

Rec A Business

A capstone project for the new business enthusiasts



Developed by – Sourin Roy

MailId – roysourin99@gmail.com

Reach me here



Kernel link – <https://www.kaggle.com/sourinroy/rec-a-business>

Report link –

Presentation link -

Introduction / Business Problem

“Rec a Business”, as the name suggests recommends a business. To put it simply, it just gives you a list of possible options one could leverage while starting a new business. This is location specific and aims to recommend the best possible business option that could help a new and enthusiastic businessman start a healthy, financially outgrowing and stable business. Alternatively, if the type of business is specified “Rec a Business” recommends the best possible location where that specific business could be set up.

New business ideas often face tough competitions from old and solid competitors, which in turn can be harmful for the new business, resulting in bankruptcy or financial instability very early in the business tenure. To overcome this, “Rec a Business” aims to provide one with the best possible location(s) to start with a new business where there is less competition. The location is selected on the basis of distance from the neighbourhood the person belongs to.

Often, it also happens, that new business enthusiasts want to set up some business in specific location but they are unsure about which business to start. “Rec a Business” also looks forward to provide one with a list of possible business options that they can start investing upon.

“Rec a Business” is meant for new business enthusiasts and aims to provide an interactive, easy and free way to get top recommendations of places or business. This project would be specifically using the location data of New Delhi, India.

Dataset(s) and Data gathering

For this project, the location data used is of New Delhi. The location data of various neighborhoods along with their borough and latitude, longitude is provided. The credit of compiling this dataset namely “**delhi_dataSet.csv**” goes to **Mr. Kumar Shaswat**, and I hereby am using this dataset for my personal project in accordance with the license provided i.e. CC BY-NC-SA 4.0. The complete dataset along with other kernels can be found on the following link - <https://www.kaggle.com/shaswatd673/delhi-neighborhood-data>. The license distribution can be found here -<https://creativecommons.org/licenses/by-nc-sa/4.0/>. The original dataset has been briefly edited and modified to help with the project. I would like to thank the creator of this dataset for having created this wonderful dataset.

Along with this dataset, I would also be using Foursquare API, and its places SDK to leverage the realtime venue data in and around the neighborhoods of Delhi. The places SDK provided by Foursquare API is free to use along with certain restrictions and can be used to get the venue data around a particular location specified by latitude and longitude. Since Foursquare API specifies in generic places like restaurants, cafe and other food related places over specific business places, the recommendations will incline more towards those.

Thus by using the Delhi neighborhood data and the Foursquare API's places SDK, I would like to provide business enthusiasts with a suitable and efficient recommendation for locations and/or business ideas.

Methodology and Exploratory Data Analysis

Initial EDA and handling missing values –

The data is first loaded using pandas function and initial exploration is done to find the number of columns and rows in the dataset. The dataframe is then checked for any missing values. Results show that the dataframe consists of 185 rows which represent different boroughs and neighbourhood. The 4 columns respectively represent the borough, the name of the neighbourhood and their respective latitudes and longitudes. The dataframe has a total of **22 missing values** in the columns of latitude and longitude. This means that the dataframe contains 22 such neighbourhoods that have no data on latitude and longitude.

```
df_delhi = pd.read_csv('/kaggle/input/delhi-neighborhood-data/delhi_dataSet.csv', index_col = 'Unnamed: 0')
```

```
df_delhi.head()
```

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Adarsh Nagar	28.614192	77.071541
1	North West Delhi	Ashok Vihar	28.699453	77.184826
2	North West Delhi	Azadpur	28.707657	77.175547
3	North West Delhi	Bawana	28.799660	77.032885
4	North West Delhi	Begum Pur	NaN	NaN

```
df_delhi.shape
```

```
(185, 4)
```

```
df_delhi.isnull().sum()
```

```
Borough      0
Neighborhood  0
latitude     22
longitude     22
dtype: int64
```

The 22 missing values need to be handled. Since the values represent actual neighbourhoods in the city of New Delhi, dropping rows would mean loss of 22 such neighbourhoods which amounts to approximately 12% of the data. Loss of 12% of the data can have severe repercussions and can harm the accuracy of the project. The missing data can't also be filled in by the normal methods since the latitude and longitude aren't continuous values and mean, median, mode or interpolation can't be used. The only option left is manually filling in the latitude and longitude of the data by using **Google maps** or by using the **geopy-geocoders** libraries. The **geopy** library performs quite well and returns the latitude and longitude values according to the search query. However, the geopy package doesn't perform very well in Indian cities and with their data since it hasn't been incorporated to that extent yet. So the only option left to handle the missing data is by replacing them with actual values of latitude and longitude manually scraped from the Google maps. To facilitate easier filling of data, the dataframe is divided into two new dataframes, one for the missing values and one for the rest. The missing values dataframe is then filled / populated manually.

```
lat = pd.Series([], dtype=float)
lng = pd.Series([], dtype=float)

#Assigning the data

#Bugum Pur
lat[0], lng[0] = 28.727248, 77.064975

#Rohini Sub City
lat[1], lng[1] = 28.741073, 77.082574

df_missing['latitude_mod'] = lat
df_missing['longitude_mod'] = lng
```

After the missing dataframe has been populated, the indexing is redone and the two dataframes are merged into one. This results in a dataframe with no missing values.

```
frames = [df_missing, df_present]
df = pd.concat(frames)
df.head()
```

	Borough	Neighborhood	latitude	longitude
0	North West Delhi	Begum Pur	28.727248	77.064975
1	North West Delhi	Rohini Sub City	28.741073	77.082574
2	North Delhi	Gulabi Bagh	28.672190	77.191620
3	North Delhi	Sadar Bazaar	28.659395	77.212782
4	North Delhi	Tees Hazari	28.665682	77.216413

```
df.isnull().sum()
#there are no missing values
```

```
Borough      0
Neighborhood  0
latitude      0
longitude     0
dtype: int64
```

Further exploration of the data and identification and handling of incorrect data –

The data set used had a few incorrect data. This could be verified from the fact that while the latitude and longitude of most of the neighbourhood could be rounded off to 28-29 and 77-78 respectively, some of them had a latitude value of more than 30 while a longitude value of -90, which makes it lie in a different country altogether. To identify and correct the data, the dataframe was again divided into two parts, and the incorrect data were replaced manually using **Google Maps**. After correcting the data, both the dataframes are merged back into one. Final integrity checks are made to assure that no missing values exist and no data was lost. The verification shows that the dataset has 183 rows and 4 columns. It may be noted that the number of rows as compared to the initial check has decreased by 2. This is because 2 neighbourhoods were dropped because they have now been merged with other neighbourhoods.

Verfying the data integrity of the final dataframe

```
df.shape
#the dataframe has 183 rows with 4 columns
```

```
(183, 4)
```

```
df.columns
#the column names are in order
```

```
Index(['Borough', 'Neighborhood', 'latitude', 'longitude'], dtype='object')
```

```
df.isnull().sum()
#there are no missing values
```

```
Borough      0
Neighborhood  0
latitude      0
longitude     0
dtype: int64
```

Using Foursquare API to create the venues dataset –

Foursquare API is a realtime location based data provider, that is used here, to gather the various venues in an around a neighbourhood. The API takes a **URL** input and the result is returned in the format of a “**.json**” file. The json file can then be converted into a dataframe that python can understand using **pandas**. The URL takes in multiple inputs like the latitude and longitude of the place, the number of places to return, the search/explore radius, etc. A function is created that returns the category of the venue. Using this function, the dataframe is iterated to find the list of all unique venue categories around each neighbourhood. A dataframe is created according to the different

```
df_venues.head()
```

Venue Category	frequency
ATM	42
Accessories Store	4
Airport	3
Airport Terminal	2
American Restaurant	11

	Venue Category	frequency
0	Coffee Shop	334
1	Indian Restaurant	322
2	Hotel	180
3	Fast Food Restaurant	138
4	Pizza Place	97
...
177	Mattress Store	1
178	Video Game Store	1
179	Whisky Bar	1
180	Spiritual Center	1
181	Record Shop	1

venue categories and their occurrence/ frequency in all the neighbourhoods. The dataframe contains all sorts of venues that are present in New Delhi. It also includes venues like Metro Station, Parks, Temple, etc which are not business ideas and can't be set up by an interested person. Hence these venues are dropped from the list so that their frequency doesn't hinder the actual order of the venue dataset. The dataframe is then sorted in decreasing order of their occurrence. A peek into the dataset shows that similar venues like Cafe and Coffee Shop, Ice cream shop and Ice cream parlour, Saloon and Gents parlour, etc occur multiple times. These data items are combined into one occurrence and the

dataframe is again sorted according to the frequency. Looking at the data, it can be said that Coffee Shops, Indian Restaurant, Hotels, Fast Food places are among the top venues, while venues like Record Shop, Whisky Bar and Video Game Store are among the rarest venues. Using the shape function it is determined that the venues dataset has **182 rows**. It is very important to note that **Foursquare API** uses **realtime** data and hence, the shape and number of data can vary according to time and day and other factors.

Setting up the UI/UX –

Once the data part of the project is complete, the python library namely **ipywidgets**, is used to generate the user interface and help the user interact with the problem. The project has two options, **recommending a business according to the place of choice** and **recommending the optimal neighbourhood for the business of choice**. For the **first option**, the user has to select a borough/district from a list of drop down. On selecting the borough, the drop down for the neighbourhood changes according to the neighbourhoods present in that borough. The user then has to select the neighbourhood, where the business is to be set up. Finally after both the borough and the neighbourhood have been selected, the program returns a list of the top 10 possible new business ideas for that specific neighbourhood.

For the **second option**, the user is first presented with a choice on which type of business he/she wants to set up, whether it is a **popular business** or a **fancy and less popular one**. On selecting the option from a drop down, the user is presented with a list of possible business ideas. Since the recommendation takes into account the distance of the neighbourhood, the user is also asked to select which neighbourhood the person resides in. Once the business idea and current neighbourhood is selected, the program returns the most ideal neighbourhood to start the new business based on its distance from the current neighbourhood.

Assumptions made for this project –

- It is assumed that the user using this project is a normal person and doesn't use this to set up any government business.
- It is assumed that the aim of the person is to earn profit and hence the recommendations are layered in that order.
- It is also assumed that a borough is sufficiently large and has more than 2 neighbourhoods and hence the optimal place of choice for a new business will always occur in that borough itself.

Limitations of this project –

- Since the project aims to recommend a business using Foursquare API, it is highly dependent on the accuracy of the data provided by Foursquare API.
- The Foursquare API is a realtime data provider and hence the results can vary accordingly.
- A person can't start a completely new business using this project since the program has no data on that specific business. Here, a completely new business means, such a business that doesn't exist in the whole state of New Delhi.
- This project doesn't take into account the financial or social status of a person.

This project has no inferential statistical testing done on it and neither is any machine learning algorithm used. The reason behind using no machine learning algorithm is because the dataset has enough data to successfully predict the outcome without the help of any extra algorithmic calculations.

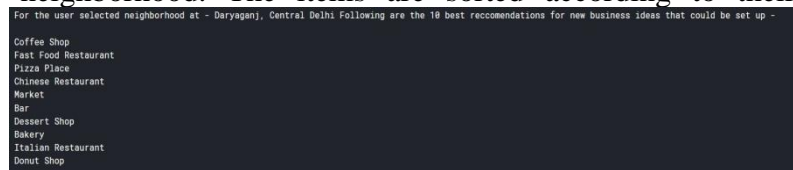
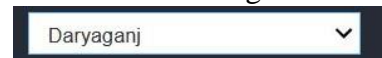
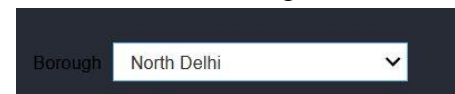
Results

The project aims to recommend a person with a business idea according to the place of choice or vice versa. Hence there isn't anything significant to draw from this project. The project however gives a good clarity that by using simple python programming a geo location data provider, it is possible to tailor the needs of a new business enthusiast so that he/she can have the best experience which they deserve.

A user's guide to using Rec a Business –

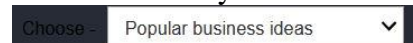
➤ Recommending business ideas :

- The user is first presented with a dropdown to choose the borough where the business is to be set up. The dropdown list contains 9 different boroughs and the user has to select the borough from the list.
- According to the borough selected, the user then has to select the neighborhood from a subsequent dropdown list that gets updated according to the borough selected.
- The program that computes and returns a list of 10 business ideas that could be set up in the selected neighborhood. The items are sorted according to their popularity in descending order.



➤ Recommending neighborhoods according to business of choice :

- The user is first presented with an option to select whether they want to start some popular business or some fancy business.
- The user is then presented with a list of business options depending on the previous choice they made.
- The user is then asked to select the borough and neighborhood they currently reside in.



- The program then a distance value for all the neighborhoods in that borough and checks to see which the closest possible region is where the business could be set up. If the residence neighborhood doesn't have the business the program returns that neighborhood itself.



Hence, this project provides a user-interactive way to help people set up new business ideas.

Project Discussion

Rec a Business provides new business enthusiasts leverage over others by specifying either the business or the location and hence he/she can invest to the best of their abilities. Through many hit and trials it has been found out that most of the business ideas can be set up in the current neighborhood itself. It has also been observed that Coffee Shops are the most popular venues in the city of New Delhi followed by Indian Restaurants and Hotels. However, there also exist very rare businesses like Korean Restaurant and Tex-Mex Restaurant among others. So all in all, New Delhi has a very diverse business culture with over 180 different types of business. So it gives one ample opportunities to set up a business if he/she really wants to.

Conclusion

This capstone project developed with Foursquare API and using python notebooks in kaggle kernel, gives the user an easy, interactive and free way to gain leverage when starting out a new business. The program doesn't use any machine learning algorithms which might make it lack luster, but its simplicity and due to the availability of plenty of data, makes this work very efficiently. The absence of machine learning algorithms also means that this project doesn't suffer from any of the problems that a machine learning algorithm suffers from. The algorithm relies heavily on Foursquare API and excluding this fact, it is almost cent percent effective when it comes to predicting a business idea or selecting the optimal business location. Hence it's safe to conclude that Rec a Business provides easy, efficient, interactive and free access to business data which when used correctly can help new business enthusiasts gain sufficient leverage over others.