Performance Analysis and Clustering Geospatial Data For Multiple Reference Point

A PROJECT REPORT

By

Paras Sibal (RA1611020010055)

Shreayan Chaudhary (RA1611020010011)

Under the guidance of

Dr. M. Ferni Ukrit

Assistant Professor

Department of Software Engineering

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

SOFTWARE ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

MAY 2020

BONAFIDE

This is to certify that this Major Project report titled “Performance Analysis and Clustering Geospatial Data For Multiple Reference Point” is the bonafide work of Paras Sibal (RA1611020010055) and Shreayan Chaudhary (RA1611020010011) who implemented and completed their project work under my supervision.

Signature of the Guide
Dr. M. Ferni Ukrit
Assistant Professor
Department of SWE
SRM University

Signature of the HOD
Dr. C. Lakshmi
Professor and Head
Department of SWE
SRM University

Internal Examiner

External Examiner

ABSTRACT

India is an extremely densely populated country (one of the densest), with more than 1.34 billion residents. It is tough to start a business here due to high real estate costs. Moreover, entrepreneurs are scared due to high risk involved and concerns whether or not they would cover the cost with profits to thrive the business. This ML model is personalized for each user by generating its own dataset and determining the latitude and longitude of a place appropriate for setting business in any particular city/state involving less risk and thereby maximizing profits by attracting more customers. Hence, the objective is to find the optimal location in any given particular city or place to help set up a business for entrepreneurs, thus saving time, money and risk. Target audience consists of entrepreneurs and small-scale businessmen/women interested in setting up the business, aiming at the corporate demographic for maximizing profits. This model is determined to be suitable for use by entrepreneurs for setting up the business and future works.

Keywords: Clustering; Geospatial Data; Multiple Reference Points; Recommender System; FourSquare API; Folium

ACKNOWLEDGEMENTS

We would like to express our most sincere gratitude to everyone who has helped us with this research directly, or indirectly; especially to our mentor, **Dr. M.Ferni Ukrit, Project Guide, Assistant Professor (Sr.G), Department of Software Engineering** for her constant inspiration and guidance along with feedback to improve the model. This work would not have been possible without the constant support from our parents and professors who helped us a lot for this research.

We would also like to thank our fellow batchmates and the faculty of the software engineering department for their continuous guidance and assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

KNN - K Nearest Neighbors

RMS - Root Mean Square

API - Application Programming Interface

GPU - Graphics Processing Unit

SRS - Software Requirements Specification

IDE- Integrated Development Environment

ER – Entity Relationship

LOC – Lines of Code

CHAPTER 1
INTRODUCTION

Establishing a new business is one of the riskiest tasks in today's world. It is very difficult to find the "best and suitable" place for entrepreneurs to establish new business and keep them running for the long term. Currently, for setting up a new business, the entrepreneurs have to consult experts which will cost additional time and money. We have proposed a solution to address this issue.

By using and analyzing geospatial data [1, 5], these techniques and technologies have made it possible to find the most optimal place. The interface will allow the user to select a city/state along with the radius in which it will look for the least competitors in the vicinity. The model will fetch the data from Google Maps and Foursquare about the corporations (business is aimed towards the corporate demographic) and the competitors nearby. After the data is fetched, the model will form various clusters and analyze it to find the optimal location for maximizing the profits.

The project will be deployed as a web application running on the cloud that includes an interactive web interface as well separately offering ML solutions through a dedicated quasi-RESTful API endpoints.

The Data Pipeline will consist of the following steps:
Data Scraping and Collection -> Data Cleaning and Preprocessing -> Clustering -> Cluster Analysis -> Selecting Best Cluster.

CHAPTER 2
PROJECT OVERVIEW

2.1 LITERATURE SURVEY

Geospatial Data Generation and Preprocessing Tools for Computing System Development

Alexey Golubev proposes a set of open source tools for generating and preprocessing of geospatial data. They are: “scatter” for generating geospatial data, “clustering” for reducing the number of geo points based on their density.

Exploratory geospatial data analysis using the GeoSOM suite

Chao Zhang proposed a methodology where the Geo-SOM implements several techniques to improve the geo-data analyzed. They are cartographic generalization.The basic idea of an SOM is to map the data patterns onto an n-dimensional grid of units

Clustering Geospatial Data for Multiple Reference Point

Ying Zhong has depicted that the multi-reference clustering (MRC) develops two algorithms, an exact search algorithm and an approximation algorithm. The exact search algorithm can compute the exact result of MRC for small kin real time, while the approximation algorithm can compute MRC with large k efficiently and accurately.

Comparative Study between K-Means and K-Medoids Clustering Algorithms

Santosh Nirmal has performed a comparative study between K-Medoid and K-Means algorithms for clustering. K-Medoid also has some disadvantages like it is more costly than k-means method. It does not scale well for large data sets. For large values of n and k, such computation becomes very costly. Also the result of the dataset shows that K-Medoids is better in all aspects such as execution time, non sensitive to outliers and reduction of noise but with the drawback that the complexity is high as compared to K-Means.

Clustering, Forecasting and Cluster Forecasting: using k-medoids, k-NNs and random forests for cluster selection

Dinesh Reddy has depicted in his work that K-Medoid is empirically proven in this study to be the best clustering method for cluster selection forecasting in between the three clustering approaches, both overall as well as for the maximum number of clusters. The comparison is fair in between the clustering methods only when the number of clusters is fixed; although obviously each method might determine different clusters.

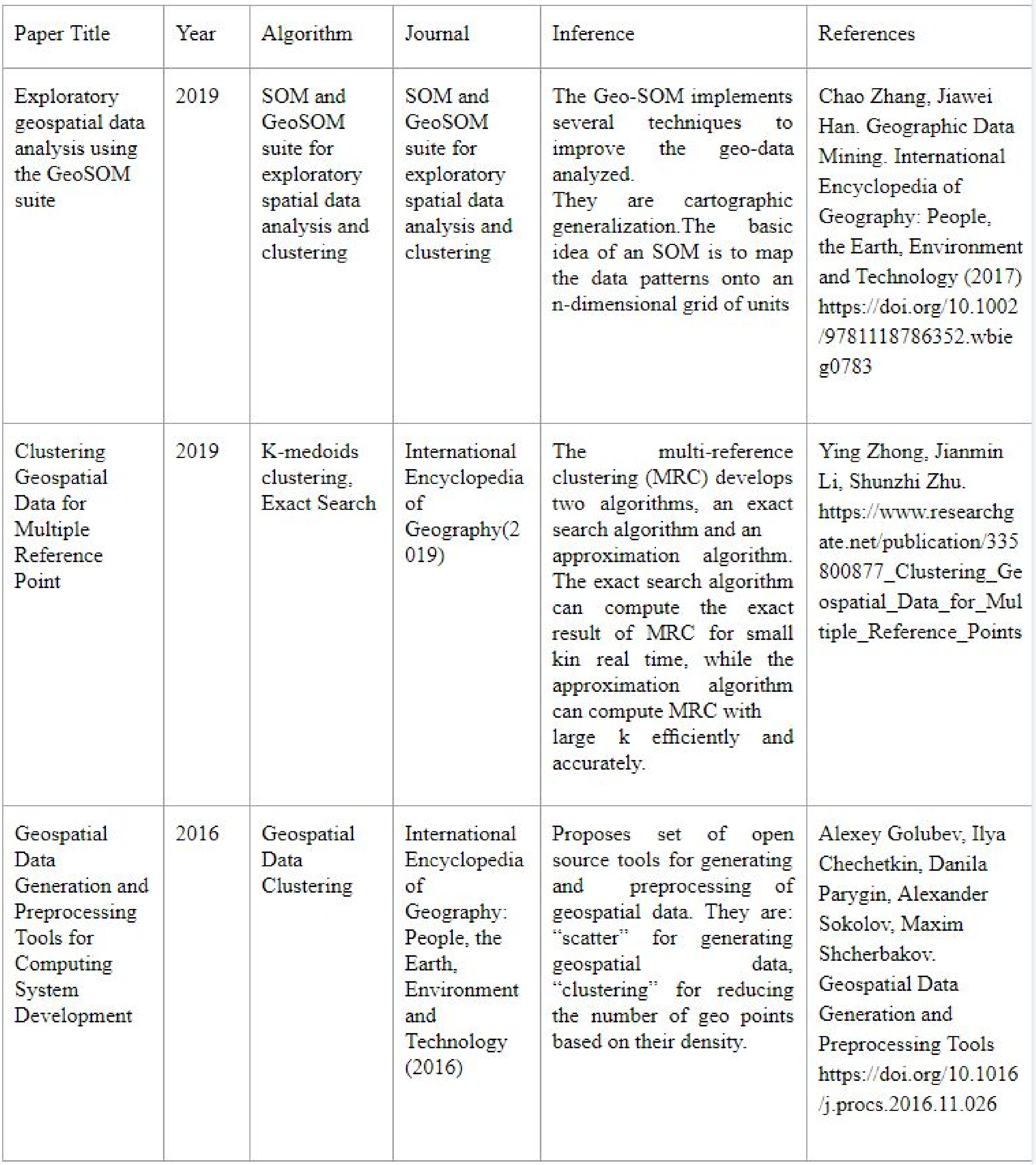
Geographic Data Mining

Rabindra Barik has studied and described all the methods of how geographical data can be mined in a standard format for scalable analysis of massive geographic and geocoded datasets.

Geospatial Data Generation and Preprocessing Tools for Computing System Development

Alexey Golubev proposes a set of open source tools for generating and preprocessing geospatial data. They are: “scatter” for generating geospatial data, “clustering” for reducing the number of geo points based on their density.

Table 2.1: Overview of Literature Survey



Paper Title	Year	Algorithm	Journal	Inference	References
Exploratory geospatial data analysis using the GeoSOM suite	2019	SOM and GeoSOM suite for exploratory spatial data analysis and clustering	SOM and GeoSOM suite for exploratory spatial data analysis and clustering	The Geo-SOM implements several techniques to improve the geo-data analyzed. They are cartographic generalization.The basic idea of an SOM is to map the data patterns onto an n-dimensional grid of units	Chao Zhang, Jiawei Han. Geographic Data Mining. International Encyclopedia of Geography: People, the Earth, Environment and Technology (2017) https://doi.org/10.1002/9781118786352.wbieg0783
Clustering Geospatial Data for Multiple Reference Point	2019	K-medoids clustering, Exact Search	International Encyclopedia of Geography(2019)	The multi-reference clustering (MRC) develops two algorithms, an exact search algorithm and an approximation algorithm. The exact search algorithm can compute the exact result of MRC for small kin real time, while the approximation algorithm can compute MRC with large k efficiently and accurately.	Ying Zhong, Jianmin Li, Shunzhi Zhu. https://www.researchgate.net/publication/335800877_Clustering_Geospatial_Data_for_Multiple_Reference_Points
Geospatial Data Generation and Preprocessing Tools for Computing System Development	2016	Geospatial Data Clustering	International Encyclopedia of Geography: People, the Earth, Environment and Technology (2016)	Proposes set of open source tools for generating and preprocessing of geospatial data. They are: "scatter" for generating geospatial data, "clustering" for reducing the number of geo points based on their density.	Alexey Golubev, Ilya Chechetkin, Danila Parygin, Alexander Sokolov, Maxim Shcherbakov. Geospatial Data Generation and Preprocessing Tools https://doi.org/10.1016/j.procs.2016.11.026

Comparative Study between K-Means and K-Medoids Clustering Algorithms	2019	K-medoids, K-means clustering	International Research Journal of Engineering and Technology (2019)	K-Medoid also has some disadvantages like it is more costly than the k-means method. It does not scale well for large data sets. For large values of n and k, such computation becomes very costly. Also the result of the dataset shows that K-Medoids is better in all aspects such as execution time, non sensitive to outliers and reduction of noise but with the drawback that the complexity is high as compared to K-Means.	Santosh Nirmal https//academia.edu.documents/59970755/IRJET-V6I315420190709-81084-ihv1ne.pdf?response-content-disposition=inline%3B%20filename%3DIRJET-_Comparative_Study_between_K-Means.pdf
Clustering, Forecasting and Cluster Forecasting: using k-medoids, k-NNs and random forests for cluster selection	2019	K-medoids, KNN, random Forest	Computers and Urban Systems Journal (2019)	K-Medoid is empirically proven in this study to be the best clustering method for cluster selection forecasting in between the three clustering approaches, both overall as well as for the maximum number of clusters. The comparison is fair in between the clustering methods only when the number of clusters is fixed5; although obviously each method might determine different clusters.	Dinesh Reddy, Konstantinos Nikolopoulos, & Konstantia Litsiou https://www.bangor.ac.uk/business/research/documents/BBSWP-19-16.pdf
Geographic Data Mining	2019	Spatial Clustering	Procedia Computer Science Journal (2017)	It studies and describes all the methods of how geographical data can be mined in a standard format for scalable analysis of massive geographic and geocoded datasets.	Rabindra Barik, Ankita Tripathi, Suraj Sharma, Vinay Kumar, Himansu Das. MistGIS: Optimizing Geospatial Data Analysis Using Mist Computing. Progress in Computing, Analytics Journal, Springer https://doi.org/10.1016/j.compenvurbsys.2011.11.003

2.2 PROBLEM DESCRIPTION

Establishing a new business is one of the riskiest tasks in today’s world. It is very difficult to find the “best and suitable” place for entrepreneurs to establish new business and keep them running for the long term. Currently, for setting up a new business, the entrepreneurs have to consult experts which will cost additional time and money.

We have proposed a solution to address this issue. By using and analyzing geospatial data [1, 5], these techniques and technologies have made it possible to find the most optimal place. The interface will allow the user to select a city/state along with the radius in which it will look for the least competitors in the vicinity.

The model will fetch the data from Google Maps and Foursquare about the corporations (business is aimed towards the corporate demographic) and the competitors nearby. After the data is fetched, the model will form various clusters and analyze it to find the optimal location for maximizing the profits.

2.3 REQUIREMENTS GATHERING

2.3.1 Questionnaire:

The preliminary requirements will be found by asking the stakeholders to fill a questionnaire. The questionnaire will provide vital information on what the stakeholders are seeking and the area of focus that needs to be tackled.

2.3.2 Interview:

The requirements will be further developed by interviewing the concerned stakeholders. This helps to clarify any misunderstood requirements as well as figure out any implied requirements.

2.3.3 Brainstorming:

A brainstorming session will be held with the developers to discuss the requirements. This step is necessary to find various approaches to solve the stakeholders' problems. Both divergent and convergent thinking will help to gather the solutions efficiently.

2.3.4 Focus Groups:

There is a discussion among the stakeholders of the product like, users or customers related to the expectations of a product.

2.4 REQUIREMENT ANALYSIS

1. High location accuracy

Prediction of the optimal location must be accurate enough to locate the place on a map accurately.

2. Fast data analysis and training time

Algorithms should form the dataset quickly and the model must locate the nearby venues and competitors quickly. Algorithms must make predictions swiftly.

Data Requirement

The coordinates of a location must have the latitude and longitude till at least 4 decimal places to locate it accurately.

3. System Requirement

Intel Core i3 (5th Generation) or higher to run effectively.

Python 3.6 with Anaconda Navigator for Spyder and Jupyter Notebooks

Framework: Django

Libraries Used: Geopy Nominatim, Foursquare, Folium, Scikit-learn, Matplotlib, Seaborn, TinyMCE, Pandas, NumPy, PostgreSQL.

2.4.1 FUNCTIONAL REQUIREMENTS

1. The software should get the city / area and the radius of the area to be searched in the vicinity from the user.
2. It should use Nominatum and Google Maps API to locate the nearby places/ competitors.
3. It should accurately predict the coordinates of the best place to set up business.
4. It should display this result on the screen of the device.
5. In case the locations are not found on Google Maps, an error message should be displayed.
6. The nearby places must be returned to the user within a tile limit of 5 seconds.
7. The application must work for all the locations / cities / areas in India.

2.4.2 NON- FUNCTIONAL REQUIREMENTS

1. It should be compatible with all devices and all platforms.
2. It should be working 24x7.
3. It must be easy to use and accessible by all people
4. The response time of application must not be more than 10 seconds.
5. Application has Backup & Disaster recovery, as all the trained models are stored in case of any malfunction.
6. The web application must be supported in all the Internet browsers like Chrome, Firefox, Safari, Internet Explorer etc.
7. The application must be flexible so that it can accomodate any changes or new features that may be added to the application in the future.

2.5 DATA SOURCE

The user will have to provide the city/state or the area in which he/she has to set up the business along with a radius (in kilometres) radius in which it will look for the least competitors in the vicinity. It has been assumed that the user only wants to target the corporate demographic for maximizing profits. The data has been scraped from Wikipedia to extract most of the corporations present in India, which can be found here: https://en.wikipedia.org/wiki/List_of_companies_of_India. Once the user enters the desired area, the exact coordinates of the place can be extracted using Geopy Nominatim library. The model will find all corporations present within the input radius.

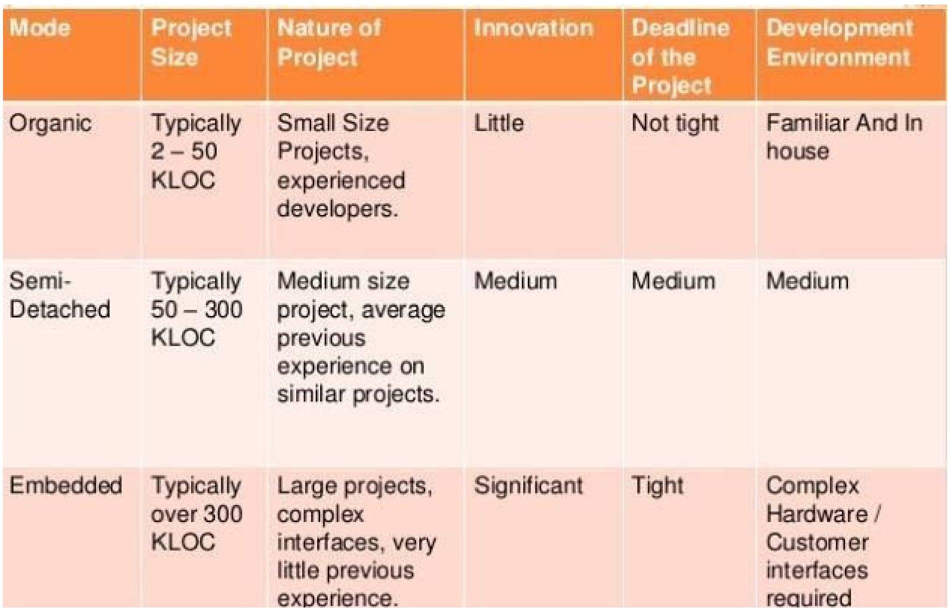
The details of the competitors will be extracted using the Foursquare API, which uses the Google Maps API to fetch the details of the places in the neighbourhood. Only the places that are available in Google Maps will be extracted. Once the data is collected systematically, it will be cleaned and only meaningful data will be extracted from it.

The Geopy Nominatim library is used to locate the latitude and longitude for the given address, cities and countries. FourSquare API is further used to find businesses within a specified distance returning venue, postal codes, and summary.

2.6 COST ESTIMATION

We will be using the COCOMO model and Wideband Delphi model to estimate the cost.

Table 2.2: Categories of COCOMO model



Mode	Project Size	Nature of Project	Innovation	Deadline of the Project	Development Environment
Organic	Typically 2 – 50 KLOC	Small Size Projects, experienced developers.	Little	Not tight	Familiar And In house
Semi-Detached	Typically 50 – 300 KLOC	Medium size project, average previous experience on similar projects.	Medium	Medium	Medium
Embedded	Typically over 300 KLOC	Large projects, complex interfaces, very little previous experience.	Significant	Tight	Complex Hardware / Customer interfaces required

Our software project is organic type in nature due to:

- The team consists of 2 members and its size is small
- The problem statement is clear and well defined
- The team members are experienced enough to tackle the problem.

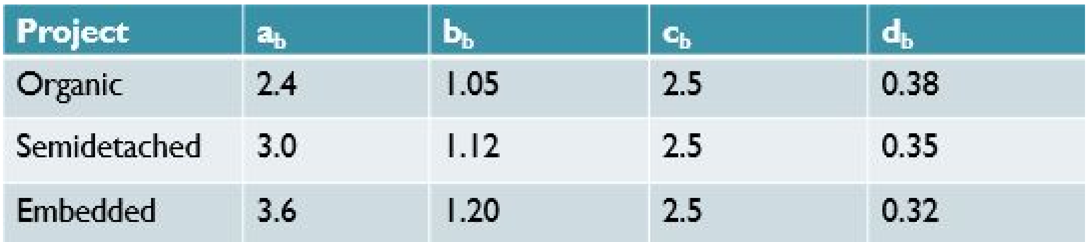
Effort (E) = a* (KLOC) ^{b} (in Person-months) (2.1)

Development Time (D) = c * (E) ^{d} (in month) (2.2)

Average staff size (SS) = E / D (in Person) (2.3)

Productivity (P) = KLOC / E (in KLOC / Person-month) (2.4)

Table 2.3: Basic coefficients of COCOMO model



Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Cost Estimation (COCOMO Model)

Text for for the Cost Estimation:

Risk Effort Applied (E) = a * (kLOC) ^ b

Development Time (D) = c * (Effort Applied) ^ d

kLOC = 3 (model + data gathering) + 0.6 (cloud + deployment + server) = 3.6kLoc

Project is organic in nature

Hence, E = 2.4 * (3.6)^1.05 = 8.9 man-months

D = 2.5 * (8.9)^0.38 = 5.73 months

Cost Estimation (Wideband Delphi)

Cost Estimation:

Effort (E) = (PE + (LE * 4) + OPE) / 6

PE = Pessimistic Estimate, LE = Likely Estimate, OPE = Optimistic Estimate

Hence, OPE = 3 months and PE = 5 months LE = (3+5)/ 2 = 4 months

E = (5 + (4*3) + 3) / 5 = 4 man-months

2.7 PROJECT SCHEDULE

2.7.1 Week 1

- Understand the problem statement and identify the possible solutions.
- Make rough architectural decisions.
- Prepare documentation.
- Brainstorm on the possible solutions.

2.7.2 Week 2

- Create a plan to collect data and generate data for a location.
- Identify possible tools / APIs that could be used to gather geospatial data of a place.
- Modify architectural diagrams.
- Select the best tool by creating a pros/cons list for the tools.

2.7.3 Week 3

- Update the documentation
- Use the tool to gather data for a particular city / location.
- Analyze the data and understand the type of business needed to be set up.
- Use the tool to find out nearby competitors and target audience in the vicinity.

2.7.4 Week 4

- Update documentation
- Use K-Means and K-Medoids to cluster the competitors with their reference points.
- Work on bug fixes in the model
- The above created model should work well for any city / place in India.

2.7.5 Week 5

- Update documentation
- Create architecture of the web application
- Create wireframe of the UI
- Start creating the Django project.

2.7.6 Week 6

- Update documentation
- Create models in the Django web application
- Create an overall design for the Model-View-Control architecture

2.7.7 Week 7

- Update documentation
- Create the homepage of the application
- Create the dashboard for the user to view his/her profile

2.7.8 Week 8

- Update documentation
- Fine tune the models and other hyperparameters
- Built user-friendly interface for the program
- Create the Sign In/ Register page for the user to login

2.7.9 Week 9

- Update documentation
- Integrate the model with the Django application
- Perform all types of Testing on the Django application to ensure proper functioning
- Work on report

2.7.10 Week 10

- Complete documentation
- Complete report
- Prepare demo or prototype of the Django application

2.7.11 Week 11

- Review documentation
- Review report
- Make necessary changes in the report and the documentation

2.7.12 Week 12

- Preparation of demo
- Preparation for hand over
- Final demo

2.8 RISK ANALYSIS

2.8.1 Strengths

The model will output the best places to set up a new business. These places balance the tradeoff between having the most target audience and minimizing the number of competitors for the same.

1. It helps to narrow down the best locations for setting up the business
2. It encourages feature reuse and improvement
3. It helps to reduce the number of competitors around the business
4. It also improves feature propagation

2.8.2 Weaknesses

The model's performance can be improved by adding a feature of the population of the place. It may be possible in the near future to determine the approximate population of a place using GPS in cell phones. By adding this feature, the model will also consider the population density of an area while it forms the clusters [4, 5]. This will improve the quality and accuracy of the clusters being formed, consequently improve the precision of the results.

2.8.3 Opportunities

Time taken will be less to extract required location data according to entrepreneur interest. Moreover, It will allow the entrepreneurs to set up high profits businesses with lower risk of failure.

There are a lot of opportunities for using Scikit-Learn for K-Means and K-Medoids and the Django framework such as:

1. There is huge community support available
2. There is a lot of documentation on the Internet. This facilitates ease of learning and development
3. The Django framework is very flexible and future changes can be easily accommodated in the application.

2.8.4 Threats

There are numerous threats that could hamper the development of the program. Some of them are:

1. Location is not available for some places.
2. Competition from contenders (having better accuracy and results) could lead to lower usage of our product.
3. Only those places that are available in the Google Maps are considered.

2.9 SRS

2.9.1 Introduction

2.9.1.1 Document purpose

The purpose of this document is to understand the complete requirements of this project. This project is made to help entrepreneurs find out the best location where they can set up their new business.

2.9.1.2 Document conventions

This SRS's format is simple. Bold letters are used for headings and the normal one for specifications. The entire document is written using the standard font, Times New Roman. Main Headings of any page are denoted by Times-14 and subheadings by Times-12.

2.9.1.3 Intended audience and reading suggestions

This particular document is expected to be read by project managers, developers, documentation writers, users, and testers. Being a technical document, terms are intended to be understood by the customer clearly. This SRS should be read starting with Introduction. The main purpose of the document is to identify if the developer is using the correct methods, requirements and other fulfilled methods. This document is used by both the developer and tester. The tester will get to know about functional and nonfunctional requirements.

2.9.1.4 Product scope

One importance of this proposed methodology is to help identify fractures easily and more accurately.

2.9.2 Overall Description

2.9.2.1 Product perspective

The software is easy to use, with a very user friendly UI and UX. The user has to provide a city / place as the input and after analyzing the nearby places, the coordinates of the best location is then displayed on the screen.

2.9.2.2 Product functions

- Product inputs the city / place from the user
- It analyzes the nearby locations and filters the relevant ones.
- Displays the result on the screen (coordinates of the optimal location)

2.9.2.3 User classes and characteristics

Administrator, Entrepreneur

2.9.2.4 Operating environment

This application can be used on any device that has a web browser.

2.9.2.5 Design and implementation constraints

No constraints at this point in time.

2.9.2.6 User documentation

- Software Requirement Specification.
- Required software.
- User manual.

2.9.3 External Interface Requirements

2.9.3.1 User interface

The software will give the user a choice to Register or Login. The user has to provide a city / place as the input and after analyzing the nearby places, the coordinates of the best location is then displayed on the screen.

2.9.3.2 Hardware interface

No hardware interface is required

2.9.3.3 Software interface

The system makes direct use of Django and Python, and can be run on any web browser like Google Chrome/ Firefox/ Safari/ Internet Explorer

2.9.3.4 Communication interfaces

The system uses a dashboard containing a report to communicate.

2.9.4 System Features

2.9.4.1 Description and Priority:

The product has a candid interface for the user. The user can use the product on any web browser on his cellphone or on the desktop.

2.9.4.2 Functional Requirements:

1. The software should get the city / area and the radius of the area to be searched in the vicinity from the user.
2. It should use Nominatum and Google Maps API to locate the nearby places/ competitors.
3. It should accurately predict the coordinates of the best place to set up business.
4. It should display this result on the screen of the device.
5. In case the locations are not found on Google Maps, an error message should be displayed.
6. The nearby places must be returned to the user within a tile limit of 5 seconds.
7. The application must work for all the locations / cities / areas in India.

2.9.4.3 Other Nonfunctional Requirements

1. The software should be compatible with all devices and all platforms.
2. It should work 24x7.
3. It must be easy to use and accessible by all people.
4. The response time of application must not be more than 10 seconds.
5. Application has Backup & Disaster recovery, as all the trained models are stored in case of any malfunction.

CHAPTER 3

ARCHITECTURE & DESIGN

3.1 SYSTEM ARCHITECTURE

The system has the following modules as part of its architecture:

- Database
- Data Model
- Cloud Service
- RESTful API
- FourSquare API
- Output

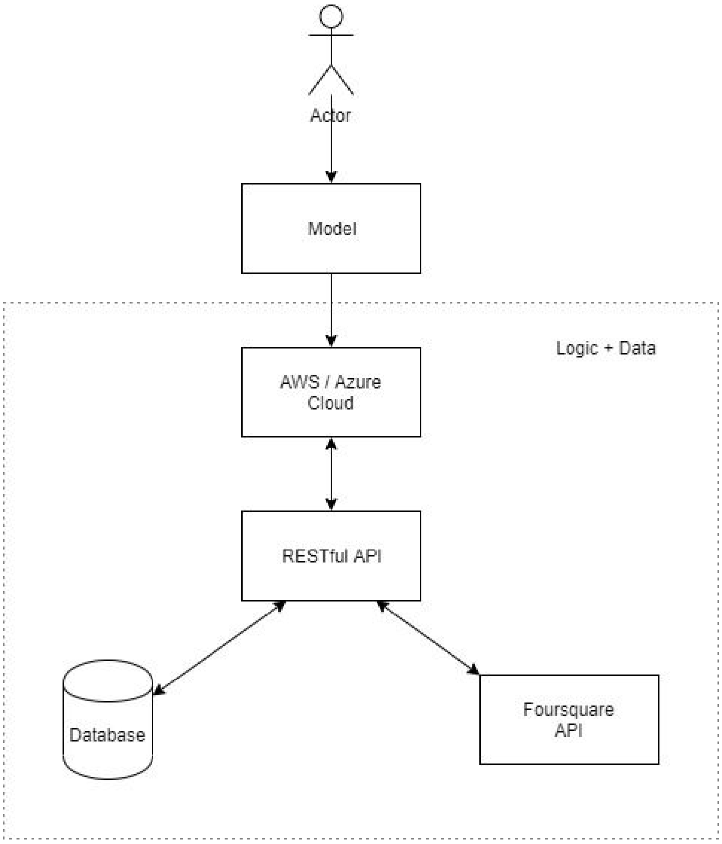


Fig 3.1: System Architecture

1. Database

This contains the data collected for training, validation and testing purposes. It generally stores user credentials and location where the user wants to open business along with the radius in km.

2.Data Model

This is used to train data using unsupervised machine learning algorithms which returns the coordinates suitable for setting up the user's business.

3.Cloud Service

The various cloud services enables the user to access and store the model and its corresponding information with ease running on cloud servers.

4.RESTful API

The project will be deployed as a web application running on the cloud that includes web interface as well separately offering ML Solution through a dedicated quasi-RESTful API endpoints. The list of corporations API is scrapped from wikipedia.

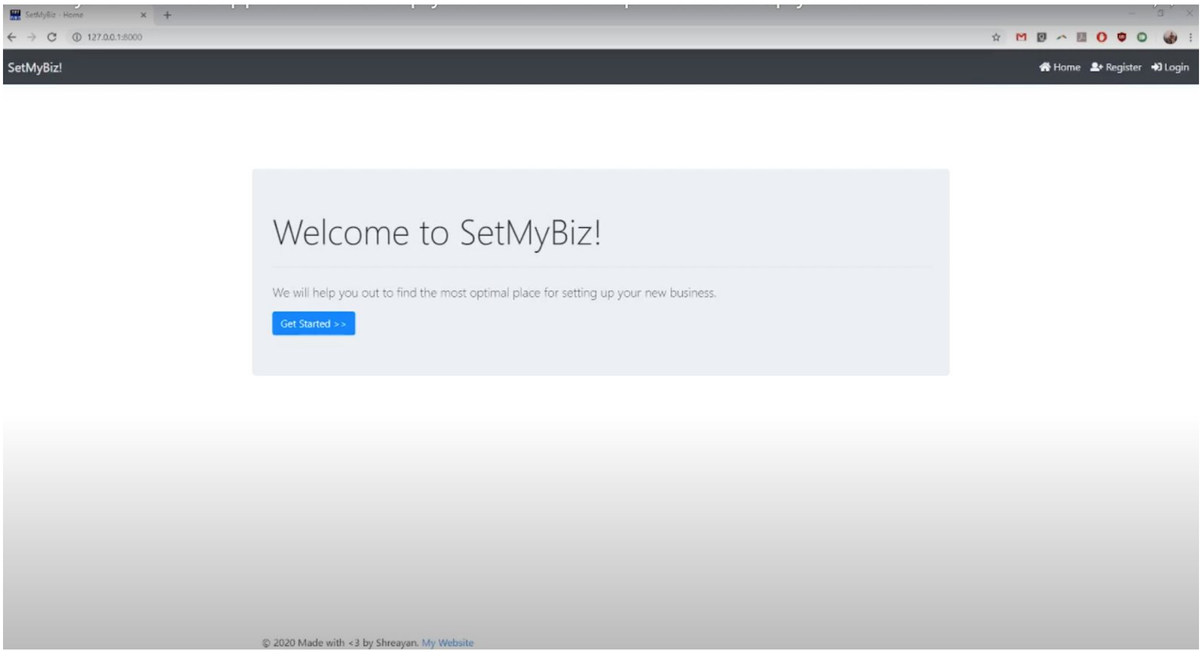
5.FourSqaure API

FourSquare API is used to determine the coordinate of all the existing restaurants within th location given by the user.

6.Output

Gives the coordinates of the location suitable for the user for setting up the business that will attract new customers thereby maximizing profits.

3.2 INTERFACE PROTOTYPING (UI)

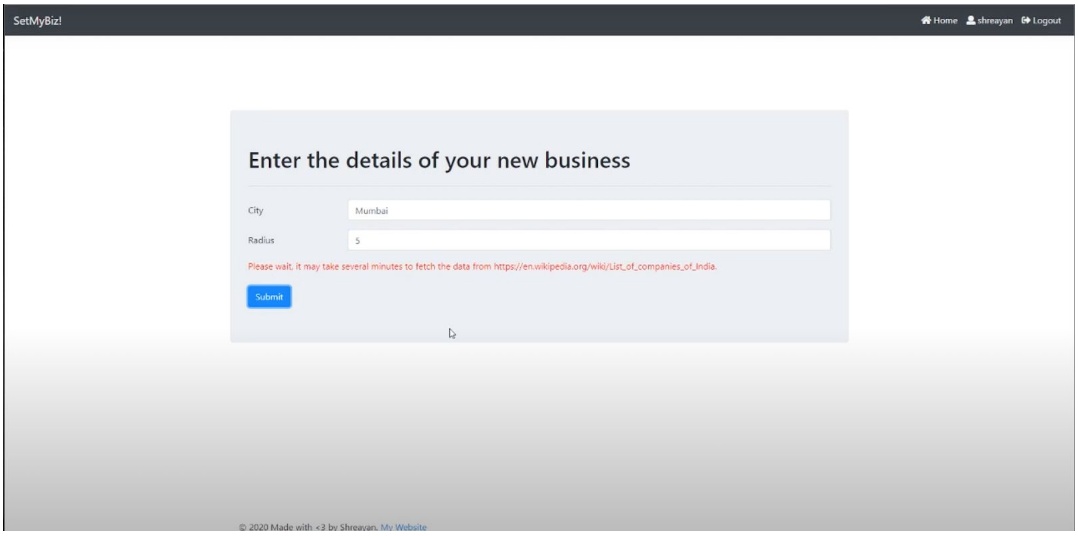


Welcome to SetMyBiz!

We will help you out to find the most optimal place for setting up your new business.

© 2020 Made with <3 by Shreayan. My Website

Fig 3.2: UI Prototype



Enter the details of your new business

City: Mumbai

Radius: 5

Please wait, it may take several minutes to fetch the data from https://en.wikipedia.org/wiki/List_of_companies_of_India.

© 2020 Made with <3 by Shreayan. My Website

Fig 3.2: UI Prototype

3.3 DATA FLOW DIAGRAM

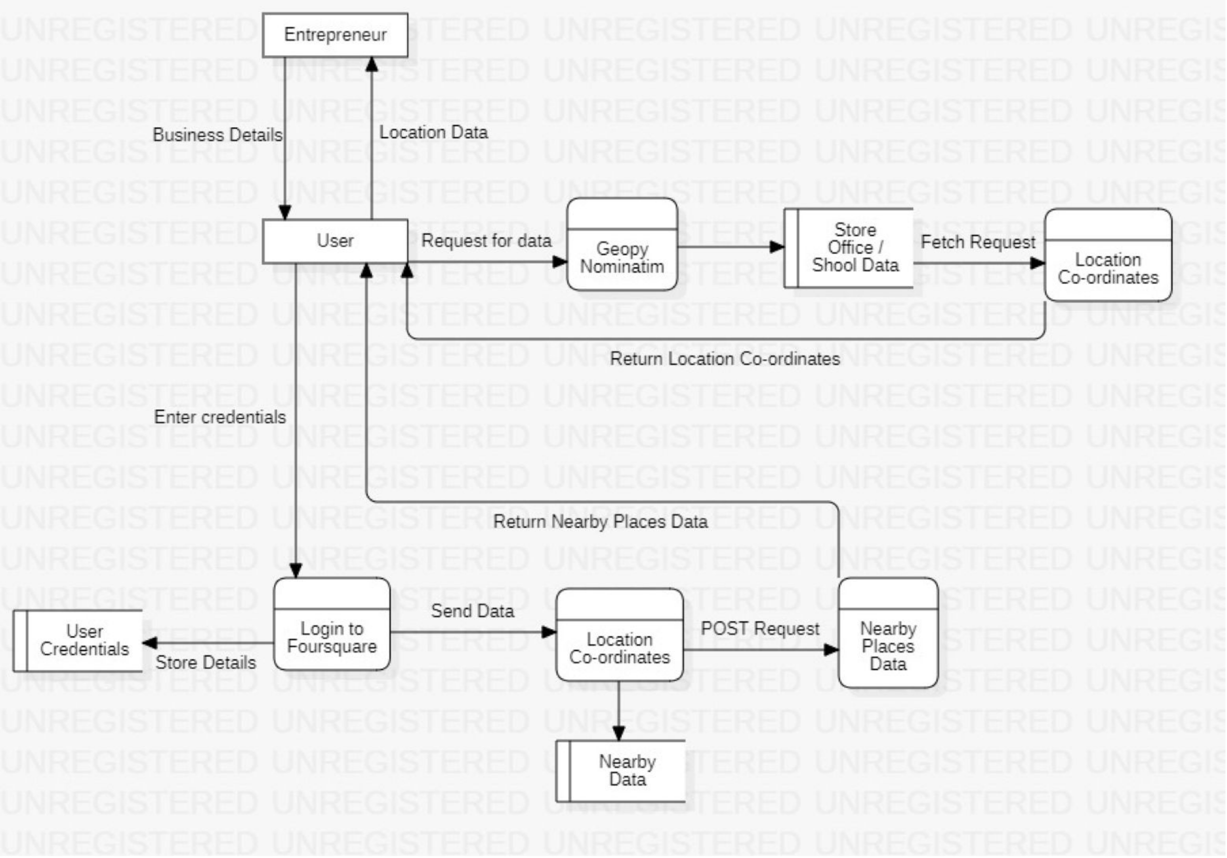


Fig 3.3: DFD Diagram

3.4 CLASS DIAGRAM

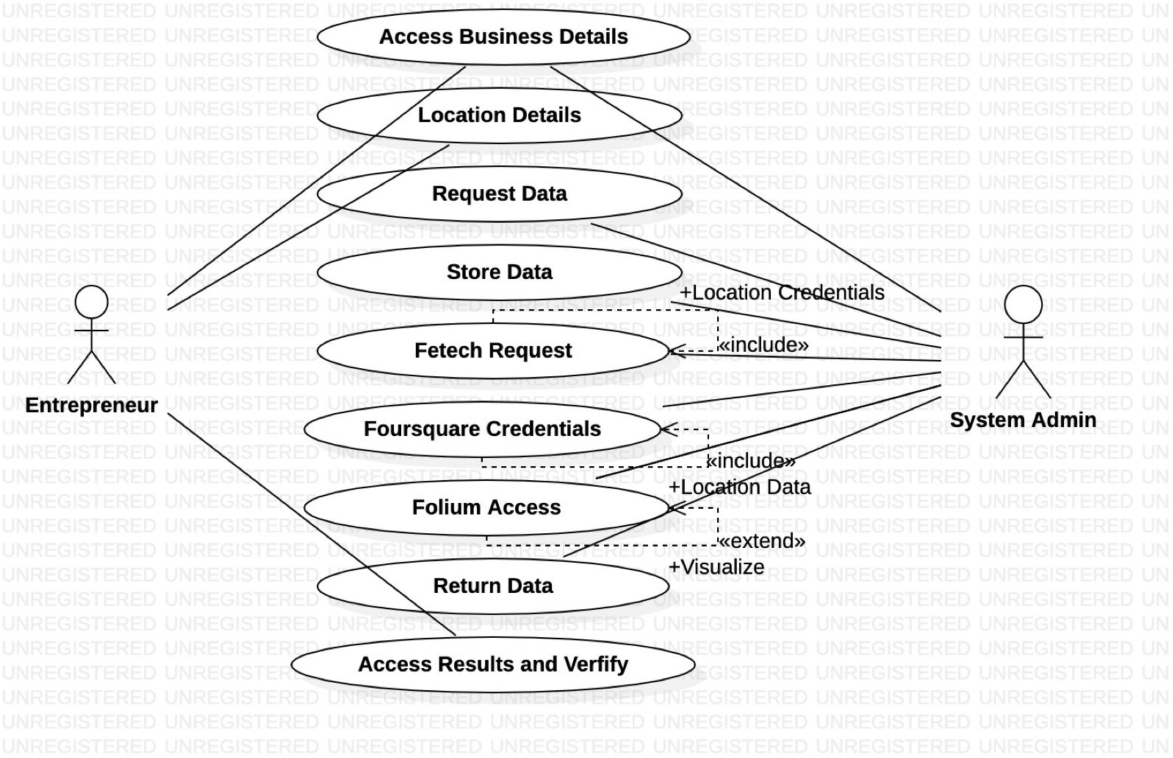


Fig 3.4: Use Case Diagram

3.5 SEQUENCE DIAGRAM

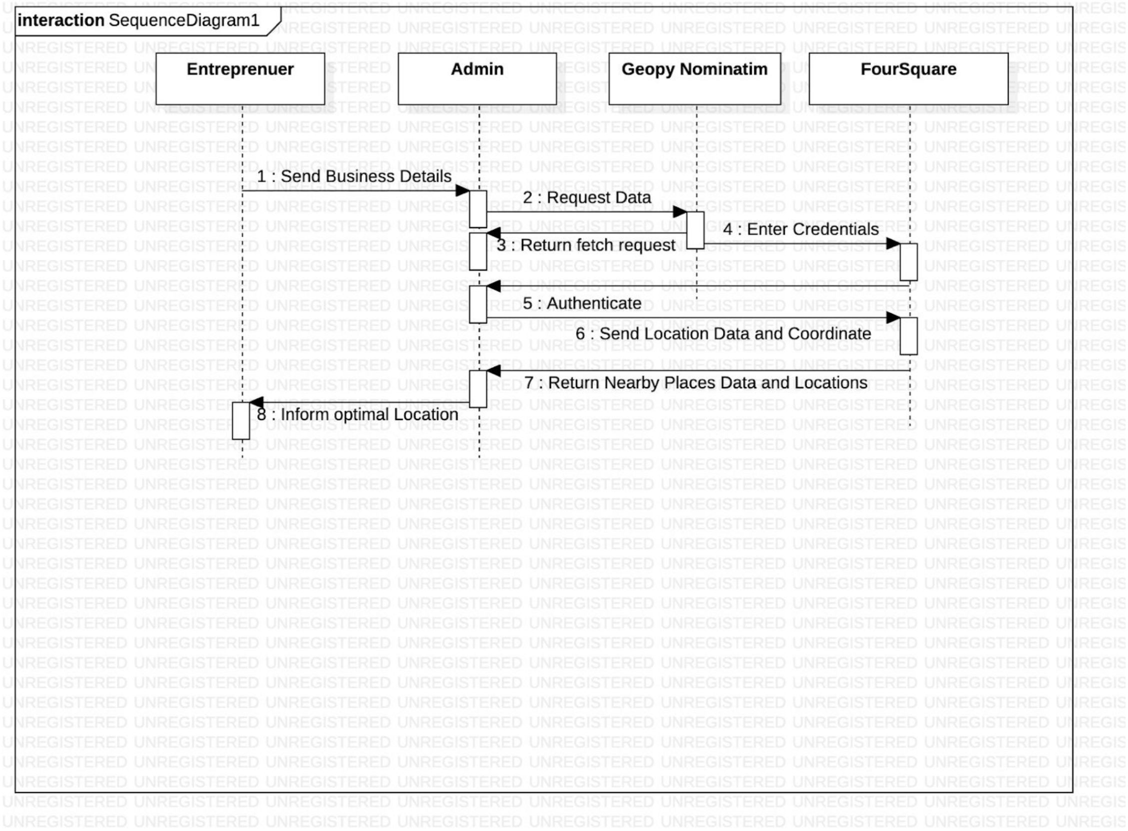

interaction SequenceDiagram1
Entreprenuer
Admin
Geopy Nominatim
FourSquare
1 : Send Business Details
2 : Request Data
4 : Enter Credentials
3 : Return fetch request
5 : Authenticate
6 : Send Location Data and Coordinate
7 : Return Nearby Places Data and Locations
8 : Inform optimal Location

Fig 3.5: Sequence Diagram

3.6 CLASS DIAGRAM

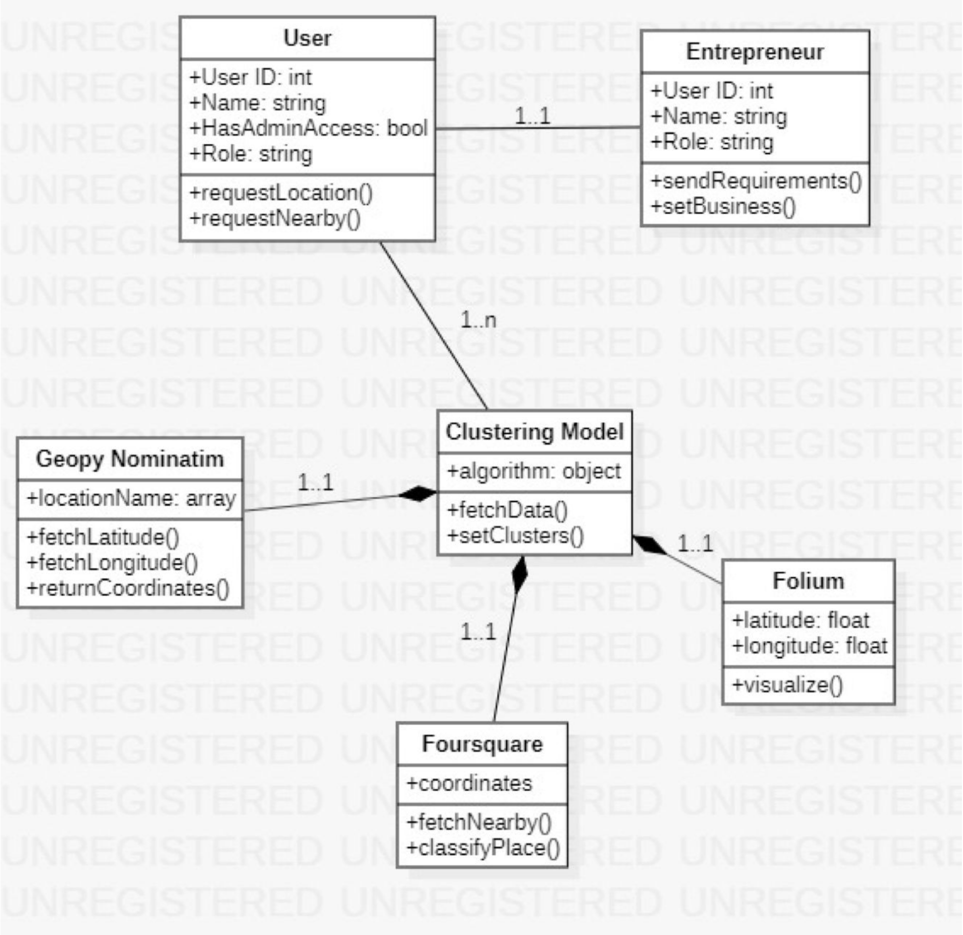


Fig 3.6: Class Diagram

3.7 INTERACTION DIAGRAM

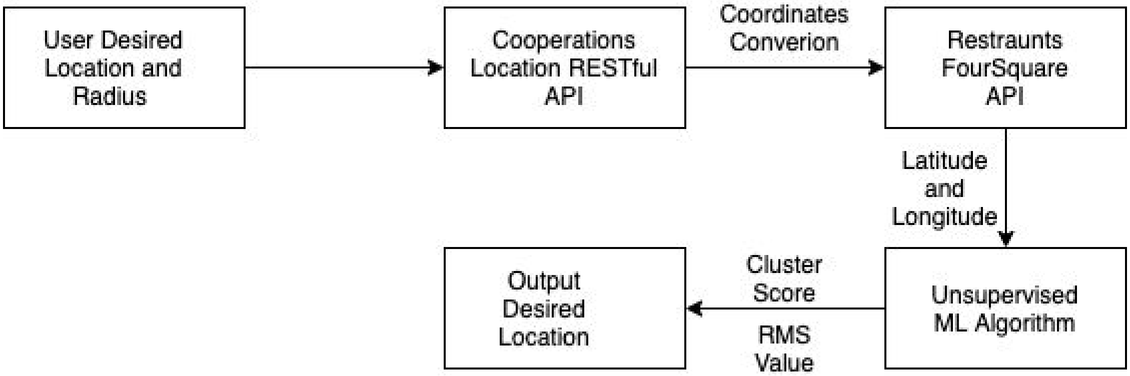

User Desired Location and Radius
Cooperations Location RESTful API
Coordinates Converion
Restraunts FourSquare API
Latitude and Longitude
Unsupervised ML Algorithm
Cluster Score
RMS Value
Output Desired Location

Fig 3.7: Interaction Diagram

3.8 COMPONENT & DEPLOYMENT DIAGRAM

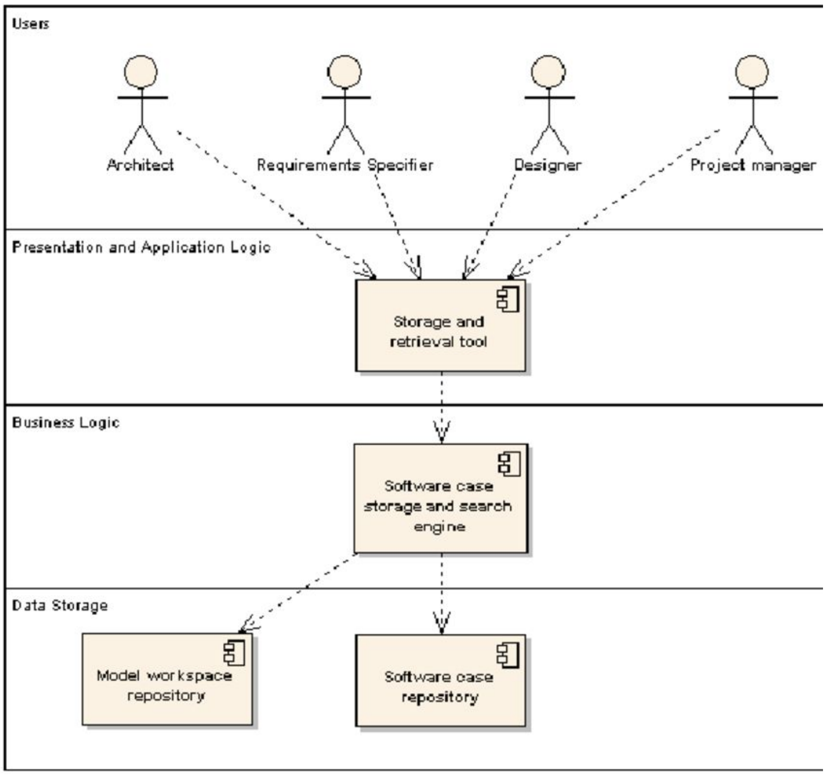

Users
Architect
Requirements Specifier
Designer
Project manager
Presentation and Application Logic
Storage and retrieval tool
Business Logic
Software case storage and search engine
Data Storage
Model workspace repository
Software case repository

Fig 3.8: Component Diagram

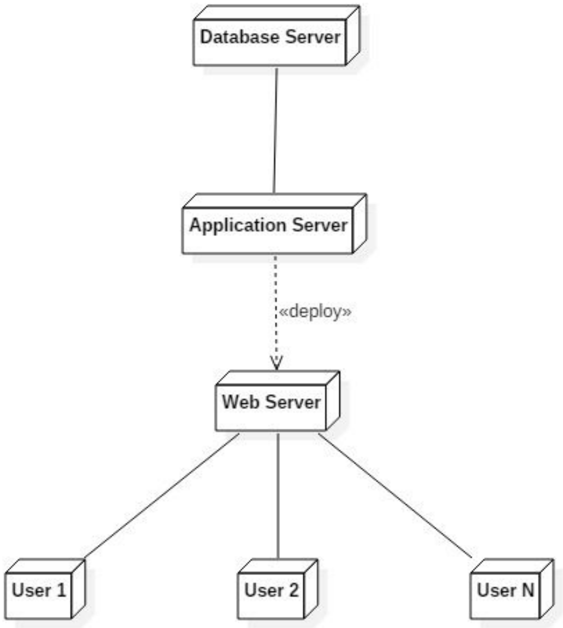


Fig 3.9: Deployment Diagram

CHAPTER 4

IMPLEMENTATION

4.1 DATABASE DESIGN

4.1.1 ER DIAGRAM

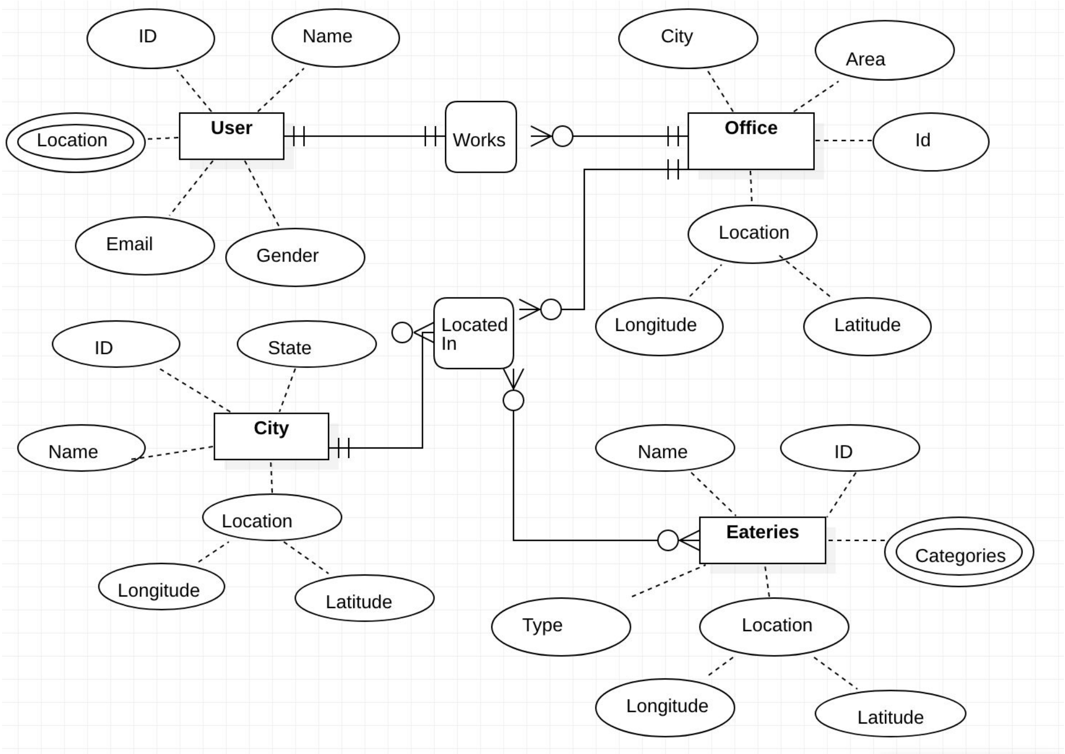


Fig 4.1: ER Diagram

4.1.2 RELATIONAL MODEL

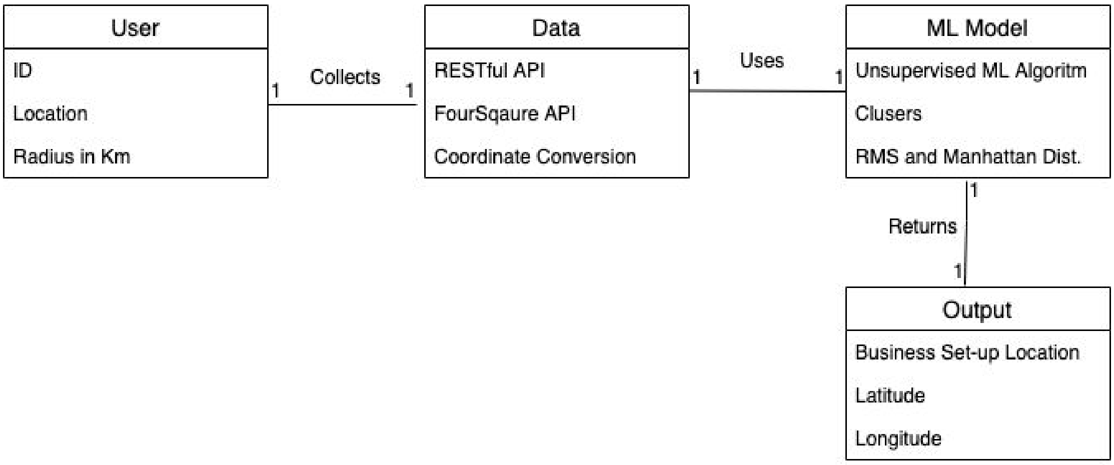
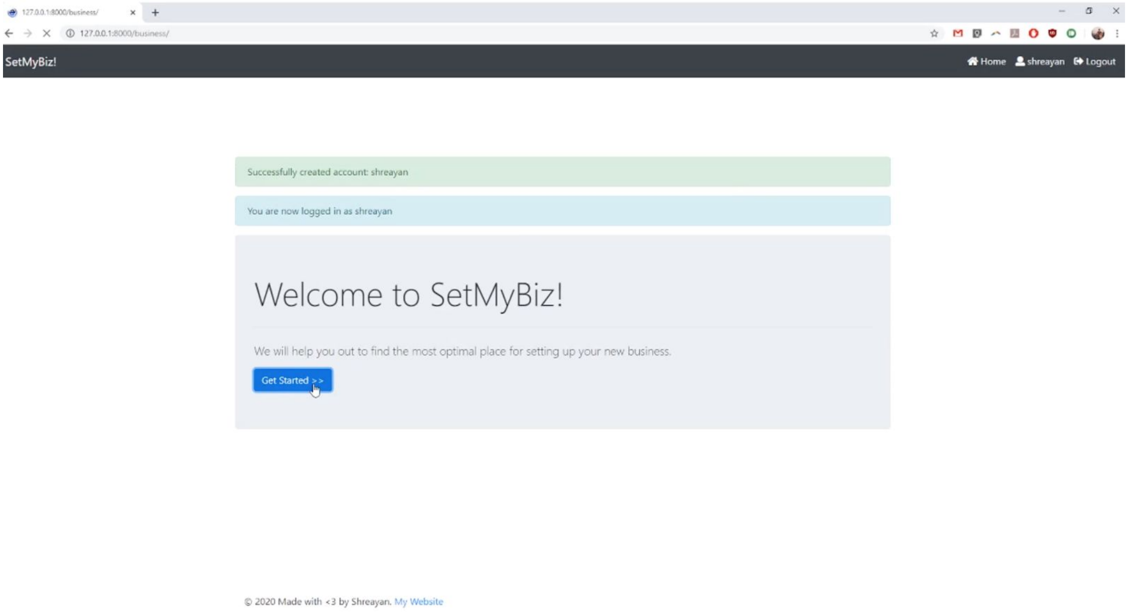


Fig 4.2: Relational Model

4.2 USER INTERFACE



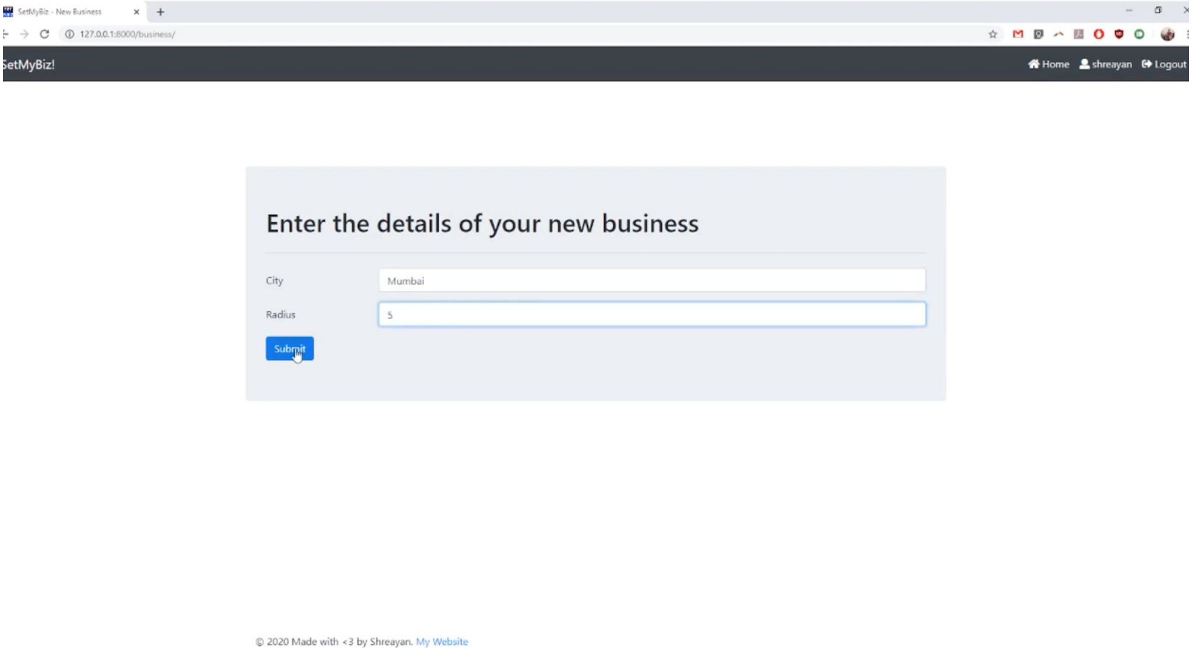
Successfully created account: shreayan

You are now logged in as shreayan

Welcome to SetMyBiz!

We will help you out to find the most optimal place for setting up your new business.

© 2020 Made with <3 by Shreayan. My Website



Enter the details of your new business

City: Mumbai

Radius: 5

© 2020 Made with <3 by Shreayan. My Website

Fig 4.3: User Interface

The project will be deployed as a web application running on the cloud that includes web interface as well separately offering ML Solution through a dedicated quasi-RESTful API.

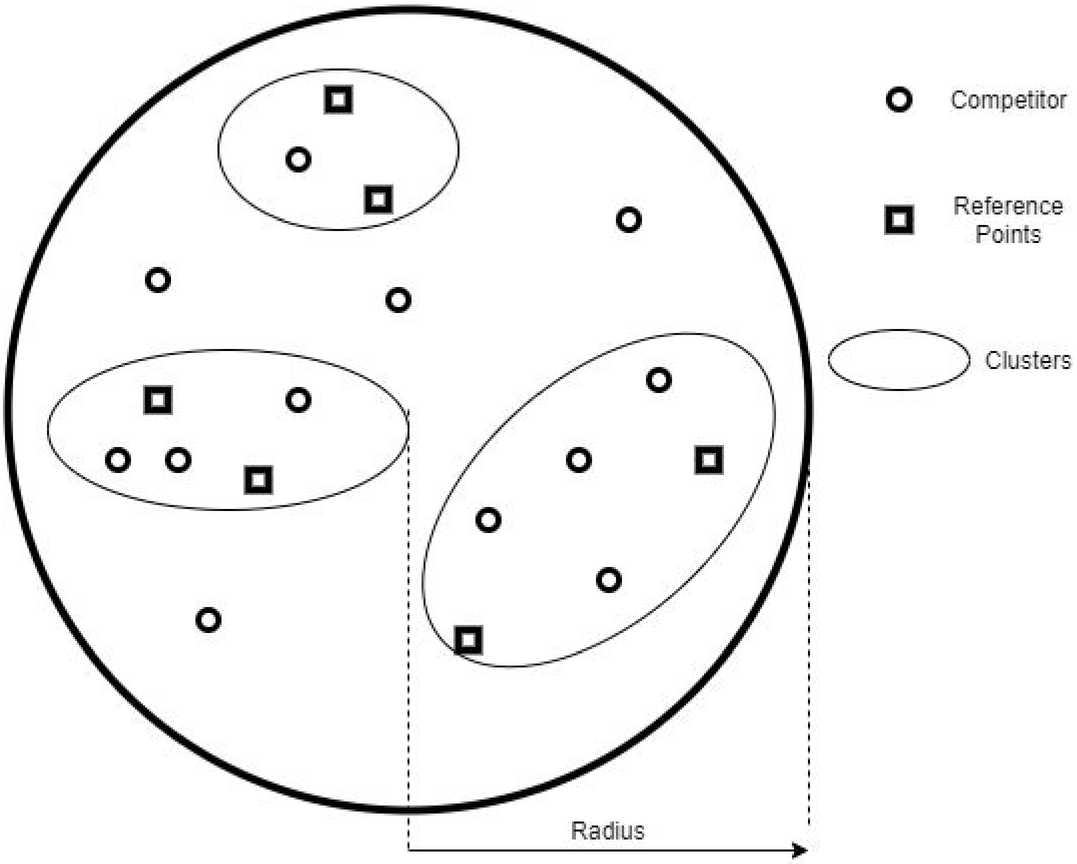


Fig.4.4: Diagrammatic Representation of how clusters will be formed to minimize the Proximity measure and the RMS distance from the cluster centre to reference points.

4.3 MIDDLEWARE

Since this is a research oriented project intended to look at practically applying a new theory, there is no middleware involved with the project. This project consists of a few simple python modules that run on the terminal/command prompt of a PC. Hence, there is no middleware involved in this project.

CHAPTER 5

VERIFICATION AND VALIDATION

5.1 UNIT TESTING

Unit testing is a type of testing in which individual modules and functions of code are individually tested to ensure that they work independently. It is done after the completion of the individual modules, classes and functions. The purpose is to make sure that every unit of the software works as intended.

Table 5.1: Unit Testing

Module Name	Module Information	Input	Expected Output	Actual Output	Result
User	It is used by users to sign up and then login user credentials.	User Login In Credentials	Form Interface Successful Log In	Form Interface Successful Log In	Pass
User	It is used by users to enter locations where they want to open business followed by radius in Km.	User enters City	City Exist	City Exist	Pass
User	It is used by users to enter locations where they want to open business followed by radius in Km.	User enters Radius in Km	Radius in Km	Radius in Km	Pass
Load Data	It is used to collect Data. Data is collected using RESTful API.	Scrapped Wikipedia Data	List of all Cooperation within the City	List of all Cooperation within the City	Pass
Load Data	It is used to collect Data. Data is collected using the FourSquare API.	FourSquare API Client ID and Status	List of all restaurants nearby all corporations	List of all restaurants nearby all corporations	Pass
Data Preparation	This module saves the various parameters taken as input as a file and results of API are converted to Coordinates.	Location Data	Coordinate Latitude and Longitude	Coordinate Latitude and Longitude	Pass
Model Data	This module uses unsupervised ML algorithms to form clusters	unsupervised ML Algorithm	Train and Clusters Data points to determine RMS	Train and Clusters Data points to determine RMS	Pass
Output Result	This module outputs the latitude and longitude suitable.	RMS and Manhattan Distance	Least Manhattan Distance	Least Manhattan Distance	Pass

5.2 INTEGRATION TESTING

Integration testing is a type of testing where all the modules and classes are meshed together and integrated as a group. The goal is to detect potential issues in the interactions between the grouped units. It is performed after unit testing. Once all the units are created and individually tested, we start combining those modules and start doing the integrated testing.

Table 5.2: Integration Testing

Test Description	Test Steps	Test Data	Expected Result (Accuracy)	Actual Result (Accuracy)
Distance Calculation	Selecting each cluster and calculating the Manhattan and RMS Distance	Data Points and Reference point within Cluster	12.96,77.67 corresponds to least Manhattan Distance	12.96,77.67 corresponds to least Manhattan Distance

5.3 USER TESTING

User Acceptance Testing is a type of testing done by the user to verify and validate whether all the requirements are met or not. The user will decide if the system is usable or not. It also ensures that the system satisfies the functional requirements.

Table 5.3: User Testing

Test Description	Test Steps	Input	Expected Result	Actual Result
Check buttons availability	Open the application and click ‘Submit’ before choosing the image	Application	‘Submit button is unavailable. Data not fetched.’	Submit button is unavailable. Data not fetched.’
Check functionality of application	click ‘Submit’ after entering login Credentials	City and Radius in Km	Cluster Formation	Cluster Formation
	click ‘Submit’ after entering City and Radius in Km	API Data Coordinates and unsupervised Model	Latitude and Longitude	Latitude and Longitude

5.4 SIZE- LOC

The project source consists of seven different files, each having their own responsibilities. The **model.py, gui.py** and **app.py** files are the most crucial files for the project.

Table 5.4: Various files and lines of code for different types of lines

File	**LOC**
model.py	274
views.py	118
base.html	24
index.html	20
app.py	114
main.css	30
main.js	25
Total	605

5.5 COST ANALYSIS

5.5.1 Schedule Cost

As we had to test the trained model on a smaller dataset, we faced multiple issues that lead to a loss of man-hours. An on-the-go learning process took time as well as there were multiple bugs that were being faced for the first time. A lot of resources and advice from members of academic background helped in providing solutions to these problems. Thus, additional man-hours were spent on learning and implementing the model correctly.

Due to lack of experience with Quasi-RESTful API and FourSquare API, multiple errors caused us to retrain the code multiple times. Due to the type and nature of bugs, there was a loss of a few day's worth of man- hours. However, this slippage due to lack of experience was taken into account during the initial risk analysis.

5.5.2 **Budget Cost**

Due to the independent research nature of this project, there was no financial cost involved. The development of the project was performed for free. However, deploying over or using the cloud services could cost the user.

5.6 DEFECT ANALYSIS

Implementation phase had the greatest number of defects. The sources of these defects were:

1. Defects from improper installation of necessary dependencies

Due to the improper installation of necessary dependencies, we faced issues like corrupted operating systems and inability to run the code. This leads to the loss of man-hours as the code could not be run and loss of system usage.

2. Defects due to Data Inconsistencies

Data inconsistencies may be present in the input data imported from google API. These API and scrapped data may contain inconsistencies such as missing details or incorrect details..

3. Defects due to incorrect or unavailable location information

Lack or wrong information could lead to incorrect results. Since population density information is unavailable, the result is calculated without it's consideration. Population density may or may not affect the results in near future.

4. Defects due to Application Server Time-out

Django server may reach max-time-out while scraping the data leading to application error and other server difficulties.

5.7 MC CALL'S QUALITY FACTORS

Product Operation Factors:

5.7.1. **Correctness-** The model will work and recommends the place suitable for setting up business in terms of latitude and longitude.

5.7.2. **Reliability-** The model will take a maximum of 1 minute to generate all results.

5.7.3. **Efficiency-** The saved model takes up 2MB of space on the system.

5.7.4. **Integrity-** System administrators can make no changes after submitting the state and radius parameters.

5.7.5. **Usability**- It requires no additional knowledge for operating the system and model; hence, it is very easy to use.

Product Revision Factors:

5.7.6. **Maintainability-** The system will need to be retrained every time a change is made in the state and radius parameter.

5.7.7. **Flexibility-** The core model will remain the same for each system version. So, changes are only made to the user interface and the model is flexible enough to work through user requirements.

5.7.8. **Testability-** System will work efficiently to recommend the place for setting up the business.

Product Transition Factors:

5.7.9. **Portability-** Since the underlying code is in Python, we can easily port the system to other operating systems by only installing Python and the required libraries. No specific hardware is needed to run the system.

5.7.10. **Reusability-** All modules of the system are reusable for time series projects and efficiently work with minimum changes.

5.7.11. **Interoperability-** The system can prove to be interoperable in the near future if changes are made to the codebase.

CHAPTER 6

EXPERIMENTS RESULTS AND ANALYSIS

6.1 RESULTS

The Model returns the latitude and longitude of the place that is suitable for entrepreneurs to set up a new business. This place involves less risk and is capable of attracting more customers thereby increasing profits. The place returned has a proper balance between the number of competitors in its vicinity and the target audience. It minimizes the number of competitors while maximizing the number of reference points in its vicinity. Each cluster will return the latitude, longitude and the RMS of the distances from the reference point to the centre of that cluster.

Given below are the metrics and formulae that have been used for evaluating the clusters:

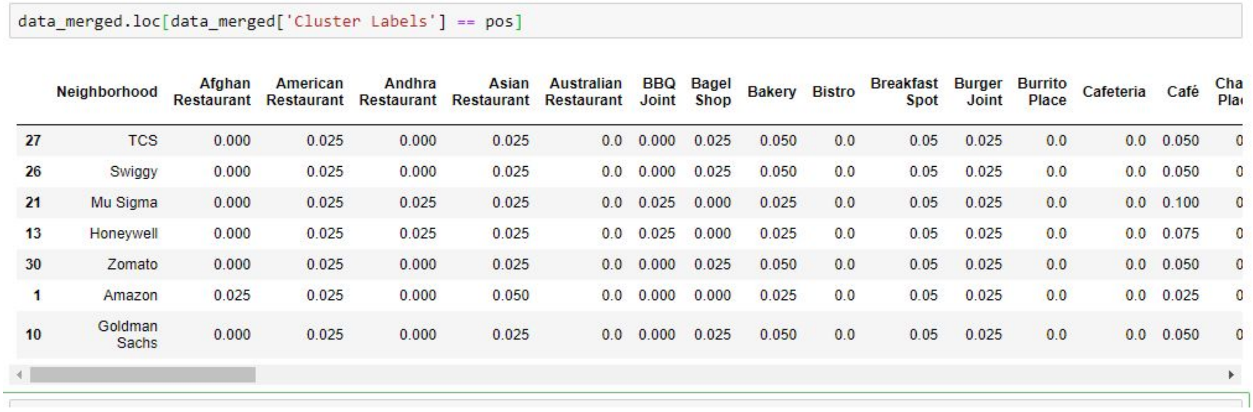
$$Haversine's\ Formula : Distance\ d\ between\ 2\ coordinates\ (x_1, y_1)\ and\ (x_2, y_2)$$

$$d\ =\ 2r\ arcsin\ (\sqrt{sin^2(\frac{x_2 - x_1}{2}) + cos(x_1)\ cos(x_2)\ sin^2(\frac{y_2 - y_1}{2})}\ )$$

$$Cluster\ Score\ =\ \frac{Proximity\ Measure}{Number\ of\ Competitors\ in\ that\ cluster}$$

$$Proximity\ Measure\ =\ \frac{RMS\ of\ elements\ in\ cluster}{Number\ of\ elements\ in\ cluster}$$

The Manhattan and RMS distance is computed for clusters and used for each cluster score calculation. The clusters that have only a single reference point in them have been neglected since their cluster score will be zero.



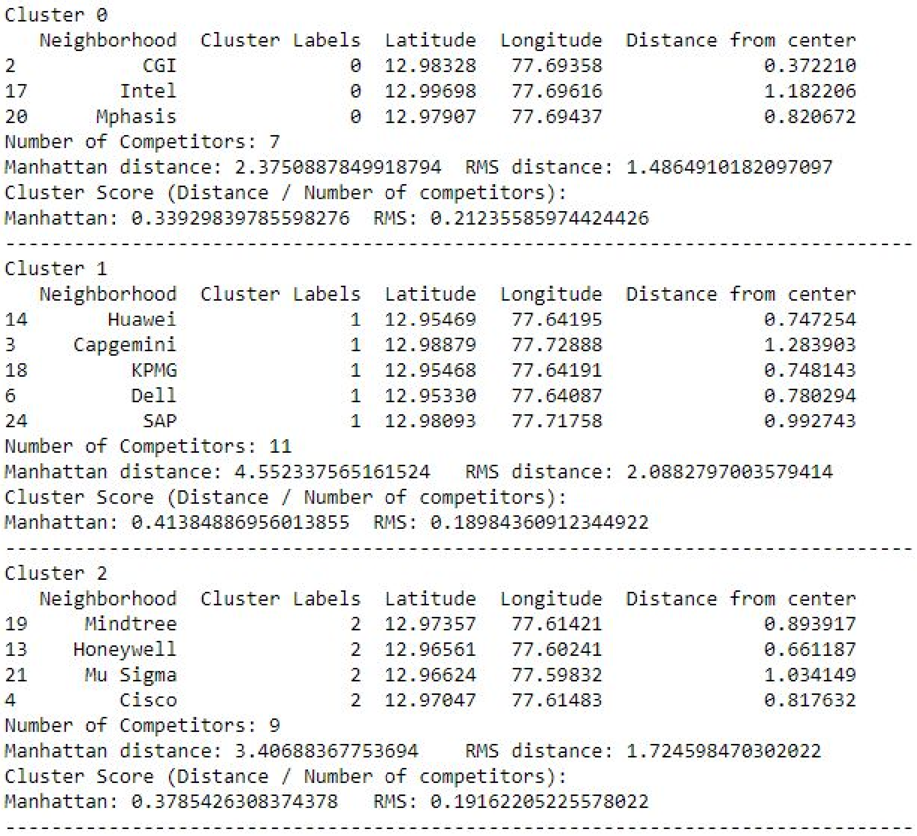
```
data_merged.loc[data_merged['Cluster Labels'] == pos]
```

	Neighborhood	Afghan Restaurant	American Restaurant	Andhra Restaurant	Asian Restaurant	Australian Restaurant	BBQ Joint	Bagel Shop	Bakery	Bistro	Breakfast Spot	Burger Joint	Burrito Place	Cafeteria	Café	Cha Pla
27	TCS	0.000	0.025	0.000	0.025	0.0	0.000	0.025	0.050	0.0	0.05	0.025	0.0	0.0	0.050	0
26	Swiggy	0.000	0.025	0.000	0.025	0.0	0.000	0.025	0.050	0.0	0.05	0.025	0.0	0.0	0.050	0
21	Mu Sigma	0.000	0.025	0.025	0.025	0.0	0.025	0.000	0.025	0.0	0.05	0.025	0.0	0.0	0.100	0
13	Honeywell	0.000	0.025	0.025	0.025	0.0	0.025	0.000	0.025	0.0	0.05	0.025	0.0	0.0	0.075	0
30	Zomato	0.000	0.025	0.000	0.025	0.0	0.000	0.025	0.050	0.0	0.05	0.025	0.0	0.0	0.050	0
1	Amazon	0.025	0.025	0.000	0.050	0.0	0.000	0.000	0.025	0.0	0.05	0.025	0.0	0.0	0.025	0
10	Goldman Sachs	0.000	0.025	0.000	0.025	0.0	0.000	0.025	0.050	0.0	0.05	0.025	0.0	0.0	0.050	0

Fig 6.1.Nearby Restaurants

6.2 RESULT ANALYSIS

We have used the Haversine formula for calculating the distance between two relative points and the cluster accuracy is calculated by using the RMS of elements in clusters and the number of elements in the cluster. The cluster score is calculated using RMS or Manhattan distance and the number of competitors in the cluster. The cluster with the minimum manhattan distance has less number of competitors in that area and suitable for setting up the business in that location.



```
Cluster 0
   Neighborhood  Cluster Labels  Latitude  Longitude  Distance from center
2           CGI               0  12.98328   77.69358              0.372210
17        Intel               0  12.99698   77.69616              1.182206
20      Mphasis               0  12.97907   77.69437              0.820672
Number of Competitors: 7
Manhattan distance: 2.3750887849918794  RMS distance: 1.4864910182097097
Cluster Score (Distance / Number of competitors):
Manhattan: 0.33929839785598276  RMS: 0.21235585974424426
------------------------------------------------------------------------------
Cluster 1
   Neighborhood  Cluster Labels  Latitude  Longitude  Distance from center
14       Huawei               1  12.95469   77.64195              0.747254
3     Capgemini               1  12.98879   77.72888              1.283903
18         KPMG               1  12.95468   77.64191              0.748143
6          Dell               1  12.95330   77.64087              0.780294
24          SAP               1  12.98093   77.71758              0.992743
Number of Competitors: 11
Manhattan distance: 4.552337565161524   RMS distance: 2.0882797003579414
Cluster Score (Distance / Number of competitors):
Manhattan: 0.41384886956013855  RMS: 0.18984360912344922
------------------------------------------------------------------------------
Cluster 2
   Neighborhood  Cluster Labels  Latitude  Longitude  Distance from center
19     Mindtree               2  12.97357   77.61421              0.893917
13    Honeywell               2  12.96561   77.60241              0.661187
21     Mu Sigma               2  12.96624   77.59832              1.034149
4         Cisco               2  12.97047   77.61483              0.817632
Number of Competitors: 9
Manhattan distance: 3.40688367753694    RMS distance: 1.724598470302022
Cluster Score (Distance / Number of competitors):
Manhattan: 0.3785426308374378   RMS: 0.19162205225578022
------------------------------------------------------------------------------
```

Fig.6.2. cluster scores calculation with a radius of 2.5km, n_clusters=10.

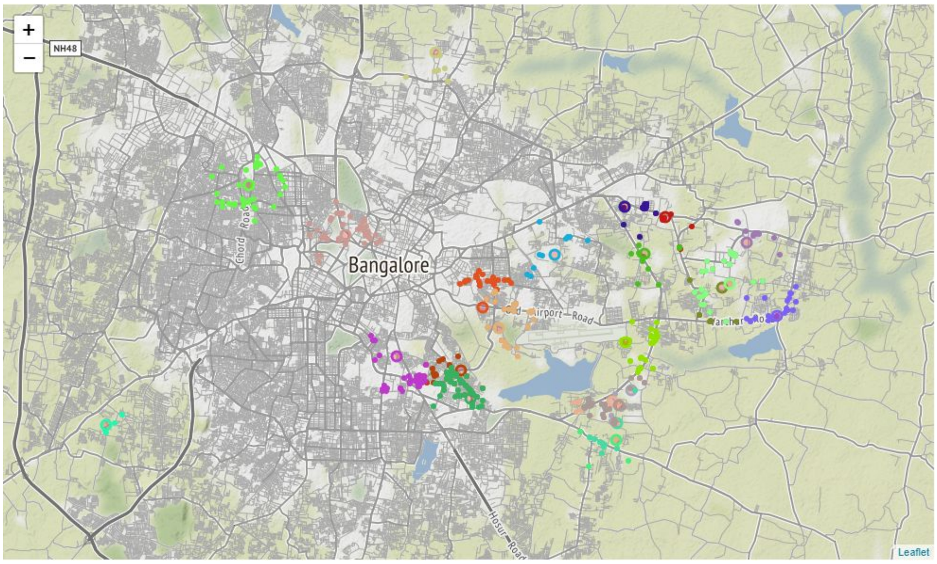


Fig 6.2: Nearby Restaurants

6.3 CONCLUSION & FUTURE WORK

Upon sorting the clusters by decreasing order of cluster score, one will get the best places to set up a new business. These places balance the tradeoff between having the most target audience and minimizing the number of competitors for the same. Hence the cluster score will be the most suitable metric to determine the most optimal place for setting up a new business.

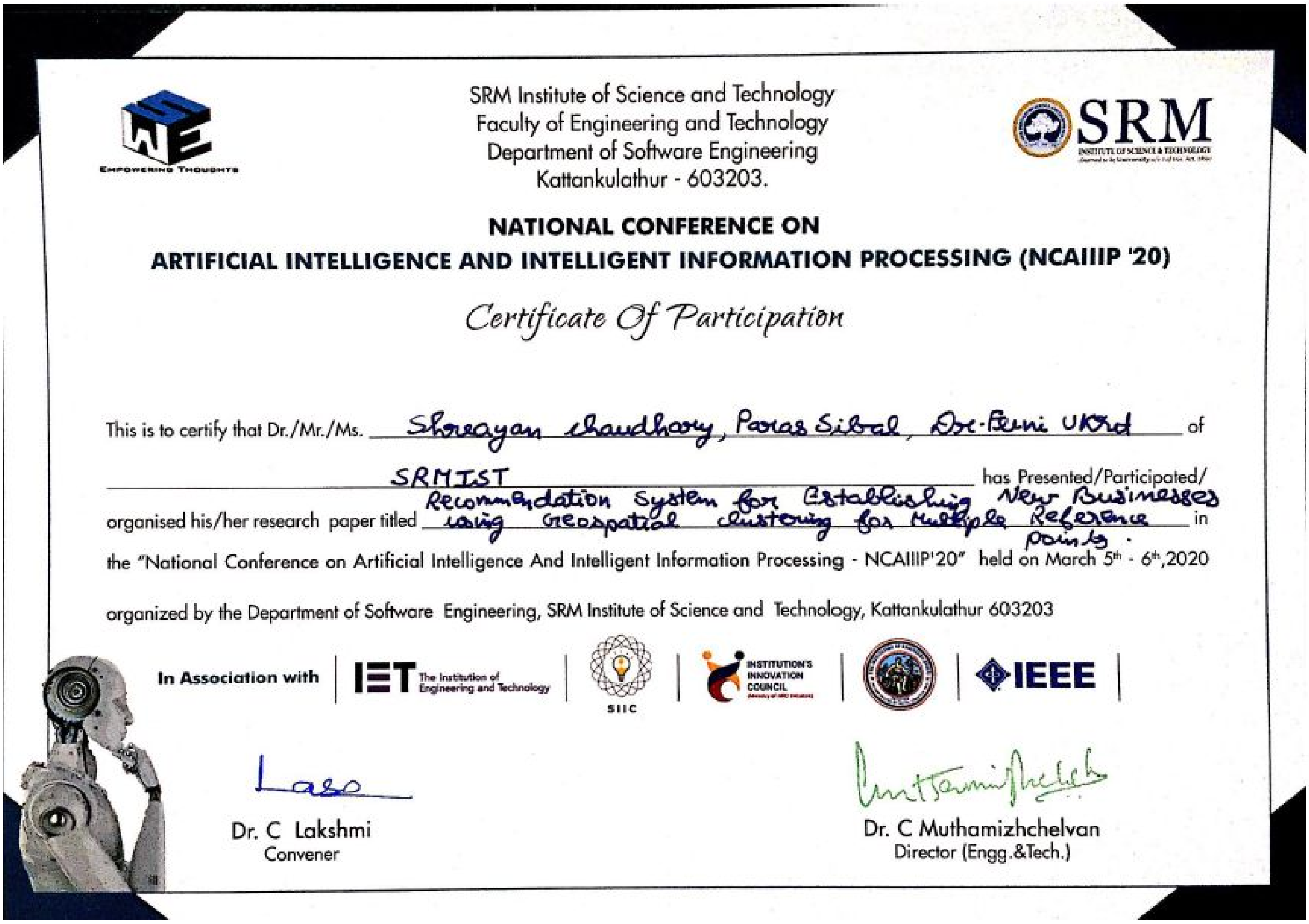
In our experiments for place = Bangalore and radius = 2.5km, we found the best place to set up the business to be around the coordinates (12.966478, 77.674238). There are several corporate offices in the vicinity and very few competitors. The place is located near Huawei, Capgemini, KPMG, Dell and SAP. It would be one of the best places to set up a new restaurant with the least risk involved and will attract more customers thereby maximizing profits.

This approach would work equally well for other kinds of businesses such as Petrol Pumps, Entertainment Places, Hospitals, ATMs, Shopping Centres, Hotels, etc. Hence it will be a convenient method which would save time, money and risk involved in the complex process of becoming an entrepreneur.

The model's performance can be improved by adding a feature of the population of the place. It may be possible in the near future to determine the approximate population of a place using GPS in cell phones. By adding this feature, the model will also consider the population density of an area while it forms the clusters [4, 5]. This will improve the quality and accuracy of the clusters being formed, consequently improve the precision of the results.

6.4 JOURNAL PAPER

The research paper was completed and presented at the National Conference on Artificial Intelligence and Intelligent Information Processing NCAIIIP 2020, on 5th and 6th March 2020.



EMPOWERING THOUGHTS

SRM Institute of Science and Technology
Faculty of Engineering and Technology
Department of Software Engineering
Kattankulathur - 603203.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY

NATIONAL CONFERENCE ON
ARTIFICIAL INTELLIGENCE AND INTELLIGENT INFORMATION PROCESSING (NCAIIIP '20)

Certificate Of Participation

This is to certify that Dr./Mr./Ms. Shreayan chaudhary, Paras Sibal, Dr. Beni Ukrd of SRMIST has Presented/Participated/ organised his/her research paper titled Recommendation System for Establishing New Businesses using Geospatial clustering for Multiple Reference points. in the "National Conference on Artificial Intelligence And Intelligent Information Processing - NCAIIIP'20" held on March 5th - 6th,2020 organized by the Department of Software Engineering, SRM Institute of Science and Technology, Kattankulathur 603203

In Association with | IET The Institution of Engineering and Technology | SIIC | INSTITUTION'S INNOVATION COUNCIL | IEEE |

Laso
Dr. C Lakshmi
Convener

Dr. C Muthamizhchelvan
Director (Engg.&Tech.)

REFERENCES

[1] Y. Zhong, J. Li and S. Zhu, "Clustering Geospatial Data for Multiple Reference Points," in IEEE Access, vol. 7, pp. 132423-132429, 2019.

[2] Rabindra Barik, Ankita Tripathi, Suraj Sharma, Vinay Kumar, Himansu Das. MistGIS: Optimizing Geospatial Data Analysis Using Mist Computing. Progress in Computing, Analytics and Networking Journal 2018 https://doi.org/10.1016/j.compenvurbsys.2011.11.003

[3] Santosh Nirmal, Comparative Study between K-Means and K-Medoids Clustering Algorithms, International Research Journal of Engineering and Technology (IRJET) Volume: 06 Issue: 03 (2019) p-ISSN: 2395-0072 e-ISSN: 2395-0056.

[4] Alexey Golubev, Ilya Chechetkin, Danila Parygin, Alexander Sokolov, Maxim Shcherbakov, Geospatial Data Generation and Preprocessing Tools for Urban Computing System Development, Procedia Computer Science Journal Volume 101, 2016, Pages 217-226 https://doi.org/10.1016/j.procs.2016.11.026

[5] Alexey Golubev, Ilya Chechetkin, Danila Parygin, Alexander Sokolov, Maxim Shcherbakov. Geospatial Data Generation and Preprocessing Tools for Computing System Development https://doi.org/10.1016/j.procs.2016.11.026 Procedia Computer Science Journal Volume 101, 2016, Pages 217-22.