Daniel Sullivan

Homework 3: SVM vs Decision Trees

Homework #3

- Read the following articles:
  - https://www.hindawi.com/journals/complexity/2021/5550344/
  - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8137961/
- Search for academic content (at least 3 articles) that compare the use of decision trees vs SVMs in your current area of expertise.
- Perform an analysis of the dataset used in Homework #2 using the SVM algorithm.
- Compare the results with the results from previous homework.
- Answer questions, such as:
  - Which algorithm is recommended to get more accurate results?
  - Is it better for classification or regression scenarios?
  - Do you agree with the recommendations?
  - Why?

Format

1. Essay (minimum 500 word document)
   Write a short essay explaining your selection of algorithms and how they relate to the data and what you are trying to do
2. Analysis using R or Python (submit code + errors + analysis as notebook or copy/paste to document)
   Include analysis R (or Python) code.

Introduction:

In homework 2 we tackled the use of decision trees within classification and even implemented ensemble methods(adaboost) in order to classify a data set. In this homework we build on that work and implement support vector machines(SVM) to approach the same classification problem. The data set I was working on was the Pokémon pokedex data set which contains every Pokémon, their types, generation, stats and whether they are legendary. I used this data set to try and train an algorithm that could predict whether or not a Pokémon is legendary. This data set is rather small with ~1000 entries and for the legendary Pokémon the data set is seriously imbalanced with only about 100 legendary Pokémon. The two articles Suggested to be tied to the homework this week brought up some very interesting points that were applicable to my dataset. Guhathakurata et.al (A novel approach to predict COVID-19 using support vector machine - PMC (nih.gov)) looked to tackle covid infection and hospitizalation rates using SVM's and Safi et al (Decision Tree Ensembles to Predict Coronavirus Disease 2019 Infection: A Comparative Study (hindawi.com)) Tackled the same issue using ensemble decision tree methods both to favorable outcomes. The SVM article worked with ~200 data points each with 8 independent variables and with about a 22:10:8 ratio for no-infection: mild infection: severe infection. The decision tree paper looked at solely predicting positive cases but had a much bigger imbalance of about 520:80 with 18 independent variables. I found both of these approaches very useful and brought up some key arguments for imbalanced data sets like mine, like not using accuracy due to the imbalance skewing the metric and how well both of these performed. Based on these studies we see the best route of analysis for my data set. SVM methods typically work better with higher dimensionality, fewer data

points, and more clearly separated data. Ensemble decision tree methods tend to work more efficiently with imbalances due to the bootstrapping involved and can work with bigger data that is less separated. The SVM covid data set shows a much less major imbalance while the decision tree data set has a large one similar to the Pokémon (my) data set. Because of this imbalance and because my data set is much larger then both of these I find that the best algorithm to implement in this instance is the ensemble decision tree method.

Model assessment:

Because I am working with an imbalanced data set I need to use the balanced accuracy since true accuracy will be skewed with about 90% of the data belonging to the non-legendary camp. The balanced accuracy is equal to (sensitivity + specificity)/2. In homework two I ran three types of models, a decision tree, a random forest, and then an adaboost algorithm. These scored fairly well with a balanced accuracy of ~.75 and ~.88 for all variables and restricted stats respectively while random forest scored a .83 and the adaboost scored a 0.90 both using all variables. The SVM algorithms scored significantly lower under multiple different kernel types ranging from 0.69-0.78 with linear being the best-balanced accuracy. When I attempted to select variables with the best separation the balanced accuracy dropped with values around 0.58 however this may be because of the loss dimensions. Altogether for this data set I found that the benefits of the decision tree methods made the most sense from the recommendation painted by the articles and the data reflected that. There was a clear winner in the ensemble method adaboost and for the most part even the weakest decision tree matched the best SVM model based on balanced accuracy.