Achieving Program Learning Goals through Course Deliverables

Benjamin Cowan

Master’s in Applied Data Science

Syracuse University

Table of Contents

[Introduction 3](#_Toc74287020)

[Section 1: IST 736: Final Project for Text Mining 3](#_Toc74287021)

[Linking Text Mining Project and Program Learning Goals: 6](#_Toc74287022)

[Section 2: IST 707: Final Project for Data Analytics 6](#_Toc74287023)

[Linking Data Analytics Project and Program Learning Goals: 10](#_Toc74287024)

[Section 3: IST 652: Final Project for Scripting for Data Analysis 10](#_Toc74287025)

[Linking Scripting for Data Analysis Project and Program Learning Goals: 13](#_Toc74287026)

[Section 4: IST 719: Final Project for Information Visualization 13](#_Toc74287027)

[Linking Information Visualization Project and Program Learning Goals: 15](#_Toc74287028)

[Conclusion: 16](#_Toc74287029)

## Introduction

In this paper, I cover course deliverables that link directly to the learning goals in the Applied Data Science Program at Syracuse University. The deliverables include four courses and their final projects: Text Mining, Data Analytics, Scripting for Data Analysis, and Information Visualization. Each project took a period of ten weeks to complete, which culminated in a final report and class presentation. To complete each assignment, I had to demonstrate my aptitude for acquiring data, cleaning and debugging data, implementing exploratory data analyses and data mining techniques, and interpreting and communicating results.

## Section 1: IST 736: Final Project for Text Mining

Broadly speaking, Text Mining requires the manipulation of unstructured text data into a structured format. Structuring the text data allows for machine learning techniques to be applied to it. For the Text Mining project, I worked with three other students to create a problem statement that our newfound text mining skills could solve. The problem statement was concerned with predicting public sentiment towards global warming, as well as predicting the year the text was made. My team thought it would be very insightful to analyze those topics to see how global warming’s public perception may have changed from 2016 to 2020. Global warming is also a huge threat to everyone living on the planet, so it was an important topic that my team and I wanted to examine. Supplementary analyses were also created to predict carbon emissions and consumption in the United States (US). The prediction of carbon emissions and consumption could provide useful information on the US’s plans to combat global warming.

Text data included user tweets from 2016 and 2020. The tweets from 2020 were obtained using the Twitter API in Python and stored in a data frame. The usernames were removed to keep the data anonymous and private. After the data frame was created, it could be saved as a CSV file for later use. Twitter API could not be used for the 2016 tweets because the API could not retrieve dates that far in the past. Tweets from 2016 were collected using vicinitas.io, which is a tool that collects tweets by hashtags from a specific date range. The tweets collected from both time periods were concatenated into one data frame for analysis. After the data frame was formed, there were several data preparation and cleaning processes to complete:

* Special characters needed to be removed because our analysis was focused only on words.
* Stop words (e.g., the, I, am, it, but) were removed since they added little value to the analysis.
* The text was focused on tweets, so hashtags needed to be removed. The hashtags were stored if needed for future analyses.

Once data preparation and cleaning were complete, the data needed to be vectorized using sklearn’s count vectorizer. Vectorizing and counting the words in each tweet allowed the unstructured text data to become a structured dataset, which allows data analytic techniques to be applied to them. Supervised learning models were then applied with the dataset to determine what year a tweet was from. For a supervised learning model, training and testing sets needed to be created. Splitting the data allowed the model to be “trained” and learn relationships within the training set and make predictions onto the testing set. The model’s accuracy was calculated from the prediction to determine how well a supervised model performed on the unseen testing data. Three-fold cross validation was also used because the results may vary based on the samples used for the training and testing sets.

The first supervised modeling techniques used was the probabilistic classifier Multinomial Naïve Bayes (MNB). The MNB model used prior and conditional probabilities of words from each document (each tweet in the vectorized dataset) to calculate a posterior probability of a tweet belonging to a specific year. However, a drawback of the MNB model is that it is “naïve”, so it does not consider any potential relationships between words. As a result, the model could miss many relationships between words that are useful for classification.

We needed to compare the MNB model with another classifier to determine if the probabilistic classifier was a decent choice for prediction. Every classifier has their own drawbacks, so it is necessary to leverage different models depending on the data. Support Vector Machine (SVM) models were implemented since most text mining classification problems are linearly separable. If the data were not linearly separable, different kernels could be implemented to transform the data into a high dimension. The three kernels used were linear, polynomial, and RBF. Another parameter to be fined-tuned was the cost. The cost parameter is necessary because it manipulates the margins being used to classify data. If the cost is tuned correctly, the model should not overfit the training data, thus making accurate predictions on the testing data.

To do sentiment analysis on the 2016 tweets, however, a lexicon tool had to be used to quantify a tweet’s sentiment. VADER lexicon was used to give the words in each text a sentiment score. Afterwards, the scores were compounded and labeled as negative, neutral, or positive. A separate column would be added to the dataset as a label for MNB and SVM models to predict each tweet’s sentiment.

After sentiment analysis was complete, I wanted to bolster my results by using Latent Dirichlet Allocation (LDA) for Topic Modeling. Sentiment analysis was useful because it pointed out specific words driving a tweet’s sentiment; however, it lacked the ability to create themes from the data. Using LDA on positive, negative, and neutral tweets allowed new themes to arise. These themes were constructed from frequent words that were present in each of the three sentiments.

The results for predicting a tweet’s year did show significantly high model accuracies for MNB and SVM models. The MNB model showed a 99% accuracy, which clearly means the model overfit the training data, making it a terrible predictor on unseen data. A new strategy had to be implemented for sampling the data, without replacement, to create an accurate model that does not overfit the training data. After reevaluating the training set, it was clearly imbalanced – there was more data from 2016 than 2020 - so sampling needed to be balanced to develop a decent classification model. Once the ideal records were obtained, models that predict a tweet’s age were created. To ensure the samples were unbiased, multiple models were run using different samples. Three of our classifiers used 200 samples without replacement. Another three classifiers used 350 samples without replacement. Different sample sizes were chosen to confirm our assumption that increasing the sample size would yield higher model accuracies. Fortunately, the model accuracies did increase to 96% accuracy.

Predicting a tweet’s sentiment, using SVMs, resulted in overfitting like the initial MNB models. However, proper sampling reduced the model’s ability to overfit the training data. Next, the cost parameter was tuned, using a loop to rerun the model at different cost values, to improve the model’s accuracy. Kernels were also changed to see which performed the best. The model using a radial basis function kernel performed the worst, while the polynomial kernel led to the highest accuracy score.

Predicting sentiment in the 2016 tweets required a similar balancing strategy. There were 811 negative, 1389 neutral, and 2800 positive tweets. If the data were sampled without any manipulation, a large proportion of the training data would be positive tweets. To avoid this, 800 tweets from each sentiment were sampled. The resulting models in the table below show the MNB model producing the most accurate prediction for a tweet’s sentiment. A normalized data frame was also created, using tf-idf statistical measure, because tweets vary in size. If tweets are larger than one another, it could potentially skew the results since words are appearing much more often.

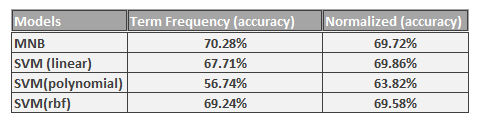


Figure 1: Table of accuracies for MNB and SVM models.

To understand what the MNB model learned, highly weighted positive and negative words were selected from a model parameter with a loop. Some of the words strongly weighted as having a positive sentiment were thank, win, acceptance, and wonderful. Words negatively weighted were stupid, denial, threat, illuminati, fracking, and war. Fracking was a very interesting word because the model used it to classify tweets as negative. This meant that fracking was being used in a negative context on twitter. Further analysis into the context surrounding these positive and negative words were done with LDA topic modeling techniques.

As stated previously, LDA identified some of the major themes from the texts within each sentiment. The figure below shows some main themes identified in the 811 negative tweets. The themes encompass mental health, fossil fuels, political figures, and much more.

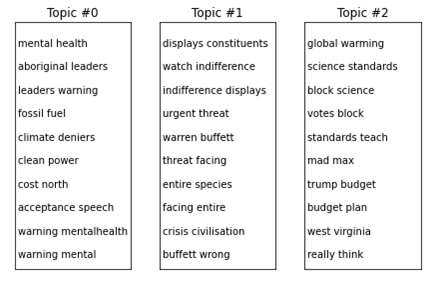


Figure 2: Three topics created using LDA Topic Modeling on tweets with negative sentiment.

In conclusion, we successfully processed unstructured text data into a labeled data frame to build classification models. These models were then used to predict a tweets sentiment and year. From the analysis, it was clear that political figures, healthcare issues, and scientific discoveries were all key topics driving global warming discussions on twitter. And, depending on the topic, global warming could have a certain positive, negative, or neutral sentiment.

## Linking Text Mining Project and Program Learning Goals:

From the Text Mining Project, I was able to utilize text and data mining techniques, as well as statistical inference on text data. I learned how to implement vectorizers, create labeled data frames, apply supervised learning models, and communicate my results. I collected twitter data using an API, processed it, and concatenated it with other text datasets. Patterns in the data, to predict a tweets year and sentiment, were explored using statistical analysis, sentiment analysis, SVMs and MNB models. As the project progressed, I identified areas to apply new algorithms such as LDA for topic modeling. Later, I would use LDA to find topics present in positive, negative, and neutral tweets. From the results, I managed to figure out topics, such as mental health and political figures, that a company could use to build their relevancy on twitter, regarding global warming. Lastly, I demonstrate practices in data science to protect user’s privacy by leaving user information out of the analysis.

## Section 2: IST 707: Final Project for Data Analytics

For the Data Analytics project, I worked with two other students to solve a real-world problem using data mining techniques. The problem was focused on weather and being able to forecast rain in different regions of Australia. We chose weather forecasts, because they are in high demand with companies. Today, companies across the world spend billions of dollars on accurate predictive models. The cost depends on how far out the forecast is, so a next day forecast costs less than a seven-day forecast. Completing the project successfully required many technical skills:

* Develop predictive models for next-day forecasts and seven-days later forecast.
* Compare the models to determine the best performing one for forecasting the data.
* Determine which variables were best for predicting rain.

The dataset we built our predictive models on was derived from the Australian Government Bureau of Meteorology. It consisted of weather data for several cities in Australia over the course of 10 years - over 23 variables and 56.5 thousand observations. Some of the variables measured were wind speed, humidity, and temperature at different times of any given day. Those values would be used as explanatory variables to predict if it would rain the next day. The response was a binary variable that stated if it rained the following day. The first hurdle to overcome was an imbalanced dataset. From the 56.5k observations, 12.5k observations had it rain the next day, while 44k instances had no rain. This imbalance is reminiscent of the sentiment problem I faced in the Text Mining project. I was able to solve this problem by developing a sampling strategy that randomly selected a balance of observations where it rained and did not rain. Five-fold cross validation was also implemented to ensure that there were no significant variations in the dataset that could alter the prediction results.

Exploratory Data Analysis (EDA) was implemented to gain insight into potential relationships, within the data, that may be helpful for modeling. I created the heatmap to show the correlations between all the numeric variables. Many of the time-associated variables, such as pressure at 3am and 9am, were perfectly correlated. It was evident these correlations were due to multicollinearity, so the associated variables were averaged and placed into new pressure, humidity, max temperature, and windspeed variables.

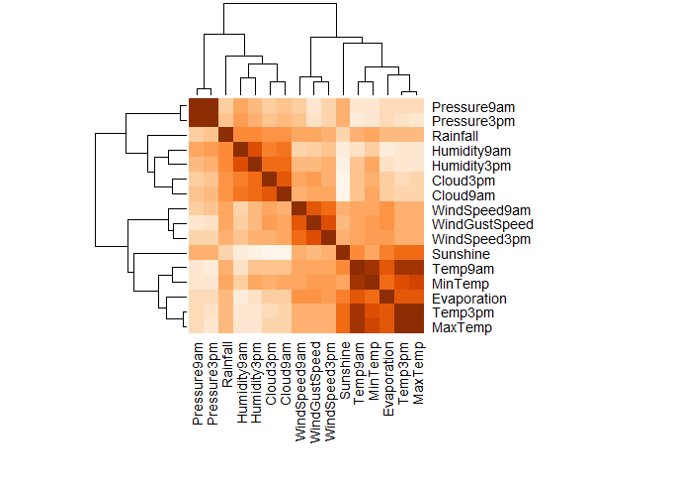


Figure 3: Heatmap of correlations between all numeric variables. The darker the color, the higher the correlation.

More EDA was created to find potential relationships or patterns in the data. An example of an EDA was the boxplots below, which illustrated fluctuations in the variation between humidity and the different areas in Australia. These fluctuations could be a great explanatory variable for the variability in our response variable. If I could improve this visualization further, I would create two separate boxplots for rain and no rain. This addition would tie the analysis directly to the response variable I wanted to predict.

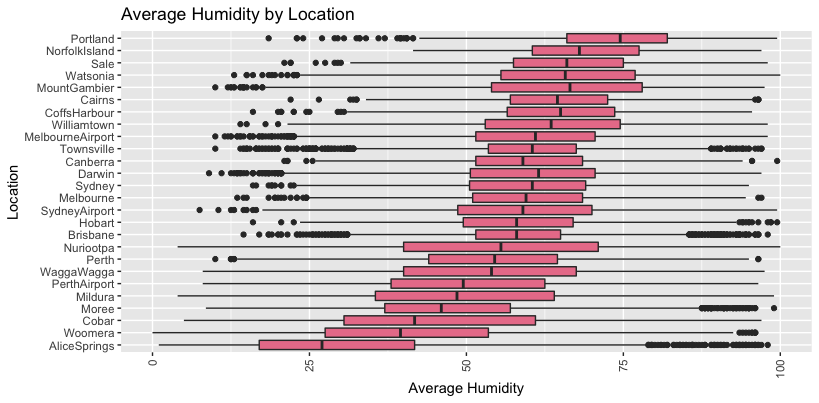


Figure 4: Distribution of Humidity for each area in Australia.

After EDA was analyzed, I implemented Association Rule Mining and several supervised learning models to demonstrate the courses learning objectives. This was also a practical experience because each algorithm has flaws and needs to be leveraged depending on the data. Having experience with each algorithm will help me make better decisions on which to use in the future. Association Rule Mining was used to find rules that connect variables within the dataset to my response variable. SVMs were also ran, using different kernels, to determine which transformation was more effective at prediction. The accuracies created from the SVMs could be used as a baseline to compare with other models. Next, Naïve Bayes, K-Nearest Neighbors, Decision Tree, and Random Forest models were created to predict rain after one day. More models would be created for predicting rain after seven days.

The Associative Rules listed specific values of sunshine, pressure, and hours of cloudiness correlated with rain the following day. These rules were not definitive, but pointed towards sunshine, humidity, pressure, and cloudiness being associated with rain prediction. They also bolster the previous exploratory data analyses that humidity may be an important explanatory variable in modeling.

SVMs were created for linear, polynomial, radial, and sigmoid kernels. The cost parameter for each kernel needed to be tuned correctly to avoid overfitting the data or being too general for making predictions. The advantage of using an SVM is its scalability for large datasets and ability to solve linearly separable and inseparable problems. The SVM accuracies for each of the kernels were calculated. The sigmoid kernel performed the best; however, each kernel performed relatively well. Accuracies ranged from 83.54% to 83.67%.

Next, one hundred Naïve Bayes models were run. Each model had a different amount of cross validation folds used for training and testing. The output below displays the accuracies of those Naïve Bayes models. A cross fold of 15 was shown to have the highest predictive capability. This method allowed me to find the cross fold with the highest accuracy, but it also gave me a range of possible accuracies. The accuracies ranged from 78.82% to 79.17%. This turned out to be a good range, because there was not any significant difference in accuracy, regardless of the number of cross folds used.

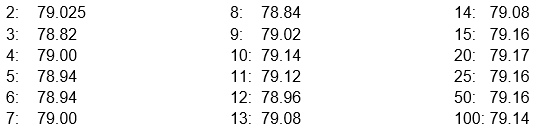


Figure 5: Number of k-folds and resulting output. For example, a model with a k-fold of 2 had an accuracy of 79.025%.

K-Nearest Neighbor (kNN) used distance measuring to make prediction on data. The strength of using kNN over other algorithms is its ability to classify testing data based on its distance from the training data. However, the number of neighbors used to classify testing data needed to be tuned to accurately predict rain. For the rain tomorrow prediction, kNN was 79.98% accurate while the rain in 7-day prediction was 72.40%. We expected the accuracy to decrease the further out the prediction got.

Later, a decision tree was made to classify rain or no rain. From the decision tree below, the amount of sunshine, pressure, rainfall, and humidity were important variables for classifying rain. This model was created using every location, so we decided to check if it applied to specific locations. After rerunning the model on specific locations, we found that Alice Springs, Portland, Darwin, and Sydney had varying accuracies due to the differences in their climates. Some were easier to predict with certain variables than others. For more accurate results, decision trees should be constructed for each location.

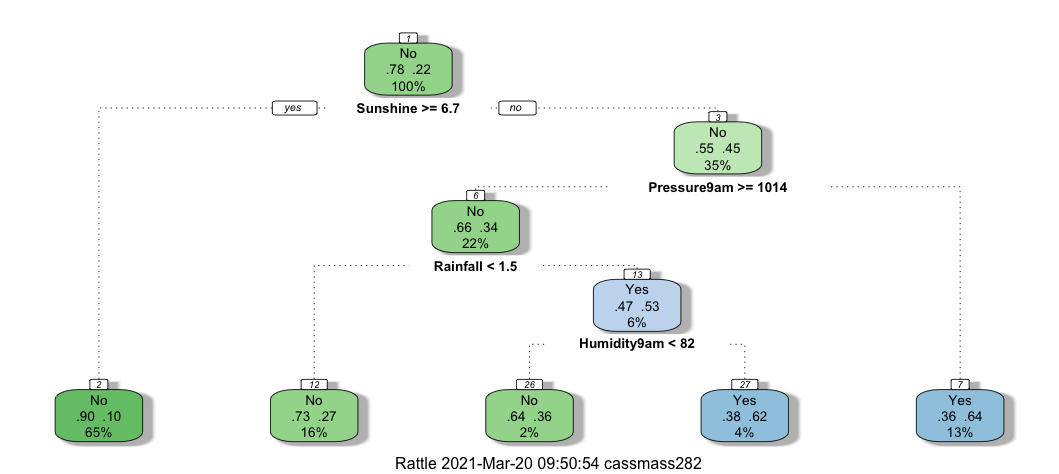


Figure 6: Decision tree modeled to predict if it rained the following day based on hours of sunshine, air pressure, amount of rainfall, and humidity.

Random Forest was the last model used to make predictions on the data set. It is an ensemble method where multiple decision trees are merged into a “forest” where majority vote results in the most accurate prediction. Different “mtry” parameters for the Random Forest models needed to be tested because it was important to try out different variables for building each model.

In conclusion, the Random Forest predictive model yielded the highest prediction for rain the next day with 83.89% accuracy. However, the SVM model had a very close prediction as well. The Random Forest also outperformed the other predictive models for rain after seven days with an accuracy of 78.53%. Once the entire dataset was analyzed, it was clear that sunshine (in hours), air pressure, and humidity were the most predictive variables from the dataset. This does not mean those are the only variables and more data should be added in the future to possible improve model outcomes. Lastly, if specific regions were subset, Decision Tree models would have the highest capability of prediction rain the following day. This was due to regions having different climates and patterns that a model can use for prediction.

## Linking Data Analytics Project and Program Learning Goals:

From the Data Analytics project, I learned several data analytical techniques and practices. This included unsupervised learning techniques such as Association Rule Mining, and several supervised learning techniques. Furthermore, I was able to apply and tune specific parameters for each learning model to improve their predictive capabilities. I was also able to compare the different models to see which model fit the weather data best. Patterns were identified beforehand, using heatmaps and distributions, and then applied to the classification models. Principle component analysis and cross validation were also implemented to reduce the dataset of redundant variables and ensure testing and training sets were representative of the entire dataset. Finally, business decisions were derived from the analysis. For instance, a business should use Random Forest models for the specific dataset when predicting next day rain or even rain after seven days. If the business wanted to make a prediction using a specific location, they should consider a Decision Tree model since it led to the highest accuracy.

## Section 3: IST 652: Final Project for Scripting for Data Analysis

The Scripting for Data Analysis project had students utilize Python scripts to access, organize and manipulate different types of data. Afterwards, the data would need to be processed into data summaries. To complete this task, I worked with structured, semi-structured, and unstructured data. The topic surrounding the data was airline customer satisfaction and identifying possible variables to predict satisfaction scores.

The structured data for airline satisfaction came in the form of a csv file consisting of 12980 rows and 28 columns. Variables included information about each customer, which included their age, gender, satisfaction score with their flight, date of their flight, etc. The challenge I had to overcome with the dataset was determining the most predictive variables for airline satisfaction. The other challenge was to do it only using Python scripts.

Semi-structured data came from tweets posted on Twitter. To compliment the structured dataset, the semi-structured data needed to be related to airline satisfaction. With that, I chose to analyze tweets about airlines. Specifically, large passenger airlines because more people may post about them on twitter. And the more traction an airline has on twitter, the likelier the chances I have at gaining a lot of flight reviews to analyze. I decided to analyze tweets about Delta, United and Southwest Airlines to compare public sentiment towards each of them. Sentiment analysis would require the text data in each tweet. Tweets were also acquired from large airline manufacturers because they might be used to predict airline customer satisfaction. A customers satisfaction level may be influenced by the manufacturer of the plane and its reputation. Next, I acquired the tweets using Tweepy API and hashtags such as #Boeing and #Airbus. After analyzing the tweets posted with those hashtags, I found many of them to be irrelevant. To find relevant tweets, I needed to research specific hashtags that were being used to review each airline. Once I collected the necessary tweets, they were stored in the form of JSON data in MongoDB. There were 900 tweets in total, split evenly between Delta, United and Southwest Airlines. Another 600 tweets were evenly split between the airline manufacturers Boeing and Airbus.

The unstructured data was in the form of each tweet’s text. The text, as well as other information in each tweet, would need to be extracted from the JSON formatted data and put into data frames for analysis. Once the text was scraped and put into separate lists in Python, text blob for sentiment analysis looped through each text in the list and calculated the overall sentiment for the airline. Additional information consisted of a Twitter user’s followers, favorites, and friends count. That data could be used in the future to compare airline manufacturer’s relevancy on Twitter.

Manipulation of the structured data set included a summary of membership statuses and possible differences in their satisfaction scores. From the output, there are differences in means and dispersions of the different customer classes. This summary statistic indicated that the status of customers may influence their satisfaction with the airline.

Airline satisfaction data was also separated and reported by the customer’s gender and age. As the summaries become more granular, we start to see more variables with the potential to influence customer satisfaction. In this case, gender may have a significant impact on the prediction, because there was variation in satisfaction scores between males and female customers. If that were the case, it would be the airline’s priority to make sure males and females were equally accommodated.

After gender was analyzed, summary statistics on age and airline status were calculated. The customers themselves were split by age range and airline status. The summaries included the mean, standard deviation, lower and upper quartiles for each group. Like gender, there was a lot of variation in satisfaction scores between older and younger customers. This was also true between customers in different airline statuses.

Next, the semi-structured data was brought into Python and summarized. The results summarized the follower, favorite, and friend counts between the Twitter users who posted about Boeing and Airbus. From the data collected, it appeared that Boeing had a much higher popularity on Twitter than Airbus. The mean number of favorites for Boeing tweets was around thirty thousand, while Airbus tweets had around twenty thousand favorites. However, more statistical analyses and data mining must be completed to be certain.

The final output, shown below, are the sentiment analyses for each of the airlines. Tweets about United Airlines appears to be more positive than Delta or Southwest. This difference should be analyzed in the future to judge customer satisfaction for each airline.

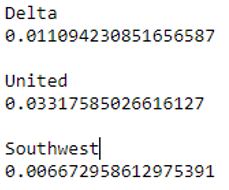


Figure 7: Sentiment scores for Delta, United, and Southwest airlines tweets. The higher the score, the more positive the sentiment.

Once all the summaries were complete, I proposed statistical analyses for the airline to take. For example, confidence intervals should be calculated to see if there is significant difference between customer satisfaction scores and the other variables in the data set. Another method could be the development of linear models with satisfaction scores as the response variable. The explanatory variables to predict satisfaction scores would be age, gender, and airline status. This method would help determine which variables were significant and explain the variation in satisfaction scores. Supervised learning models, such as SVMs, could also be implemented to help determine customer satisfaction. Finally, if there was a significant difference between customer satisfaction and age, I suggested the airline run a survey. The survey would ask an age group what they liked most about their experience at the airline. Afterwards, the airline would run an A/B test using the survey data. The control group for the test would be the previous customer satisfaction data. New testing data would be the customer satisfaction scores after the airline made the appropriate changes. Then statistical tests, such as confidence intervals to detect the mean difference between control and test data, would be calculated. Results from the test would show if the experiment led to a significant difference in customer satisfaction.

In conclusion, the summary statistics about the structured data depicted a customer’s airline status, gender, and age as potential predictors of customer satisfaction. The semi-structured data showed that Boeing garnered more attention on Twitter than Airbus. This could be a potential variable to analyze when predicting customer satisfaction. Finally, sentiment analysis between Delta, Southwest, and United Airlines resulted in United Airlines having the more positive score. This may be indicative of United Airlines being a more enjoyable experience to customers than other airlines.

## Linking Scripting for Data Analysis Project and Program Learning Goals:

In the final project for Scripting for Data Analysis, I demonstrated my ability to manipulate and organize structured, semi-structured, and unstructured data in Python. I also used statistical analyses to identify potential variables that could be used to predict customer satisfaction. Based on the semi-structured data, I extracted new variables, such as a follower count, that a business could use for future predictions. New strategies were implemented to analyze airlines, such as sentiment analysis on tweets for each airline manufacturer. Finally, storing tweets in MongoDB emphasized the importance of privacy and how data is managed. For example, storing data in MongoDB may have different user permissions than in SQL Server. This could cause sensitive data to get leaked or tampered with, so it is imperative that data is managed and secured along the data pipeline.

## Section 4: IST 719: Final Project for Information Visualization

The final project for Information Visualization required students to analyze a dataset in R, develop a story about the data, and present it using a poster. I chose to analyze a dataset about online chess matches. The dataset contained over 20,000 chess matches with 16 variables. Variables included the number of turns in each match, player rankings, opening names, etc. I chose a chess dataset because streaming chess matches has become more popular in recent years. Ranked chess has also grown in popularity so I wanted to craft a data-driven story for people interested. I also wanted to engage my audience by including questions on my poster that could be answered through plots I made using the dataset.

Before I created my plots, I needed to clean the data. The raw dataset contained ranked and non-ranked match. These non-ranked matches would need to be filtered from my analysis so I could focus on the ranked chess games. Any missing data was also removed to generate my plots. Next, I created rating for each match by averaging the player ratings. This average allowed me to create interesting EDA to put in my story. I factored the new ratings column into low, medium, and high player rankings. This would allow my audience to get a clear picture of how many matches were in each category.

To analyze the distribution of chess match ratings, I created the following histogram. The histogram shows that ranked matches range from ~900 to ~2500 with the mean ~1500.

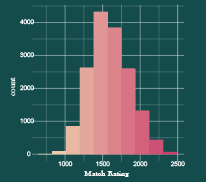


Figure 8: Histogram of match ratings.

I was going to create a bar plot of the frequencies of each chess opening, however, there were too many openings to create a visually appealing and informative graphic. To include the bar plot, I changed my strategy and decided to include the 10 most frequent chess openings.

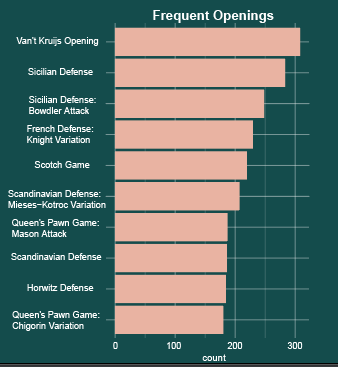


Figure 9: Bar plot that displays the openings played most from the ranked online chess matches.

Next, I created two dimensional plots to convey relationships within the data. The scatterplot below shows the relationship between players on each side of the board (Black and White). As the rating of players on one side increased, the other side, generally, increased as well. I fit a linear model against the visualization to highlight the relationship. In addition, I created a variable of the absolute difference between player ratings and added it as a color gradient. The data had to be continuous if I were to use a color gradient. The gradient showed that many of the matches between players were not fair - many of the player’s opponents had much higher rankings - and show potential flaws in how the online program was matching players.

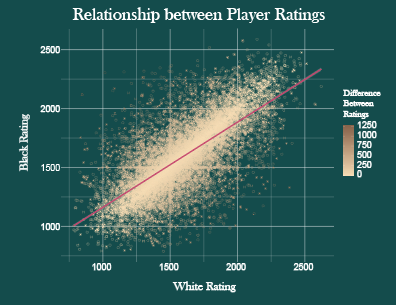


Figure 10: Two-dimensional plot – player ratings for both sides of the board.

My last two-dimensional plot showed densities between the victory status of each match and who won. For the density to work, I needed to use the jitter function in R to properly display each category. From the scatterplot below, I was able to illustrate that most game ended in resigns compared to checkmate, out-of-time, or draw.

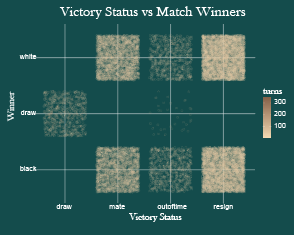


Figure 11: Two-dimensional plot – The Victory Status and corresponding Winner for each ranked chess match.

Once my plots were created, I needed to craft a layout for my poster. I split my plots into an EDA section (pie chart and histogram) on the left side and relationships section (two dimensional plots) on the right. This format allowed my audience to read my poster like they were reading a book. Starting from the left, my simple EDA plots presented some high-level data which eased my audience into the poster. As the presentation progressed towards the two-dimensional plots, I intentionally created whitespace in the layout. This blank space was necessary to let my audience rest their eyes and not be overwhelmed by all the information presented on the poster. Additional information and questions were located at the bottom as part of my poster’s hierarchy.

Besides layout and hierarchy, I needed to choose a color scheme that fit my chess theme. I chose a dark background and brighter brown and magenta colors for key areas on my poster. This contrast was used to draw the attention of my audience to specific areas of the poster, notably the two-dimensional plots.

## Linking Information Visualization Project and Program Learning Goals:

This project taught me concepts to implement into any data visualization. I learned how useful layout, hierarchy, and contrast were in presenting a data-driven story. I also developed skills in R to manipulate data and find patterns within it using visualizations, such as one-dimensional and two-dimensional plots. Finally, I developed new strategies, based on the data, which were critical to my poster. Examples included the ratings variable and rating difference between players.

## Conclusion:

The most difficult challenge I had to overcome was leveraging each analytical technique appropriately. This challenge was gradually overcome as I got exposed to the major areas of practice in data science. Taking on several different topics and data structures has also bolstered my ability to communicate my analyses in clear, concise, and thought-provoking ways. Overall, it has been a great experience to engage with the expert in the Applied Data Science program at Syracuse University and apply their teachings to real-world problems.