Daniel Sullivan

Homework #1

Pre-work

1. Visit the following website and explore the range of sizes of this dataset (from 100 to 5 million records):
   https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/ or (new) https://www.kaggle.com/datasets
2. Select 2 files to download
   Based on your computer's capabilities (memory, CPU), select 2 files you can handle (recommended one small, one large)
3. Download the files
4. Review the structure and content of the tables, and think about the data sets (structure, size, dependencies, labels, etc)
5. Consider the similarities and differences in the two data sets you have downloaded
6. Think about how to analyze and predict an outcome based on the datasets available
7. Based on the data you have, think which two machine learning algorithms presented so far could be used to analyze the data

Deliverable

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. Exploratory Analysis using R or Python (submit code + errors + analysis as notebook or copy/paste to document)
   Explore how to analyze and predict an outcome based on the data available. This will be an exploratory exercise, so feel free to show errors and warnings that raise during the analysis. Test the code with both datasets selected and compare the results.

Answer questions such as:

1. Are the columns of your data correlated?
2. Are there labels in your data? Did that impact your choice of algorithm?
3. What are the pros and cons of each algorithm you selected?
4. How your choice of algorithm relates to the datasets (was your choice of algorithm impacted by the datasets you chose)?
5. Which result will you trust if you need to make a business decision?
6. Do you think an analysis could be prone to errors when using too much data, or when using the least amount possible?
7. How does the analysis between data sets compare?

The two data sets I chose were the air quality data set shown here: Local Air Quality in North Carolina | Kaggle and a data set containing every known Pokémon up until the most recent Pokémon release this last fall found here: Pokemon Stats | Kaggle.

Air quality:

The Air quality data set contains over 11000 data points collected over the span of September 2018 to November 2019 so a little over one year in chapel hill north Carolina. The dataset contains 22 columns per entry and these are latitude, longitude, last read, temperature, humidity, pressure, site label, inside/outside, uptime, wifi signal, whether there were hardware issues, age of data, and the air particulate ppm of the current read and then each 10min, 30min, 1hr, 6hr, 24hr, and 1 week averages for each reading. For this data set I wanted to see if I could trim down the dataset and see if I could predict the month the reeding took place in when presented with the ppm readings. Because I wanted to avoid the curse of dimensionality and an dips in accuracy I decided to drop the data points below.

inside/outside, site label, Latitude, and longitude: these data points were dropped due to having the same value entered for every observation since they are constant I did not deem them too critical to the analysis.

Wifi signal: I excluded this variable because while it may pose significant for the overall study the WiFi signal for the most part remains constant and should not interfere with any actual measurements just the rate at which the data can be downloaded or uploaded from the device.

Up time and age of data: I didn't get exactly why these data points were included because if this was an ongoing time trial these metrics don't hold too much value sp were excluded.

Hardware issues: While this is a clear indication of an issue with the machine I didn't find any particular patterns with the data. Also the study only lists error codes taking place at the time and while this is useful it is a variable that for most of the data was entered as NA and I have no way to tell what each code means. For this reason I dropped the variable

From the remaining values I broke out the date time format into months for each individual time point and proceeded. Because I wanted to do a classification problem trying to select for the month two methods jumped out at me, gradient boosting and using decision trees in a random forest. The reason I implemented these two algorithms was that the data for all ppm measurements showed two clear groupings starting around 10/19 where a new pattern was developed. Additionally most of the ppm measurements follow what seems to be a cyclical pattern throughout each day as a sort of wave. Because of this I don't think that many of the parametric methods such as linear/logistic regression would do well as the data does not meet many of the assumptions needed. One additional thing I noticed in regard to the wave pattern found in the data was that while this existed across all ppm measurements the non-averaged measurement had the most variance and was significantly higher however as the averaged time got bigger the variance dropped and the pattern was more clear so the 24hr and one week time points showed the pattern more.

Correlation is very prevalent in this data set which is a bit expected as the ppm reader is taking averages some of the data gathered for the hour and 6 hour time point average probably over lap and

thus give a very strong correlation. The exception to this is the current ppm reading has almost no correlation with any of the time points. Additionally as the difference between the time averaged data increases the less correlated the value is i.e. the 10 min time point is more correlated to the 1hr time point then the 24 hr time point. This could be a significant hindarence as most algorithms do not perform nearly as well when there is high levels of correlation between variables.

Pokémon:

My second data set I collected was the complete Pokémon pokedex from Kaggle. Each entry pertained to one pokemon and the dataframe had the number of the Pokémon, the generation, the primary and secondary types, health points, attack, defense, special attack (sp attack), special defense (sp defense), speed, the sum of all stats, and whether it was legendary or not. All of this was very well documented and with minor errors aside there wasn't a single value missing. Knowing a little bit of the background about Pokémon I decided to construct a model that would classify a Pokémon's primary type. Generally, each Pokémon type has specific stats it gravitates towards. Ghost, fairy and psychic typically have higher sp attack and sp defense, rock and steel favor physical attack and physical defense, while electric and flying typically are the speediest Pokémon. In order to tackle this question I tried to implement two separate methods. First, I tried to classify based on KNN and then implemented a gradient boosted method. I chose these because I was expecting similar types to be in general proximity to each other in the case of KNN and after that knew that gradient boosting is a good method for non-binary classification and that there may be too much variability in the data for many of the regression methods as well as too many variables.

This data hardly had to be cleaned and the only modification I made to the data frame were fixing a few mistakes and dropping the secondary type and legendary fields. As far as correlation is concerned the only stats that have considerable correlation is the stat sum. This is expected since as the total goes up for each Pokémon each stat should also increase. Aside from that the other data points were not heavily correlated with only slight correlation.

Pros and cons of Algorithms:

Decision Trees/Random Forest:

One of the most advantageous aspects of using decision trees is that these algorithms tend to be very flexible and can classify well when the data does not follow any assumption needed for parametric methods and that is one reason I chose it for the air quality data set. However one reason why I was concerned about using decision trees was that the tend to have very low accuracy. This is largely explained by how they are prone to overfitting and to off set this need to stay small and therefore can be too general when classifying. One way I circumvented this was by deploying the decision trees in a random forest. This is an ensemble method that uses many small trees to increase the accuracy. This trade off does make the model less interpretable but that wont really matter too

much for our use. This method does have some cons, in most cases it has restricted use for regression, it can hyper focus on categorical variables, and can be computationally intensive and slow. However where we have a moderately sized data set, no categorical variables and are not looking for a regression this model works very well for our data set.

K Nearest Neighbor:

KNN analysis is a “lazy learner” algorithm which uses nearby points to categorize the new data. It is a very simple algorithm with high predictive power and works very well on both small and large datasets. It does not rely on assumptions going into the training like many of the parametric making it highly flexible and works best with non-linear data. However, It does have many issues with dimensionality. As more variables are added the computational load grows exponentially and accuracy can drop. With our Pokémon data set this is not a massive issue since the data set is small with only about one thousand entries even while using multiple variables.

Gradient Boosting:

Gradient boosting is an ensemble method that uses many week learners sequentially. It is extraordinarily robust and dramatically increases the accuracy of decision trees. It works for both classification and regression modeling however is computationally intensive and typically performs better with larger amounts of data. One major flaw of this type of algorithm though is that it is prone to over emphasizing ouliers and the presence of outliers can severely skew accuracy.

<u>Conclusions:</u>

Ultimately most of the models performed exceedingly well on the datasets. For the airquality we were able to get a month prediction accuracy of 68% through gradient boosting and 74% using the random forest algorithm. For the Pokémon data set the primary type classification had an accuracy of 20% for KNN and 99.5% for gradient boosting. If I were to use any of these models in a business setting I would definitely lean more on the random forest air quality model and the gradient boosted Pokémon model. This clearly had the best performance of all of the models showing the best accuracy for their respective data sets. I would imagine that the KNN model really struggled with the multiple dimensions of the pokedex and I would not use this model however while the random forest model beat out the gradient boosted model for air quality both are still very reliable models.