in the homework assignment I decided that for the project it would take too long to cross validation to find the best forest size so I hard coded some values for the size and tested the accuracy on some different sizes. Best submission for Random Forest was approximately 50 for a forest of size 10.

Adaboost from Scikit-learn

Using a third-party library for one of the Kaggle submissions was something I enjoyed. It was a change of pace. I decided to test for the Adaboost model simply because some classmates I have spoken with advised that I use it since they received good results with it. Scikit-learn made it simple to use. All I had to do was give it the data in libsvm format, build the classifier and get the predictions. It was pretty straight forward. My best submission for Adaboost was 82.2.

Ideas Explored

I tested several ideas from class in the project. The main ideas and concepts that I used from the lectures were, cross-validation, hyper parameter testing, data transformation and discretization, random seeds and bias folding on some models. I mostly used the models that were directly given in the homework and just altered the code to work with the Kaggle data. I also made use of the data transformation files that were given in the homework's for the libsvm data and the csv data. It assisted in creating the data into a workable state so the model could learn and test on it.

Conclusion

The main takeaways from the project for me are that machine learning is difficult. I can understand the concepts of machine learning and understand how the models/classifiers work in each case, however, I am not very good when it comes to implementation of machine learning models. I sometimes found it difficult to understand how to translate the concepts to code, especially in python, since I have never used it before this class. I have learned that I am an amateur at machine learning and it takes a lot

of practice and trial and error to get these models performing well. When using Scikit-learn, I was able to make a well performing Adaboost classifier without any problems. If given more time, I would have liked to improve my models and continue to learn from my mistakes.

This class had a high learning curve, but I feel that I have walked away with an in depth understanding of what machine learning is and how to do it. It was a difficult class to keep on top of due to the hours of debugging and trying to understand how to implement the models.

Comment to Evaluators:

I hope that there can be some leniency in grading of the project. I personally found it difficult to do the project and the homework together, especially towards the end of the semester. I am taking 4 CS courses and 1 math 5000 level course. Take 5 courses was a difficult decision and it didn't allow me to use as much time as I would have liked to towards this course. I also work 30+ hours a week while attending school full-time. I greatly appreciate all the help from the TA's and professor with this course.