# Portfolio Milestone

**Syracuse University, School of Information Studies M.S. Applied Data Science**

Ryan Ondocin
SUID: 83072-6905
School Email: rjondoci@syr.edu
Data Science Portfolio Website:
https://ryanondocin2020.github.io/ryan_ondocin_portfolio/

# Table of Contents

# 1. Introduction

Over the past two years of earning my Master's in Applied Data Science from Syracuse University's School of Information Studies, I've endowed myself with the ability to analyze, and develop insights from data via multiple sources and programming languages. Courses such as Database Administration, Natural Language Processing, and Big Data Analytics, helped me hone in on these qualities due to the creation of machine learning models, reports and actionable insights via the use of SQL Server Management Studio, Python, R and MS Excel. The program curriculum is notable for providing students with the knowledge needed to generate value within any organization.

The Applied Data Science Program has the following learning objectives which will be demonstrated throughout the remainder of this portfolio. For each project I will demonstrate the ways in which each learning objective was achieved. The conclusion of this Portfolio will fully synthesize how each of the points below were executed.

1. Describe a broad overview of the major practice areas in data science.
2. Collect and organize data.
3. Identify patterns in data via visualization, statistical analysis, and data mining.
4. Develop alternative strategies based on the data.
5. Develop a plan of action to implement the business decisions derived from the analyses.
6. Demonstrate communication skills regarding data and its analysis for relevant professionals in their organization.
7. Synthesize the ethical dimensions of data science practice (privacy)

Without further ado, let's dive into each major project for all of the aforementioned courses to evaluate my completion of the program.

## 2. IST 659: Database Administration - Hospital RDBMS

a. Abstract/Description

For my final project, a Hospital database management system was organized to manage patients, billing, costs, and staffing. In the context of healthcare, workflow congestion could literally translate to life or death for patients which means there is no room for inefficiencies when establishing an RDMS. Data management is crucial for assessing patient info, arranging patient/doctor schedules and accounting for the financial needs of the involved parties. The best method for attacking this issue is to establish a database management system that can adequately deal with problems relating to file security and information retrieval/updates.

The proposed solution is to create a relational database that efficiently, securely and succinctly processes medical information. The system would be able to digitally store patient medical records, appointment schedules, medical staff directories, room assignments and billing information. This approach would be less time consuming and more secure than a file processing system which affords the user a much better experience. Automating the process would also improve healthcare management and staff/patient interactions as greater precision would be conducted in regards to planning. The patient would be able to plan their appointments, view their billing statements and access their medical records on one secure platform ahead of time. The beauty in this approach is that by minimizing logistical congestion in the hospital, you can optimize the time spent being treated and cared for by medical personnel. Our system would also allow doctors to quickly retrieve/update medical notes which can ensure that the best diagnostic practices are being put to use.

Conceptual/logical models were crafted to organize the relationships between patients, doctors and administrative members, medications, diagnostics, room assignments and billing (Fig. 1). Tables were created in SQL Server Management Studio while data population was accomplished using Microsoft Access, which also facilitated the creation of a fully functional interface for use. Reports and triggers were created to automate the room assignment process for management.Business questions were posed for this project to realize it’s full potential and utility in the context of the hospital workflow.

**Fig. 1:** Logical Model, (Ondocin, "IST 659,", 2021).

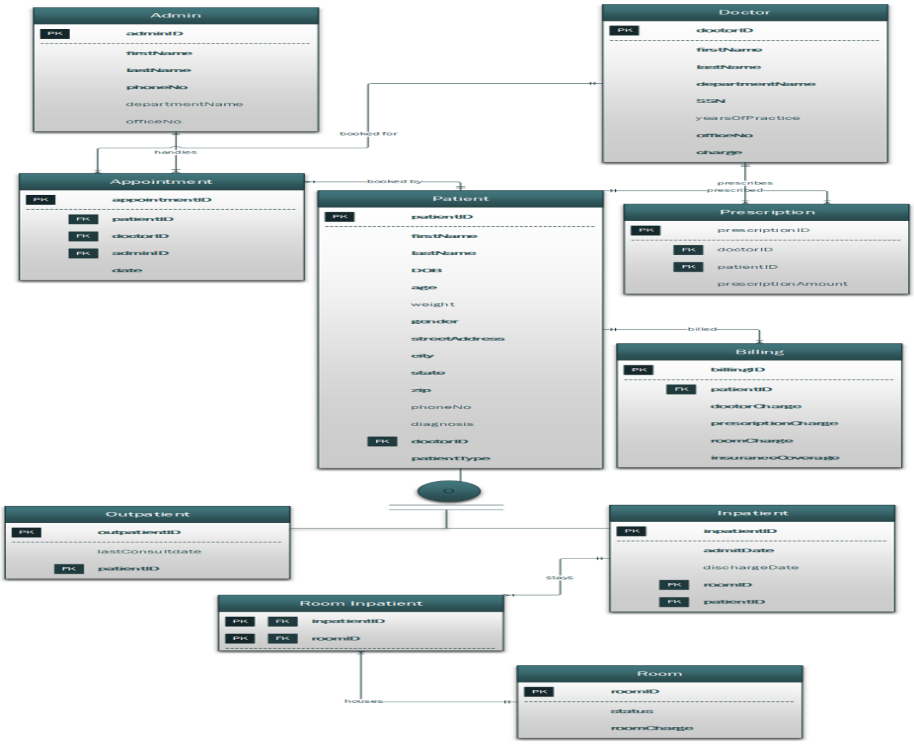



**Fig. 2:** Reports

totalBill_query_report

| billingID | patientID | patientType | billingDate | doctorCharge | prescriptionCha | roomCharge | NumberOfDays | talRoomCharge | insuranceCover | TotalBill |
|---|---|---|---|---|---|---|---|---|---|---|
| 100000111 | 1 | I | 9/9/2019 | 250 | 40 | 0 | 1 | 0 | 80 | 58 |
| 100000222 | 2 | I | 10/6/2019 | 125 | 400 | 100 | 1 | 100 | 60 | 250 |
| 100000333 | 3 | I | 10/6/2019 | 300 | 80 | 0 | 1 | 0 | 40 | 228 |
| 100000444 | 4 | I | 4/8/2019 | 530 | 0 | 0 | 1 | 0 | 10 | 477 |
| 100000555 | 5 | I | 8/18/2019 | 260 | 90 | 100 | 1 | 100 | 0 | 450 |
| 100000666 | 6 | O | 5/13/2019 | 120 | 0 | | | | 55 | 54 |
| 100000777 | 7 | O | 11/12/2019 | 1000 | 15 | | | | 100 | 0 |
| 100000888 | 8 | O | 11/11/2019 | 400 | 0 | | | | 15 | 340 |
| 100000999 | 9 | O | 11/11/2019 | 550 | 100 | | | | 45 | 357.5 |

Tuesday, December 3, 2019 Page 1 of 1

## NumberOfDoctors_report

| departmentName | CountOfdoctor |
|---|---|
| Cardiology | 1 |
| Dermatology | 1 |
| Endocrinology | 1 |
| Gastroenterology | 1 |
| General Internal Medicine | 1 |
| Nephrology | 1 |
| Oncology | 1 |
| Pharmacology | 1 |
| Pulmonology | 1 |

## dbo_ROOM_report

| roomStatus | roomID | roomCharge |
|---|---|---|
| Occupied | | |
| | 1 | 100 |
| Vacant | | |
| | 2 | 100 |
| | 3 | 100 |
| | 4 | 100 |
| | 5 | 100 |
| | 6 | 100 |
| | 7 | 100 |
| | 8 | 100 |
| | 9 | 100 |
| | 10 | 100 |
| | 11 | 100 |
| | 12 | 100 |
| | 13 | 100 |
| | 14 | 100 |
| | 15 | 100 |
| | 16 | 100 |
| | 17 | 100 |
| | 18 | 100 |
| | 19 | 100 |
| | 20 | 100 |

## Maximum_Insurance

| insuranceCoveragePercentage | patientID | firstName | lastName |
|---|---|---|---|
| 100 | 7 | Jamal | Badger |
| 80 | 1 | Timothy | Gamble |
| 60 | 2 | Chris | Richards |
| 55 | 6 | Lucy | Puro |
| 45 | 9 | Sally | Baker |
| 40 | 3 | Chase | Roberts |
| 15 | 8 | Rick | Carlton |
| 10 | 4 | Nancy | Frechette |
| 0 | 5 | Elvira | Robinson |

## Common_diagnosis_query_report

| CountOfdiagnosisID | diagnosisCategory |
|---|---|
| 2 | Hypertension |
| 1 | Hypothyroidism |
| 1 | Obesity |
| 1 | Osteoarthritis |
| 1 | Acute bronchitis |
| 1 | Allergic rhinitis |
| 1 | Anxiety |
| 1 | Back Pain |
| 1 | Diabetes |

## Maximum_Insurance

| insuranceCoveragePercentage | patientID | firstName | lastName |
|---|---|---|---|
| 100 | 7 | Jamal | Badger |
| 80 | 1 | Timothy | Gamble |
| 60 | 2 | Chris | Richards |
| 55 | 6 | Lucy | Puro |
| 45 | 9 | Sally | Baker |
| 40 | 3 | Chase | Roberts |
| 15 | 8 | Rick | Carlton |
| 10 | 4 | Nancy | Frechette |
| 0 | 5 | Elvira | Robinson |

## ExperienceDoctor_report1

| doctorID | firstName | lastName | yearsOfPractice | departmentName |
|---|---|---|---|---|
| 8 | Jeffrey | Carpenter | 23 | Pharmacology |
| 1 | Susan | Grey | 10 | General Internal Medicine |
| 6 | Phil | Kinsella | 9 | Endocrinology |
| 7 | Patricia | Smith | 8 | Pulmonology |
| 4 | Beth | Rettinger | 7 | Oncology |
| 3 | John | Noble | 4 | Dermatology |
| 2 | Chris | Billinson | 2 | Cardiology |
| 5 | Amy | Cote | 1 | Gastroenterology |
| 9 | Amanda | Shock | 1 | Nephrology |

Tuesday, December 3, 2019

Page 1 of 1

**b. Reflection & Learning Goals**

**The GUI created as a result has three levels of users:**

- **Admin: Schedule appointments/verify patient information/book rooms/administer bills**
- **Doctors: view/update personal and patient medical information**
- **Patients: view treatment notes/update personal information/view medical costs'**

While the project was an interesting exploration into maintaining the efficacy of hospital database management systems, the work was not nearly robust enough to account for all of the functions that occur in a fully operational medical unit. A few considerations that were chosen to omit are things such as hospital inventory and payroll management to name a few. Similarly, we will never be able to fully encompass all of the intricacies associated with insurance claim processing or financial assistance programs designed for accommodating certain patients due to their intrinsic complexities.

That being said, a system was created that can fundamentally deal with the central processes that are vital for clinical administrations. The scope was limited by not focusing on unstructured data such as CT/PET-Scans/X-Rays/MRIs generated from lab results because of their difficulty in obtaining and interpreting on a high-level. In order to omit unstructured data from our relational database we will solely be dealing with medical records/diagnostics in a textual form. The system was composed of doctors, patients, diagnoses, billing information, inpatient room planning and administrators. Admin will be able to administer billing statements, assign inpatients to rooms based on availability, confirm/modify/delete appointments and edit patient/staff details. Patients will be able to view their diagnosis/billing charges and schedule appointments. Doctors will be able to modify/view patient medical records, update their consultation fees and diagnose patients based on their symptoms.

Given HIPAA considerations and the confidential nature of medical information, all users will simply not have equal access to the data. Patients did not have the ability to share their medical records with outside users nor did doctors. These safeguards will protect the management system from data leaks/breaches and ensure that the most ethical practices are utilized. In order for our proposal to be helpful we are making the assumption that our 'hospital' is still using a file processing system. The data intake process is as follows. There is a medical form that is filled out ahead of time by the patient online. This form would capture the bare-essentials necessary for proper treatment to take place such as patient information and medical history(name, address, height, weight etc. etc..). Admin would be responsible for entering this information into a system that would update if the patient has visited a hospital before and be created if it's a patient's first visit. Appointments are scheduled ahead of time, via a phone call by the patient. The administrator will determine when an appointment can take place based on the doctor's availability(offline). After an appointment has occurred the doctor provides a proper diagnosis to the patient and he/she enters this diagnosis into the system effectively updating the patient's medical records. Following the visit, the service charges will be totaled and issued to the patient one day after they have left the hospital. If the patient has been admitted

then room availability is checked by an administrator and the patient is assigned a room. After the patient has been discharged their 'type' is changed from inpatient to outpatient by an administrator. The bill is administered the day after a patient has been discharged.

The exercise of developing a hospital RDBMS was an excellent exercise for illuminating how data is stored and accessed, which is obviously a critical skill for any data analyst/data scientist to have. More complex data architectures, reports and tuning metrics were additionally explored in IST 722 to assist me in making the healthiest business decisions with regards to database management systems. Automation of operational tasks such as inpatient room assignments and billing management is an easily reached goal with the skills developed in the first semester of the Applied Data Science program. This project contributed to the successful application of the learning goals through the exercise of collecting and managing data, as well as the identification of patterns using statistical analysis; these observations are leveraged to reveal insights from within the music database which are delivered using reporting tools and are easily understood by relevant professionals.

To summarize this project had a heavy emphasis on collecting and organizing patient data as well as developing a plan of action to implement the RDBMS based on the business rules developed early on in the course of this project. The video demonstration linked on my github goes in depth on different user experiences; showing how Doctors could manage prescriptions/diagnoses, Administrators could view appointments and manage inpatient room assignments and how outpatients could view their bill/their medical charts securely.

**Administration:**
https://drive.google.com/file/d/18pnaTqtVoPvsHuzG1NNKovYy4GSOtp2d/view?usp=sharing
**Doctors:**
https://drive.google.com/file/d/1SA_AUbqy6Z9ruazvaz9L4Qc_d6wCok1b/view?usp=sharing
**Patients:**
https://drive.google.com/file/d/1ork8HxD26R5_ACVRDtpicoGsrF2KhYwr/view?usp=sharing

The ethical dimension of this project cannot be overstated as we considered HIPAA compliance regulations, and SSO authentication for administrative privileges. (Ondocin, "IST 615,", 2021)

**3. Textual Clustering of Covid-19 Related Tweets**

a. Abstract/Course Achievements

During my semester studying Natural Language Processing, various data mining techniques were introduced to analyze text and develop insights from unstructured data. In the final project, coronavirus related tweets were topically clustered via the use of a variety of NLP and ML techniques, in order to offer a new lens into the COVID-19 pandemic that can help users digest information in a much simpler manner. An Interactive bokeh plot was created to aid in the understanding of information dynamics between Twitter users and the coronavirus. The

modeling process was inspired by a publication on Covid-19 Literature Clustering. Many of the ideas were molded from their works on health-care literature made available in the COVID-19 Research Dataset Challenge on Kaggle (Eren, E., COVID-19 Literature Clustering, 2020). Tweets were clustered using KMeans and Dimensionality Reduction was performed using Principal Component Analysis. This approach can be seen as a means of noise reduction amidst the deafening social buzz of the virus. Relying on metadata (retweets/mentions/location/favorites) would've deducted from the uniquely text-based approach we attempted but it could serve as a means of model validation in the future.

Tweets are highly distinguished from regular textual data given their 140 character-limit and ubiquitous use of slang, hashtags, and emojis. Furthermore, the use of embedded metadata such as hyper-links, GIFS, and images enables users to devour a higher volume of content much faster than ever before. Twitter's platform has effectively restructured the way in which we process data: rewarding users with the currency of retweets, favorites, and verification. This change calls for a few fundamentally different approaches to traditional NLP techniques to help us understand information transmission. This idea was elaborated and explored throughout the course of this review. Hyperlinks, special characters, and emojis were removed from the tweets following preprocessing. Due to the white-space generated, white spaces and newline characters also needed to be filtered out. The data was lower-cased as a means of standardizing our data to aid in initial NLP preprocessing. After the preprocessing step machine learning techniques were approached in order to devise an unsupervised learning algorithm to visualize the transmission of information. To find a low-dimensional representation of our tweets, we elected to use Tf-IDf in conjunction with t-SNE. TF-IDF was then utilized as a vectorization technique, as the algorithms require numerically based data in order to process the information conveyed by unstructured data. Dimensionality reduction via PCA Principal Component Analysis was used to reduce the dimensions of our Tf-Idf vectorized features while maintaining 90% variance( Sklearn.decomposition.PCA, 2018). This was lowered from the default value (95%) in order to obtain ~ 290 features (opposed to 379). Our data originally contained 1195 features (very high), so PCA didn't compromise the loss of too much information, luckily. 290 features is still a relatively large number of dimensions for t-SNE to process however this method was still able to remove some noisy outliers from our vectorized tweets to make the clustering process more digestible for k-means.

K-Means is an unsupervised learning algorithm so the model will not distinguish between what the true validity of our class labels actually are. The whole point of this experiment is to work without labeled data to view information transmission of tweets purely on a textual basis. Therefore, instead of validating our model on things such as LDA key-word topic modeling, we created a Bokeh plot that allows us to visually assess the coherency of clusters based on content. The elbow curve didn't yield desirable results. We don't see an extremely clear inflection point which is problematic. Instead of extensively grid searching our model, we decided to plot out the data using t-SNE to intuitively view the optimal number of clusters. This elbow curve is jagged and suggests that we may want to train the model on a different

vectorization technique in the future, such as Word2Vec. The above figure proposes that K values are optimized between 15-20 clusters. Following 20 clusters, the decrease in distortion is not as significant. We will use 17 for this model. Given that this is an unsupervised algorithm, we may have multiple optimal clusters based on when the data was scraped from Twitter [May,1]. As mentioned, we proceeded with the process and decided to see how t-SNE was handling our optimal number of clusters. t-Stochastic-Neighbor Embedding was then imported from the sklearn manifold package in order to visually map our 290 features onto a 2-dimensional plane. This is done by minimizing the KL divergence between joint probabilities of the low-dimensional (1500, 290) and high-dimensional embedding (1500,1195) ("T-Distributed Stochastic Neighbor Embedding.", 2020) t-SNE's cost function will keep most of the context-based word vectors that we want without being too large to process. t-SNE will use the original feature vector X (1500,1195) that was obtained via tf-idf on the preprocessed text.

Following this, we have a visual representation of our tweets! The goal of this technique is to reduce our high dimensional feature vector to 2 dimensions. We can plot the tweets themselves by using these dimensions as x, y coordinates. In other words, similar tweets will be closer together, and dissimilar ones farther apart. There are approximately 17 natural clusters that we can instantly recognize. This value was used to update our k-means labels to compensate for the inadequacy of our elbow-curve method. t-SNE did well in terms of dimensionality reduction, but labeled data generated by k-means could help us in opening up the hood and examining our tweets through a content-based lens. Clusters developed by k-means aided in the generation of labels to help visually separate tweets containing different feature vectors. Validation, in this experiment, is then intuitive and manual. t-SNE clusters could be compared to the color of k-means labels to search for any discrepancies. For example, there are some deviations in the label generated by k-means and tweets that are spread out in Figure 9 (see orange points w/ label = 2). This occurs because the labels and the points do not have proportionate resemblance between themselves and the higher dimensional data. This most likely signifies the computational loss we see from tf-Idf vectorization. This isn't all bad considering we simply want popular retweets to have a high concentration of coherently clustered and consistently labeled points. Points with more spread probably correspond, then, to posts that contain most of the same COVID-related keywords (death, cases, tests, trump) but fundamentally different content.

By scraping a few thousand COVID-related tweets within a specified time range we hoped to give the reader an idea of information transmission (mostly retweets). Visualizing the spread of articles/rumors/stories/opinions during this time of crisis can serve the purpose of noise reduction which could ease some of the anxiety we get from social media. The modeling process was inspired by techniques used in the COVID-19 Literature Clustering publication, authored by Eren, E. Maksim. Solovyev, Nick. Nicholas, and Charles. Raff, Edward. The primary difference in their work is its application, which clusters academic literature by topic using LDA to help health professionals keep up on field-specific information related to the virus. Their idea being that by clustering similar research articles they could help simplify the search for related

Publications. While some of the modeling and preprocessing techniques were similar to our approach, we have modified and repurposed this motivation to understand how Twitter data can be clustered within a given time frame. The clustering of tweets can be better represented in the form of a labeled plot. t-SNE was used to intuitively visualize the optimal number of clusters in our data, which is cheating the model to a certain extent, however, upon further review the dimensionality reduction steps seemed to sufficiently cluster together the information in a way that was appropriate for visualization. Following PCA and dimensionality reduction, our data was mapped from 1195 to 290 to 2 features respectively. Although t-SNE had an extremely high number of features to map on the coordinate plane, it didn't seem to compromise too much information in our data.

To reiterate the limitations of our work, the free version of tweepy only allowed us to scrape 1500 tweets every 15 minutes or so, which is a very small number of observations for a robust machine-learning algorithm to be trained on. Furthermore, retweets were displayed in a truncated mode that was followed by an ellipsis character. Additionally, it was clear that this the form of sharing was extremely popular in our data. This gave a disproportionate amount of weight to retweets given that they contained "..." even after preprocessing. In the future, more extensive data cleaning could be carried out on this work to give us a better idea of the variations and additional commentary people use for RTs. The use of tf-Idf for vectorization yielded a sparse matrix which is known to have a very high computational loss in the field of textual clustering. Perhaps sentiment analysis could've been investigated more in depth after dimensionality reduction took place but our plates were still full throughout this study. Finally, Twitter data is uniquely distinguished from regular text given its ubiquitous use of hashtags, slang, and emojis. Perhaps in the future, we could utilize a tool such as VADER as a means of processing these emojis to extract more meaningful information from the text. Even in light of all of these shortcomings, our exploration still yielded some pretty remarkable results that fascinated us time and time again. We will now open up the hood of our model and investigate our interactive Bokeh plot to make connections between the apparent clusters generated from t-SNE, the k-means labels and the tweets themselves.

Following t-SNE, we generated an interactive Bokeh plot to investigate our data. Note that this process has been refined and modified from the Covid-19 Literature Clustering challenge. Only slight changes such as the removal of the slider (# of clusters) and keywords generated from LDA were made. In the cluster highlighted in Figure 10 , there is a wide variety of similar tweets combined together. With this combination, we can easily determine and analyze the tweets which are related to Coronavirus. This dashboard will remove the need to manually scroll through tweets and provide the user with an interactive template.

As observed above, there are various coherent and incoherent structures being formed by the tweets. An observation about the coherent clusters' formation is that they occur due to the retweet of the same twitter post without any additional content. While in the case of incoherent clusters' formation, there are retweets of the same twitter post but here the retweet is attached

with some form of additional information such as text, images, or other embedded metadata. Some tweets contain a larger length of text due to which some of the text data inside the retweet is ignored. If we had access to the full ‘extended’ form of tweets, we would expect clusters to be slightly less coherent or natural, overall. Our data contains tweets from May 1st to May 3rd, 2020. If we select a different date, we would procure a drastically different data frame that would yield a completely different visualization. That being said, after running the code 40-50 times, we still observed similar patterns in the transmission of information. Most of these tweets tend to be politically focused, which is consistent with our word frequency distribution seen in Figure 6 . Trump has obviously dominated the social spotlight when it comes to COVID. When a tweet is posted by a major, verified source such as a celebrity or a politician, they have better reach and therefore have a larger cluster around them due to an increased number of retweets from their original post. The final interactive bokeh plot was posted on my github pages and is available on my data science portfolio website.

**Fig. 4.** Word Frequency

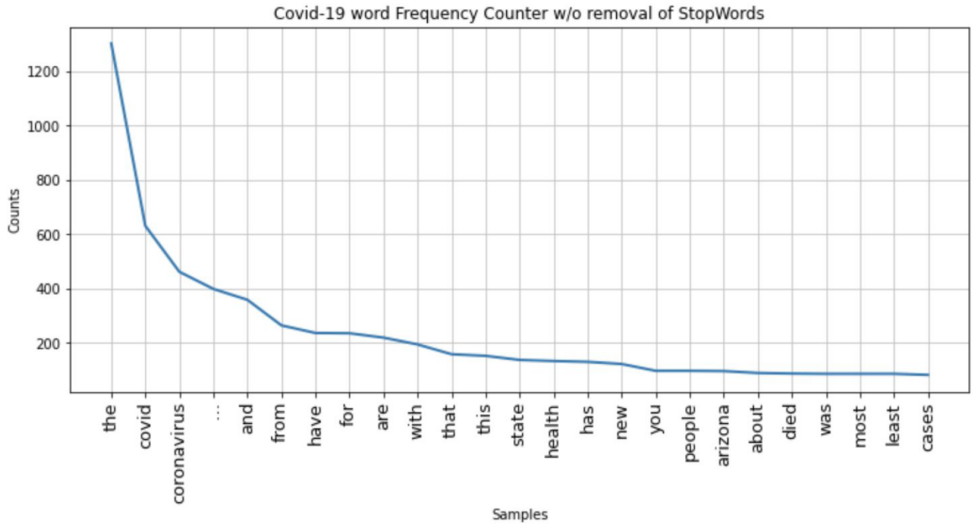


**Fig. 5.** Elbow

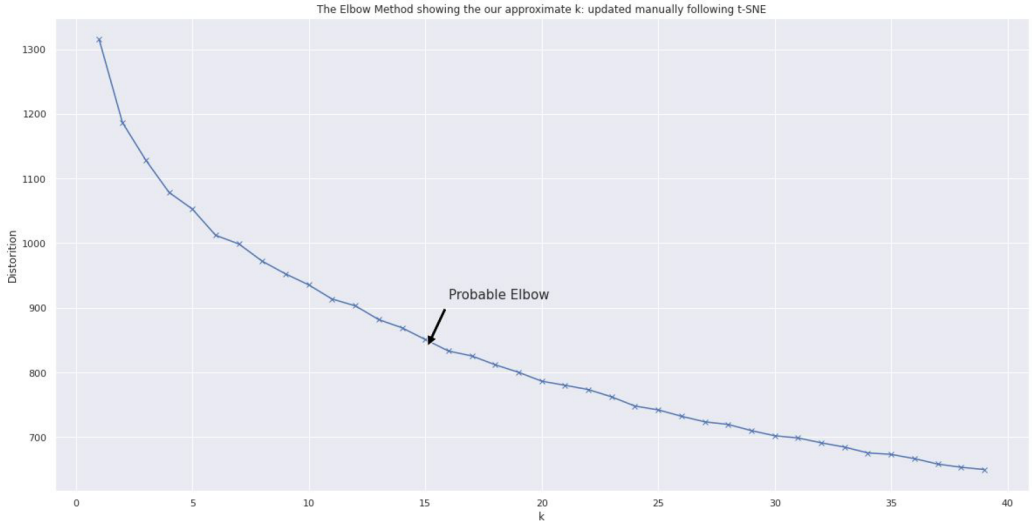



**Fig. 6.** tsne

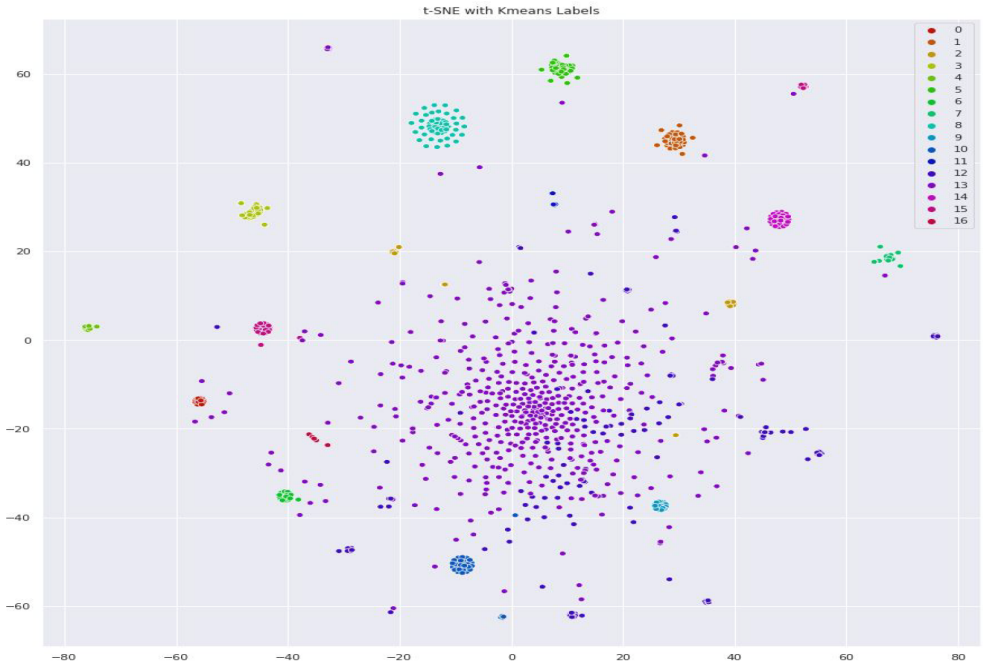
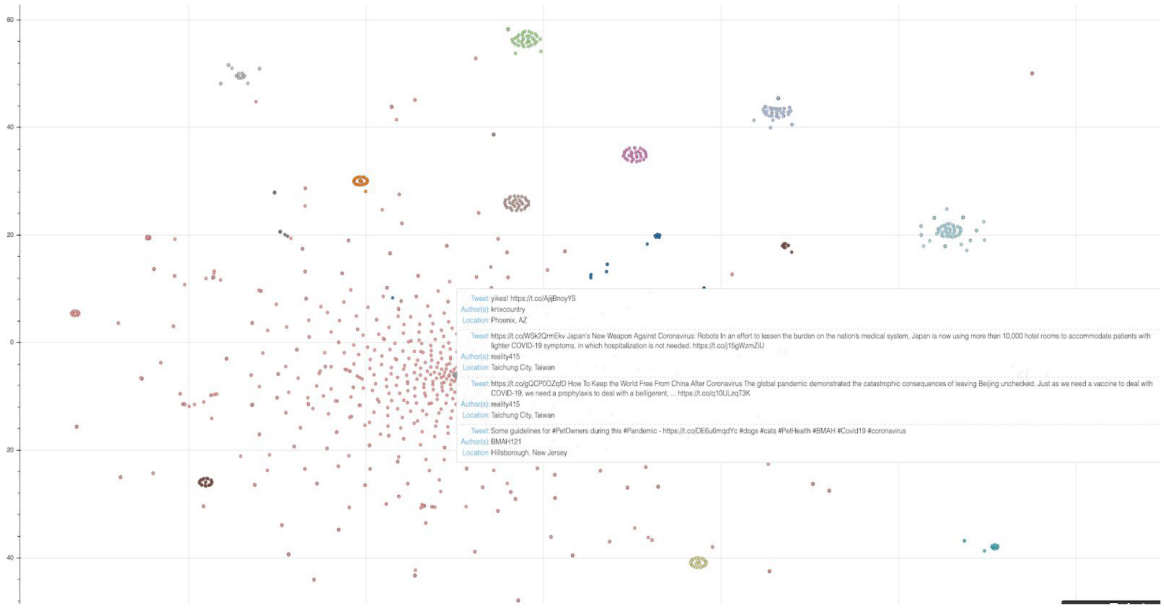


**Fig. 6.** Bokeh Plot




**b. Reflection & Learning Goals**

This exercise provided the opportunity for the collection, exploration and transformation of externally sourced twitter data via tweepy. PCA and t-SNE were useful for developing a low-dimensional representation of clustered Covid-19 related tweets which helped in developing insights into the information flow of users regarding the pandemic. (Ondocin, "IST 664,")

This project contributed to the successful implementation of the learning objectives where real-user data was scraped and modeled using natural language processing and machine learning techniques. This in turn helped to illuminate patterns and insights within the data that speak to the ecosystem of one of our most vital communication platforms.

**4. Sentiment Analysis of 2020 Election Twitter data using PySpark**

a. Abstract

Social media has played an integral role in presidential elections since 2016; with the advent of Cambridge Analytica and nuanced psycho-political campaigning tactics. Twitter can be seen as a platform for candidates and users to gain substantial outreach in order to showcase their views to the world. Thus it is important to analyze and understand the role Twitter can play in gauging sentiment surrounding hot button issues that voters use in deciding which candidate is fit for leading the United States for the next four to eight years.

The aim of this project was to perform sentiment analysis on tweets pertaining to the 2020 US Presidential election for candidates Joe Biden and Donald Trump. Our objective was to unearth contrasting information in a one week period before and after the election results were announced. Using Principal Component Analysis (PCA) and KMeans clustering, we tried to

analyze the presence of polarity or communities in the dataset. Furthermore, we performed a hashtag distribution analysis on our clusters to validate and inspect user sentiment. We applied Logistic Regression and Random Forest Classification to generate the top words, in terms of feature importance, responsible for predicting tweet sentiment and we looked for any change in the top words between these two time periods.

I also analyzed patterns of outlier users; most of which belonged to either news channels providing constant election updates, or provocative users who were supporting/slandering the two candidates. From the results of KMeans Clustering and PCA, we saw an increase in the optimal number of clusters from 2 to 3 after the election's results were announced. We feel this can sufficiently prove that before the election, the general public was focused on divisive issues that characterized users into one of two political camps. However, after the election, many new topics emerged into the picture suggesting a period of reorientation in American discourse for issues such as gun reform, voter fraud, and racial inequality. Hashtag Analysis afforded us the opportunity to analyze the discrepancies in the distribution of hashtags within our clusters. We discerned that Biden was popular on Twitter both before and after the election, which supports the idea that Twitter is a fairly left-leaning platform according to the sample's insights. Based on our modeling results, we identified words such as "corrupt, stupid, idiots, win, great", etc.. that had the highest feature importance for predicting the sentimental label of tweets. We found substantial congruity in the most important words for predicting sentiment before and after the election, which can attest to our model's ability to generalize. One of the underlying naive assumptions we made was to ignore the presence of sarcasm. When a user writes #Biden2020, we assume they are in favor of Biden but there is also a slight possibility of Trump supporters using the same hashtag with undertones of sarcasm. Therefore, in the future, we hope to implement more robust models for detecting sarcasm in the hopes of improving the model's inference abilities. We used TextBlob, a third-party tool, to generate sentiment to train our machine learning algorithms. Therefore, the results obtained from our machine learning models do have a distinct possibility of being misrepresented as the generated labels cannot be considered as 'ground truth'. Hand generated labels in this context would even fall short of this assumption because there is a degree of subjectivity that goes into the interpretation of tweet sentiment. In the future, we plan on employing several methods to accurately generate sentiment labels. Important words generated by the machine learning models cannot portray an objective picture of the sentiment surrounding this election. Therefore, we will account for bigrams and trigrams in future work as well as different vectorization techniques to avoid losing contextual information surrounding tweets, allowing for better model inference.

**b. Reflection and Learning Outcomes**

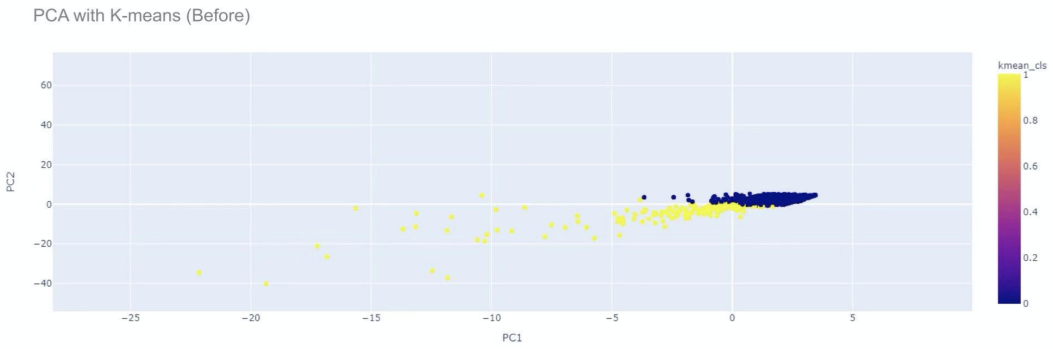
IST 718 was by far one of the most valuable courses taken at the iSchool due to its use of modern programming paradigms for solving some of today's most pressing issues. This project served as an adequate exercise in analyzing the sentiment of tweets surrounding the 2020 US

Presidential election. By scraping twitter's API for data containing search words pertaining to the candidates of the election, we obtained a sufficient number of tweets for clustering users and figuring out the most important words for predicting the sentiment. We settled on TextBlob to assign polarity and subjectivity scores to our observations which aided in our investigation of clustered data as a means of gauging a high-level sentimental overview of tweets within the platform's ecosystem. We also chose to remove retweeted data because we felt it could lead to disproportionate weights being assigned to the IDF score, leading us to make illogical inferences and conclusions with respect to label sentiment.

We then performed PCA as a means of dimensionality reduction. Proactive users who were either Pro-Trump or Pro-Biden accounted for most of the time typical outliers observed. Since we performed user aggregation and count vectorizer, its term frequency increased which pushed them apart in the PC1 and PC2 dimensions of space. Similar outliers were detected on the after election results data. Following PCA, we applied KMeans to make inferences about the nature of our clustered data. We found an increase in the optimal number of clusters from 2, to 4 which suggests that more topics were being used to characterize clusters and discussions after the election was called, whereas before it was simply left vs. right. It seems that users reoriented and opened up to other topics such as Georgia, fake elections, voter fraud, etc.

Our findings from the hashtag analysis show that Biden was leading before the election was announced but took an even larger lead in terms of frequency distributions after the election was announced which is perfectly logical given the outcome. At any rate, this was a fascinating exercise for analyzing one of the most important Presidential elections to date and we look forward to improving our code in the future. And our results, following Logistic Regression and Random Forest showed significant overlap in the most important words for determining sentiment labels some of which were win, stupid, and corrupt, suggesting the high ability for model generalization. At any rate, we enjoyed this exploration and look forward to improving our methodology to try this code on future events that are discussed on Twitter's platform.

**Figure 7.** PCA with K-means for before and after election data



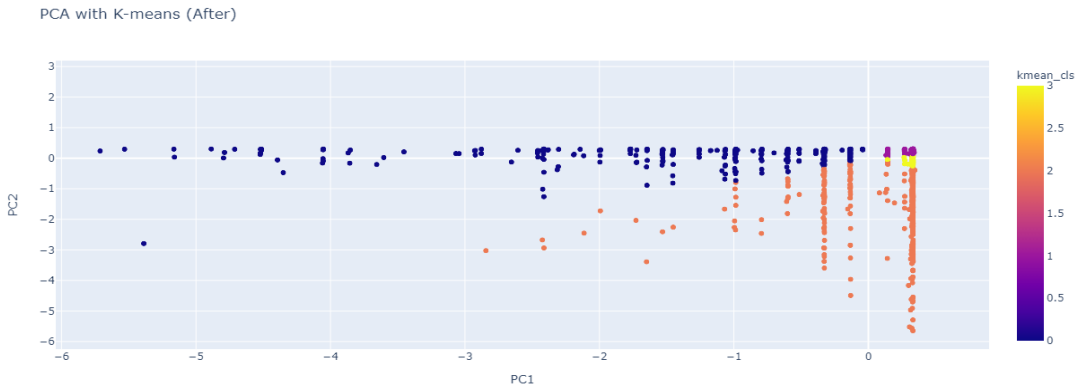



**Figure 8.** Word Importance of Logistic Regression: Before (left 2), After (right 2)

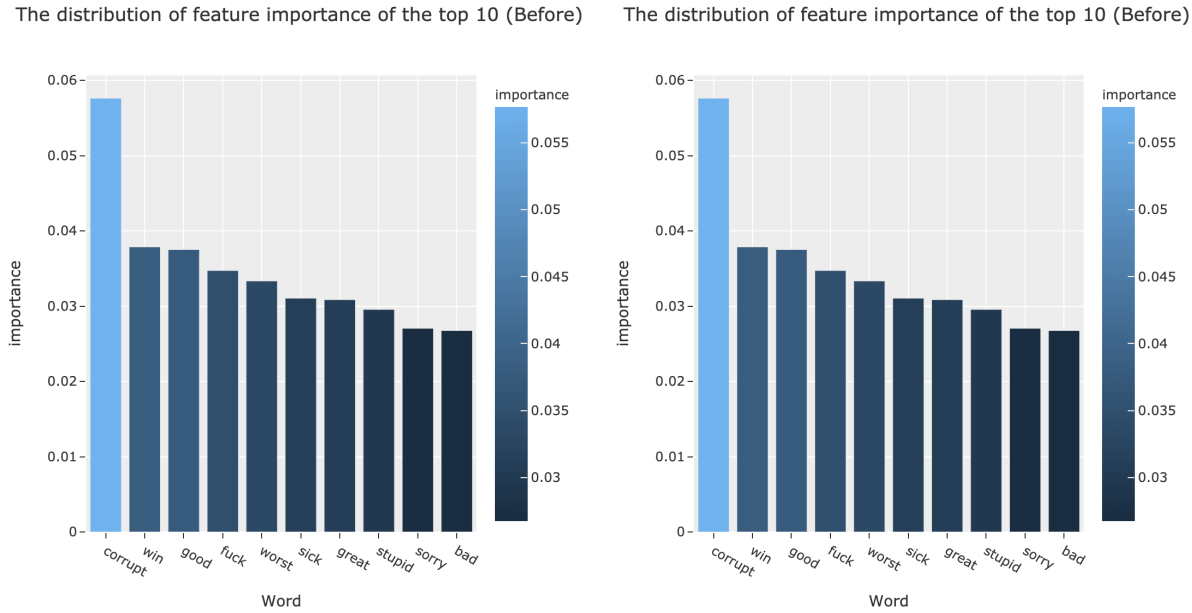
| | Negative word | score |
|---|---|---|
| 0 | worst | -0.144012 |
| 1 | corrupt | -0.132141 |
| 2 | outrageous | -0.128522 |
| 3 | boring | -0.127508 |
| 4 | insane | -0.126952 |
| 5 | pathetic | -0.124976 |
| 6 | terrible | -0.122350 |
| 7 | disgusting | -0.122205 |
| 8 | disappointed | -0.121327 |
| 9 | crap | -0.121073 |

| | Positive word | score |
|---|---|---|
| 0 | proud | 0.072794 |
| 1 | right | 0.072888 |
| 2 | nice | 0.074059 |
| 3 | lol | 0.076070 |
| 4 | good | 0.078195 |
| 5 | better | 0.081684 |
| 6 | win | 0.087409 |
| 7 | love | 0.089324 |
| 8 | best | 0.096129 |
| 9 | great | 0.101759 |

| | Negative word | score |
|---|---|---|
| 0 | fucking | -0.140101 |
| 1 | moron | -0.138700 |
| 2 | violent | -0.133689 |
| 3 | insane | -0.130664 |
| 4 | fake | -0.129026 |
| 5 | sorry | -0.127712 |
| 6 | false | -0.126255 |
| 7 | crap | -0.126070 |
| 8 | bad | -0.125830 |
| 9 | stupidity | -0.125397 |

| | Positive word | score |
|---|---|---|
| 0 | free | 0.069341 |
| 1 | interesting | 0.070580 |
| 2 | proud | 0.072433 |
| 3 | happy | 0.072708 |
| 4 | nice | 0.074199 |
| 5 | great | 0.085121 |
| 6 | good | 0.088222 |
| 7 | best | 0.088897 |
| 8 | elect | 0.089673 |
| 9 | win | 0.093586 |

**Figure 9.** Feature Importance for Random Forest



Through studying Big Data Analytics under the direction of Willard Williamson, data modeling concepts pertaining to programming paradigms such as PySpark were introduced which inspired the aforementioned final presentation. The required tools for this course go above and beyond what prospective employers seek out in future employees in the field of data science.

## 6. Conclusion

This portfolio served as an anthology of the works I've synthesized throughout my time studying Applied Data Science at Syracuse University. Each section successfully demonstrates the implementation of the crucial learning objectives that constitute the fundamental areas and major practices of data science. While data science can be described as a broad field; the skills that require mastery can be synthesized to successfully transfer any curious mind into a skilled critical thinker who is able to make data speak with grace. Database management is a fundamental system that accommodates the maintenance of rapidly changing data. For this to occur, logical models and relationships between entities need to ensure that data integrity is upheld at all times. Along the same avenue, data warehouses require pipelines that can extract data from multiple sources to be molded into an environment when analytical insights can be deduced and reported. This sort of skill falls under the subject of data engineering, allowing for the synthesis of both structured and unstructured data to be utilized and understood by a competent analytical team. Concepts of ETL in conjunction with Spark, Python and SQL constitute some of the core tools anyone dealing with a Data Warehouse must understand. These concepts were not only reinforced in IST 722 but they were made and exercised in real time with courses such as IST 615 (Cloud Management) and the execution of such methods was carried out cogently in section 2 of this portfolio with the creation of a Hospital RDBMS(Ondocin, "IST 659,").

Data Mining is also a major area of data science that was emphatically exercised throughout the curriculum of the ADS program with courses like IST 707, Data Analytics. This course emphasized the importance of extracting insights from data, void of any hypothesis, to help businesses make data driven decisions. Discovering the patterns in data allowed me to transform it into meaningful structures that aligned with business values throughout a multitude of case studies and projects. The clustering of data was an evident theme throughout this portfolio and could be seen during the textual clustering of Covid-19 related tweets (Ondocin, "IST 664,") and sentiment analysis of 2020, Twitter Based Election Data(Ondocin, "IST 718,").

Business Intelligence is another area of data science that requires one to find patterns in historical data to create concise reports that communicate actionable insights. While examples of BI aren't mentioned in this report, courses such as MBC 638 and SCM 651 reinforced this aspect of Data science.

Machine Learning, or the art of making predictions based on data lies, in my opinion, at the core of this program. After data has been cleaned/curated by a data engineer and perhaps scrutinized by a business analyst, a Machine Learning Engineer would be responsible for constructing pipelines and predictive models to help an organization make better decisions. The field can be divided into the subcategories of supervised, unsupervised and reinforcement learning. Examples of ML can be seen in both my project on Twitter Election Sentiment, and the Textual Clustering of COVID-19 tweets. This serves as an excellent segue into one of my favorite practices of data science: Natural Language Processing.

Natural Language processing can transform raw text data into vectors that can be useful in the field of predictive analytics. My COVID-19 project is a perfect example of this in which I cleaned and vectorized textual information to cluster data points on a two-dimensional plane based on similarities generated from retweets. Data was collected and managed using API's to be analyzed using state of the art statistical and data mining techniques for tasks including classification, regression and clustering. (Ondocin, "IST 664,"; "IST 718,").

Data visualization techniques using plotly, bokeh, matplotlib, seaborn and gg plot were utilized frequently with different clustering techniques in order to mine for data patterns and arrive at interesting conclusions from samples of data. When given a dataset, a competent data scientist is expected to be able to visualize the distribution of columns, detect and deal with outliers in terms of imputation or removal and generate meaningful reports to pass on to upper-levels of management. The importance of presenting your insights cannot be overstated. In some cases, organizations have difficulty building machine learning models, so instead they rely on simple and informative visualization for generating value and seeing perhaps the flow of data.

Above I mentioned only a few of the major practice areas of data science that are covered in-depth in the preceding sections of this portfolio. Deep learning, Business Intelligence and Cloud Computing are also emerging areas of the field that consider, at a large scale, how data can be efficiently accessed, stored and how computation times can be optimized.

The collection and organization of data are evident in the projects described in detail, above. During IST 659, the data was manually collected and inputted into a database with predefined fields as well as a logical model. ERD's were useful for validating users at each level of the organization and ensuring that triggers (for vacant hospital rooms for instance) were set off to ensure proper management (Ondocin, "IST 659,"). In my second project involving the textual clustering of COVID-19 related tweets; Twitter's API was scraped using Tweepy to obtain data including a range of dates and hashtags. After pulling this data into a csv; hyperlinks, special characters, and emojis were removed using Pandas during the preprocessing step.(Ganesan, K., 2019). Due to the white-space generated, white spaces and newline characters also needed to be filtered out. The data was lower-cased, lemmatized and stemmed as a means of standardizing our data to aid in initial NLP preprocessing (Ondocin, "IST664"). A very similar process was executed in the final project for IST 718, however the organization and cleaning of data was executed using PySpark. Hashtags were extracted into a new column as a means of validation for

our model results, and regular expressions were utilized to leave the alphabetical content of each tweet (Ondocin, “IST 718”).

The identification of data patterns in visualization, statistical analysis and data mining can also be seen in the last two projects. An intuitive way to detect “bot” users on social media is to plot out their tweet frequency over a small range of days. If someone were to tweet 300 times in 15 minutes, this could certainly be classified as possible bot activity. Word frequency graphs were useful for identifying the most important key words, and stop words that needed to be removed with additional cleaning. Seaborn was also utilized to create a histogram of the word count of each tweet. Twitter has a character limit, which is why the median of our normally distributed graph averaged to about 20 words and 15 unique words. KMeans clustering was implemented in both of the NLP related projects in order to identify which twitter users were most similar. When dealing with election data where the outcome of a candidate is binary, we found an optimal number of two clusters. For COVID-related tweets the elbow method, which plots distortion vs the optimal number of clusters, told us that there were approximately 15 unique groups of tweets or opinions related to the coronavirus. t-SNE and PCA were useful for reducing the dimensionality of our vectorized data set and plotting on a two dimensional plane that a human can easily interpret. The plot revealed several coherently lumped clusters that represented retweeted data (in addition to some sort of additional text). By identifying users at the center of these clusters as well as the number of followers each account had, we could create a social network of key players when it comes to spreading information relative to the coronavirus (Ondocin, “IST 664”).

Alternative strategies were iteratively updated throughout each of these projects in order to deduce analytical insights that return the highest amount of value. In regards to Twitter data, which is rife with hashtags, slang and emojis, instead of using these for sentiment analysis we decided to completely clean the data and take a uniquely textual approach to the problem.

After creating a t-SNE plot that clusters coronavirus related tweets based on similarity, a plan was implemented to create a live and interactive bokeh plot on my github so users could hover over data points and see which tweets were popular at the time as well as which users appeared most frequently in this analyses (Ondocin, “IST 664”). The purpose of this, as mentioned before is to allow users to digest social media from a birds-eye view instead of by scrolling for hours on end. The latter of which induces (as I can only imagine) anxiety and paranoia in the face of the world's largest crisis in the last 50 years or so. The plot can be accessed at the following link below:

https://ryanondocin2019.github.io/

Communication skills regarding data and its analysis was clearly developed, and demonstrated following the final presentation for each course. Technical concepts and insights were communicated simplistically in a language that everyone could easily understand in each

organization. The results of analyses were translated into actionable business decisions from conclusions that were methodically derived.

The ethical dilemmas that a data science practitioner faces are unique in every situation and must be dealt with in a manner that shows compliance with regulatory and business standards. During the construction of the Hospital RDBMS, HIPAA standards were reviewed and enforced at every level (Ondocin, “IST 659”). We ensure that only relevant data was included in the system and in analytical cases, that personal information was anonymized and scrubbed. Currently I am working as a data entry specialist at the nation's second largest Vaccination center in which electronic patient data must be entered, organized and analyzed. I utilize Tableau to forecast leftover daily dosages, and create interactive visualization dashboards that reflect, in real time, which counties across the country patients are traveling from and deactivating those spots once their appointment has been completed. Duplicate appointments were also deleted using methods learned throughout MBC 638. Before we would rely on data entry specialists to write down patients on an individual basis who scheduled multiple appointments. These slips would be collected every 15 minutes for 12 hours and deleted manually by myself (administrative privileges). Instead, I devised a method that cross references a report of completed appointments with future appointments and utilized VLOOKUP commands to automatically identify duplicate appointments and delete thousands within minutes. After each step of the process, sensitive patient data is either shredded or deleted from the system so I ensured that during each step of analysis, I was only including aggregated, anonymized data that helped administrators run the operation in the most effective way possible. This patient data also included questions and diagnostics on their medical history which could automatically disqualify them from receiving a vaccine. By generating reports with the CDMS software and parsing the data using delimiters, I was able to contact patients who would have potentially traveled for hours to reach the site to instead reschedule or cancel their appointment overall. This optimized the efficiency of the organization and increased the retention rate of appointments by over 10% by accounting for no-shows/ineligibility disqualifications. Circling back to the operational efficiency before I started working, I discovered that 30% of data entry specialists were using the current date to filter appointments. This method prevented us from detecting duplicate appointments, which is why I communicated this to the over 300 employees at the Expo center to ensure that we functioned as a well-oiled machine.

These projects in conjunction with my recent work experience is representative of the satisfactory completion of the critical learning objectives of the Applied Data Science program. They demonstrate that I have developed the necessary skills to become a successful practitioner in the field of data science. My time here has allowed me to learn how to synthesize, collect, visualize, manage and analyze data. The completion of the aforementioned steps also allows me to deliver actionable insights to any organization that I’ve ever had the pleasure of working with.

The ability to solve complex problems from both structured and unstructured data has endowed me with the competence to cultivate methodical strategies to boost operational

efficiency and management. Using the methods required in this portfolio, I've equipped myself with a tool kit that allows me to solve a wide variety of problems in a multitude of domains that can make me an asset to almost any organization.

**References**

Eren, E. Maksim. Solovyev, Nick. Nicholas, Charles. Raff, Edward, COVID-19 Literature Clustering, 2020,April, location = University of Maryland Baltimore County (UMBC), Baltimore, MD, USA, https://github.com/MaksimEkin/COVID19-Literature-Clustering

Ganesan, K. (2019, April). All you need to know about text preprocessing for NLP and Machine Learning. Retrieved November 30, 2020, from: https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

Ondocin, R. J. (n.d.). (2021) IST 659: Database Administration. Retrieved from https://github.com/Ryanondocin2019/MSADS_PortfolioMilestone/tree/main/IST659/IST659

Ondocin, R. J. (n.d.). (2021) IST 664: Natural Language Processing. Retrieved from https://github.com/Ryanondocin2019/MSADS_PortfolioMilestone/tree/main/IST664/IST664

Ondocin, R. J. (n.d.). (2021) IST 718: Big Data Analytics. Retrieved from https://github.com/Ryanondocin2019/MSADS_PortfolioMilestone/tree/main/IST718/IST718

"Sklearn.decomposition.PCA¶." Scikit, scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

"T-Distributed Stochastic Neighbor Embedding." Wikipedia, Wikimedia Foundation, 16 Apr. 2020, en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding.