Nick Kriz

Professor Tom

12/12/2020

Final Machine Learning Report

Throughout the completion of this project, I encountered many difficulties such as parsing through text, changing variables to categorical variables in order for the machine to understand the variables as numerical values that it can use to predict future stars for a business. Additionally, I faced many difficulties with needing to type cast certain variables to make them more compatible for the machine such as changing the star values to ints rather than doubles since for some reason I could not find a way for the machine to accept the variable type that the program read stars in as. In addition, I faced issues with needing to ensure that the dataset that the machine was learning was compatible with the dataset that the machine would be used to predict. However, after all was done, I noticed that my prediction was not as accurate as I had hoped, however I learned more about machine learning and using certain models than I ever thought I could. And the best part was that I needed to figure more of the problems out by myself.

When considering reading the dataset into the machine, there was a lot that needed to be done once I read over what the json file looked like. I read in the data using the example you gave in class, however, there was a lot more I wanted to do and try. I first created the df in my program using the append function that helps with reading in json lines. I wanted to include things that I knew would matter in predict what the star rating of a business would be. This included things like postal code, state, business category type, etc. I decided that coordinates would present similar findings to the state and zip code, so I decided to leave it out. I also left out the address and city for the same reasons. I remember that you had mentioned we could use the city to access google and check its surroundings to learn more, however I found this to be too difficult at my level, so I excluded it from the data. The last thing I left out was the attributes since I found that too many were very different and wouldn’t result in very accurate predictions if these were considered for the data. In terms of handling the states and categories, I decided to create a categorical variable by using the Categorical() function in python. This allowed me to add these newly created variables to a new column for the dataset. I called this category type and state code. This also made sure that if there were any that repeated or where associated with the same name, they would equal the same value. Next, I decided to handle the hours data. This was by far the most complicated part for me as it caused many obstacles that I was determined to get through. I attempted to use isin functions and str.find functions, as well as trying to use a OneHotEncoder to try and encode the categories of hours into numerical values. Once I realized this would present some issues, I thought it might be beneficial to know if the company was open on a Saturday as this would surely result in higher ratings for the business. I used a for loop and the function list() to change the dictionary ‘hours’ into a list in which I could more easily examine the keys that are associated with ‘hours’. I used an if statement to set a new column of values to 1 if Saturday if present in the hours or to a 0 if there was no mentioning of a Saturday in the hours of the business. I then deleted the rest of the unusable data from the dataset so I could make it more visually appealing and easier to work with. Lastly, I wanted to make sure the machine could understand all values as it originally had difficulty learning doubles. So, I used a type cast method to change the column ‘stars’ in the dataset to be all values that are of the int type. Next, I set the target variable equal to the star ratings and then the data equal to the remaining data in the dataset.

I used the kfold analysis that we learned in class to train to different models using the machine. However, prior to this, I made sure to include the sklearn metrics and every other sklearn that would be used throughout the program to test the accuracy of the model and give us the confusion matrices. In this for loop that was used on the kfold object that I created using the associated data. Due to my inability to make the incoming data compatible with the previous data used to train the machine. I decided it would be best to reduce the size of data being used, therefore making each dataset more comparable. Additionally, I wanted to ensure that the incoming data could be predicted just as well as the data was learned. This meant that I would use the similar for loop earlier to read this data in and change it to proper formatting that can be used to predict more accurate star ratings. I used accuracy scores to check the accuracy of my prediction which results in 0.26. This wasn’t as high as I had hoped however considering that amount of data I used vs the amount that I actually wanted to use, it seems fairly good to me. In other words, I would have liked to include the length of hours on a weekend day, specifics of where the business was located, etc. but these were variables that made it too difficult for the program to work properly. I also used a confusion matrix to show how well my machine was able to predict using the associated data. I noticed that the confusion matrix for the logistical model showed that there were noticeable errors in my data that I attributed to leaving out other data values that occurred later in the data file. These values were left out as I said earlier due to making the dataset compatible with the dataset being tested by the program.

I decided to use a logistical regression and a linear regression to predict using these two types of models. As expected, I found that the logistical model was far more accurate than the linear model. In the linear model, it caused the incoming data values to be plotted on a linear model that caused the star values to replicate that meaning they progressively increased as it went through the 90 businesses being tested. However, when I looked at the data involved with the logistical model, it presented me with fairly accurate findings. First, it was less continuous and linear like the linear model. This meant that it was more likely to be accurate than the linear model as it should not be so that the corresponding star ratings move in a linear way with the number of businesses increasing. The results of the prediction using the logistical model return star ratings that are whole numbers as it should have since this was the type of data that was trained in the model. However, the linear model gives us a result that has star values including decimals meaning that data doesn’t make too much sense when compared to what we inputted. Despite this, the logistic model also gave us much more accurate data as we can look at one business more specifically. A business that was given a higher star rating in the prediction was found to have a zip code that places it in more of an urban area. This reflects that data used to train the model as a business that was listed with a more urban zip code had a higher star rating. Furthermore, the logistical model that I used should be considered good due to the fact that it presented very good findings from the confusion matrix. Although there were notable errors that can be seen the random high values, the majority of them are very stable and show that our model resulted in findings that could be seen as more accurate. Especially when compared to the linear model, my logistical model gives predicted star values that are more accurate due to the non-linear nature of these values.