CSC 177: Data Warehousing and Data Mining

Project 1: Data Preprocessing

Jagan Chidella

Group: An Lam, Jimmy Le, Tom Amir,
Dianna Melendez, Amrit Singh, Talal Jawaid, Min Li

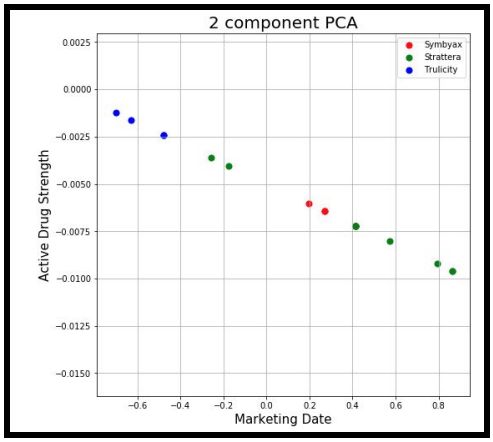
Before starting this project, the group went through Tutorial 4 for Data Preprocessing. Each method of Preprocessing given to us was analyzed for our personal understanding, and documented in the "P1_Data" document. After realizing the capabilities of the various methods of Data Preprocessing, we began to seek out our data sets online. After searching online, we chose two data sets that were interesting that we were going to thoroughly analyze for Part 3B of our project:

1) **Drugs**: In this zip file, we are using the "Product" Excel Sheet. We made sure to save it as a ".csv" file before proceeding applying the Pre-Processing techniques that we learned.
2) **London Air**: In this data set, we found that we were able to apply the more interesting Pre-Processing techniques, such as: missing values, and saving a dataframe.

Since the purpose of this project is Data Pre-Processing, we wanted to take data sets and clean them up in such a way where it was an easily readable format. In addition to being easily read, we wanted less attributes to deal with, while dropping unnecessary data. The techniques that we applied were: Missing Values, Outliers, Duplicate Data, Shuffling Dataframes, Sorting Dataframes, Saving a Dataframes, Dropping Fields, Calculated Fields, Feature Normalization, Concatenation.

One issue we ran into was with Aggregation. While we attempted to follow the code provided in the tutorial 4, we were unable to successfully run the aggregation preprocessing technique. We attempted multiple times on different data sets, including the London Air data set and the Brazilian rain measurement data set, but we were unable to reproduce a similar output to the tutorial example. We saw that the London Air dataset had datetime values in a format that was dissimilar to the precipitation dataset in the example. The dataset provided had the datetime in the format YYYY-MM-DD while the London Air data set that we were using had datetime in YYYY-MM-DD HH:MM:SS. We successfully attempted to strip the time using both excel and python split() method, however this did not solve our problem. We found this difficult to debug since the code was not throwing an error, it was just failing to produce an output. We consulted with Professor Chidella and TA Siddharth Chittora but were ultimately unable to produce an output for the aggregation preprocessing technique.

One thing that was interesting to analyze is the PCA performed on drugs Symbyax, Strattera, and Trulicity. PCA performed on active drug strength relative starting marketing date. From the PCA, the data suggests that as the Standardized (Z-Score) Marketing Date increases, the Strattera was manufactured at a much lower strength. When compared to Symbyax, Trulicity continued to stay unchanged. It was interesting to see a negative correlation in a large set of data.



Through the Pre-Processing of our data, the number of rows and columns were significantly reduced. Since we "scrubbed" the data so that it became cleaner, our confidence in the legitimacy of the data increased. We got rid of NaN data, blank data, unnecessary fields, duplicate data, outliers, and sorted the data in such a way where the format was easily readable. After condensing the data into what we deemed necessary, the amount of information to look at is significantly less for us to read, and less for our program to parse through. This will result in quicker, and more accurate predictions, because of the removal of junk data and dimensions.

Although we pre-processed multiple data sets, we decided to focus on the London Air dataset to extract a more meaningful calculation from the resulting sets. We decided to split the data as follows:

Training Data: First 80% of the data after Pre-Processing

Test Data: Last 20% of the data after Pre-Processing

After partitioning the data, we focused specifically on the "value" column and found that the mean and standard deviation were as follows:

Training Data:

Mean = 48 Standard Deviation = 54

Test Data:

Mean = 47 Standard Deviation = 54

What we can draw from this data is that splitting data into two consequent sets is a vital step in creating a machine learning model. The training data receives the majority share of the partitions since this is the data that the model is built on, whereas the remaining test set will be

used to validate the training data against. In our case, it is apparent that the two sets of randomized data provide consistent results in terms of the mean and standard deviation of the “value” attribute.

Pre-Processing is an important step in Data Mining, but it is also long and very tedious. Covering all these steps took a lot of work that needed to be done in a specific way. Although tedious, we received a clean data set that we are confident in using for the next couple of projects. Seeing as splitting the data up into test data, and training data resulted in an almost identical mean and standard deviation shows that the data isn’t skewed across the board and that there is no longer any missing data or outliers that have a huge effect on the set that split up. Our data is clean and we are ready to proceed in the process of Data Mining.