Peer-graded Assignment:
Capstone Project - The Battle of Neighborhoods

Phung Hoan Anh NGUYEN

1. Introduction

New York City's demographics show that it is a large and ethnically diverse metropolis. It is the largest city in the United States with a long history of international immigration. New York City was home to nearly 8.5 million people in 2014, accounting for over 40% of the population of New York State and a slightly lower percentage of the New York metropolitan area, home to approximately 23.6 million. Over the last decade the city has been growing faster than the region. The New York region continues to be by far the leading metropolitan gateway for legal immigrants admitted into the United States.

Throughout its history, New York City has been a major point of entry for immigrants; the term "melting pot" was coined to describe densely populated immigrant neighborhoods on the Lower East Side. As many as 800 languages are spoken in New York, making it the most linguistically diverse city in the world. English remains the most widely spoken language, although there are areas in the outer boroughs in which up to 25% of people speak English as an alternate language, and/or have limited or no English language fluency. English is least spoken in neighborhoods such as Flushing, Sunset Park, and Corona.

With it's diverse culture , comes diverse food items. There are many resturants in New york City, each beloning to different categories like Chinese , Indian , French etc.

So as part of this project , we will list and visualize all major parts of New York City that has great indian resturants.

2. Data

Data For this project we need the following data :

New York City data that contains list Boroughs, Neighborhoods along with their latitude and longitude. Data source : https://cocl.us/new_york_dataset Description : This data set contains the required information. And we will use this data set to explore various neighborhoods of new york city. Indian resturants in each neighborhood of new york city. Data source : Fousquare API Description : By using this api we will get all the venues in each neighborhood. We can filter these venues to get only indian resturants. GeoSpace data Data source : https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm Description : By using this geo space data we will get the New york Borough boundaries that will help us visualize choropleth map.

3. Approach

Collect the new york city data from https://cocl.us/new_york_dataset

Using FourSquare API we will find all venues for each neighborhood.

Filter out all venues that are Indian Resturants.

Find rating , tips and like count for each Indian Resturants using FourSquare API.
Using rating for each resturant , we will sort that data.
Visualize the Ranking of neighborhoods using folium library(python)

4. Data acquisition and cleaning

4.1 Data sources

Most player stats, position, age, and draft position data can be found in two Kaggle datasets here and here. These two datasets, however, lack data for certain years. For example, the player stats dataset ends in 2017, and the player draft dataset starts in 1978 and ends in 2015. To complement these two datasets, I scraped basketball-reference.com for player season stats of 2018 and player draft positions of 1965-1977 and 2016-2017

4.2 Data cleaning

Data downloaded or scraped from multiple sources were combined into one table. There were a lot of missing values from earlier seasons, because of lack of record keeping. I decided to only use data from 1980 season and after, because of later seasons have fewer missing values and basketball was a lot different in the early years from today's game.
There are several problems with the datasets. First, players were identified by their names. However, there were different players with the same names, which cause their data to mix with each other's. Though it was possible to separate some of them based on the years, teams, and positions they played, I decided that it was not worth the large effort to do so, because such players only accounted for ~1% of the data. Therefore, players with duplicate names were removed.
Second, multiple entries existed for players who changed teams mid-season. This cause their seasonal data to represent multiple samples with incomplete data. I wrote script to extract total season stats for these players, and discarded partial season rows.
Third, there were two short seasons in recent NBA history, during which less than the normal 82 games were played. This has caused stats in those seasons to be artificially smaller than other seasons. To correct that, I normalized cumulative features such as points, rebounds, etc. as if 82 games were played.
After fixing these problems, I checked for outliers in the data. I found there were some extreme outliers, mostly caused by some types of small sample size problem. For example, some players had only played a few games or a few minutes the entire season, and had performed extremely well or poor in those minutes. Therefore, seasons during which less than 20 games or 100 minutes were played were dropped from the dataset. Similarly, there were players who only took one 3-point shot, but made it, therefore had 100% shot accuracy. I changed the shot accuracies for players who shot less than 10 shots to missing values.
There were 4 features which had missing values. Games started were imputed from minutes played because starters usually play more minutes. Missing 3-point accuracies were imputed with a very small value (0.05) because if a player rarely shoots 3s, it is probably because he is not very good at it. Missing free throw accuracies were imputed using the mean of all players. Missing draft positions, meaning undrafted, were imputed using position 61 (the position after the last position in the draft, 60th).

4.3 Feature selection

After data cleaning, there were 13,378 samples and 49 features in the data. Upon examining the meaning of each feature, it was clear that there was some redundancy in the features. For example, there was a feature of the number of rebounds a player collected, and another feature of the rate of rebounds he collected. These two features contained very similar information (a player's ability to rebound), with the difference being that the former feature increased with playing time, while the latter feature did not. Such total vs. rate relationship also existed between other features. These features are problematic for two reasons: (1) A player's certain abilities were duplicated in two features. (2) A player's playing time were duplicated in multiple features. In order to fix this, I decided to keep all features that were rates in nature, and drop their cumulative counterparts (Table 1).

There were also other redundancies, such as that total rebounds are the sum of offensive rebounds and defensive rebounds. For features that can be calculated by sum of other features, I decided to drop them (Table 1).

After discarding redundant features, I inspected the correlation of independent variables, and found several pairs that were highly correlated (Pearson correlation coefficient > 0.9). For example, shots attempted, shots made, and points scored were highly correlated. This makes sense, after all, you score points by making shots. From these highly correlated features, only one was kept, others were dropped from the dataset. After all, 24 features were selected.

Slightly different features that depict the same overall abilities of players.

5. Exploratory Data Analysis

Player improvement year over year was not a feature in the dataset, and had to be calculated. I chose to calculate the difference of win shares between two consecutive years as the target

variable. Win shares were chosen out of a few metrics because it is the most interpretable, after all, we play basketball to win. Calculated player improvement had a normal distribution centered around 0, with most values between -6 and 6. To verify if this calculation is consistent with people's eye-test of player improvement, I plotted the rank of improvement of past Most Improved Players winners among all players, and found that in most cases, they were among the most improved players (Figure 1). This suggested that the chosen metric of player improvement, was a reasonable one.