README

Envision Chicago

EnvisionChicagoFramework.py is the main executable file.

Libraries required: pandas,numpy, LabelEncoder, tree, train_test_split, re, SequenceMatcher, datetime, matplotlib.pyplot, csv, TextBlob, pandasql, nltk, sklearn

Query 1
Description: Using this query, the application reports the type of a crime, whether assault, battery etc, within 3 blocks of a grocery store, a school and a restaurant.
Approach: We have integrated the restaurant data with business data using tech similarity obtained by SequenceMatcher. This file is already generated and placed in data folder. We have considered the precision for the 3 blocks based on latitude and longitude such that it covers all the four directions. We have validated our computation by testing it on Google Maps. For the rest of the part, We have used pandas.
Input command: python3 EnvisionChicagoFramework.py -q query1
Output: The output is generated in the results directory as query_1_result.csv

Query 2
Description: The application trains a model to learn crime statistics and predict the probability of different types of crime for a given set of addresses using supervised learning. The results of logistic regression and decision trees are compared.
Approach: We used decision tree and multinomial logistic regression for building model. The input data has four features - block address, male percentage in total population, female percentage in total population and median age. The target classifiers is crime type. This query outputs crime type for given address and probability.
Input command: python3 EnvisionChicagoFramework.py -q query1 -args [‘Block Address’]
Example: python3 EnvisionChicagoFramework.py -q query1 -args “004xx S Financial Pl”
Output: The output is displayed on the console with crime type, probability and technique for the given address.

Query 3
Description: A graph plot is created that compares Crime statistics for each age group by census block which in turn shows intersection of parameters from multiple datasets.
Approach: We have fetched the age group according to the census block group through an API available at census.gov. We have also integrated census tracts to the crime data based on the latitude and longitude available in both the data sets. We used these two generated files to yield statistics about the crime with respect to the census block groups.
Input command: python3 EnvisionChicagoFramework.py -q query3

Output: The output is generated in the results directory as query_3_result.csv

Query 4
Description: The application determines the relationship between the average review rating, whether increases or decreases, and food inspection result for a restaurant.
Approach: A dictionary with keys, id, name, address is created for the given restaurants CSV file. This dictionary is integrated with food inspections CSV file to find inspection result. Jaccard is used on address and name field in restaurant and business licenses for integrating.
Input command: python3 EnvisionChicagoFramework.py -q query4
Output: The output is generated in the results directory as query_4_result.csv

Query 5
Description: For each Yelp reviews of each restaurant, sentiment analysis is performed with ratings 1,2 and 3 as negative sentiments and 4 and 5 as positive sentiments.
Approach: Initially we cleaned the reviews file as few of the reviews were divided because of double quotes. A uniform reviews is made. We use pandas to integrate restaurants and reviews using review id. We iterated through the combined file and for each review, we predicted sentiment using TextBlob library.
Input command: python3 EnvisionChicagoFramework.py -q query5
Output: The output is generated in the results directory as query_5_result.csv

Query 6
Description: The application plots a graph average ratings against most frequent overall sentiments. This visualizes how sentiments are related to review ratings.
Approach: The file is grouped according to restaurants. For each, we found average user rating. We found out maximum repeated sentiment for each restaurant. Using this, we created a dataframe which contains average rating and maximum repeated sentiments for each restaurant.
Input command: python3 EnvisionChicagoFramework.py -q query6
Output: The output is generated in the results directory as query_6_result.csv

Query 7
Description: The application predicts review rating using text from Yelp reviews.
Approach: We used nltk library to vectorise the review and also to remove stop words. Then this is passed to classifier. We used Naive bayes multinomial classification to predict the rating.
Input command: python3 EnvisionChicagoFramework.py -q query7 -args [‘Review Text’]
Output: The output is generated in the results directory as query_7_result.csv

Query 8
Description: The application shows the number of years the restaurant has been active after a failed food inspection.

Approach: For this query, we have integrated food inspection and business licenses data. The runtime of business after a failed inspection is calculated as:

1. If the license associated with the failed inspection has been cancelled or revoked
2. If there is no new license issued after the failure of inspection for two consecutive years

Note: The dataset for this query consists of restaurants located from zipcode 60601- 60607
Input command: python3 EnvisionChicagoFramework.py -q query8
Output: The output is generated in the results directory as query_8_result.csv

Query 9
Description: The influence of liquor license on crime incidents in a neighbourhood is predicted.
Approach: We cleaned the census file to find the average latitude and longitude for tract block. We integrated the census with business licenses and cleaned it to remove restaurants which do not have business licenses. We integrated the crimes file and census file using the tract id. Finally, we integrate both intermediate files generated. In the final file, we grouped according to tract id and count number of restaurants which sells liquor, count the number the crimes, count the number of arrests using pandas.

- Census_File_Cleaning.py- This takes census file as input and returns cleaned file which contains average of latitude and longitude in Chicago.
- BL_Census_final_integration.py- It integrates business licenses and census data.
- Crimes_Census_final_integration.py- It integrates crime and census data.

These 3 generated files are used as input for final output generation for query9
Input command: python3 EnvisionChicagoFramework.py -q query9
Output: The output is generated in the results directory as query_9_result.csv

Query 10
Description: The probability of robbery(all types) for summer 2018 is predicted using the past crime data and weather data for a zip code.
Approach: We extracted the weather data from wunderground for each zip each day from last 10 years. We created a manual file which maps zipcode to tracts. We mapped this manual file with the extracted weather file. This combined file is now integrated with census file using the tracts. We integrated the crime file with census file using latitude and longitude. This file is integrated with previous combined file. This results in a file which contains crime and weather data with intermediate census data. The final generated file is fit into Naive bayes classifier. This is used to predict the future probability of robbery.

- WeatherData_Extraction.py- It is the URL of wunderground to generate a weather data for the past few years for each day.

Note: To predict a future crime and its probability query_10_crime_weather_predict.py edit the test_x1 or test_x2 or test_x3. Enter a tract id, year, month and mean temperature. The predicted results can be found in output file.
Input command: python3 EnvisionChicagoFramework.py -q query_10
Output: The output is generated in the results directory as query10_result.csv