Rishabhsinh Virpura

12/13/2023

STAT 4185: Final Project Analysis

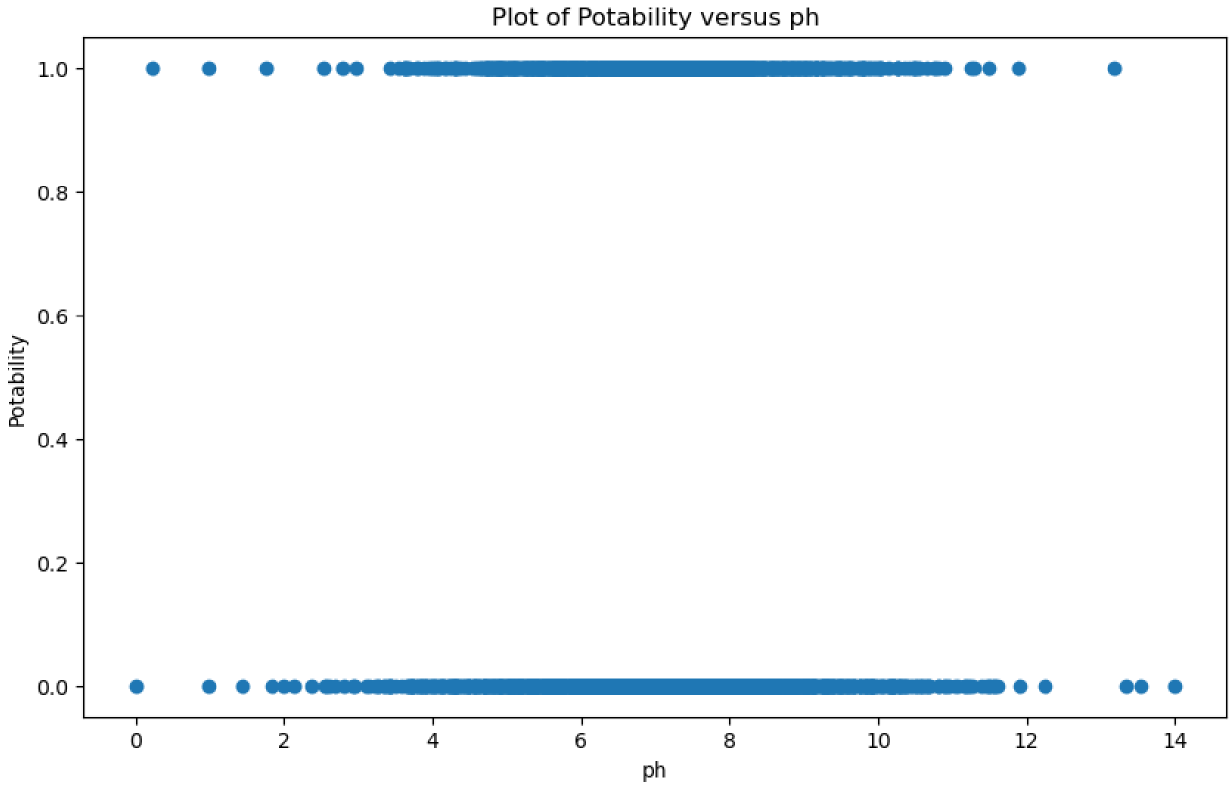
In this analysis, I am going to go over my process for what I did in my jupyter notebook, and how I decided to analyze this dataset.

First of all, I would like to re-state my goal for this project. Given a water potability dataset I found from Kaggle (that is included in my github report and in my final project proposal), my goal was to make an ML model that could predict whether or not a given sample of water was potable. The dataset included 10 features for the sample, including ph, hardness, amount of sulfate, etc. The final feature is potability which is a binary output of either 1 or 0 to indicate whether the water is potable or not.

At the beginning I mostly imported packages I planned on using, and I turned the csv file I am using into a pandas dataframe. Then, I checked and saw how many null values were in the dataframe. I noted that only 3 features had null values, and the potability column did not have any null values at all. I likely would have gotten rid of any rows that had a null value for potability (since there is no use in imputing the feature we are trying to predict), but luckily I didn't have to. I then deleted any rows that had two or more missing values. I wanted some sort of cutoff for when a row was missing too much data, and I decided this was the best solution that did not delete too many rows.

Finally, I finished cleaning up my dataset by imputing any remaining missing values with the median of the column it is from. For example, if there was still a remaining ph value missing, I would have imputed it with the median of all ph values in the row. I decided to input with the median instead of the mean, because it gives the data a better normal distribution. One visualization I included was the distribution of the sulfate column after imputing with the mean, and the distribution of the same column after imputing with the median. They ended up being very similar, since their values were actually very similar. The distribution was not actually impacted too much, and they both had a roughly normal distribution.

After this, I included a couple of scatterplots. The x axis was one of the features of the dataset, and the y axis was the potability. The goal was to see if any outliers in the feature would affect the potability. I expected to see some kind of division in values. For example, if the feature was below a certain value, then the potability may largely be 0. And if the feature was above a certain value, then the potability would largely be 1. However, the results surprised me. I made several scatterplots like this, but the spread between potability values that were 0 and potability values that were 1 always seemed to be equal. Here is one of the scatterplots I made:



In this figure, I would have expected water potability to be 0 for ph values that were outliers, and the water potability to be 1 around the middle ph values. However, both the potability values that are 0 and the potability values that are 1 seem to have an even spread. I think this mostly shows that in a dataset that has as many as 9 features, one feature alone will not decide the water potability.

After this, I split and trained my dataset. I split my training data into 80% of the dataset and the test data into 20% of the dataset. I chose this ratio because this is a smaller dataset that only has a few thousand rows. So, I thought it would be more important to have more training data to make the model more accurate, rather than having more testing data. I also used the standardscaler to scale the dataset, because I could not think of a need to use any other more specific method. I scaled every column in the X dataset, since none of them were categorical.

Then, I finally created the model I would be training. I decided to use the random forest model, since I am the most familiar with that model from in class examples and assignments. I also tried to use adaboost initially, but the random forest model seemed to be working better, so I decided to move forward with that one. After I created the model and had it predict the outcome of the test data, I printed the accuracy, precision, recall, and f1 score. They were 0.7, 0.7, 0.42, and 0.5 respectively. Although it was fairly accurate and precise, the recall and f1 scores were a bit lackluster. So, I tried to tune some hyperparameters as the last step in my project. I used gridsearchCV and tried to tune the following parameters: n_estimators, max_depth, min_samples_split, min_smaples_leaf, and bootstrap. I used these hyperparameters after researching online which ones were the best to test, and looking at previous examples in class.

After tuning the parameters, however, the scores did not seem to improve much. I tried a few things to improve the metrics, and they worked to some effect. This analysis is the result of me going through and tweaking my process over time to improve my accuracy, precision, recall, and f1 scores. For example, I originally used ada boost instead of random forest. But random forest seemed to be working much better, so I switched the type of model I was using. I also originally did not delete any data when I was working on the project. But I noticed my model was working better if I was more selective on the data I was imputing, and outright deleted rows that were missing too much data. Those were the main ways I tuned my model, but I also think that if I had more time to mess around with the hyperparameters that it would work better. Unfortunately my computer becomes really slow when I try to tune the hyperparameters, and it takes a long time to do so. I really need to use my computer to study for my finals, so I tried to limit the amount of time I spent tuning hyperparameters. If I was to make a next step though, this is probably what I would focus on.

Overall, I think it was very interesting to apply what I learned throughout this project. I had fun going through the different steps of the data science pipeline and applying what I learned in class. When I first started this class, I think I would have been very overwhelmed by a project like this. But after doing the homeworks and playing around with pandas and other ML models, I was able to do this project without too much trouble. I think I may tweak the model I created a bit more in the future, but this class was very helpful in getting me to understand how I can use python and libraries like matplotlib, sklearn, pandas, etc. to analyze datasets.