IST 782 Portfolio Milestone

Syracuse University

The Master of Science in Applied Data Science program at Syracuse University has prepared me to enter the workforce as a data scientist. Through it, I have acquired a high level of technical skill along with an ability to reason about data, make decisions based on data, and present data; a convergence of learning that has progressively manifested itself with each course and has become more discernable as I approach graduation. At the surface of it all lies a piece of paper with my name on it but behind that is a much greater body of work. Among it, I have selected several of my top projects to unpack in this paper to holistically showcase my efficacy as a data scientist. More specifically, I will describe the applications of knowledge, utilities used, challenges that I encountered and how I overcame them, the skills that I have gained as a result, and the major takeaways as it relates to the learning objectives set forth by the iSchool faculty.

The first project that I will discuss is my final project for IST 659 Data Administration Concepts and Database Management which is an introductory SQL and database management course and one of the core classes in the curriculum that every student must take. Rightfully so, because the material taught in this course is fundamental to computer information systems and ubiquitously practical given that almost every organization nowadays relies on some kind of database to store their data. Not only that, but in a lot of cases these databases are responsible for feeding data in and out of data science pipelines and thus, makes it a fitting place to start knowing that databases are an essential part of the day to day for anybody who practices in the field of data science.

In this project, I designed and coded a relational database in Microsoft SQL Server for storing fitness related data such as exercise history and nutrition logs. Being that I am a fitness

enthusiast and certified personal trainer myself, I approached this from the perspective of a personal trainer who just signed on a new client. The goal being to store fitness data for the client in a database so that it can be used for a dashboard showing their ongoing progress. I find this to be an interesting use case because this could be helpful for keeping clients engaged and motivated. It also allows for making data driven decisions in a streamlined manner such as modifying an exercise or diet plan based on how the client is responding.

During the conceptual phase of the database development process, I defined entities and a number of business rules centered around the client training program. A couple of examples of these rules would be that the client is not permitted to workout two days in a row or that the client cannot perform more than three sets of the same muscle group in the same workout. Subsequently, the logical modeling phase consisted of mapping the relationships, forming associative entities, and normalizing the database model into third normal form. Next, during physical database design and data definition language, the entities became tables and the business rules were implemented as integrity constraints. I collected real data on myself as if I were the client and inserted it into the database via transactions and stored procedures. Lastly, I wrote select statements, indexed them, and made views from which I built a dashboard displaying summary statistics and trend charts.

Something that I realized while completing this project is the importance of flexibility in a database. As the database grows in size and complexity it can become difficult to make changes later on. This is why the database should always be designed expecting that there could be restructuring down the line. For example, new data may come to light and entirely new tables need to be added. I experienced this when I got an Apple watch mid way through

the project because I could now incorporate data like sleep quality, heart rate, step counting, or body temperature. By the same token, I designed the database for weightlifting based training but what if I decided to switch the training style to cardio or crossfit? I may have designed the database differently had I thought of this in the early stages but it was not apparent at first. These kinds of unforeseen events are bound to happen sooner or later which is why flexibility should always be a point of emphasis.

In other words, prepare for the unexpected. One way of doing this is to conduct informational interviews with future users of the database and ask them a lot of questions. Doing so can help to understand the bigger picture and perceive things through their point of view, which may uncover potential areas of concern that can be taken into consideration during database design, thus preventing oversights and ensuring that nothing slips through the cracks. Any information gathered should be recorded in the database documentation so that it can be referred to later to recall why certain decisions were made. Thorough database documentation is crucial, especially with complicated database schemas. Eventually the torch will be passed on so this makes it seamless for the next person to step in. In summary, not everything can be planned for in advance but these are a couple of measures that can be taken.

Before taking this course, it often times felt like the data I was using was coming from a black box, but I no longer feel that way now that I have an in depth understanding of databases and how they work. Furthermore, I can build my own database from scratch, optimize it, secure it, and administer it. Despite having little knowledge to begin with, the content in this course resonated with me, which is part of the reason why I chose data platforms and pipelines as my

secondary core track. At this point in the program, the groundwork has been laid and I am starting to see how the path forward is being shaped.

The second project that I would like to present is my final project for IST 707 Introduction to Machine Learning. I will call this part 1 of the project because I later expanded on it in my final project for a different course called IST 736 Text Mining which I will call part 2. But together I will consider this as one collective project. IST 707 is a course focused in the space of data mining and programming with R. Some of the algorithms that I studied in this course include logistic regression, decision tree, clustering, support vector machine, and naïve bayes. While these are some of the more basic models, they are also sometimes the most effective and are more interpretable than complicated models such as boosting models and neural networks. IST 736 crosses over with a lot of the same ideas as IST 707 but specifically with text data and instead of using the R programming language the Python programming language is used.

In part 1 of the project, the goal was to predict popularity of songs based on how they sound. The dataset that I used was made up of two components. The first component was a dataset from Kaggle consisting of thousands of songs and various attributes about them. The second component was additional metadata about the songs that I collected by web scraping Wikipedia pages. The dependent variable was the song popularity expressed as a numeric value between 1 and 100. The independent variables include song tempo, duration, loudness, speechiness, danceability, and energy, among other similar audio features meant to quantify how a song sounds. Additional independent variables that I collected from Wikipedia include

whether the song was a single, the year the song was released in, how old the artist was at the time of release, and the genre associated with the song.

Some of the questions that I wanted to answer were how popular songs have changed overtime, which attributes correlate with popular songs, and if the popularity of a song can be predicted based on its attributes. I conducted exploratory data analysis with a variety of data visualizations as well as a data cleaning phase to handle outliers and missing data before moving into a modeling phase where I trained and tested various machine learning models and tuned their hyperparameters. I evaluated the models according to their percentage of accuracy on the testing dataset and reviewed the largest errors to see if there was room for improvement. Some of the models performed better than others but they were all fairly close. The results were significant enough to indicate that it is possible to predict the popularity of a song based on its attributes.

While the results of part 1 of the project were promising, one of the obstacles that I was faced with was poor quality of data. As it turned out, the sample of songs provided in the Kaggle dataset was biased towards recently released pop songs. This means that the sample likely was not representative of the population. Additionally, the data that I collected from web scraping was incomplete. There were missing data points because some songs did not have all of the data available on Wikipedia. I was able to fill in some of the missing pieces by using techniques such as k nearest neighbors and undersampling, but the data issues ultimately ended up hindering the accuracy of the models. This just goes to show that no matter how good a model is, if the data going in is bad then the data coming out is also likely bad. There is

only so much that can be done. This was a valuable lesson for me on the importance of data quality.

In part 2 of the project, I restaged the problem as an unsupervised classification problem and tried to create clusters of songs based on their attributes. I addressed the data quality shortcoming from part 1 by collecting better data by using the Spotify API to collect a random sample of songs. I was also able to get the same audio features that I used in part 1 from the Spotify API. I gathered the song lyrics for each song by web scraping Genius.com. Based on the song lyrics, I engineered additional features in the dataset through LDA topic modeling and sentiment analysis. Each topic from the LDA model became a column in the dataset. I iterated through every word in every song and added plus one to the corresponding LDA column for whichever topic the word fell into, resulting in a number representing the degree of prevalence of the topic. The sentiment analysis was translated into one column expressed as a polarity score measuring the negativity or positivity of the song lyrics ranging from negative one to positive one.

After putting everything together, the dataset contained how the song sounds (from the audio features), what the song is about (from the topic modeling), and whether the song is positive or negative (from the sentiment analysis). In theory, all of the ingredients were there to be able to create groupings of similar songs. I used principal component analysis to compress the dataset into four dimensions accounting for approximately 80% of the cumulative variance. I used a k means clustering algorithm to do the clustering and experimented with different arbitrary numbers of clusters. My approach for deciding the best number of clusters was mainly subjective. For each number of clusters, I graphed the first two principal components on a

scatterplot and color coded the points based on their designated cluster. I chose the one with the least visible overlap between clusters as an indication that the model was able to differentiate between the clusters. Finally, I looked through the clusters and listened to the songs in them; the clusters made sense but every here and there a song would be out of place.

One of the challenges that I was faced with in part 2 of the project was dealing with ambiguity. When it comes to text data, it is not always clear what is meant by certain text. I found this to be especially true with song lyrics. Ask ten people to rate the lyrics of a song on a scale of 1 to 10 for how negative or positive it is and you will get ten different answers. This is why with text mining, most general purpose and out of the box solutions need to be further tailored for the task at hand. For example, I noticed that the sentiment analyzer I used had a tendency to interpret songs as neutral as a way of compensating for uncertainty. A better way to go about this would be train a custom classifier. Amazon Mechanical Turk is a great tool for crowdsourcing labels for data like how positive or negative a bit of text is. With that being said, the summary of my thoughts are as follows. While yes, as technology continues to evolve and natural language processing becomes smarter, ambiguity in text data is becoming less of a problem, but it still needs to be treated on a case by case basis and it is not a one size fits all.

The third project that I will discuss is my final project for a course called IST 718 Big Data Analytics. This course is a Python based programming course that traverses the end to end data science pipeline including obtaining, scrubbing, exploring, modeling, and interpreting data, with a focus on large datasets and combining data from multiple sources. I am currently in the process of completing the project, being that this is my last course in the program. At the time of this writing, however, the project is nearing its completion. The data collection and data

exploration parts are done; the data modeling is just started, leaving just the remaining data modeling and conclusion. Even though it is not completed, it is mature enough where there are insights to be drawn, so I will elaborate as much as I can on the preliminary results.

In this project, the goal is to create a forecast of future prices for over 150 different cryptocurrencies which falls into the multivariate time series regression category. Cryptocurrency is a rapidly growing space, especially as the digital push towards blockchain, web3, and the metaverse picks up momentum. And now, thanks to apps such as Coinbase, Binance, and Robinhood, it is easier than ever to trade cryptocurrency. Although cryptocurrency has become more steady recently due to an increasing number of investors, there are still risks associated with cryptocurrency, and there is a lot of debate over what will happen with the cryptocurrency market. Nonetheless, in this project, I decided to set out for myself to study the cryptocurrency market.

The data collection process has consisted of wrangling data from multiple different sources using APIs and web scraping. I started off by narrowing the scope down to just the cryptocurrencies that are tradeable on Coinbase. There is a Coinbase web page that contains an updated list of these cryptocurrencies, so the code that I wrote goes to this web page programatically and gets the cryptocurrencies on that list. Then, using the yfinance Python package, I have collected historical price data on each cryptocurrency dating as far back as the data available on Yahoo Finance. This forms the core of the dataset, from which everything else is left joined on, including additional attributes such as the circulating supply overtime, maximum supply, number of tweets, YouTube activity, and Google Trends interest level.

Time series problems can be callenging in of themselves but what makes this project challenging in particular is the volatility of the cryptocurrency market. With cryptocurrency, most of the variance is explained by randomness and less by trend, seasonality, or cycle. Although things are beginning to stabalize, the market has historically been subject to extreme fluctuations. In this situation, most forecast models will not produce a good forecast unless the historical data is properly cleansed. Another thing that can be done is to limit the history altogether to more recent and more reliable history. This is not a trivial task given that this needs to be done individually for each cryptocurrency, but afterwards, I look forward to trying out some new models that I have not used yet.

As I take a step back and reflect, in this paper, I have highlighted some of my most notable projects that define my experience as a pupil of data science and how they have shaped my learning. Each project has uniquely contributed to my learning in its own way but at the same time they have all built on eachother. Leading up to this moment, it has felt like with each one I have grown a little bit more. And not just in terms of technical skills. Having the tools at my disposal is one thing, but it is just as much about coding as it is about problem solving, communication, and leadership. I will walk away from this program as a well rounded data scientist. There is no telling what the future holds, but what I do know is that a lot of doors have been opened. It is time to close out this chapter, but the story will continue to be written.