Final Portfolio

Matthew Ryder

Syracuse University iSchool

Applied Data Science

[Matthew.ryder@syr.edu](mailto:Matthew.ryder@syr.edu)

SUID#884895956

Contents

[Introduction 3](#_Toc74141600)

[Hearthstone Battlegrounds Database 4](#_Toc74141601)

[Introduction 4](#_Toc74141602)

[Methods 4](#_Toc74141603)

[Results 6](#_Toc74141604)

[Conclusion 6](#_Toc74141605)

[Learning Goals 7](#_Toc74141606)

[Automating Loan Eligibility 8](#_Toc74141607)

[Introduction 8](#_Toc74141608)

[Methods 8](#_Toc74141609)

[Results 9](#_Toc74141610)

[Conclusion 10](#_Toc74141611)

[Learning Goals 11](#_Toc74141612)

[Heart Disease Prediction 11](#_Toc74141613)

[Introduction 11](#_Toc74141614)

[Methods 12](#_Toc74141615)

[Results 13](#_Toc74141616)

[Conclusion 14](#_Toc74141617)

[Learning Goals 15](#_Toc74141618)

[Spotify Song Prediction 15](#_Toc74141619)

[Introduction 15](#_Toc74141620)

[Methods 16](#_Toc74141621)

[Conclusion 18](#_Toc74141622)

[Learning Goals 18](#_Toc74141623)

[Conclusion 19](#_Toc74141624)

# Introduction

Data Science is a field which is still in infancy. The current age of data is constantly changing, and the amount of data being gathered is always increasing. In a field so young and dynamic, professionals are required to be adaptable and more important rely on a solid foundation to succeed in any environment. As the field grows and expands, more disciplines arise providing more and more opportunities for people who are skilled handling large quantities of data in any capacity. This is where the iSchool’s Applied Data Science program sets students up to succeed. As an undergraduate student studying Data Analytics, the scope of my coursework was very narrow, using very few tools and only scratching the surface of what was needed in industry. This made me want to use higher education to develop an in-depth knowledge of data gathering, cleaning, modeling, and visualization techniques. Throughout this program all of these areas are not only practiced but taught so that they can be understood and not simply recreated, allowing graduates to remain confident in a changing industry. Throughout my time in the Applied Data Science program my coursework and projects have challenged me to understand what goes on behind the scenes, not just how to write the code. As highlighted by the four chosen projects, this program has allowed me to develop a wide variety of skills, all of which will contribute to consistent growth as a professional and prepare to navigate the changing world of Data Science.

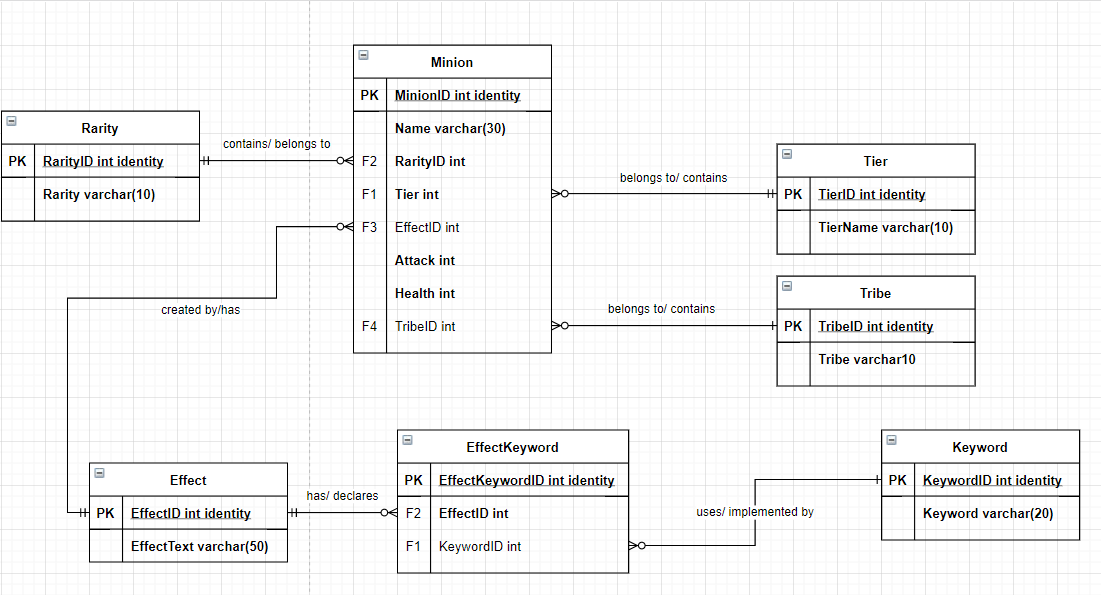
# Hearthstone Battlegrounds Database

## Introduction

The first project I have chosen to highlight was the first project I completed in the program. For this project I had built a database for the popular Battlegrounds game mode within Hearthstone, my favorite digital trading card game. My goal for the project was to use the database design principles I had learned to create a database with a user-friendly interface that could be used by players like myself to look at, and theorize the best strategies for their games, before getting into a game. This would allow for players to work to develop better strategies before playing the game, increasing their chances of winning due to their preparedness. The database was designed within SQL Server and utilized Microsoft Access for the user interface.

## Methods

First before any code could be written, business rules had to be written and a logical model had to define the schema of the database in first normal form. The business rules simply defined what features were required or could make up a single minion. For example, “All minions must have a name,” applied to every minion will not allow a new minion to be created if the *Name* field does not exist. However, there are also rules that allow minions to have features but are not required such as “Minions have an effect.” This rule did not enforce a required field in the database a certain minions may not have any effect associated with them. Once the rules were defined the following model was produced for the database schema:

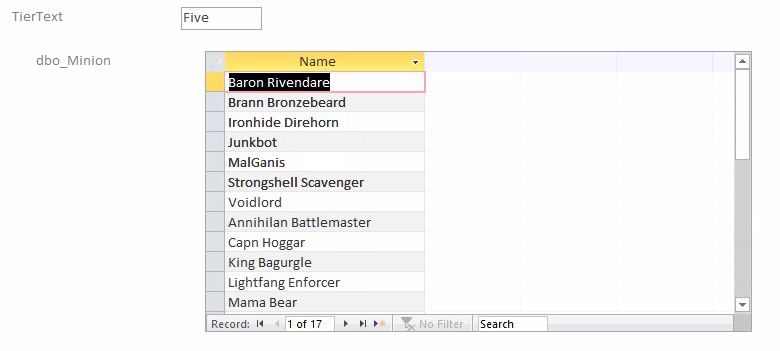


Next, using SQL the tables and their relations can be built based on the logical model above. Then when we have data that needs to be added to the minion table we can use the foreign keys which have been created to populate the Minion table. The code above added the ‘Kindly Grandmother’ minion into our database and using the foreign ID’s it is appropriately relating to all the other entities in the database. Similarly for the other entities not otherwise mentioned procedures were developed and then utilized until all the data was inputted into the database. Also, the *Minion* table had to be populated last so that all the necessary foreign keys existed when I created all the minions.

Now that all the data is in the database, the UI can be created in Microsoft Access. There are many different features of the data that player may want to see so three primary methods were built for accessing the data. First, player could simply look at all minions, with the foreign keys eliminated for their more readable values. Second, players could query by minion tier, as part of the game minions were placed in tiers so players may want to know what some of best minions are at each tier. Lastly, players could query the minions by their tribe. Tribes are one way in which similar minions synergize with one another and a player may want to play a game using exclusively one tribe, so this provides a way to look at a certain tribe of interest. Using a combination of SQL and Access the database had been loaded and was ready to be utilized.

## Results

The results of the Microsoft Access UI created an easy way for people to query and view the dataset. When players would query the data, they would be able to search for minions based on their tribe or tier. A list of all the different minion names would then be returned for the player.



Both query results successfully identify the information that the player would be looking for and fulfilled the purpose it was designed for. Using my developed ability to manage and design the database the results were exactly as I had hoped.

## Conclusion

The goals of this database were to build a database which included all the minions in the Hearthstone Battlegrounds minion pool and to be able to find minions based on Tribe, Keyword, or Tier so that player could theorize without ever playing a game. With the UI implementation in Access the appropriate forms exist so that players could do exactly that, assuming they had access to Access. I was able to implement procedures for data entry with constraints making data entry as consistent as possible. Since this data uses data from a video game, it allows for continuous updates and monitoring. As games grow and develop existing cards are removed and changed, and new cards are added. On top of those opportunities for continuous development, the Battlegrounds minions’ data does not represent the appropriate data for all game modes, and even does not cover all data in the battleground game mode itself. The database did, however, successfully show how the data could be maintained in first normal form and left a good template for adding other game mode data or simply keeping up with the many Hearthstone Battlegrounds updates.

## Learning Goals

This project helped me to develop four different learning goals. First, this project allowed me to explore the database focused area of data science. As many people rely on data, it is crucial to understand how to design a database model that suits your problem, in this case the normalized model was use. With that understanding of how the database should be designed, decisions regarding key fields and data storage can be made. Then, different layers of security can be set up by implementing user permissions and developing views and procedures which can only be accessed by certain users or user groups. Secondly, the learning objective of collecting and organizing data was explored in this project. The data for this project was manually collected and input into the database. The schema for this database was designed specifically to meet the needs of this project to ensure the data was maintained appropriately. Next, the ability to communicate what is in the data was developed using the Microsoft Access UI. This UI allowed for non-technical users of the data to have an easily digestible look at the data itself, assuming they understood the content. Lastly, some of the ethical dilemmas surrounding data privacy were a factor in this project. The data that was used is publicly available with web facing interface, and although this database did not require user specifications as I was the only user, the different procedures could have user specific permissions associated with them to ensure the inflow of data was not corrupted.

# Automating Loan Eligibility

## Introduction

The next project I will explore was during my second term in the program for Introduction to Data Science. The goal of this project was to build prediction models which can determine whether someone will be approved for a bank loan. To do this the data will contain information about the loan as well as the individual. Using multiple modeling techniques in R, different models can be built and will allow for the bank to simply approve or deny applicant based on the models and eliminate the time and cost of using people for initial decisions. This would also allow for applicants to very quick results rather than having to wait for a manual decision.

## Methods

The data utilized contained a variety of data that would be used for the loan approval process. First, there was data relating to the person who is applying for the loan such as credit score, date of birth, marital status, gender, degree, industry they work in, and their financial information. Next, there was preapproval information including the amount and term someone was preapproved for. Lastly, there was information relating to the loan itself, this included when it was applied for, the term of the loan, and amount requested. One key aspect of this data was that although each case referred directly to one person there was PII in the dataset and every person was identified with only an ID number.

To build the best model for this dataset, three different modeling techniques were utilized. These models used credit score, requested loan term, requested loan amount, preapproved loan amount, preapproved loan term, person degree type desc, actual net income, monthly debt capacity, and age to predict the approval status of a loan application. First, was the *naiveBayes* algorithm from the e1071 package in R. The Naïve Bayes algorithm provided a viable approach to this problem as it relies on conditional probabilities to make predictions on one class or another. This applies well to this problem because the data included several variables that are likely to have stronger weights than other and if, for example, we know that people with a certain education level are highly likely to be approved for loans then our model will be able to use that information to influence its decisions.

The next technique used was a random forest model from R’s *randomForest* package. The random forest algorithm can be effective in splitting the categorical so that certain groups, like the different education levels, can be evaluated differently if that variable has a high impact on the approval decision. This model also can produce strong results with very little compute power, so it is a good algorithm for most classification and regression problems.

The last algorithm was a support vector machine, using the *kernlab* package to build the model. Support vector machines attempt to build hyperplanes to separate the data and will allows for more variables to be compared simultaneously than the random forest. This model will be the model which takes the most time to train so without superior results will not be the preferred model.

## Results

The worst performing model was the naïve bayes model. The model did still provide a strong accuracy incorrectly predicting 14 out of 284 observations in the test data for an accuracy of 95%. As shown below the data the Naïve bayes model struggled most with was people with a credit score greater than 750.

Next the random forest model was the highest performing model with an accuracy of 99.29%. One of the errors for this model was denying someone with a net income of $0, even though they were approved in the testing data set, this is likely due to a data input error and is not a bad sign for our model.

The last model was the Support Vector Machine model. This model performed in between the other two with an accuracy of 97.88%. Again, the model shows errors for zero values and in this case predicts that two people with a credit score of zero to be approved for a loan, suggesting that other variables had more influence on this model, but this is an error in the model that is a cause of concern.Chart, scatter chart

Description automatically generated

## Conclusion

Overall, all 3 classifiers produced very strong results with the best model being the random forest model. With the goal of producing a model that could automate the approval process all three models showed promising results for this process. The random forest model producing the best results is understandable as the decision tree is likely a similar process to how a human would make their decisions. They would focus on one or two key variables then check the remaining the aspects of the applicant and/or the loan until a decision can be made.

## Learning Goals

The primary learning goal demonstrated in this project was the ability to identify patterns in data. Utilizing both statistical analyses, through model development, and visualizing the results to better understand the predictions from those models. Secondly, the success of the models provided a clear plan of action for implementing the models for automatic approval decisions. The results from the random forest showed that this model is clearly strong enough to be used for these decisions with only two errors on the testing data, one of which likely had a data input error. Also, using visualization to report modeling results helps to communicate those results whoever wants to understand them. By plotting the results and showing which variables impacted success can help make it clear to a subject matter expert if the model is reliable or not. Lastly, when working with personal information ensuring that people are not identifiable is of the upmost priority. In this case the dataset, as well as the variables used for the models do not constitute enough information for someone to be able to tie back to any one person, ensuring applicants information is safe.

# Heart Disease Prediction

## Introduction

Now, I will look at a more complex classification problem, from my third term in the program. The goal of this project was to utilize various classification techniques in R to predict whether someone is, or is not, at a high risk of heart disease. The motivation for this project was based on a similar research project that I attempted, with little success, during my undergrad and was hopeful that the techniques learned in this class would allow me to use similar data to better predict heart disease. This project utilizes patient information, such as blood pressure, and binary value indicating whether the patient is high risk or not, taken from Kaggle. After the data is processed, 4 different algorithms will be used to attempt to build the best predictor possible.

## Methods

Before different modeling techniques could be applied the data needed to be preprocessed. This dataset did not contain missing data that needed to be addressed, allowing the data splitting to be the only step necessary. Using random sampling, the dataset was divided into a training and testing set that will be used for every model to ensure all models are provided with the same data.

The first modeling technique used was a random forest from the *randomForest* package in R. The random forest algorithm uses a large quantity of decision trees to make predictions. Each tree can use a subset of the explanatory variables this way trees will all produce different responses, and then in the case of this problem the class that is predicted more times will be what the forest predicts. This project utilized three different random forest models with 500, 750, and 1000 trees. Using more trees should increase prediction accuracy, but after a certain point the small gains, if any, will not be worth the compute time. Using these three different values we are likely to find a strong model that requires very little time to train.

The second algorithm used was the k-Nearest Neighbors algorithm from the *class* package in R. A kNN classifier will plot the data and using distance calculations determine the class that a certain data point belongs to based on its N nearest neighbors. This problem the model is being used to differentiate between two classes so three classifiers will be built using 3, 5, and 7 nearest neighbors ensuring there will be no ties. This model was appealing to this problem as it is very plausible that there is a variable(s) that are dramatically different between people who are and aren’t high risk.

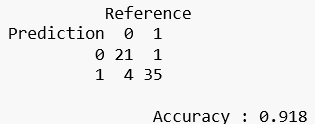
Next, Support Vector Machines were built using the *e1071* package in R. SVMs plot the data in as many dimensions as the data requires and attempts to build a single separator to isolate one class from another making it an excellent candidate for this binary classification problem. Then different models can be built which will impact the mathematical formula used to build each separator. For example, in two dimensions, a linear kernel will uses take on the form of mx + b while a polynomial kernel can use higher powers of x to build a more sophisticated separator. In this case there are 13 explanatory variables so the SVM will build four different hyperplanes using a linear, polynomial, radial, and sigmoid kernel. Each of which producing a different hyperplane and results.

Lastly, from the *caret* package the Extreme Gradient Boosting algorithm was used. This algorithm was used as it is a trendy algorithm in the data science field. XGBoost, as it is commonly known, uses gradient boosted decision tree which can produce very strong prediction results at very low computational time and often outperforms random forest models. The models generated from the xgboost, if they showed promising results would likely have the lowest compute time making them a viable option to use in production.

## Results

When evaluating these models, I used confusion matrices to evaluate overall accuracy, sensitivity, and specificity. Primarily, the focus of this problem was the accuracy, ensuring it is above the no-information rate and providing results that would not endanger the lives of patients. After, determining the model(s) with the best accuracy we would also want to consider the sensitivity, or true positive rate, to ensure that we are choosing a model that best predicts people with heart disease because it is preferred to identify more people who are high risk at the expense of people who are low risk being misclassified.

The worst performing models were the three k-Nearest Neighbor models, with the 7-neighbor model producing the strongest results with a prediction accuracy of 75.41%. Then, the Support Vector Machines performed similarly with the strongest one, with the linear kernel, having an 83.61% accuracy. In this group we also see the worst performing models which are the sigmoid and radial kernels as they both predicted every test case as being high risk. Next, the xgboost model produced a classification accuracy of 90.16%. This is a strong prediction accuracy but was outperformed by the 750 and 1000 tree random forest models. Both models had an accuracy of 91.8% and predicted positive cases 97.22% correctly and can be seen below.



Given these models high performance the best model to put into production based on these results would be the 750-tree model as it is less computationally intensive than the 1000-tree, but as more data would be collected the model would need to be reevaluated to ensure we have the best model.

## Conclusion

My goal was to produce a model that could accurately predict patients who were high risk for heart disease. Using a variety of classification techniques 11 different models were tested to ensure that the best model was being developed. Using a random forest model patients were correctly classified 91.8% of the time with the high-risk group being classified even more accurately at a rate of 97.22%. These results were an improvement of similar projects I had done in the past and utilized a variety of algorithms. To continuously improve upon this work more data would need to be collected and the models would need to be retested. Also, there is no shortage of algorithms or programming languages that could be used to attempt to improve the results. In a problem like this, the cost of failure is so high that constant monitoring and improvement of your predictions is critical when a model like this goes into production.

## Learning Goals

This project focused on the area of data science focused on predictive analytics. For many industries, the ability to use data to predict the future, or in this case diagnose patients, can provide key advantages making data modeling a key skill for any data scientist. Although for this project the data gathering process was trivial, without the data being collected and prepared appropriately the models may not have the success they did, make data collection a key step in the project. Another learning goal that this project displays is the ability to develop a plan of action to implement business decisions. By evaluating every model and using not only accuracy to determine the best model, as this problem required high-risk patients to be identified as well as possible. Lastly, using these results would be used to communicate with other business users so that managers could understand why each model was being chosen, development teams could ensure the right models were being put into production, and most importantly doctors could understand the results coming from the models and use the results to best serve the patients relying on the results.

# Spotify Song Prediction

## Introduction

The last project I will explore will utilize a variety of machine learning techniques as part of Big Data Analytics during this semester, Spring 2021 and the final code will be completed on or before June 16th and will be uploaded to the projects repository. The goal of this project was to predict popularity and genre from a song dataset. This data contained a variety of features about the song, which can be gathered from the Spotify API and information about the song. Using both regression techniques and NLP techniques the goal is to build models which can predict the popularity or genre of the song using either the lyrics or song features. Using this information artists will be able to determine what is most important for a popular song and what features have more influence in certain genres.

## Methods

The dataset for this project can be found on Kaggle, it contains the song’s name, artist, lyrics, genre, and release data. There is also a variety of audio features relating to each song, such as loudness, energy, and danceability. Then to prepare the data, 4 unique datasets were created for each of the models we are trying to build. By using unique datasets, each one can be manipulated, if necessary, without having any impact on the other models and its data. Once all the datasets are ready, models can be developed for all 4 models.

The first model will attempt to predict the popularity of a song based on certain audio features. If this model can accurately predict popularity, we will be able to inform artists of what features are important to making their songs more popular. This is very important for artists with their music on Sportify because popularity is a direct measure of the number of listens a song has and artists on Spotify are compensated based on the number of times their songs are streamed by users. To do this, three different regression techniques will be used. The first will be ordinary linear regression. OLR is a good preliminary modeling technique as it will account for certain features directly correlating with popularity. This way is higher energy songs, for example, have the highest popularity scores the regression model will make that clear to us. The next model will be random forest model. This model will allow popularity to be determined based on some combination of features, without requires correlations between popularity and an explanatory variable. Lastly, I will use and Multilayer Perceptron model. A multilayer perceptron will be the best model for identifying and understanding relationships between the different audio features. This model will likely have the longest compute time, however, requiring higher performance than other models to be worth the extra time. Overall, these three models should produce a variety of results which will give us different insights on how audio features can impact song popularity.

The next model will use the same audio features as the problem above but will instead attempt to predict the song genre. By being able to classify genre and better understand features which belong in similar groups, companies like Spotify can better build their automated playlists and recommender systems. With this being a classification problem, instead of a regression problem, different algorithms will be used. The first model will be k-Nearest neighbors. This model should be able to cluster the different genres if songs in the same genre have similar features and thus there will not be much distance between them in high dimensional space. The next model will be a naïve bayes model. The naïve bayes utilizing conditional probabilities should also work well to isolate each genre if certain values for a feature are specific to only one genre. However, this model may have low testing accuracy if the features prove to be similar amongst all genres. Lastly, we will use a Support Vector Machine. The support vector machine can build a variety of functions to build hyperplanes which can separate the data. This SVM allows for us to train and store a model which is very valuable for when a model is placed into production because the kNN model does not train and test it will simply use all the data and continuously build on itself, also known as lazy learner. The three models should all be able to determine genre which can be valuable for recommender systems.

Next, we will attempt the same two problems as above, classifying genre and predicting popularity, but this time using song lyrics. To do this the same algorithms will be used for the same respective problems. By using the same algorithms there will be very clear comparisons between the two problems and how using audio features and lyrics influence both genre and popularity. Before the algorithms can be implement on the lyrics they first need to be tokenized. By doing this the resulting data will no longer contain the lyrics but instead contain counts of each word as it occurs in the song, with the column headers being words in our dictionary. The dictionary will be all the words the occur in every song so that dataset will have zeros for each of the words that do not occur in one song but do in another. The tokenized data is required for the different algorithms which need numeric data to run and cannot use text. For predicting popularity, the same ordinary linear regression, random forest, and multilayer perceptron techniques will be used. Then for the genre classification problem we will again use k-Nearest Neighbors, Naïve Bayes, and Support Vector Machines. By using the same techniques with different data, we should be able to understand which algorithms work better with which data, which techniques work best for each problem, and it will give us a wide variety of models to choose from when determining which model is best for each problem.

## Conclusion

Overall, this project relied on a variety of machine learning techniques in python and used skills from multiple courses. By using text data and numeric data there was a vide variety of data processing and modeling techniques that impacted the project. These different models will allow for comparisons of the different tools and their effectiveness on different data and use in different problems. This project also utilized the largest dataset, in comparison to the other 3 reinforcing the necessity of efficient code and understand of your models to ensure the problems examined can be completed efficiently.

## Learning Goals

The project, I believe, developed most of the learning goals for this program. First, worked specifically on the machine learning area of data science. With this being in Python, the dominant programming language for machine learning, this reinforced techniques and knowledge used in this practice area. Next, this project used data organization techniques specific to the text data and data subsetting. Having split the data into 4 data frames so it could be manipulated to each problem’s specifications and tokenizing text data, so it was in a usable format were both different data preparation techniques which had not been used in any of my other projects. Next, the project utilized visualizations to better understand the data before beginning to use different models to solve each of our problems. Using these visualizations, you can see how the data is composed and potentially and bias that it would introduce into our models. Then, by developing 12 models in total this will create a variety of options for determining what the best way is to approach genre classification and song popularity identification. By analyzing the model results there will be several options impacting how the best model is determined. To build on that, after analyzing the models and their results we will be able to identify what model, if any, should be developed further and used to inform our decision making. Lastly, after all the models have tested and their results analyzed we will be able to communicate with different business groups how these models can be used to make better decisions going forward regarding music creation and music classification.

# Conclusion

Throughout this program, I have grown immeasurable and feel well prepared for work in the data field. Having studied data analytics during my undergraduate studies I felt very comfortable with R and statistics but did not have the knowledge you need to succeed in the field. With my main interests being in machine learning, this is the area I feel I have grown most. Before, I understood how to use an algorithm to build a model and, in some cases, I could look at the documentation and understand how change certain parameters could impact my outcome, but I never understood what they did. Having the knowledge of what these algorithms are doing behind the scenes, even if some of the math is too complicated to attempt by hand, is an invaluable skill because you can better explain to others why you chose one model over another and you are not just throwing a dart at a wall and picking whatever algorithm it lands on. Another area I feel I grew in was the database side of the field. No matter what sector of the data science field you work in you need to understand how the data is stored. Prior to program I had seen example of databases using foreign keys and referencing those keys rather than the string values in certain tables and I never understood why. I had looked at this normalized database form and thought, no one can read this why bother using at all? I have learned how to maintain a normalized model in not only a single database, but also in a data warehouse. Managing this way helps to prevent errors in data input and helps build restraints on what can be entered into the database. The last area I feel I developed very strong was in programming. For any data scientist being able to read and write code is critical. Before this program I felt I was strong in R and had a beginner knowledge of Python. Having taken several classes that used R with different professors I was shown variety of ways to do things in R. By seeing things like splitting the data into a training and testing set in different ways then I had done them before had forced me to think about why I was doing thing the ways I was, and if there was a way for me doing things more efficiently. For example, I remember being taught to use loops for a variety of tasks in R, even though this way inefficient. Then in my Introduction to Data Science we were given asked to find all rows where a column had some value and a loop is incredibly slow for this, but a simple which operator can perform the same task far more efficiently. Next, I knew very little in python, but understood basic syntax and could follow a template well enough to feel like I understood it. Through the Text Mining, Natural Language Processing, and Big Data Analytics courses I feel far more comfortable in python. I have used python for a variety of data cleaning steps which I had not before, I’ve even learned how to use APIs to get data. Even things as simple as the homework assignment in some of my classes I would not have had a clue as to how to attempt prior to this program, but now the code took less time to write than it did to run. Those three areas are the main area I feel I have grown during this program, and they all are highly transferrable to my future career. My goal is to work on machine learning model. The challenge of building a highly accurate prediction model is what got my initial interest in data in the first place. Upon completion of the program, I will begin to look for this type of role and likely my growth in Python as well as my understanding of algorithms will best prepare for this role. Lastly, this career has shown the value of always learning. When you look at the data industry, the amount of growth and change in the industry is incredible. The cost of data storage has decreased dramatically and there is no shortage of data for the foreseeable future. Then with the boom of social media there is more unlabeled text data than any other form of data and as this type of data continues to be available people will continue to develop new techniques to parse and extract value form text. Lastly, the technology and algorithms are always improving. With companies always want to get ahead and data science competitions available for anyone to join people are always coming up with new ways to model data. Most recently this development has been seen with the xgboost algorithm and its success was notable amongst different competition winners. With all these constant changes, if you are not eager to learn and grow as the field demands you fall behind and fail. Throughout the program I have learned than I could have imagined. All I have learned have prepared me for a career in the data science field and shown the importance of always wanting to learn the “new thing” because you never know when the industry will be ready to evolve.