Advanced Data Analysis
DATA71200
Summer 2024

Final Report: Machine Learning Project Takeaways

I. Introduction

The three machine learning projects produced for this class were ultimately quite challenging for me--largely due to critical errors I made early on when selecting my dataset. I was eager to take this course, and particularly excited when I realized that our primary focus would be machine learning for categorical data. As someone with a professional and educational background focused on visual art, I have frequently found myself veering towards more categorical variables in the datasets that I have selected for projects throughout my time in the Grad Center's Data Analysis & Visualization master's program.

Ultimately, I think the quality of the work I produced for this class and my understanding of the machine learning processes we studied would have been much improved had I recognized that I needed to pivot datasets sooner and spent more time selecting an alternative option once I realized it was necessary.

II. Dataset Selection

I've been thinking a lot about what topic I'd like to do for my thesis, and I was eager to test out working on a similar dataset for this course. Given my arts background, I have been considering doing an exploratory project using the information available in the FBI's Lost Art Database. I thought this class would be a perfect opportunity to get to know that dataset while considering whether or not to work with it for my thesis.

I had naively planned on scrapping the data directly from the database website, and I assumed that it would be a simple process. Unfortunately, when I went to scrap the data I quickly found out that the FBI had protections on its website that blocked it from traditional Python data-scrapping techniques (or at least the ones that I'm familiar with). I proceeded to spend another week trying to copy/paste the data and organize it by hand (which I still may complete for my thesis later on).

Unfortunately, in the quickened space of a summer course, the time I lost stubbornly trying to make the FBI data work left me backed into a dataset corner. Finally, I accepted that I needed to pick a new dataset to begin work on my first project for this class. So, I turned to data resources that I was most familiar with. I wanted to find a widely accessible data source with a least 10 features and a generally useful categorical predictive variable. Naturally, I found myself digging through options on NYC Open Data.

A few summers ago, I visited Mona Chalabi's "The Gray-Green Divide," installation on the exterior of the Brooklyn Museum. Her work studied the density of trees in NYC against neighborhood heat bubbles and wealth inequities. I remembered reading that she had collected some of the data from NYC Open Data, and I decided that it might be interesting to see how the NYC Open Data's Hyper-Local Temperature Monitoring dataset might be used to further predict/show the distinction between neighborhood and dangerous heat levels.

The moment that I switched courses from trying to use the FBI data, which would have been filled with blanks and been almost entirely categorical, to the Hyper-Local Temperature Monitoring dataset, my set of challenges was entirely inverted. This data was very accessible, but also huge. The entire version has about 2.1 million rows. Off the bat, I needed to limit to something that I could load into Github. So, I reduced it to only locations within Brooklyn in 2019. Despite that, I still needed to split the dataset into two separate CSV files before I could load them into Github without it crashing.

This should have been a sign that my dataset was going to be too big to do some of the later processes (but I will get to that soon).

III. Data Cleaning & Transformations

When getting set with my data, I first loaded my two primary CSV files from the NYC Open Data and concatenated them into a single pandas dataframe. Later, during Project #2, I also decided to merge another dataset into my research. This was for two reasons. Firstly, to have a greater variety of features to select from and, secondly, to further study Mona Chalabi's work on the impacts of climate change in various neighborhoods of New York City. The CVS file that I merged with the NYC Open Data was a spreadsheet with neighborhood names and median household incomes to correspond to the neighborhood borders around the Lat/Longs provided in the Hyper-Local Temperature Monitoring data.

Once I had the comprehensive dataframe ready in my notebook, I next checked for any missing data. I saw that some rows were missing the air temperature. So, I created a quick visualization to confirm that the air temperature data was normally distributed. Upon confirming that it was, I decided to replace all missing temperature data with the appropriate mean.

As this course focused on working with categorical data, I decided to make my predictive variable the level of heat advisory instead of the specific temperature. I also hoped that would allow for more leeway in predictions and more robust outcomes.

So, I created a new column for the heat advisory levels associated with each temperature by applying a function to the dataframe's air temperature column. I also removed the existing date column and replaced it with separate month and day columns to better suit the necessary structures for the machine learning processes down the line.

From there, I assigned all the dataframe columns, except the Air Temp and the Advisory into the features variable, and I assigned the Advisory level to the target variable. Then, to accommodate the necessary machine learning data structures, I did One Hot Encoding on my features data and Label Encoding on my target data.

When I did the label encoder on the advisory level variable, I also checked to see the value counts for each advisory level. There was a big discrepancy between the number of occurrences for each variable. Given that, when I went to split my dataset into training and test data, I made sure to conduct a stratified split so that I wouldn't end up with all the "Extreme Danger" occurrences in just the test or the training data set.

IV. Data Visualization: Key Takeaways

As I mentioned previously, data visualization was helpful from the start of Project #1, as it allowed me to confirm the normal distribution of the air temperature feature before deciding if it made more sense to replace the missing values with means, medians, or to simply drop them.

Once I had more data fully cleaned and organized, exploring my data visually also helped me realize another issue with my dataset--specifically the lack of variety in the neighborhoods represented in it. Given the scale of the overall Hyper-Local Temperature Monitoring dataset as well as its "hyper-local" name, I assumed that many neighborhoods of Brooklyn would be represented and that more of the borough would be covered geographically. However, when visualized, I realized that the air temperature monitors used in the data set were primarily condensed into only a few areas.

During class, we used a scatterplot on a set of real estate data from California and charted the Lat/Longs in that data. The result allowed us to create a map of the state. I expected a similar outcome to happen for Brooklyn when I charted the Lat/Longs in my temperature data. Unfortunately, what I saw instead was five condensed clusters showing me how limited the areas of Brooklyn being represented in my data truly were. This lack of diversity in the neighborhoods represented would limit the potential of the neighborhood feature being useful. That would get visualized later in Project #2 when I saw that none of the encoded neighborhoods registered as important on a bar chart of feature importances that I created when exploring using the decision tree model on my data.

V. Experiments with Supervised Learning Algorithms

I used two supervised learning models on my data, k-nearest neighbors (KNN) and a decision tree. I also later went back and briefly ran a random forest model, but it proved to be less effective than the decision tree or the KNN.

A. K-Nearest Neighbors

The KNN approach is a simple model that makes predictions by finding the closest data points, e.g. the nearest neighbors within a data set. The "k" in KNN stands for the number of neighbors being used to assign a class. This means that when the model seeks to classify a new data point, it looks to the k/set number of closest neighboring data points, and then uses the majority class among those data points to assign the predicted class for the new data point being fed to the model.

I initially ran the KNN model on my training and test data by setting k to a fairly standard kick-off point of 3. With k=3, I was able to get an accuracy of 93.09% on the test data. Then, I used the grid search to iterate through potential k values ranging from 1 through 10. The grid search tested the KNN model for each value of k in the grid search. From that exercise, the best-performing k value performed slightly higher with k=7 resulting in an accuracy level of 93.84% on the test data.

B. Decision Tree

For my second supervised learning model, I decided to try the decision tree model on my data. I've always loved a good flow chart, so I wanted to try to decision tree model because I appreciated the logic in its approach. A decision tree's model is structured to make decisions/predictions by splitting data into subset groups based on a series of conditions. Like a reverse flowchart, the decision tree model is best visualized like a dendrogram. The widest point is at the bottom where the smallest subsets are divided into the terminus leaves. The prediction is the root at the top, answering the key question about the data by trailing backward up the leaves.

Using the decision tree model ended up being very comparable to the results I got with my best KNN model (k=7). For the decision tree, I got an accuracy of 93.9% accuracy. It was when I went to visualize the most important features in the decision tree model that I saw how truly ineffective my efforts had been in considering the neighborhood in terms of heat advisory. Perhaps if I were to do this again, I would try shrinking my predictions into census tracks instead of neighborhoods, which would have differentiated the temperature collection locations further.

VI. Experimenting with PCA for Feature Selection

In the third/final project, I worked a lot on using PCA feature selection. Unfortunately, my data performed poorly under the PCA selection process, especially when I used unscaled data. On the scaled data, the result I got said that only one feature explained 95% of the variance and that was median household income, which I knew from understanding the topic could not be true. I got an inverted result on the scaled data which said that nine features explained 95% of the variance, which made much more sense (month, day, hour, etc.).

VII. PCA as Pre-Processing for Clustering

Unfortunately, when it came to doing the clustering, my data was just too large to load and run on the scale that my home computer could conduct. I tried several approaches, even purchasing premium GPUs, but it still kept crashing. In the end, I entered the code into the notebook but accepted that it wouldn't be able to load.

Next, I moved on to trying to run the clustering process on the alternative suggested dataset, the sklearn breast cancer data. The breast cancer data was ultimately more appropriate in size and subject. The breast cancer dataset has more separable data points and features. Using the PCA approach, 10 features explained 95% of the variance in the outcome variable, which makes sense since this data was less ordinal and more clearly categorical, and with less noise.

VIII. Key Learning Outcomes

Ultimately, I think the supervised learning models are a little easier to wrap my head around. I hope to be able to come back to trying these technics on the FBI Lost Art Database information--once I figure out a way to export it and clean it.