1: Overview

For my final project I am competing in the Kaggle competitive project. I currently have a **0.63699** classification accuracy on Kaggle. I have multiple strategies I will use to improve this score, as there isn't a huge margin between my score and the baseline. The baseline for this competition seems to be right around 0.5. To discover this I made a submission that labels everything as positive and received about a 0.5 classification accuracy.

2: Approach 1

My first approach was to train the classic perceptron algorithm with the training examples and use it to classify the test examples. This was done with no pre-processing of the data. I performed cross-validation on the testing set to determine the best *epoch* and *learning rate* to use with the perceptron algorithm.

I found an epoch of 5 and a learning rate of 0.1 to be performing slightly better than other choices, but not my much. This approach gave me a classification accuracy on Kaggle of about **0.55**.

3: Exploring the Data

After my first approach coming up with undesirable results, I began to explore the data.

Each example is defined by *360* possible features. But I discovered only *130* of these features appear in the training examples. Of these *130* there are a great deal of them that appear in nearly every training example and four features that appear in every training example. The *65* features with the highest frequency appear in at least half of the training examples.

Another discovery was the values a feature can hold vary greatly. I collected the minimum value, maximum value, range, and average value each feature can hold.

Additionally, the value ranges of each feature, when compared to the other features, showed a large variety. I decided to discretize the feature values into buckets for my second approach.

4: Approach 2

With the values of each feature put into buckets, I decided to try out training a decision tree using the ID3 algorithm. This lead to an undesirable classification accuracy on the Kaggle test examples though.

I decided to return to using a linear classifier. Running trails using the classic perceptron, margin perceptron, and the aggressive margin perceptron. I saw the best results with the margin perceptron so decided to move forward with it.

The margin perceptron is what I used to accomplish my current Kaggle classification accuracy of **0.63699**. I performed cross-validation to decide an *epoch* of 7, a *learning rate* of 0.1, and feature values discretized into one of 20 bins.

3: Future Plans

As I complete homework assignment five my next approach will be to use a support vector machine (SVM). I saw decent results with the margin perceptron and am optimistic about using an SVM the more I learn about them. As will all my attempted approaches thus far, I will be performing cross-validation to determine the best hyper parameters.

I didn't experience good results by training a single decision tree. That said, I will experiment with using an ensemble of small decision trees. I am in the process of implementing this for homework five and will test it for this project once I finish the implementation.