Decision Tree Project

Brandon Maness

**Introduction**

This assignment attempts to give a diagnosis of diabetes in a patient based on 8 tests. I wish this assignment was more fully-fleshed, but the dataset is within one specific set of individuals, the female Pima Indians, so this analysis is only accurate within their genetic sets.

**Method**

The model used in the prediction is a decision tree, which is well suited for the small feature space of the data. Not a lot of pre-processing was needed as the data was already very clean. I did notice that the pre-defined columns left out the ‘skin’ variable, I’m not sure the reasoning for that but I left it as it was there for some reason. Using helper functions, I split the data into true and false rows to match the ‘question’ then partitioned. I had to compute the Gini impurity to get the entropy, then use the entropy to gauge information gain to see how well the model would perform. Then I used the partition data rows to build a branched tree

**Experiment**

**Data Description**

This dataset was pulled from a Kaggle competition. The data consists of diagnostic results of Pima Indian woman, a Native American tribe. The variables are as follows:   
**Pregnant**: Number of pregnancies the patient has had   
**Insulin**: Insulin level at time of diagnostic   
**BMI**: Body Mass Index   
**Age**: Age at time of diagnostic   
**Glucose**: Glucose levels from oral test   
**BP**: Blood Pressure at time of diagnostic   
**Pedigree**: Genetic diabetic history   
**Skin**: Thickness of the skin on the triceps (omitted for some reason)  
**Outcome**: 1 if diabetic, 0 if non-diabetic

**Results**

The gini impurity of my set was 0.4544, I’m not a huge fan of such a high likelihood, but I feel like with more features and a larger sample space, this could be minimized. Entropy came out to 0.9331 which is fantastic, the dataset was very pure and had very low variance. The end prediction tests proved 100% accurate, although they were single class tests. This is still a success in my book as the tree works properly.

**Observations/Analysis**

I noticed we were given two info gain functions. I ran both in my split. While entropy came out to a higher value, after some research it seems they’re basically the same but entropy is more computationally intensive, so I stuck with the gini info gain.

**Conclusion**

I was very pleased with the performance of the model and was very surprised at how easy it was to implement the tree structure. I have a much better grip of the concepts of info gain, entropy, and the backbone structure that is necessary to produce the branching in the tree.