Data Science Capstone Project Report: My Pipe Dream of Opening a NY Deli in North Carolina

***Lauren Crowe***

1. **Introduction**

* 1. Background:

I have always desired to open a NY Deli and Sandwich shop, as I grew up waitressing in one. I loved the people and the atmosphere. North Carolina is overrun with chain deli shops like Jimmy Johns, Jersey Mike’s, etc. so I would like to find a location close to family where a small business would thrive without excess competition.

I am from Greensboro, NC and have the option of opening this shop in Charlotte (where I currently live) or Greensboro NC.

Since this decision would require a substantial financial input on my part I need to determine which city and neighborhood within the city selected has reasonable rent costs. Once I select a city, I want to understand which areas of the city have the most Deli/Sandwich shops so that I can select an area that doesn't have an oversaturated market.

* 1. Problem:

There is a tremendous amount of research that could go into opening a business that would require hours of someone’s time including travel and physically visiting locations. Though I am local to NC, I do not have the time to physically drive and do manual research across the two cities of interest to determine the best location to open a NY Deli based on surrounding venues and costs. The Application of Data Science for this project aims to allow me to perform this analysis from behind my keyboard to make an educated decision without all the manual work/driving/phone calls.

* 1. Interest:

Even though this is a pipe dream of mine, anyone who is interested in opening a restaurant would benefit in leveraging Data Science to make an educated decision before selecting a location to open any restaurant or venue.

2**. Data Acquisition and Cleaning**

2.1 Data Sources:

* Web based search on Top 50 - Highest Zip Codes in NC for Rent (downloaded as csv) and imported from [this](https://www.rentcafe.com/blog/rental-market/real-estate-news/expensive-zip-codes-north-carolina/) site.
* Geospatial Data for all US Zip Codes from [this](https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/table/) site (downloaded as csv) to determine all zip codes lat/longs to be leveraged for FourSquare Venue search, and Clustering Analysis.
* FourSquare Venue search based on Greensboro Zip Code as input for Clustering Analysis.
* As a bonus I also researched best living neighborhoods in Greensboro and location (geospatial) to compare again the cluster analysis to try to select a zip code for a Deli/Sandwich shop that was also close to a good place to live (house). My web search took me to [this](https://www.sparefoot.com/moving/moving-to-greensboro-nc/the-5-best-neighborhoods-in-greensboro-nc/) site to find the top 5 neighborhoods and locations in Greensboro, NC.

2.2 Data Cleaning:

Data downloaded from the Top 50 - Highest Zip Codes in NC for Rent was pulled down directly and saved off as a CSV from the site. It was uploaded to the IBM Watson Project as a CSV in files and imported using Python code. It was sorted from highest to lowest rent cost by zip code and narrowed down to the top 20 Zip Codes by City to be visualized in a descending bar chart.

Geospatial Data for all US Zip Codes was also downloaded directly as CSV from the site. It was uploaded to the IBM Watson Project as a CSV in files and imported using Python code. The data set was reduced by filtering only on Greensboro NC zip codes and putting the data into a dataframe to be leveraged as an input for a map and Cluster Analysis. Number of venues and unique venue categories were analyzed and several venue categories like Jewelry Stores and Gas Stations were excluded because they were not pertinent to the analysis. The Greensboro zip code and nearby venue data was then transformed using onehot encoding to get a count of unique venue categories and mean frequency of occurrence for each category by zip code. The top 10 venues by mean frequency of occurrence by zip code was then analyzed, put into a pandas dataframe and sorted to derive a 1st – 10th most common venue column for each Greensboro, NC zip code. Any Greensboro, NC zip codes with NaN values for most common venues were removed.

As part of the next step of K-Means clustering analysis using the zip code & foursquare venues sorted dataframe the data was analyzed using a loop to determine the optimal number of clusters using the Elbow Method. Input of clusters from the Elbow Method was then used to create and fit a Kmean cluster algorithm with an output of an array of cluster values by zip code.

Foursquare API data was pulled using the central point of Greensboro NC as the basis for the venue search, with a limit of 1,000 and a radius of 10,000. Venue categories were determined and the json data was cleaned and structured into a pandas dataframe. A function with a loop was created to get nearby venues for all the Greensboro, NC zip codes and then merged back to the Greensboro zip code dataframe. Zip code values had to be converted back to integers to join the zip code data back to the venue sorted data with an added column to identify the cluster label. This merged data was leveraged to create a Folium map of all the Greensboro zip codes and their corresponding clusters. Some of the zip codes had identical lat long values. The Folium map and cluster analysis drove me to narrow my list of zip codes for my Deli/Sandwich shop further, and I chose to plot both the counts of occurrences of venue categories of “Deli/Bodega” and “Sandwich Place” based on the patterns of the clusters and my desire to focus in on the critical few zip codes I had narrowed my search to. To do this I created two new dataframes to capture and visualize Total Count and Mean Frequency of occurrence for the sum of “Deli/Bodega” and “Sandwich Place” by zip code to visualize.

2.3 Feature Selection:

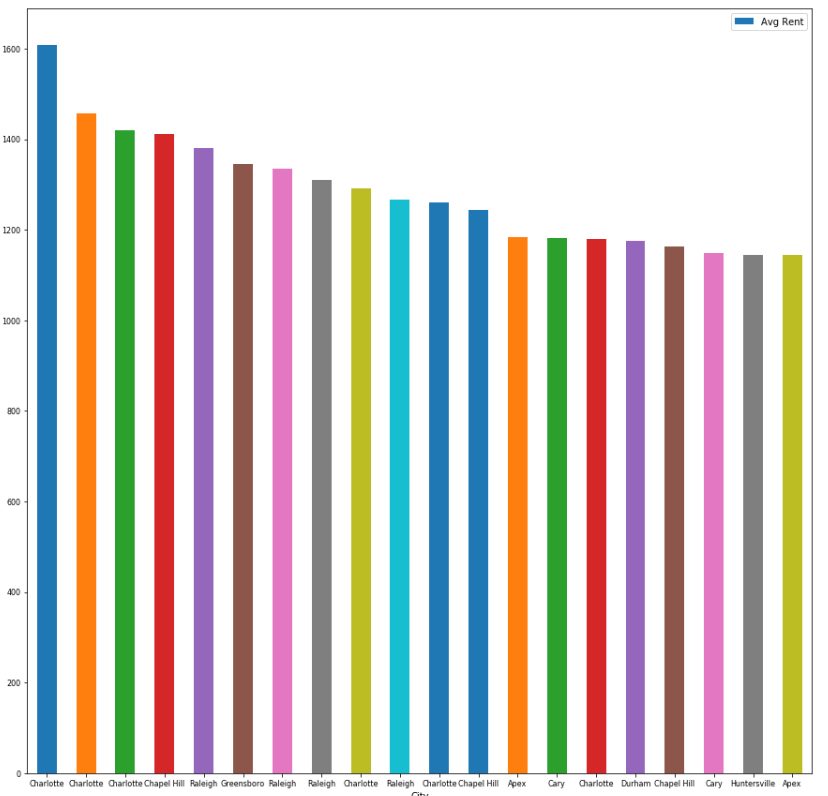
The most critical features for this clustering analysis came from the following data attributes across data sources:

* Greensboro NC Zip Code at/long attributes
* Nearby Venues and Venue Categories for each Zip Code from Foursquare API

3. **Exploratory Data Analysis:**

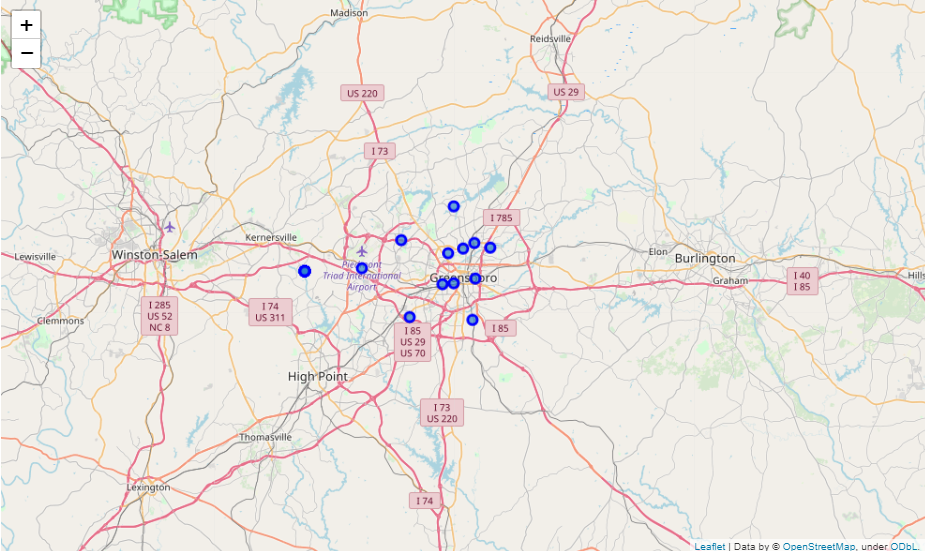
3.1 Visualizing the top 20 Zip Codes in NC proved a good exercise where it was found that 7 of the top 20 Zip Codes for Rent Cost are in Charlotte, and 1 is in Greensboro. Considering this data it was easy to make the determination to focus my analysis on a location for my Deli/Sandwich shop in Greensboro, NC and chose to avoid zip code 27403 in Greensboro.

Figure 1. Descending Bar Chart Top 20 Zip Codes for Rent Cost by City



3.2 Finding all the Greensboro, NC zip codes was critical to my Foursquare venue search as these were the “landmarks” I needed to search and cluster the areas of Greensboro, NC by most common venue types.

Figure 2. Folium map of all Greensboro Zip Code Markers



3.3 Tying the Greensboro zip code geospatial data to the Foursquare venue data in a dataframe was critical to allow for levering K-means clustering unsupervised learning techniques and visualizing most common venues by zip code to narrow my search to an area not over populated by other Deli/Sandwich shops.

Figure 3. Head of Merged Zip Code and Venue Category and Lat/Long Dataframe.



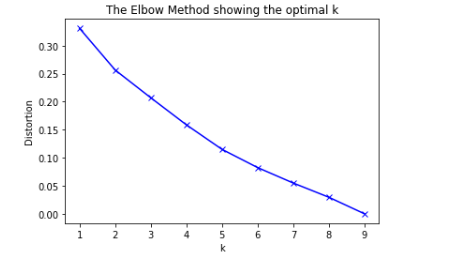
3.4 Creating a dataframe leveraging one hot encoding to determine the top 10 most common venues per zip code based on mean frequency of occurrence allowed me to visualize the need to go back and exclude certain venue types like “Gas Station” and “Jewelry Store” from my Foursquare output, as it was the 1st and second most common venue across most zip codes and was NOT the target of my analysis. Upon excluding the irrelevant venue categories and re-running the onehot encoding and creating the dataframe I had a dataframe ready for Cluster Analysis.



Figure 4. Dataframe of Zip Code and Venue 1st – 10th most common venues

3.4 Elbow Method revealed that 2-3 clusters were optimal for the K-means fit algorithm. Note that there is no sharp elbow in this case so K-means was ran under both 2 & 3 clusters and the cluster outputs were analyzed. Based on practicality and # of zip codes I was analyzing I chose 3 clusters.

Figure 5. Output of K-Means Loop to determine number of clusters for K-means fit



3.5 Cluster Analysis and Map revealed patterns as follows. Cluster #2 for Zip Code 27402 had the most common venue type as a “Sandwich Place” so I immediately ruled it out. Cluster #1 and #0 along with corresponding zip codes reveled other restaurant types in their top 10 venues, so I didn’t rule them out immediately and chose to analyze some further where the top 5 venues didn’t include a “Deli/Bodega” or “Sandwich Place”. The most common restaurant type across 1-10 and neighborhoods was a Pizza Restaurant, which poses little conflict since I would be opening a different restaurant type. K-means cluster analysis narrowed down my search to the following zip codes: 27412, 27410, 27408, 27420, 27403, 27407.

Figure 6. K-Means Cluster Folium Map (Green = Cluster 2, Purple = Cluster 1, Red = Cluster 0)

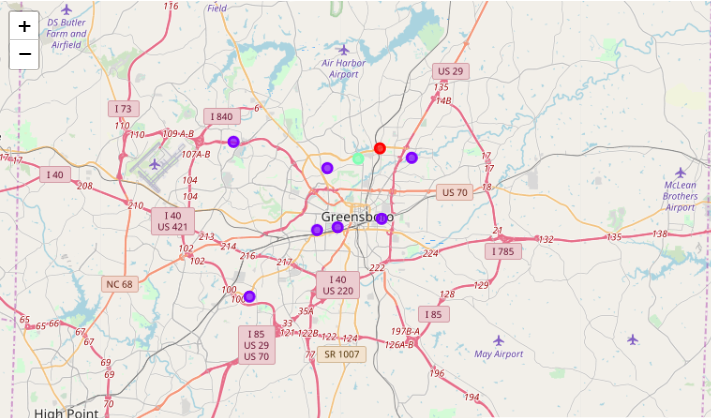


Figure 7. K-Means Clustering DataFrame with Zip Code, Cluster Label and Top Ten Venues by Venue Category



3.6. Narrowed search to select zip codes and plotting of mean frequency of “Deli/Bodega” and “Sandwich Place” aggregated revealed zip codes where there was little/no presence of the type of restaurant I wanted to open. This allowed me to narrow my search even further.

Figure 1. Rank of delta-win-share of Most Improved Players winners among all players of each year

Figure 2. Box plot of improvement of players of different ages.

3.3 ​ ​Relationship between improvement and overall ability The hypothesis here is that players who are already stars don’t have much room to improve, while a mediocre player can still improve. Our data were consistent with this hypothesis.Using win share per 48 minutes (WS/48) as a measure of a player’s overall ability, I observed a negative relationship between a player’soverallabilityandhisimprovementnextseason(Figure 3). The mean improvement of star players (WS/48>0.2),solidplayers(WS/48between0.1and 0.2), rotational players (WS/48 between 0 and 0.1), and “scrubs” (WS/48 below 0) were significantly different from each other (z-test, p<0.001) (Figure 4).

3.4 Relationship between improvement and minutes played I hypothesized that players with less playing time might be more likely to improve. ​If a team recognizes a player's positive contribution during his limited time, he is likely to get more playing time, and therefore increase his production and/or improvehisskills.Ontheotherhand, if a good player is already a starter, he is already playing a lot of minutes and can't get more playing time.Afterinspectingthedata,itwastruethatplayerswhoplayedlessthan25minutesa game had statistically higher improvement than those who played more than25minutesagame (z-test, p<0.001). However, the actual difference of mean between the two groups was small (~0.7).

3.5 Relationship between improvement and games played I observed anegativerelationshipbetweenplayerimprovementandthegamesplayed(Figure5). If a good player missed significant numbers of games, it was probably because ofinjury,which might have negatively impacted his performance. He might return to his former form next season, and therefore improve. Players who played fewer than 50 games were more likely to improve than those who played more than 50 games. (z-test, p<0.001, difference of mean=1.3).

Figure 3. Scatter plot of improvement and player overall ability (measured by win share per 48 minutes)

Figure 4. Histogram of player improvement separated into 4 groups based on how good a player is.

Figure 5. Scatter plot of player improvement and games played.

3.6 Relationship between improvement and positions There is this myth among NBAfansthatfrontcourtplayerstakelongertoadapttotheNBAthan backcourt players, therefore they would have smaller improvement in the first few years. I transformed the feature of player position intoabinaryfeature(frontcourtvs.backcourtplayers) and found that there was no difference between frontcourt and backcourt players in their improvements, even in their first 2 years (z-test, p=0.34)

3.7 Relationship between improvement and last year’s improvement I hypothesized that a player’s improvement might be correlated with hispreviousimprovement, because younger players might improve continuously for a few years, and older players might decline for a few years straight. It turned out that the relationship between improvement and prior improvement was negative (Figure 6). In other words, more often than not, a player will “regress to the mean” rather than continuously improve or decline.

Figure 6. Scatter plot of player improvement and that of last season

3.8 Relationship between improvement and draft positions I, as many other basketball fans, thought that players drafted earlier are generally moretalented and therefore more likely to improve than players drafted later, at least in their early years. It turned out this was only true for a few really young and talented players (Figure 7) . Players under the age 20 with different draft positions did not have statistically different improvement (z-test, p=0.16).

3.9 Relationship between improvement and teams I engineered two features based on team information: was a player on a good or bad team, and did the player change team next season. Player improvement and team strength (measured by total win shares) hadaveryweaknegativerelationship.Playersthatchangedteamswereslightly more likely to improve than players that stayed on the same team(z-test,p<0.001,differenceof mean = 0.2).

Figure 7. Box plot of player improvement among different draft groups and ages

4. Predictive Modeling

There are two types of models, regression and classification, that can be used to predict player improvement. Regression models can provide additional information on the amount of improvement, while classification modelsfocusontheprobabilitiesaplayermightimprove.The underlying algorithms are similar between regression and classification models, but different

audience might prefer one over the other. For example, an NBA team executive might be more interested in the amount of improvement (regression models), butageneralNBAfanmightfind the results of classification models moreinterpretable.Therefore,inthisstudy,Icarriedoutboth regression and classification modeling.

4.1 Regression models 4.1.1 Applying standard algorithms and their problems I applied linear models (linear regression, Ridge regression, and Lasso regression), support vector machines (SVM), random forest, and gradient boost models to the dataset, using root mean squared error (RMSE) as the tuning and evaluation metric. The results all had the same problems. The predicted values had much narrow range thantheactualvalues(Figure8),andas a result, the prediction errors were larger as the actual values deviatedfurtherfromzero(Figure 9). These results were not acceptable, because players with large improvement/decline were arguably more important for NBA teams to predict than players with little change in performance. Having larger errors on those predictions was obviously not desirable.

4.1.2 Solution to the problems The reason behind these problems were the uneven distribution of player improvement, in that players with little improvement/decline were more common than players with big improvement/decline (Figure 8). Therefore, the models tried to prioritize minimizing errors on players with little improvement/decline when RMSE was used as the evaluation metric. My solution to thisproblemwastoassignweightstosamplesbasedontheinverseoftheabundances of target values. In other words, players with large improvement/decline would have higher weights in model training and evaluation because they were more rare. Using this method, all models predicted target values with similar range and distribution as the actual target values (Figure 10).

Figure 8. Distribution of actual and predicted improvement using linear regression with equal weights of samples.

Figure 9. Scatterplot of prediction errors vs. actual target values using linear regression with equal weights of samples.

Figure 10. Distribution of actual and predicted improvement using linear regression with different weights of samples based on inverse of sample abundance.

4.1.3 Performances of different models Using the new approach of different sample weights, I built linear regression, SVM, random forest, and gradient boost models using weighted root mean squared error as the evaluation metric. For each model, hyperparameters were tunedusingthesamemetricandcrossvalidation. For comparison, I also built a simple linear regression model withjustoneindependentvariable (age) as the benchmark model. SVM had the best performance among all models, which had ~26% less error than the benchmark model (Table 2). The predicted improvements had linear relationship with the actual improvements (Figure 11).

Table 2. Performance of the regression models. Benchmark (one feature) Linear Regression SVM Random Forest Gradient Boost

Weighted RMSE 3.84 2.98 2.86 2.93 2.96

4.2 Classification models The application of classification models was much more straightforward. I dividedthesamples into two classes (improvement>=0 or <0). The number of samples in each class were about the same.Ichoselogarithmiclossasthemetricherebecausetheresultswouldprobablybepresented with probabilities and logarithmic loss puts more emphasis on the probabilities than other metrics. Logistic regression, SVM, random forest, gradient boost models and a voting model weretunedandbuilt.Amongtheindividualmodels,theSVMmodelperformedthebest(~67.5%

accuracy), and voting model performed similarly as the SVM model (Table 3), though the differences between models were small.

Figure 11. Scatter plot of predicted and actual player improvements of the SVM model.

Table 3. Performance of classification models. Best performance labeled in red. Logistic Regression SVM Random Forest Gradient Boost Voting Model

Log Loss 0.605 0.603 0.612 0.613 0.603

Accuracy 0.675 0.675 0.672 0.672 0.675

No. of True Positives 835 830 810 815 838

No. of False Positives 413 406 396 400 416

No. of False Negatives 438 443 463 458 435

No. of True Negatives 929 936 946 942 926

Figure 12. A section of ROC curves of different classification models.

I also evaluated the models using their ROC curves. In this particular problem, lower false positive rate is moreimportantthanhighertruepositiverate.Inotherwords,itismoreimportant to be sure that a player will improve as predicted, rather than predict all players who will improve, simply because a team can only have limited number of players. In the ROC curves with low false-positive rate, the voting model had slightly higher true positive rates than other models (Figure 12).

5. Conclusions In this study, I analyzed the relationship between NBA players’ improvement/decline and their performance and biographic data. I identified age, win share, minutes/games played, improvement last season among the most important features that affect a player’s improvement next season. I built both regressionmodelsandclassificationmodelstopredictwhetherandhow much a player would improve/decline. These models can be very useful in helping NBA team management in a number of ways. For example, it could help identify players to acquire, estimate the sizeofthecontracttoofferplayers,planforperformancechangesofplayersalready on the team, etc.

6. Future directions I was able to achieve ~26% improvement from the benchmarkmodelintheregressionproblem, and ~68% accuracy in the classification problem. However, there was still significant variance that could not be predicted by the models in this study. I think the models could use more improvements on capturing players’ individual traits. For example, two players might have similar performance metrics, but one might be more physical and the other might be more finesse. The future performance of these two types of players might be different. Another example is that players whose contracts are expiring might play harder/better than players who just signed hefty contracts. More data, especially data of different types, would help improve model performances significantly. Models in this study mainly focused on individual features. However, interactions with teammates, coaches, might also contribute to a player’s performance. For example, if a player had a new teammate who is a superstar at the same position, his performance is likely to suffer because of competition. These interactions data are obviously more difficult to extract and quantify, but if optimized, could bring significant improvements to the models.