Distribution of basic amenities in COVID-19 pandemic and prediction of potential emergency areas

Submitted By: Team 23

Submitted to:

Venkataramanan Subramanian
Amrut Joshi
Prof. Prakash Yalla
Prof. Vasudeva Varma

Problem Statement

In view of the current COVID-19 pandemic situation, we are trying to:

Provide essential services to people stuck in pandemic.

Providing a web interface for people to access the services the need like:

1) Ambulance Service
2) Medical Service
3) Grocery services
4) Drinking Water Supply

as per their needs.

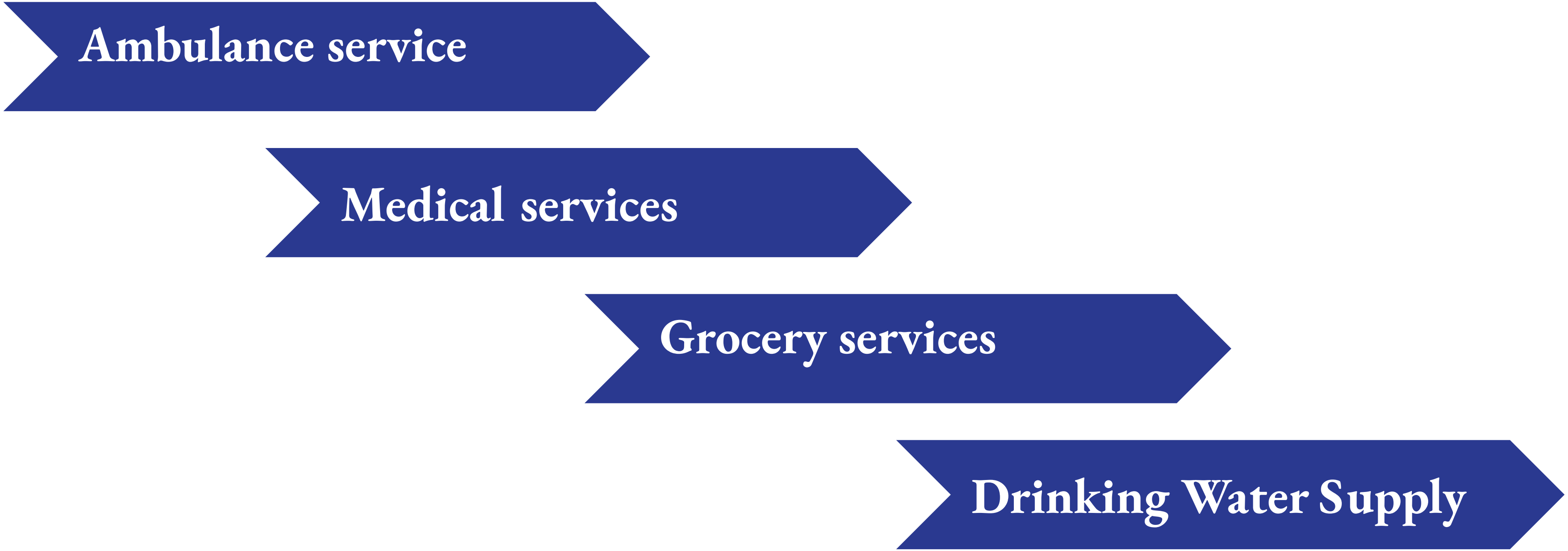
Predict potential emergency areas, so as to provide necessary services in those areas.

Using models like:

1) Linear regression
2) Neural network
3) Perceptron model
4) LSTM
5) Multinomial Naive Bayes
6) Gaussian process regression

On the data collected for India, emergency areas are predicted.

Services Provided By Our Project

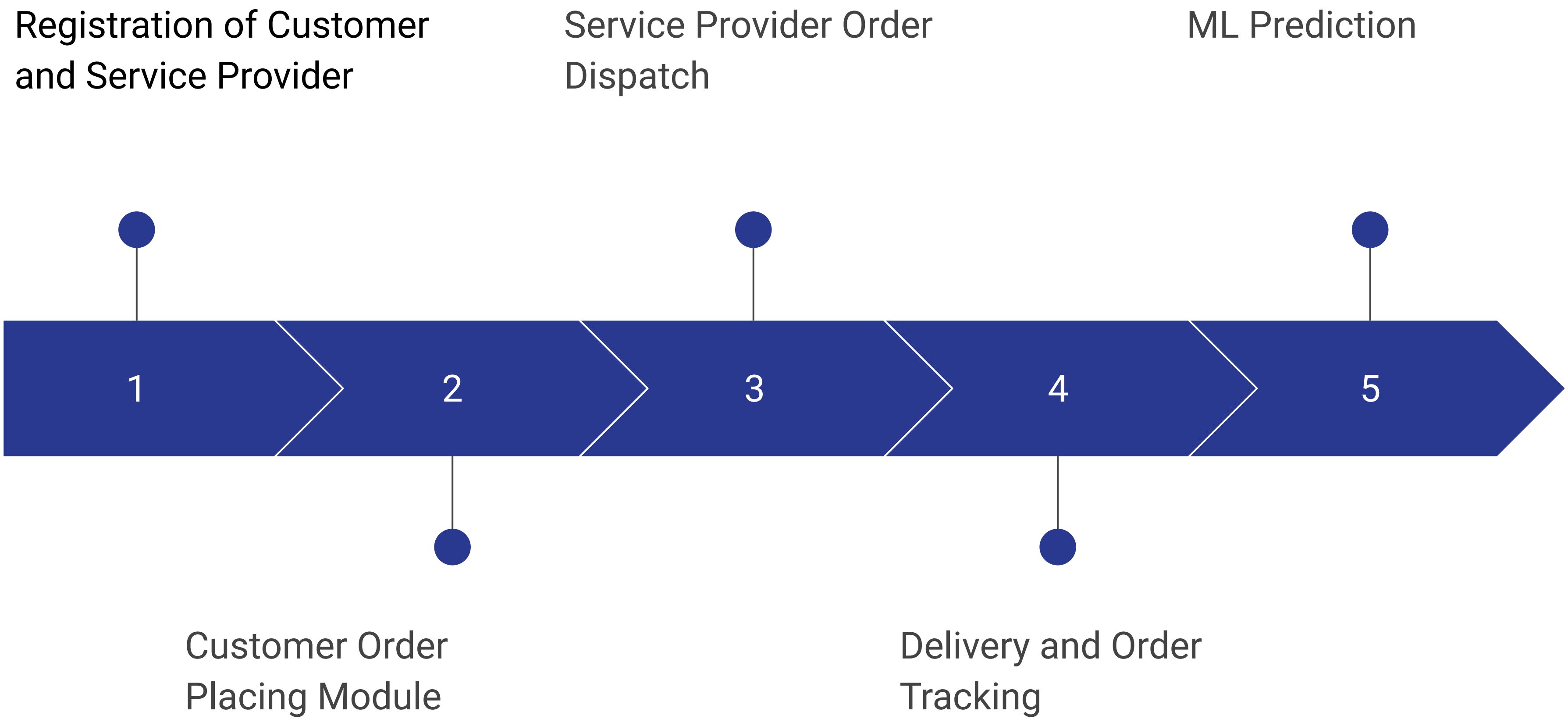


- Ambulance service
- Medical services
- Grocery services
- Drinking Water Supply

Solution

We designed a web application which provides interface to different kinds of users for accessing the services the want to access or provide.

Implementation Of Web Application



Registration of Customer and Service Provider

Customer Order Placing Module

Service Provider Order Dispatch

Delivery and Order Tracking

ML Prediction

Registration

- Registration can be done by three kinds of users, namely
    - End user
    - Service Provider
    - Delivery Provider
- The following are the mandatory fields for registering end user:
    - Email address, username
    - Password
    - Latitude, Longitude
- The following are the mandatory fields for registering service provider:
    - Email Address and user name
    - Password
    - Latitude, Longitude
    - Service Type
- The following are the mandatory fields for registering delivery person:
    - Email Address and User name
    - Password

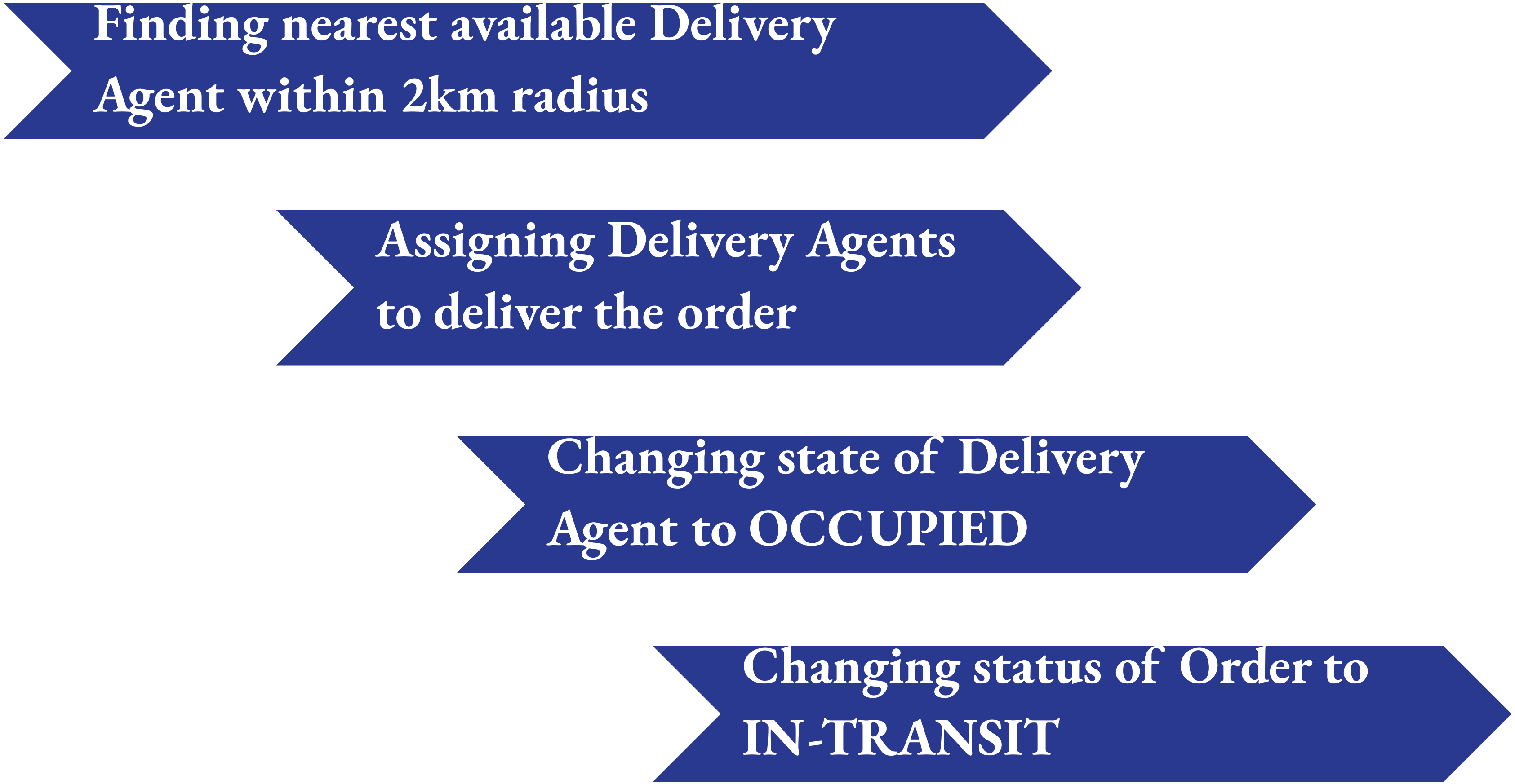
Service and Dispatch

Service Providers can only provide service based on its service type.
Four types of service providers:

1) **Groceries**
2) **Ambulance**
3) **Medical Suppy**
4) **Water Supply**

Orders which have not been dispatched yet will be shown to a service provider.
Each service provider has a service radius, order coming from within that radius will only appear under the **Order List** of that particular provider.
Orders can be dispatched by the service providers.
Orders which have been dispatched no longer appears on the **Order List** of service providers.
After dispatching the order, an entry is made in Transaction table OrderDispatch which is further used by the **Delivery System**.

Delivery and Order Tracking



Finding nearest available Delivery Agent within 2km radius

Assigning Delivery Agents to deliver the order

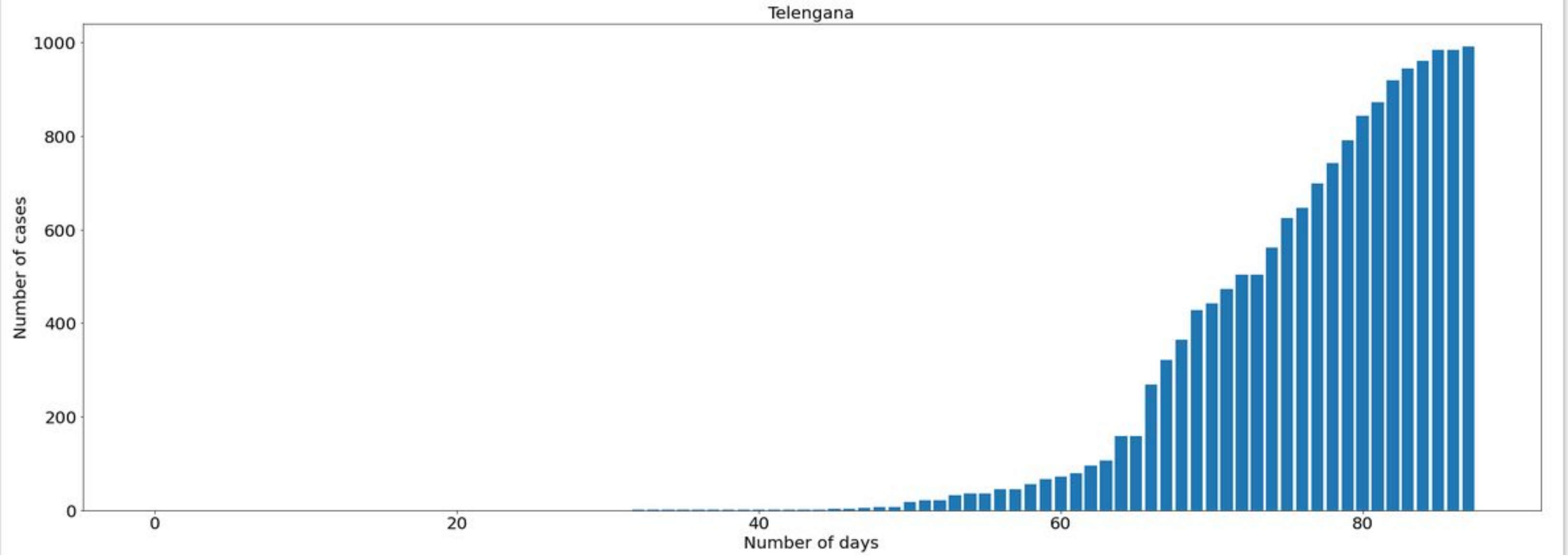
Changing state of Delivery Agent to OCCUPIED

Changing status of Order to IN-TRANSIT

Machine Learning Models Implementations

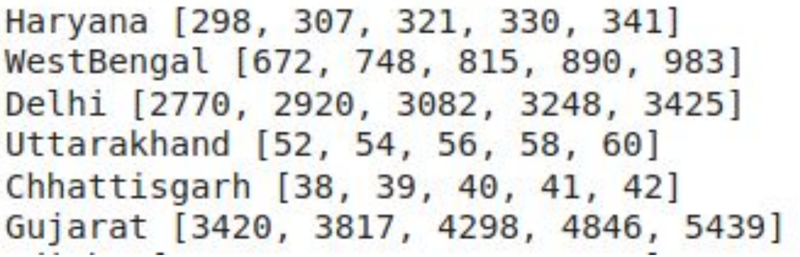
Data Preprocessing

- We first load the dataset.
- It has the number of cases that have come in the different states in different days between 30th January, 2020 to 26th April, 2020.
- After removing states having no cases, we plot the graphs of number of cases vs days for the different states. An example graph for Telengana is shown below.



Training and predicting using regression models

- We model this as a regression time series model. In this setion, we used Linear Regression and Multi Layer Perceptron model variations.
- We tested using different window sizes. Let's take an example of 5.
- So, we are using the data for 5 days in order to predict the number of cases for the 6th day.

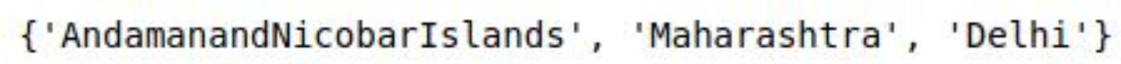


```
Haryana [298, 307, 321, 330, 341]
WestBengal [672, 748, 815, 890, 983]
Delhi [2770, 2920, 3082, 3248, 3425]
Uttarakhand [52, 54, 56, 58, 60]
Chhattisgarh [38, 39, 40, 41, 42]
Gujarat [3420, 3817, 4298, 4846, 5439]
```

- This is an example output which we get. It represents that the number of cases predicted in Haryana will be 298, 307 and so on for 27th April, 28th April respectively.

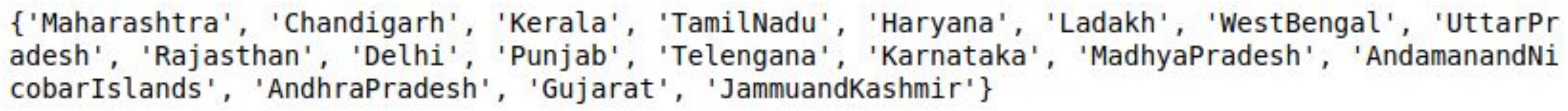
Predicting the states in emergency situation

- We use the population data as well in order to predict the states that may enter into potential emergency situation in the next few days
- We use a concept of threshold in order to do that.
- Let’s say for example if we take a threshold of 0.0001, it means that we will predict a state to be in a emergency condition if it’s number of cases in the next 5 days is more that (1/10000) times it’s population.
- We see that using a threshold of 0.001 we get an output like :



```
{'AndamanandNicobarIslands', 'Maharashtra', 'Delhi'}
```

- But on reducing the threshold and thus making it more strict, many more states are predicted as emergency. Output for threshold = 0.00001 is shown below



```
{'Maharashtra', 'Chandigarh', 'Kerala', 'TamilNadu', 'Haryana', 'Ladakh', 'WestBengal', 'UttarPr
adesh', 'Rajasthan', 'Delhi', 'Punjab', 'Telengana', 'Karnataka', 'MadhyaPradesh', 'AndamanandNi
cobarIslands', 'AndhraPradesh', 'Gujarat', 'JammuandKashmir'}
```

Multilayer Perceptron Model

- Using MLPClassifier from sklearn.neural_network library.
- Name of the region is taken as input, and data corresponding to that place is taken into account, having Date and number of confirmed cases on that date.
- Data is split in train and test. Train data is used to train MLPClassifier model. On the trained model predictions are done using test data.
- R2 Score is calculated to check the efficiency of the model.
- R2 Score for Andhra Pradesh came out to be -0.244.
- As we can see R2 Score < 0, which means that the model does not follow the trend of data. Hence, we move on to next model.

Long Short Term Memory

- Used LSTM from keras.layers library in python
- For the specified place, data is extracted, and the Date column is used as index of the dataset.
- LSTM is trained on train data using:
  - batch size =1
  - 500 epochs
  - time step =1
- R2 Score = 0.955
- As R2 Score > 0, it turned out to be a good model.

Multinomial Naive Bayes

- The multinomial Naive Bayes classifier is suitable for classification with discrete features. The multinomial distribution normally requires integer feature counts.
- Used MultinomialNB from sklearn.naive_bayes
- Using the Train and test split data, the model was trained.
- R2 Score = 0.307
- As R2 Score > 0, it turned out to be a good model.

Gaussian process regression

- **Gaussian Processes (GP)** are a generic supervised learning method designed to solve *regression* and *probabilistic classification* problems.
- Used GaussianProcessRegressor from sklearn.gaussian_process
- Using the Train and test split data, the model was trained.
- R2 Score = 0.9602
- As R2 Score > 0, it turned out to be a good model.
- Out of all the models used Gaussian process regression gave best R2 Score.
- It turned out to be best model for this case.

Thank you