INTERNSHIP REPORT

Intership Duration
19-06-2023 to 28-07-2023

DONE AT

CENTRAL POLLUTION CONTROL BOARD

Ministry of Environment, Forest and Climate Change
Government of India

PROJECTS

Multivariate Multi-step PM2.5 Time Series Forecasting & Complaint Category Classification And Location Extraction Using NLP

Submitted by

Harsh Bansal
Enrollment No. A25305221034
B.Tech. Computer Science And Engineering 3rd Year
Amity University Punjab

INTERNSHIP REPORT

Intership Duration
19-06-2023 to 28-07-2023

DONE AT

CENTRAL POLLUTION CONTROL BOARD

Ministry of Environment, Forest and Climate Change

Government of India

PROJECTS

Multivariate Multi-step PM2.5 Time Series Forecasting & Complaint Category Classification And Location Extraction Using NLP

Submitted by:

Harsh Bansal

A25305221034

B.Tech CSE

Amity University Punjab

Submitted to:

Divisional Head

I.T. Division

Central Pollution Control Board

MoEFCC, Govt. of India

Acknowledgement

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of this internship. Their unwavering support, guidance, and encouragement have been invaluable in shaping this endeavor.

I wish to express my special thanks to **Sh. B. Vinod Babu** for granting me this opportunity to work in the I.T. Division of the Central Pollution Control Board.

I would like to express my sincere appreciation to **Sh. Sharandeep Singh** and **Sh. Anurag Sharma**, my dedicated and patient supervisors at the Central Pollution Control Board. Their unwavering support, guidance, and technical expertise have played a pivotal role in enhancing my skills and successfully completing my projects during the internship. I am truly grateful for their mentorship and constant encouragement throughout this enriching experience.

I am immensely grateful to **Smt. Garima Sharma** for giving due consideration to my application and facilitating its approval, without which securing this internship would not have been possible. Her support and belief in my abilities have been instrumental in this opportunity, and I sincerely appreciate her role in making this happen.

I am deeply grateful to the **entire team of I.T. Division, CPCB**, for their warm welcome, cooperation, and willingness to share their knowledge with me during my internship. Their expertise and passion for their work have been a constant source of inspiration throughout this journey.

To all those mentioned above, and to those who may not be named but played a role in this experience, I offer my heartfelt thanks. This report is a reflection of your collective efforts and support, and I am proud to have had the privilege of working with such wonderful people.

Harsh Bansal

Abstract

This internship report presents a comprehensive overview of two major projects undertaken during the internship at the Central Pollution Control Board (CPCB), India. The Central Pollution Control Board, a statutory organization under the Ministry of Environment, Forest and Climate Change, is responsible for gathering air quality data through diverse stations and initiatives such as National Air Quality Monitoring Programme (NAMP), Central Control Room (CCR) for Air Quality Management, and Continuous Ambient Air Quality Monitoring Stations(CAAQMS), spanning the entire country. The first project involved PM2.5 Multivariate Multistep Time Series Forecasting, utilizing time series forecasting models and techniques to predict PM2.5 levels. The second project focused on Natural Language Processing (NLP) combined with Machine Learning to categorize complaints received on the SAMEER App, including the extraction of locations from text using named entity recognition (NER). The report outlines the objectives, dataset descriptions, methodologies, and discussions and results of both projects, highlighting the contribution of these efforts to environmental monitoring and complaint management systems at CPCB.

Table of Contents

List of Figures

List of Tables

Chapter 1

Project - PM2.5 Multivariate Multistep Time Series Forecasting

1.1 Objective

The need for this project stems from the increasing concern about air pollution and its detrimental effects on human health and the environment. PM2.5, which refers to fine particulate matter with a diameter of 2.5 micrometres or less, is a major air pollutant known to have significant health implications, including respiratory issues, cardiovascular problems, and even premature death. Accurate forecasting of PM2.5 levels is crucial for implementing effective mitigation strategies, developing timely interventions, and informing policy decisions aimed at reducing air pollution and its associated health risks.

By analysing the Air Quality data from CPCB and developing a reliable forecasting model, this project aims to provide valuable insights into the temporal patterns and trends of PM2.5 levels. The ability to forecast future PM2.5 levels at an hourly frequency can enable proactive measures to be taken, such as issuing timely health advisories, adjusting emission control strategies, and implementing pollution abatement measures in high-risk areas. Moreover, the availability of an accurate forecasting model can assist in resource allocation and planning, allowing government agencies and environmental organizations to allocate their limited resources effectively.

Furthermore, the scarcity of reference material and research in the domain of multi-variate multi-step time series forecasting specifically for air quality data highlights the importance of this project. By addressing this research gap and developing a model tailored to handle the complexities of the dataset, this project can contribute to the advancement of forecasting techniques in the field of air pollution monitoring. The insights gained from this project can also serve as a foundation for future research and further exploration in the field, aiding in the development of more sophisticated and comprehensive forecasting models for air quality management.

Overall, the objective of this project is to develop an accurate and reliable forecasting model for PM2.5 levels based on the analysis of Air Quality data, which is capable of predicting the next 72 hours of PM2.5 Levels for a particular station / state. Though a dynamic model is expected, lack of sufficient computation resources might be a possible challenge in its development due to increasing complexities in the model. The need for this project arises from the urgency to address the detrimental effects of air pollution, the significance of timely and precise forecasting for informed decision-making, and the opportunity to advance research in the domain of multi-variate multi-step time series forecasting for air quality data.

1.2 Dataset Description

The dataset primarily used for analysis in this project was obtained from the CAAQM (Continuous Ambient Air Quality Monitoring) portal of the Central Pollution Control Board (CPCB). The dataset focused on the air quality measurements at the ANAND VIHAR station. The data spanned from the start date of 01/04/2023 to the end date of 20/06/2023, with a frequency of hourly measurements. The total number of data points in this dataset was 1945.

Initially, the dataset consisted of nine features: PM2.5, PM10, NOx, RH (Relative Humidity), SR (Solar Radiation), WS (Wind Speed), WD (Wind Direction), and merged date. These features were selected based on their relevance to air quality monitoring and the availability of data.

However, later in the project, after conducting initial research and consulting with my manager, a larger dataset was chosen to enhance the scope of the analysis. The extended dataset covered a longer time period, from 24-06-2019 to 24-06-2023, with the same ANAND VIHAR station and hourly frequency. This expanded dataset contained a total of 35,065 data points.

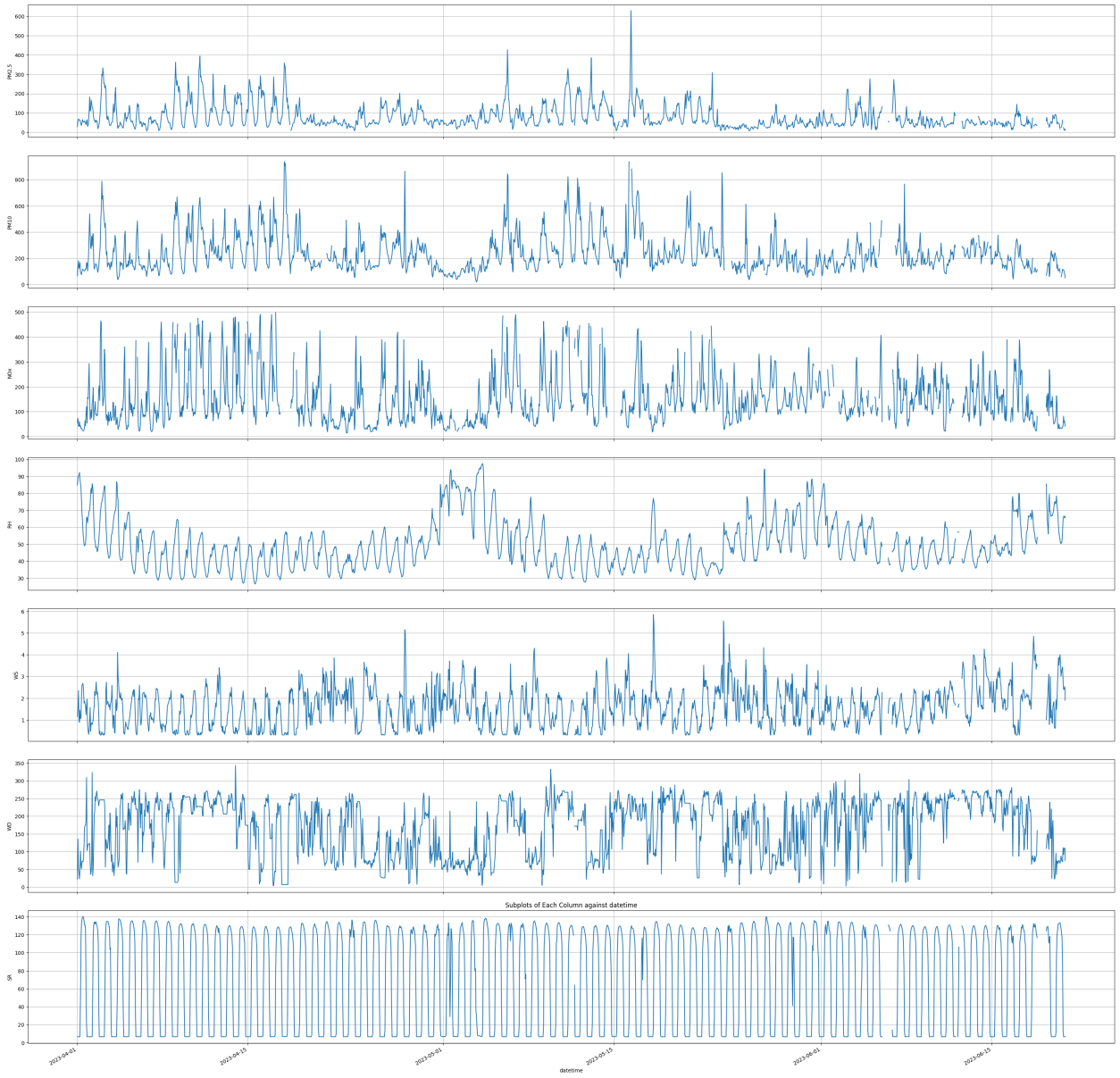


Figure 1.1: Line Plot of Various Columns in Dataset 1

The features in the larger dataset were further refined and processed, resulting in a total of nine features after processing: PM2.5, PM10, NO, NO2, NH3, NOx, SO2, CO, and merged date. The selection of these features was based on their significance in assessing air quality parameters and aligning with the project's objectives.

It is important to note that the chosen datasets focused on a single station, ANAND VIHAR, as this internship aimed to prioritize learning and skill development rather than jumping directly to a complex analysis involving multiple stations. Though it was already made clear, that the final aim was to work on creating models and applications that could actually be used by CPCB at a proper level for public use. The datasets utilized in this project were publicly available through the CAAQM portal of CPCB, ensuring transparency and accessibility of the data for research purposes.

Overall, the datasets employed in this project consisted of hourly air quality measurements from the ANAND VIHAR station. The initial dataset spanned from 01/04/2023 to 20/06/2023, with 1945 data points and initially nine features. Subsequently, a larger dataset covering the period from 24-06-2019 to 24-06-2023 was used, consisting of 35,065 data points and nine refined features. The selection of features was guided by their relevance to air quality assessment, consultation with the manager, and research-based methodologies.

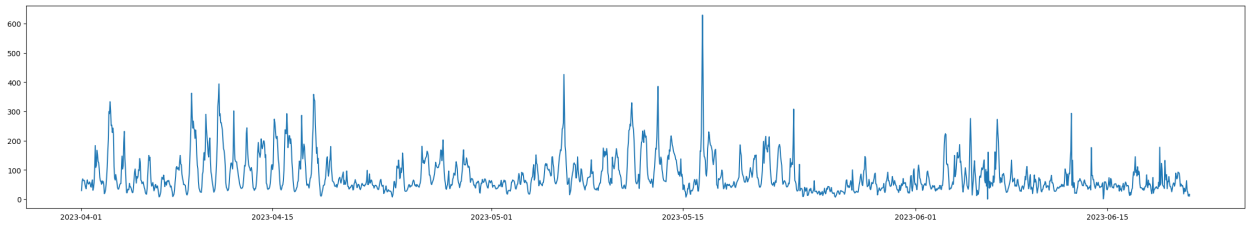


Figure 1.2: Plot of PM2.5 after Imputation

1.3 Methodologies

The task of Machine Learning follows some standard steps, i.e. Choosing a dataset, understanding the data, cleaning the data and applying any other required preprocessing, deciding a model and choosing suitable parameters, and finally evaluating performance and tuning parameters to improve the results.

1.3.1 Dataset 1 (1945 Points) with AR, MA Models

Understanding Data

Therefore, the initial step involved examining the dataset to gather insights. The dataset's limitations, such as start and end dates, were identified, and each column was scrutinized for missing values, outliers, and potential duplicate or overlapping entries. The key findings from the analysis are as follows:

- The dataset contained data for a total of 1945 hours, spanning from 01/04/2023 to 20/06/2023.
- The original dataset had a Date/Time column, which was subsequently divided into separate "From Date" and "To Date" columns.
- With the exception of the date column, every column in the dataset had missing values. Notably, the NOx column had the highest number of missing values, totalling 188 entries. To establish a standardized date format in a single column, the "To Date" column was dropped, and the "From Date" column was converted to the datetime standard for easier interpretation, considering the temporal nature of the data.

Handling Missing Data

Dealing with missing values posed a significant challenge. After reviewing research papers and various other projects, I discovered that imputation and interpolation were commonly used methods to handle missing values, each with its own array of strategies to choose from. In this project, I experimented with multiple strategies, each having its own rationale and potential issues.

- To address the issue of missing values, linear interpolation was performed using B-Fill, F-Fill, and Mean Interpolation. These methods were selected because they are commonly employed as general techniques for fixing missing values in various projects. However, it was observed that the use of straight lines distorted the true time-dependent behavior of the data, as the variability of the data was lost and replaced with constant values.
- K-Nearest Neighbors (KNN) Imputer initially seemed like a promising option, but for some reason, it failed to work on the dataset. Due to insufficient knowledge and understanding, the decision was made to drop the idea and postpone its discussion to another module. This failure was attributed to the consistent absence of values across all columns within a specific time range.
- The approach of using Linear Regression with Random Values was then adopted. Random values were first filled in all columns except for the target column's missing data points. Subsequently, the filled dataset was utilized as a whole to predict the missing values in the target column. This iterative process was performed for each column, resulting in a complete dataset.

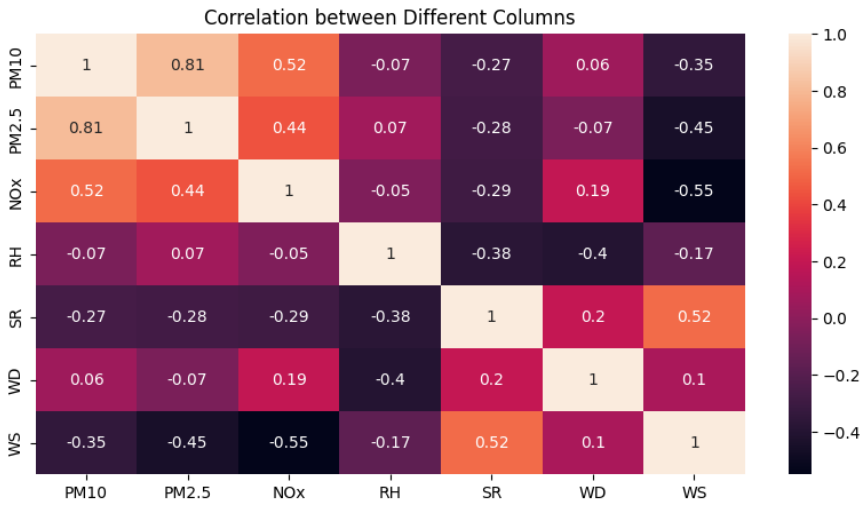


Figure 1.3: Correlation Plot (Dataset 1)

Time Series Analysis

In time series analysis and modeling, correlations, stationarity, and causality tests play a crucial role. Correlation is used to examine the relationships between two different parameters. Autocorrelation helps identify the relationship between different instances of the same parameter, separated by a time difference or lag. Partial autocorrelation at lag k is the correlation observed after removing the influence of correlations at shorter lags. Both the autocorrelation function (ACF) and partial autocorrelation function (PACF) are helpful in determining the orders for autoregressive (AR) and moving average (MA) models.

To identify variables with either high or low correlation, a correlation plot was generated, considering the aim of the analysis. The observation revealed that relative humidity (RH) and wind direction (WD) had very low correlations with PM2.5 (0.07 and -0.07 respectively), while PM10 had the highest correlation of 0.81 on a scale of 0 to 1. Please refer Figure 1.3 for details. ACF and PACF plots were created, and they are included in the report. These plots provide insights into the correlation structure and lag effects present in the data.

Granger causality tests were conducted to determine if one variable can be used for prediction of another variable. The results indicated that all the variables were useful in predicting PM2.5 values, as evidenced by p-values lower than 0.05 in all cases.

Additionally, an Augmented Dickey-Fuller (ADF) test was performed on each column, not just on PM2.5. The purpose of this test is to assess stationarity. It was found that all columns exhibited stationarity, as indicated by p-values below 0.05 for each column.

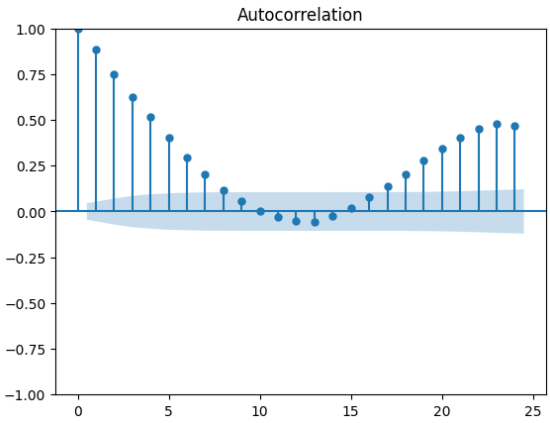
Testing Auto Regressive and Moving Average Models

Although it was not initially part of my project, I became interested in exploring univariate time series modeling, incorporating other variables as exogenous variables in different autoregressive (AR) and moving average (MA) models where exogenous variables were allowed.

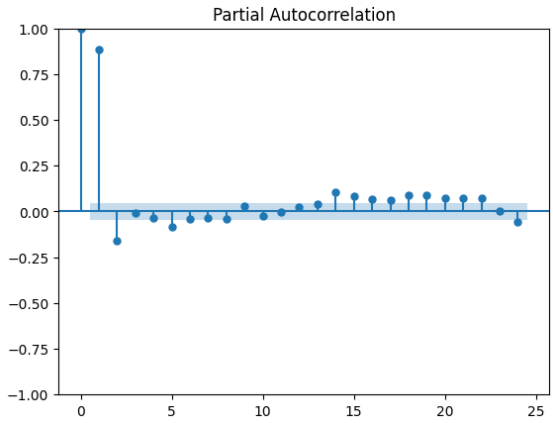
After analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, as well as employing the auto-ARIMA method, we identified two promising sets of p and q values: (2,7) and (9,0).

- First, I tested the ARIMA (2,0,7) (p,d,q) model, which solely relies on the PM2.5 data to make future predictions. Unfortunately, this model performed poorly, resulting in nearly a straight line. Consequently, I decided to discard this model.
- Next, I experimented with the SARIMAX (1,0,1)(2,0,0,24) model, utilizing orders determined by the auto-ARIMA approach. In this model, the exogenous variables included all the parameters except PM2.5.

The results of the SARIMAX model are given in the table 1.1 below.



(a) ACF Plot of PM2.5 on Dataset 1



(b) PACF Plot of PM2.5 on Dataset 1

Figure 1.4: ACF and PACF Plots of PM2.5 on Dataset 1

Table 1.1: SARIMAX (1,0,1)(2,0,0,24) Results

Parameter	**MAE**	**MSE**	**RMSE**
Value	21.086	1103.255	33.215

Compared to the ARIMA model, the SARIMAX model demonstrated greatly improved predictive accuracy. However, it should be noted that the inclusion of exogenous variables introduces a dependency on future predictions, requiring both current data and past PM2.5 values for accurate forecasting.

In summary, incorporating exogenous variables in the SARIMAX model yielded better predictions than the ARIMA model alone.

Testing Vector Autoregrssive (VAR) Models

In multiple time series forecasting, VAR (Vector Autoregression) models are commonly employed. Unlike general autoregressive (AR) or moving average (MA) models, VAR models consider not only the target variable to be predicted but also other variables that influence it. By considering the interdependencies among the variables, VAR models can forecast all the variables in the dataset using values from each other. Although our main focus in this project was to forecast PM2.5, utilizing VAR was essential as we wanted to capture the impact of other parameters as well.

For VARMAX modelling, the endogenous variables included PM2.5, PM10, and NOx, while the exogenous variables were WD (Wind Direction) and WS (Wind Speed). It's worth noting that exogenous variables are necessary for future predictions in VARMAX models.

Results for VARMAX (9,0) and VARMAX(2,7) were as follows:

Table 1.2: VARMAX Evaluation Results

	MAE	**MSE**	**RMSE**
VARMAX (9,0)	34.29	2025.72	45.00
VARMAX (2,7)	35.07	2056.53	45.34

However, these models had two drawbacks. Firstly, they provided unnecessary predictions for other data that were not required for our target variable (PM2.5). Secondly, exogenous variables were required for multi-step future predictions. To overcome these limitations, we decided to explore Stacked LSTM (Long Short-Term Memory) models.

1.3.2 Large Dataset (35025 Points) with Stacked LSTM

Due to limited knowledge about Neural Networks, the approach taken in this part involved trial and error strategies, along with references from various papers and online tutorials.

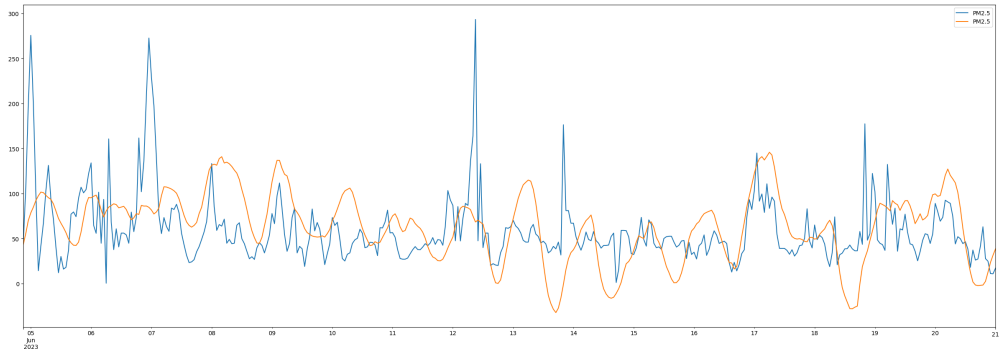


Figure 1.5: VARMAX (9,0) Test Dataset vs Prediction Plot

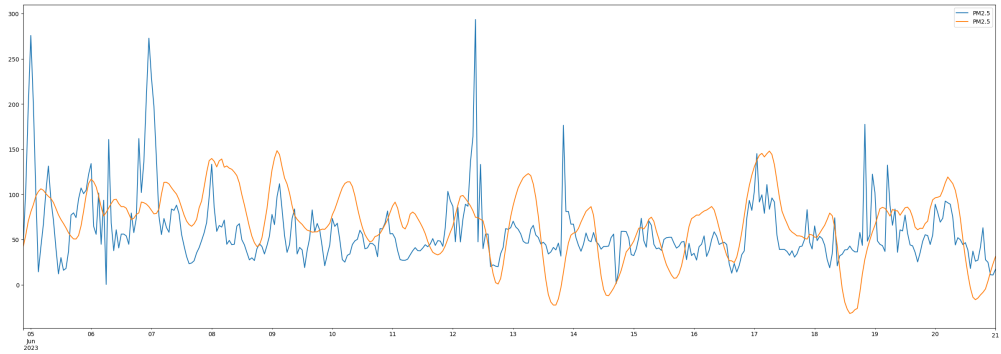


Figure 1.6: VARMAX (2,7) Test Dataset vs Prediction Plot

Dataset Description

To leverage the power of Neural Networks and prevent overfitting, it was crucial to increase the dataset. Additionally, some new features were introduced in comparison to the previous dataset. The current dataset comprised Air Quality Data from 24-06-2019 to 24-06-2023 for the Anand Vihar Station in New Delhi. The new features included PM2.5, PM10, NO, NO2, NH3, NOX, SO2, and CO, along with two date-time columns. The date-time columns were merged into a single column. As a result, the new dataset contained a total of 35,065 entries, representing hourly Air Quality data recorded at Anand Vihar Station.

Outlier Detection and Removal

Outliers, which are observations that deviate significantly from the rest of the dataset, can have a significant impact on the analysis results. In this project, the interquartile range (IQR) was utilized to identify outliers.

The IQR is calculated as the difference between the 25th percentile (Q1) and the 75th percentile (Q3) of a dataset, representing the spread of the middle 50% of values. A commonly employed method for outlier detection is to consider observations as outliers if their values are 1.5 times greater than the IQR or 1.5 times less than the IQR.

Applying this approach to the current dataset, outliers were identified and subsequently removed from all columns. Specifically, focusing on the PM2.5 column, the range of values changed from (1.0 - 978.0) to (1.0 - 336.75), indicating the removal of extreme values. Furthermore, the mean of the PM2.5 column shifted from 123.99 to 103.79 after the removal of outliers.

By detecting and addressing outliers, the dataset was refined, leading to a more representative and reliable analysis.

Handling Missing Values using KNN Imputer

Building on the lessons learned from processing the previous dataset, a different approach was adopted to handle missing values in this project. KNN Imputer was employed, using a K value of 535.

KNN Imputer is a method that leverages the concept of K-nearest neighbors to estimate missing values based on the values of the nearest neighbors. By considering the characteristics of similar instances, KNN Imputer provides a robust strategy for imputing missing values.

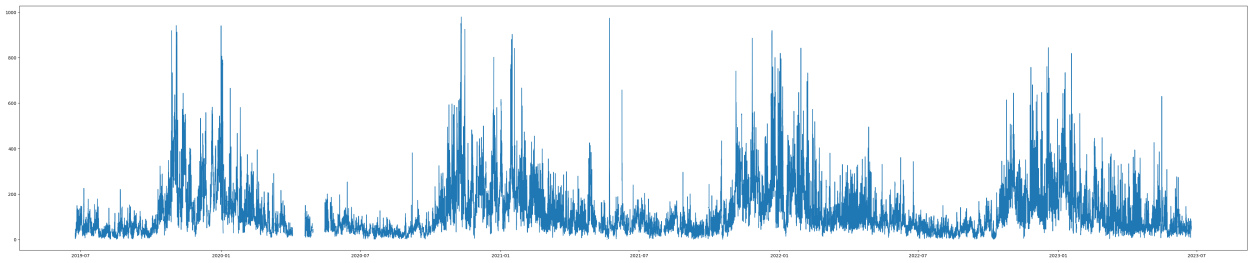


Figure 1.7: Line Plot of PM2.5 Data in Dataset 2 (35025 Entries)

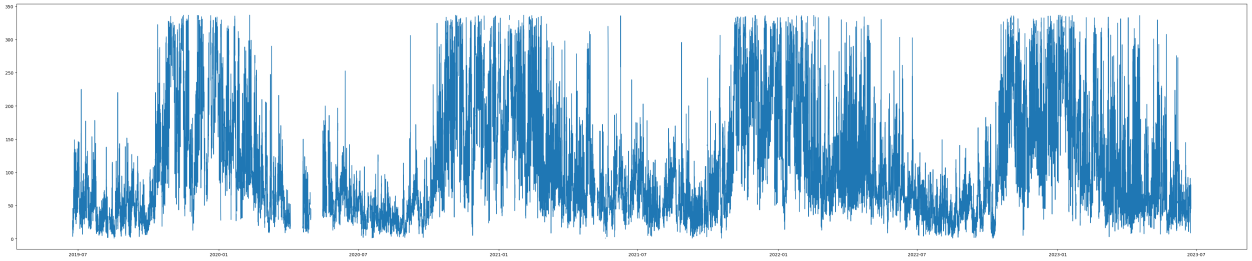


Figure 1.8: After Outlier Removal using IQR - Line Plot of PM2.5 Data in Dataset 2

In this project, the K value of 10 was chosen based on experimentation and analysis. By incorporating a larger number of neighbors, the imputation process benefited from a more extensive range of information and observations.

Using KNN Imputer helped address the missing values in the dataset, enhancing the completeness and integrity of the data for subsequent analysis and modeling.

Correlation Analysis

A correlation heatmap was generated using the interpolated dataset. Once again, PM10 exhibited the highest correlation (0.73) with PM2.5, while NH3 showed the least correlation (0.14) with PM2.5.

Model Architecture

For this project, a Stacked LSTM model was chosen as it has been widely used in time series forecasting projects. The model consisted of three LSTM layers with 128, 64 and 32 units respectively, each of first two followed by a dropout layer with a rate of 0.2. Finally, a dense layer was added. The model incorporated 24 timesteps of the entire dataset, including PM2.5, to generate a 12-step forecast for PM2.5.

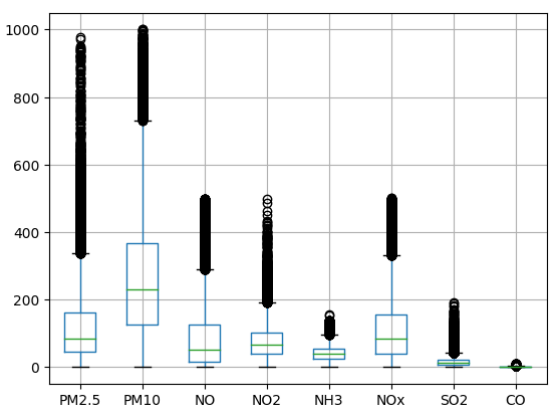
Data Splitting and Model Training Configuration

The dataset was divided into 70% for training, 15% for validation, and 15% for testing. The data was not shuffled to maintain the order of the data. The ADAM optimizer with a default learning rate of 0.0005 was utilized. The batch size was set to 8, and the model was trained for 20 epochs. The best model (least loss) was used for the evaluation in further steps.

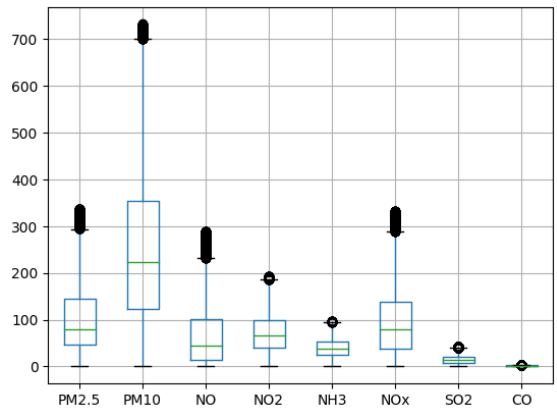
Model Evaluation

The final evaluation of the models was conducted using two different approaches:

- **Direct Evaluation:** The first approach involved directly flattening the predicted values and applying relevant metrics to assess the performance of the models. Various evaluation metrics, such as mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE), were used to quantify the accuracy and precision of the predictions.
- **Overlapping Evaluation:** The second approach involved considering the overlapping values between the predicted timesteps. Specifically, each predicted timestep (t) was compared with the previous timestep (t-1) of the subsequent prediction, as well as with the timestep t-2 of the prediction after that, and so on until 12 timesteps (adjusted for cases with fewer than 12 timesteps). This approach allowed for a comprehensive assessment of the model's performance over consecutive overlapping intervals.



(a) Boxplot of Original Dataset



(b) Boxplot After Outlier Removal

Figure 1.9: Boxplot Comparison before and after outlier removal for Dataset 2

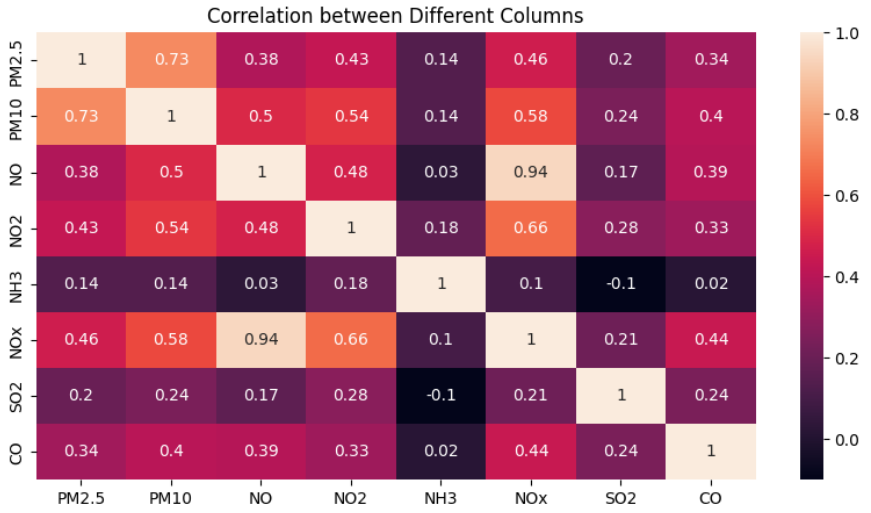


Figure 1.10: Correlation Plot (Dataset 2)

By employing these two evaluation methods, a comprehensive understanding of the models' predictive capabilities was achieved. These evaluation techniques facilitated the comparison of different models and provided valuable insights into their strengths and weaknesses in forecasting PM2.5 values.

The results can be observed in Table 1.3 given below.

Table 1.3: Evaluation of Results for Stacked LSTM Model on Dataset 2

Metric	Original	Mean of Overlapping Values
MAE	37.504512	31.645958
MSE	2641.341788	1986.839607
RMSE	51.393986	44.573979
R-squared	0.566594	0.673976
IOA	0.847579	0.890032
MB	-1.559531	-1.799331

1.4 Results and Discussion

The project encompassed a comprehensive analysis of time series data, focusing on predicting PM2.5 values. Various approaches were employed, starting with dataset analysis to identify limits and investigate missing values, outliers, and duplicate columns. Techniques such as linear interpolation and imputation methods like K-Nearest Neighbors and Linear Regression with Random Values were explored to handle missing values. The choice of handling missing values had a significant impact on the subsequent analyses and modeling.

Correlation analysis played a crucial role in understanding the relationships between variables. Correlation heatmaps highlighted the varying degrees of association between PM2.5 and other parameters. PM10 consistently exhibited the highest correlation with PM2.5, emphasizing its influence on PM2.5 levels. These findings informed subsequent modeling decisions, ensuring the consideration of relevant variables in forecasting.

The project delved into traditional time series models such as autoregressive (AR), moving average (MA), and vector autoregression (VAR). The inclusion of exogenous variables in VAR models allowed for a more comprehensive understanding of the factors influencing PM2.5. While the VAR models provided valuable insights and predictions, they had limitations such as unnecessary predictions and a dependency on exogenous variables for future forecasts. To overcome these limitations, the project explored neural networks and employed a stacked LSTM model. This model demonstrated promising results in forecasting PM2.5 values by incorporating the entire dataset and leveraging the temporal dependencies captured by LSTM layers. The LSTM model allowed for more accurate predictions, considering the complex dynamics and interdependencies within the time series data. The comparitive analysis of the performance of all the various models employed can be observed in Table 1.4 and Table 1.3.

Table 1.4: Performance Evaluation of Models Trained on Dataset 1

Dataset 1 — Anand Vihar ***01/04/2023 to 20/06/2023 — Hourly*** ***PM10, PM2.5, NOx, RH, SR, WS, WD***				
Model	**Hyperparameters**	**MAE**	**MSE**	**RMSE**
SARIMAX	(1,0,1)(2,0,0,24)	21.086	1103.255	33.215
VARMAX	(9,0)	34.29	2025.72	45.00
VARMAX	(2,7)	35.07	2056.53	45.34

Overall, this project showcased the iterative and exploratory nature of time series analysis and modeling. It underscored the importance of preprocessing, handling missing values, understanding correlations, and selecting appropriate models. The results obtained from different approaches provided valuable insights into PM2.5 prediction, highlighting the significance of variables such as PM10, NOx, and exogenous factors like wind direction and speed. Further refinements and research can be pursued to optimize the modeling techniques and explore additional features that may improve PM2.5 forecasting accuracy in real-world applications.

1.5 Conclusion

In conclusion, this thread discussed various aspects of a data analysis and modeling project, focusing on time series data. The initial steps involved dataset analysis, identifying data limits, and examining missing values, outliers, and duplicate columns. Correlations, stationarity, and causality tests were performed to gain insights into the data.

Different techniques were explored to handle missing values, including linear interpolation and imputation methods such as K-Nearest Neighbors (KNN) and Linear Regression with Random Values. Each method was evaluated based on its impact on the data and correlations between variables.

The project then delved into time series modeling, considering autoregressive (AR), moving average (MA), and vector autoregression (VAR) models. The use of exogenous variables in VAR models was particularly highlighted. While the models showed varying levels of success in predicting PM2.5 values, certain limitations were identified, such as unnecessary predictions and dependence on exogenous variables for future forecasts.

To address these limitations, the project ventured into the realm of neural networks and employed a stacked LSTM (Long Short-Term Memory) model. The dataset was expanded to include additional features, and linear interpolation was used to handle missing values. Correlation analysis revealed the relationships between variables.The LSTM model architecture included multiple LSTM layers, dropout layers for regularization, and a dense layer for prediction. The model demonstrated promising results in forecasting PM2.5 values. The dataset was split into training, validation, and testing sets, and the model was trained using the ADAM optimizer for a specified number of epochs.

In summary, this project showcased the journey of analyzing and modeling time series data, exploring different techniques and models to understand and predict PM2.5 values. It emphasized the importance of data preprocessing, handling missing values, understanding correlations, and selecting appropriate models for accurate predictions. Further research and fine-tuning can be pursued to enhance the forecasting capabilities and practical applications of the developed models.

Chapter 2

Project - Complaint Categorization and Location Extraction

2.1 Objective

In the modern world, organizations and businesses rely on customer feedback to gain insights into the functioning of their various modules. Customer complaints serve as a valuable source of feedback and establish a direct connection between the organization and its users, providing a means for addressing their concerns. The Central Pollution Control Board has developed an app called Sameer, which offers hourly updates on the National Air Quality Index (AQI) published by the CPCB. The Air Quality Index provides a simplified representation of air quality through a single number, nomenclature, and color. The app allows the public to submit complaints, accompanied by pictures, and offer valuable suggestions. It is available on both the Google Play Store and Apple's App Store.

However, a significant challenge arises in manually categorizing the complaints into specific categories chosen by the users. These categories, such as Leaf Burning, Construction or Demolition Activity, etc., are used to classify the complaints and assign tasks to various departments. Unfortunately, users sometimes struggle to accurately categorize their complaints or become confused by multiple complaint categories. Additionally, while the app captures the complaint's location during submission, there is a desire to explore the possibility of extracting location information from the complaint description itself. The goal is to convert these locations into coordinates and plot them on a map, enabling the visual identification of hotspots based on a combination of complaint categories and locations.

To address these issues, the project is divided into two phases. The first phase focuses on developing a Classification Model capable of categorizing the data into different complaint categories. The second phase involves developing a Named Entity Recognition (NER) model to extract location data from the complaint descriptions and determine possible coordinates. This task presents a challenge due to the presence of Code-Mixed Complaint Descriptions, where two languages are mixed together. The dataset contains complaints written in various languages, necessitating data standardization.

The overall objective of this project is to improve the complaint management system by automating the categorization process and extracting location information accurately. By achieving these goals, we aim to enhance the efficiency of addressing customer complaints and identifying pollution hotspots for targeted interventions.

2.2 Dataset Description

For this project, we utilized a proprietary dataset containing a diverse range of complaints. The dataset was sourced directly from the organization and is not publicly available. The complaints within the dataset were collected during the year 2020. The dataset consisted of two subsets: one with 2,517 complaints and another with 6,986 complaints.

Each complaint in the dataset was associated with various fields, including:

- Complaint ID: A unique identifier for each complaint.
- Description: The textual content describing the complaint.
- Category: The assigned category for the complaint, indicating the nature of the issue.
- Assigned By: Information about the individual or entity responsible for assigning the complaint.
- Assigned To: Details of the person or department assigned to address the complaint.

- State: The state in which the complaint was lodged.
- Lodged On: The date and time when the complaint was lodged.

In our analysis, we focused primarily on the Description and Category fields. The additional fields, such as Assigned By, Assigned To, State, and Lodged On, were not relevant to our specific objectives and were therefore discarded during the preprocessing stage.

To build our models, we initially used the dataset containing 6,986 complaints. This dataset served as the foundation for model development and evaluation. Additionally, we reserved the smaller dataset consisting of 2,517 complaints for further testing and validation. The intent was to iteratively improve the models by incorporating a larger dataset in future iterations.

It's important to note that the dataset contains complaints in various languages and includes instances of Code-Mixed Complaint Descriptions, where multiple languages are mixed within a single complaint. Standardizing the data and addressing the language variation presented an additional challenge that needed to be tackled during the project.

2.3 Methodology

The tasks of developing a classification model for complaint categorization and a Named Entity Recognition (NER) model for extracting location data involved several common steps. These steps included standardizing the data to a single language, performing data cleaning, and applying general Natural Language Processing (NLP) preprocessing techniques. The following subsections provide a detailed discussion of these steps. Note that these steps were common for both the sub-datasets.

2.3.1 Dataset Cleaning

During the manual inspection of the dataset, two distinct types of unnecessary data were identified in some complaints: irrelevant complaints and low-information complaints. These were addressed through a cleaning process, resulting in two separate lists: one for cleaned complaints and another for actions taken. The specific actions taken are described below:

Irrelevant Complaints

Some complaints were deemed irrelevant due to their content. The following types of complaints were identified and removed from the dataset:

- Complaints of length 10 with just a phone number.
- Empty complaints or complaints containing only special characters like dots (.), commas (,), or spaces (' ').

The cleaned complaints list contained complaints that were considered relevant and contained useful information for further analysis.

Low-information Complaints

Additionally, complaints that provided minimal usable information due to their very short length were also identified. These complaints were unlikely to contribute meaningfully to the analysis. Therefore, complaints with less than 4 words were excluded from the dataset thus filtering out low-information complaints.

By applying these cleaning actions, we aimed to ensure that the dataset primarily consisted of relevant complaints with meaningful content, enhancing the effectiveness of subsequent processing and modeling steps. As a result, the dataset 1 now had 2377 entries, and dataset 2 and 6605 entries in dataset 2.

Duplicate Complaints

During the manual observation of the dataset, it was noticed that certain complaints were duplicated, where the same complaint was submitted multiple times in a row. These duplicate complaints offered no additional value and had the potential to introduce bias during the model training process. To mitigate this issue, the duplicate complaints were identified and treated as follows:

- Removal of duplicate complaints: Multiple copies of the same complaint were identified and removed from the dataset. This step aimed to eliminate redundancy and ensure that each unique complaint had a fair representation in the dataset.

By addressing the presence of duplicate complaints, we reduced the overall complaint count to 2,098 for Dataset 1 and 6,142 for Dataset 2. This reduction in the number of complaints helped to mitigate potential bias and ensure a more balanced representation of complaint categories during model training.

2.3.2 Language Detection

Processing datasets with multiple languages requires careful consideration. In this study, the decision was made to standardize the data into one language. However, it was essential to analyze the languages present in the complaints and classify each complaint accordingly for easier translation.

To detect the languages present in the complaints, a combination of Spacy and Langid libraries was utilized. Spacy, a Python library, provides language detection pipelines that were employed in this analysis. The entire complaint description was fed into the pipeline, and the resulting language was recorded. It was observed that the majority of complaints were in Hindi, English, Marathi, and Hinglish, with very few instances of other languages.

Initially, the direct approach of language detection using Spacy resulted in suboptimal performance due to the presence of a diverse range of comments, many of which contained a mixture of Devanagari script and English within the same comment. This type of language mixing differs from code-mixed languages and caused the model to struggle in identifying the correct language. Additionally, smaller complaints posed challenges for accurate language detection.

To address these issues, an alternative approach was adopted. First, the language of the entire complaint was identified using Spacy, as discussed above. If the result was categorized as "Other," indicating that the complaint was not in Hindi, English, or Marathi, the complaint was split into individual words. Language detection was then performed on each word using the Langid.py library. The number of words falling into each language category was tracked, and if the number of words belonging to one of the above languages exceeded half of the total complaint length, the complaint was classified into that language category. Otherwise, it was categorized as "Other".

An interesting observation during the language detection process was the similarity between Hinglish comments and two languages, Malay and Swahili. It was noticed that some complaints identified as Malay or Swahili were actually Hinglish complaints. To address this, a modification was made to the language detection approach.

In the modified approach, complaints initially classified as Malay or Swahili were directly categorized as Hinglish (Other) complaints. This adjustment resulted in significantly improved accuracy for identifying Hinglish complaints. By incorporating this modification into the overall language detection process, we achieved more precise classification of complaints in terms of language.

This approach yielded more accurate classification of the complaints in terms of language. It was thoroughly checked manually multiple times to ensure its robustness. The use of two different language detection models, Spacy and Langid, contributed to improved language detection accuracy. The final distribution of complaints across languages is summarized in Table 2.1.

Table 2.1: Count of Data-points for each Language

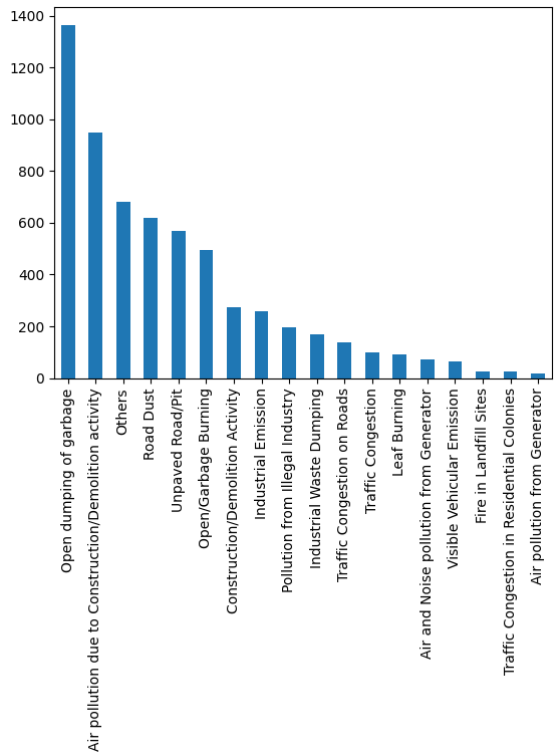
Language	Dataset 1	Dataset 2
English	1965	5810
Hindi	78	204
Other	31	127
Marathi	24	1

Through this language detection and classification process, we achieved a more accurate understanding of the language composition within the dataset, which was crucial for subsequent translation tasks.

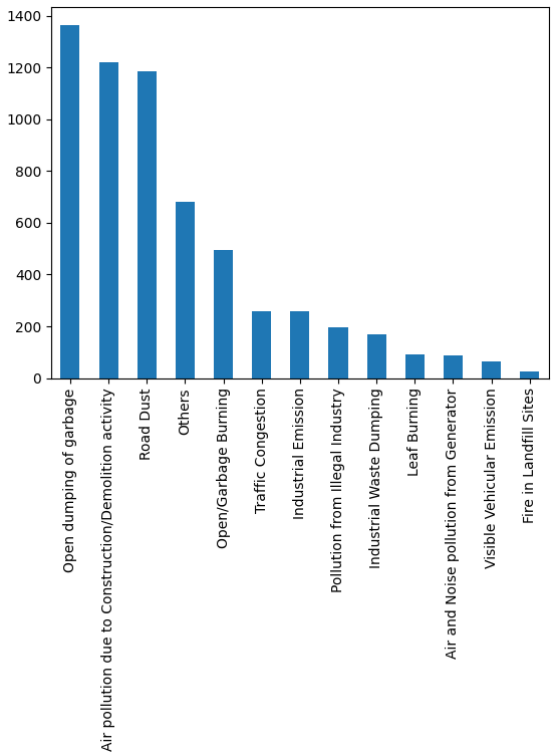
2.3.3 Translation and Standardization of Complaints

Once the dataset was labeled with complaint categories and languages, it was necessary to standardize the complaints into a single language for improved classification. Based on the language distribution analysis (as discussed in previous Section 2.3.2), English emerged as the majority language. Therefore, it was decided to standardize all Hinglish, Hindi, and Marathi complaints into English.

Various attempts were made to standardize the complaints before settling on the final approach. These attempts included transliteration, using pre-trained translation models from libraries like HuggingFace, and leveraging free translation libraries available for Python. However, these attempts yielded poor results. Transliterations were not accurate, and free libraries often failed to preserve the intended meaning of the complaints. Thus, it was necessary to explore alternative options, including paid translation APIs.



(a) Number of Datapoints vs Labels - Original



(b) Number of Datapoints vs Labels - After Merging

Figure 2.1: Bar Plots showing number of datapoints available for each label in Dataset 2 - before and after merging.

For the translation of non-English comments, the Azure Translator API was employed. Microsoft's Azure Translator API proved to be a robust and suitable tool for handling diverse translation tasks, including code-mixed complaints. Although a few Hinglish complaints were tested using Google Translate, it failed to provide accurate translations compared to Microsoft Translate, further solidifying the decision to utilize the Azure Translator API.

The language identification performed earlier proved beneficial at this stage for two reasons:

- Specifying the "FROM" language in the Azure API request significantly improved translation accuracy. Without specifying the source language, the API's auto-detection sometimes resulted in incorrect translations.
- The Microsoft Azure Translator API offers 2 million characters in translation per month within its free-tier account. To stay within the free-tier limits and avoid additional costs, only the non-English complaints were translated, ensuring efficient resource usage.

During the translation process, the FROM language was set to "hi" (Hindi) for Hindi and Hinglish comments, and "mr" (Marathi) for Marathi comments.

The final translations were manually evaluated to ensure their accuracy, and the results were satisfactory. The translated complaints, along with their respective categories, were stored in a separate dataset for subsequent steps, such as category classification and location extraction. By standardizing the complaints into a single language, we ensured consistency in the dataset, enabling more accurate analysis and modeling tasks.

2.3.4 Complaint Category Analysis

For the task of complaint categorization, the larger dataset consisting of 6,142 complaints was utilized to train the model (mentioned in Section 2.2). The smaller dataset containing 2,098 complaints was reserved for additional testing and validation purposes at a later stage. This division ensured that the model was trained on a substantial amount of data while allowing for independent evaluation on unseen complaints. By utilizing both datasets, we aimed to enhance the robustness and generalizability of the complaint categorization model.

Category Data Analysis

The analysis of the data revealed the following information regarding complaint categorization:

- There were a total of 18 categories, including 'Air and Noise pollution from Generator,' 'Air pollution due to Construction/Demolition activity,' 'Air pollution from Generator,' 'Construction/Demolition Activity,' 'Fire in Landfill Sites,' 'Industrial Emission,' 'Industrial Waste Dumping,' 'Leaf Burning,' 'Open dumping of garbage,' 'Open/Garbage Burning,' 'Others,' 'Pollution from Illegal Industry,' 'Road Dust,' 'Traffic Congestion,' 'Traffic Congestion in Residential Colonies,' 'Traffic Congestion on Roads,' 'Unpaved Road/Pit,' and 'Visible Vehicular Emission.'
- The dataset exhibited class imbalance, as evident from the distribution shown in Table 2.2.
- Some categories displayed overlap or ambiguity, including:
  - 'Pollution from Illegal Industry' overlapped with 'Industrial Emission.'
  - 'Unpaved Road/Pit' was related to 'Road Dust.'
  - 'Traffic Congestion on Roads' and 'Traffic Congestion in Residential Colonies' were variations of the broader category 'Traffic Congestion.'
  - 'Air pollution from Generator' fell under 'Air and Noise pollution from Generator.'
  - 'Construction/Demolition Activity' was associated with 'Air pollution due to Construction/Demolition activity.'

Table 2.2: Category Wise Availability of Data

Data Available for Each Category		
Category	**Dataset 1**	**Dataset 2**
Road Dust	313	618
Open/Garbage Burning	299	496
Industrial Emission	279	258
Others	236	682
Unpaved Road/Pit	235	567
Open dumping of garbage	223	1366
Air pollution due to Construction/Demolition activity	206	948
Pollution from Illegal Industry	138	195
Leaf Burning	60	90
Air and Noise pollution from Generator	40	72
Construction/Demolition Activity	38	272
Industrial Waste Dumping	36	169
Visible Vehicular Emission	23	63
Fire in Landfill Sites	16	24
Traffic Congestion	12	98
Air pollution from Generator	12	17
Traffic Congestion on Roads	8	138
Traffic Congestion in Residential Colonies	4	24

These observations provided valuable insights into the structure and characteristics of the complaint categories. Understanding the category distribution and identifying potential overlap or ambiguity aided in developing a more accurate classification model.

2.3.5 Model Creation for Complaint Categorization Task

Text Preprocessing

Text preprocessing is a crucial step in natural language processing tasks to clean and prepare the text data for further analysis. The following steps were performed as part of the text preprocessing phase:

- Convert text to lowercase: The text was converted to lowercase to ensure consistency and avoid treating the same words with different cases as different entities.

- Remove punctuation: Punctuation marks were removed from the text using Python's built-in string.punctuation module.
- Tokenization: The text was tokenized into individual words using the word_tokenize function from the nltk library.
- Remove stop words: Stop words are common words that do not carry significant meaning and can be safely removed from the text. Stop words were eliminated from the tokens using the set of stopwords provided by the nltk library for the English language.
- Remove numbers and non-English words: Tokens that consisted solely of numbers or non-English characters were discarded from the text. Regular expressions were used to match and filter out such tokens.
- Lemmatization: Lemmatization is the process of reducing words to their base or root form. The WordNetLemmatizer from the nltk library was employed to lemmatize the tokens, reducing them to their canonical forms.
- Join tokens: The preprocessed tokens were then joined back into a single string, creating the final processed text representation.

The text preprocessing steps outlined above aimed to standardize the text data, remove noise, and retain relevant information for complaint categorization. By applying these preprocessing techniques, we ensured that the input text was in a clean and consistent format, ready for further analysis and feature extraction.

TF-IDF Vectorization

Vector semantics is a technique used for word and sequence analysis that aims to define the semantic meaning of words and interpret their features, such as similarity and oppositeness. It achieves this by representing words in a multi-dimensional vector space, where words with similar contextual usage are closer to each other.

One commonly used method for vectorizing text is TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF is an algorithm that transforms text into a meaningful numerical representation, which can then be used to train machine learning models for prediction tasks. The TfidfVectorizer class is an implementation of TF-IDF vectorization, which converts a collection of raw documents into a matrix of TF-IDF features. Each document is represented as a set of words, and the number of times each word appears in the collection is used to compute its TF-IDF score.

In the context of complaint categorization, TF-IDF vectorization was employed to represent the complaints as numerical feature vectors. This process involved converting the raw text data into a structured format that could be utilized by machine learning algorithms. By leveraging TF-IDF vectorization, the model could capture the importance of specific terms within each complaint and use them as features for classification.

The TF-IDF vectorization step played a crucial role in preparing the complaint data for further analysis and classification tasks. It transformed the raw text into a numerical representation that could be fed into the machine learning model for training and prediction.

Train-Test Split

The preprocessed data, obtained after TF-IDF vectorization, was divided into a train-test split with an 80/20 ratio. This means that 80% of the data was allocated for training the models, while the remaining 20% was reserved for testing the model's performance. To ensure a representative distribution of samples across the training and testing datasets, stratification was applied. This approach aimed to maintain the proportion of each complaint category in both the training and testing datasets, helping to avoid scenarios where certain categories have no samples in the testing set.

Model Testing

In the initial phase of model testing, the Support Vector Machine (SVM) algorithm was employed. The preprocessed dataset was fitted into the SVM model using default parameters. The performance of the model was evaluated using metrics such as accuracy, precision, recall, and F1 score. The results of the SVM model are presented in Table 2.3.

During this testing phase, the SVM model's performance was assessed to gain insights into its effectiveness in categorizing the complaints. These metrics provided an overall assessment of the model's accuracy and its ability to correctly classify complaints into their respective categories.

Table 2.3: Evaluation of SVM on Dataset with 18 Categories (Before Merging)

Metric	***Value***
Accuracy	0.643
Precision	0.667
Recall	0.643
F1 Score	0.625

Further, after analysing the poor performance of the model through the use of confusion matrix (Figure 2.2), it was pretty evident that the inferences (as mentioned in Section 2.3.4) about category overlapping were valid. Thus, those categories were merged into each other, to reduce the category count from 18 to 13. The count of entries available for each category after merging is available in Table 2.4.

Table 2.4: Category Wise Availability of Data after Merging

Data Available for Each Category (After Merging)	
Category	**Data Points**
Road Dust	1185
Open/Garbage Burning	496
Industrial Emission	258
Others	682
Open dumping of garbage	1366
Air pollution due to Construction/Demolition activity	1220
Pollution from Illegal Industry	195
Leaf Burning	90
Air and Noise pollution from Generator	89
Industrial Waste Dumping	169
Visible Vehicular Emission	63
Fire in Landfill Sites	24
Traffic Congestion	260

In the next phase, four different algorithms were employed to train and test the models on the merged dataset. The algorithms used were Support Vector Machine (SVM), Logistic Regression (LR), Random Forest Classifier (RFC), and Multinomial Naive Bayes (M-NB). Each model was trained individually using the same TF-IDF vectorizer and the training dataset. Stratification was applied during the splitting process to ensure a balanced representation of complaint categories in the training and testing set.

The performance of each model was evaluated using key metrics such as accuracy, recall, F1 score, and precision. These metrics provided insights into how well each model performed in categorizing the complaints. The results of the model testing phase, including the accuracy, recall, F1 score, and precision for each algorithm, are presented in Table 2.5.

Table 2.5: Evaluation of Models - Before and After Merging

Model	***Accuracy***	***Precision***	***Recall***	***F1 Score***
SVM (Before Merging)	0.643	0.667	0.643	0.625
SVM	0.736	0.743	0.736	0.730
LR	0.723	0.726	0.723	0.718
M-NB	0.555	0.575	0.555	0.479
RFC	0.680	0.680	0.680	0.674

Upsampling using SMOTE

The analysis of Tables 2.2 and 2.4 revealed that the dataset suffered from class imbalance, as mentioned in Section 2.3.4. To address this issue, an upsampling technique called SMOTE (synthetic minority oversampling technique) was employed.

SMOTE is a popular oversampling method used to tackle imbalanced datasets. It aims to balance the class distribution by creating synthetic minority class examples. The technique generates new minority instances by interpolating between existing minority instances. This oversampling process involves randomly selecting one or more of the k-nearest neighbors for each example in the minority class and creating synthetic training records. By replicating and synthesizing new instances, SMOTE helps to balance the dataset and improve the representation of the minority class.

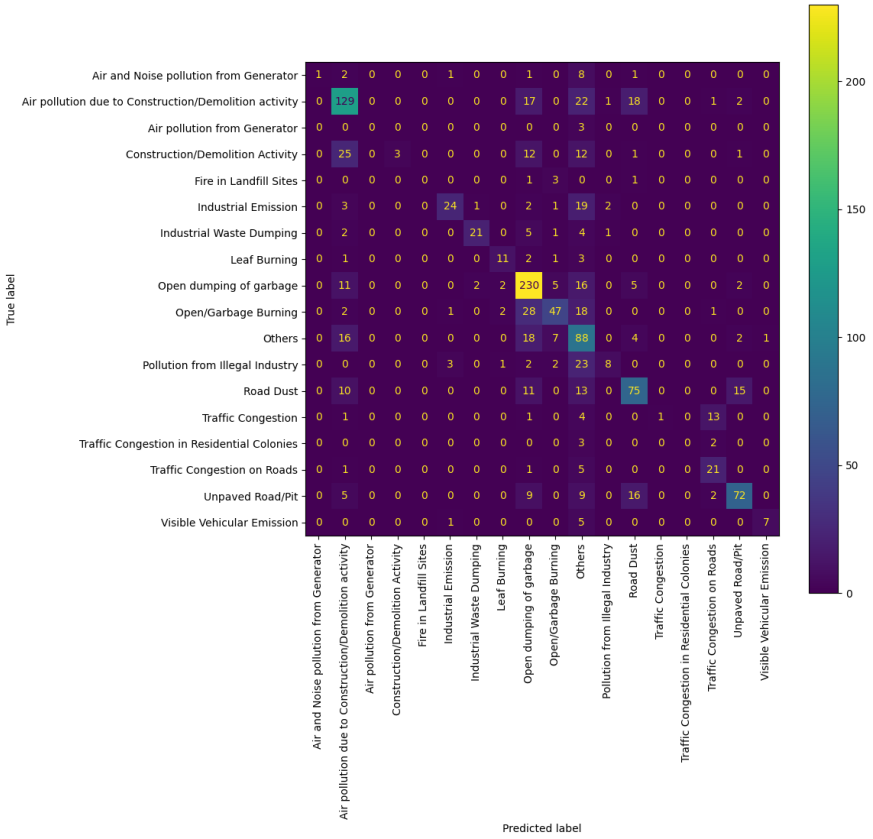


Figure 2.2: Confusion Matrix - SVM before Merging Ambiguous / Related Labels

Model Evaluation on Upsampled Data

To apply SMOTE, a custom sampling strategy was implemented. The mean number of observations across all categories was calculated and rounded to 470. A k value of 5 was chosen for the k-neighbors parameter. According to the sampling strategy, any labels with less than 470 data points were upsampled to 470 using SMOTE. It's important to note that SMOTE was only applied to the training dataset, not the testing dataset.

After upsampling the dataset, the two top-performing models from the previous experiments, Logistic Regression (LR) and Support Vector Machine (SVM), were retrained using this new upsampled dataset. The evaluation results of these models can be observed in Table 2.6. This evaluation aimed to assess the impact of upsampling on the performance of the LR and SVM models, considering the improved balance in the training data.

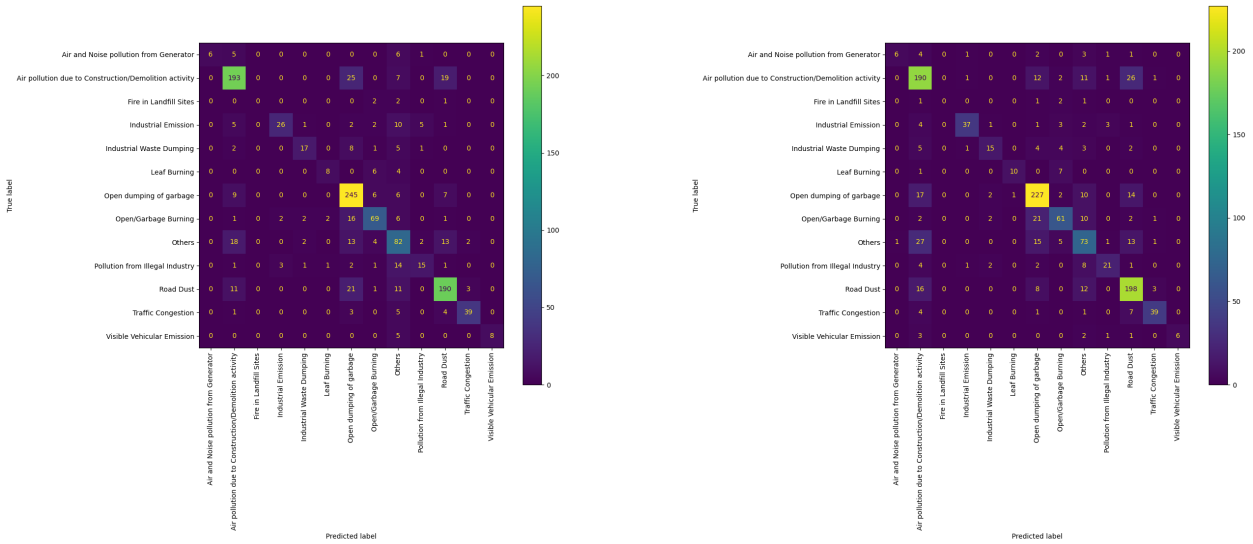
Table 2.6: Evaluation of All Models developed in this Project

Model	***Status of Dataset***	***Accuracy***	***Precision***	***Recall***	***F1 Score***
SVM	Original Dataset	0.643	0.667	0.643	0.625
SVM	Merged	0.736	0.743	0.736	0.730
LR	Merged	0.723	0.726	0.723	0.718
M-NB	Merged	0.555	0.575	0.555	0.479
RFC	Merged	0.680	0.680	0.680	0.674
SVM	Merged and Upsampled	0.757	0.760	0.757	0.751
LR	Merged and Upsampled	0.766	0.763	0.766	0.762

2.3.6 Extraction of Location-related Keywords from Text

The second part of the project focused on extracting location details solely from the complaint descriptions. Due to resource constraints, the smaller dataset consisting of 2,098 entries was utilized for this task.

Named Entity Recognition (NER) is a widely used technique in data preprocessing that involves identifying key information in text and classifying it into predefined categories. NER enables the identification of



(a) Confusion Matrix - SVM after Merging

(b) Confusion Matrix - LR after Merging

Figure 2.3: Comparison of Confusion Matrices on Dataset with Ambiguous Labels Merged

entities consistently mentioned or referred to in the text.

To perform NER on the complaint dataset, the Python library **Stanza** developed by Stanford's NLP Group was employed. Using the library's English NER model, which was trained on the OntoNotes Corpus, four different types of entities were initially extracted from each complaint:

- Location (LOC) - Non-GPE locations, mountain ranges, bodies of water
- Facility (FAC) - Buildings, airports, highways, bridges, etc.
- Geopolitical Entity (GPE) - Countries, cities, states
- Organization (ORG) - Companies, agencies, institutions, etc.

However, upon evaluating the output, it was observed that the NER model could extract location keywords from only a limited number of complaints, approximately 82 out of the total dataset.

Further examination revealed that the model's performance was unsatisfactory due to the nature of Indian locality names. These names often included characteristics that the NER models were not trained on, such as names of individuals in the locality names (e.g., Lodhi Colony or Kasturba Gandhi Marg). Additionally, certain words like "Marg," "Path," and "Vihar," commonly used in complaints to denote specific locations, were not identified by the NER models.

Despite these challenges, efforts were made to extract location-related keywords from the complaints. The limitations and difficulties encountered during this process highlighted the need for further exploration and refinement of the NER models to better capture the specific characteristics of Indian locality names and improve the extraction of location information from the complaint descriptions.

To address the challenges faced with the existing NER approach, several improvements were implemented. The new approach involved a series of steps to extract location-related keywords from the complaint descriptions. These steps were executed in a specific order:

1. Entities of type LOC, FAC, GPE, or ORG were identified using the Stanza library. Duplicate entities within the same complaint were eliminated to ensure uniqueness.
2. If no entities were recognized in the previous step, each word in the complaint description was checked for LOC or ORG entities using the Polyglot NLP library. Again, duplicate entities were removed.
3. Any words in the complaint ending with "pur" or "garh" (common in Indian location names) were directly added to the recognized entities.
4. Any six-digit integer found in the complaint was considered a potential pincode and added to the recognized entities.
5. Directions such as north, south, east, or west were directly added to the recognized entities.

6. Bigrams (pairs of adjacent words) were extracted from each complaint, and specific patterns were used to identify relevant location-related bigrams. For example, if the second word in a bigram matched keywords like nagar, vihar, metro, village, road, apartments, bagh, or garden, and the length of the first word was greater than or equal to 4, it was added to the recognized entities. Additionally, bigrams where the first word was "Sec" or "Sector" were also included.

7. Trigrams (three adjacent words) with "near" as the first word and a second word of length greater than 3 were extracted and considered as potential location-related keywords.

8. All the extracted set of keywords were joint together to obtain one single location string.

These steps aimed to capture location-related information from the complaint descriptions by leveraging a combination of NLP libraries, pattern matching, and heuristics. The iterative approach increased the likelihood of identifying relevant location keywords, addressing the limitations encountered with the initial NER approach.

This refined approach was developed iteratively through multiple attempts to enhance the extraction of location information from the complaints. The goal was to increase the number of complaints with identified locations beyond the initial count of 82. Various improvements and modifications were made to the approach in order to achieve this objective.

After implementing the revised approach, a significant improvement was observed, resulting in a total of 1,124 complaints with successfully identified locations. This marked progress in expanding the dataset with location information, enabling further analysis and visualization of complaint hotspots based on categories and geographic locations. The iterative nature of the approach allowed for continuous refinement and optimization, ultimately leading to a substantial increase in the number of complaints with associated location details.

2.3.7 Geocoding Complaints - Converting Locations to Coordinates for Visual Analysis

In the previous section, location keywords were extracted from the complaints and organized into a separate dataset. The next step was to convert these location-like keywords into precise coordinates that could be plotted on a map for visual analysis.

Geocoding is the process of transforming a location description, such as an address or place name, into geographic coordinates on the Earth's surface. To achieve this, various geocoding libraries and APIs were explored. Open-source options like OpenStreetMaps were initially considered, but they struggled to accurately identify the fuzzy locations obtained from the extracted keywords.

To address this issue, enterprise-level Geocoding APIs, such as Azure Maps API and Google Maps API, were evaluated. In this project, the Azure Maps API was selected due to its ability to accurately geocode fuzzy locations and provide reliable coordinates. The Fuzzy Search method of the Azure API was used, running recursively to obtain coordinates for each location keyword. The resulting coordinates were then stored in the dataset for future use.

Subsequently, the obtained coordinates were plotted on an Indian map using the Folium library in Python. The results were manually checked for accuracy, and while there were some instances where locations were not identified correctly, overall, the error was minimal and satisfactory.

By geocoding the extracted location keywords and visualizing them on a map, the project enabled a spatial analysis of complaint hotspots based on category and geographic location. This allowed for a better understanding of the distribution and concentration of complaints across different areas, aiding in the identification of potential problem areas and facilitating targeted interventions.

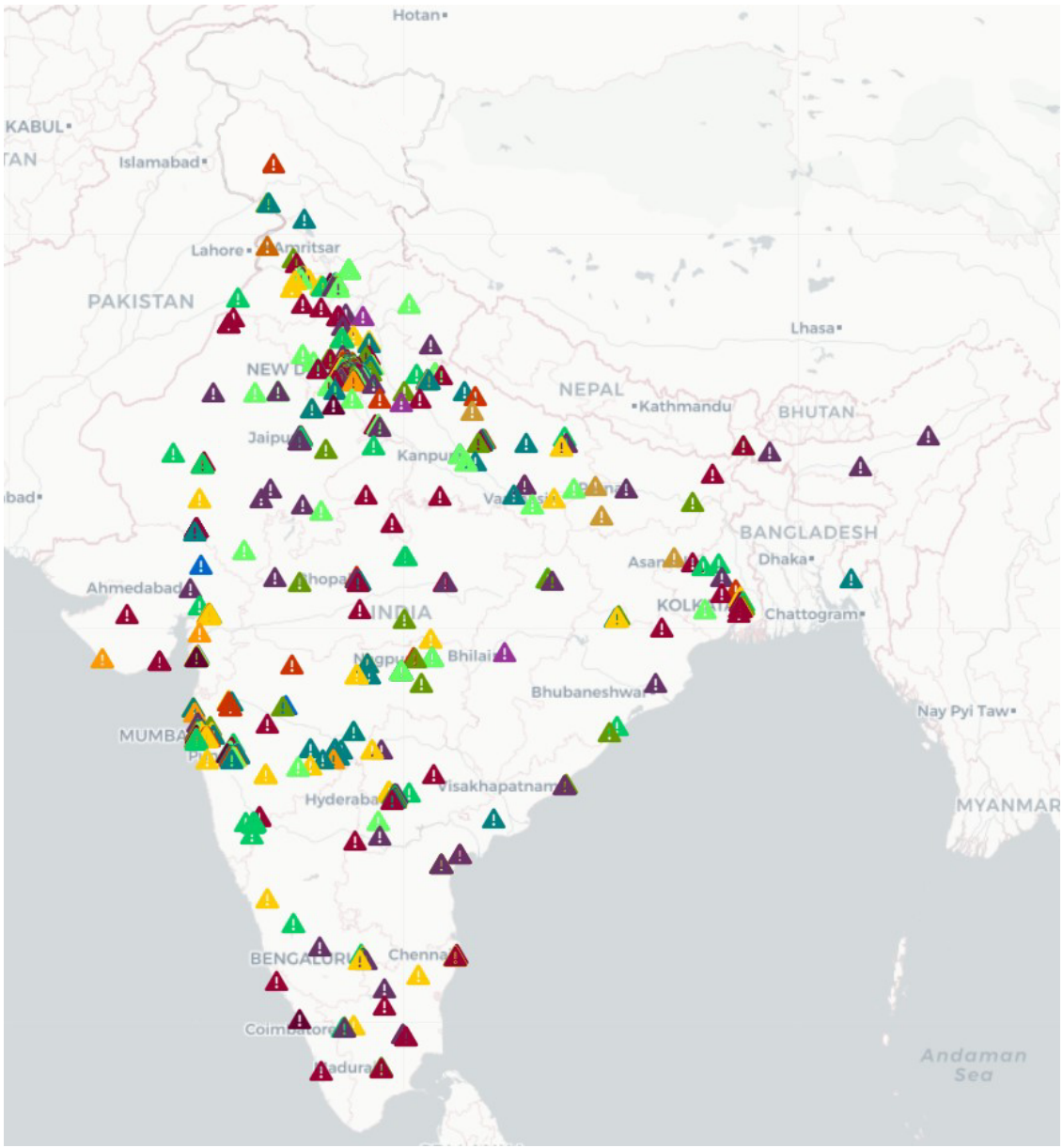


Figure 2.4: Map of India with Identified Locations - Color Coded Category Wise

Category	Color
Open/Garbage Burning	
Open dumping of garbage	
Industrial Emission	
Others	
Air pollution due to Construction/Demolition activity	
Leaf Burning	
Road Dust	
Unpaved Road/Pit	
Traffic Congestion in Residential Colonies	
Fire in Landfill Sites	
Industrial Waste Dumping	
Pollution from Illegal Industry	
Traffic Congestion on Roads	
Air and Noise pollution from Generator	

Figure 2.5: Legend for Map 2.4

2.4 Results and Discussion

The project involved two main tasks: complaint categorization and location identification. In this section, we present the results obtained for each task and provide a discussion on the findings.

2.4.1 Complaint Categorization

For the complaint categorization task, the performance of four different machine learning algorithms was evaluated: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest Classifier (RFC), and Multinomial Naive Bayes (M-NB). The models were trained on a dataset consisting of 6,142 complaints, and their performance was assessed using accuracy, recall, precision, and F1 score metrics.

The results, as shown in Table 2.5, indicate that all four models achieved relatively high accuracy in categorizing the complaints. SVM and LR demonstrated the highest overall accuracy, with scores of 0.736 and 0.723, respectively. These results suggest that the trained models were able to effectively classify the complaints into their respective categories.

However, it is important to note that the dataset used for training the models was imbalanced, with some categories having significantly fewer samples than others. This could lead to biased performance and lower recall for categories with fewer samples. To address this issue, an upsampling technique called SMOTE was employed, resulting in a more balanced dataset. The performance of SVM and LR models was then evaluated on this upsampled dataset, as shown in Table 2.6. It can be observed that the performance improved for most categories, with higher recall and F1 scores, indicating a better ability to correctly classify complaints across different categories.

Overall, the results of complaint categorization demonstrate the effectiveness of the machine learning models in accurately classifying complaints. The inclusion of the SMOTE technique for upsampling further enhanced the models' performance, particularly for categories with fewer samples.

2.4.2 Location Identification

The task of location identification aimed to extract location-related keywords from the complaint descriptions and convert them into geographical coordinates for visual analysis. The approach involved multiple steps, including NER (Named Entity Recognition) using the Stanza library, pattern matching, and heuristic techniques.

Initially, the NER model was applied to extract entities such as location, facility, geopolitical entity, and organization. However, due to the specific characteristics of Indian locality names and the presence of non-standardized language usage in complaints, the NER model struggled to identify location keywords accurately. As a result, only a limited number of complaints (82 out of 2,098) had recognized location entities.

To overcome this limitation, an iterative approach was adopted, combining various techniques such as Polyglot NLP, keyword matching, and extraction of location-related bigrams and trigrams. These steps significantly improved the extraction of location information from the complaints, resulting in a total of 1,124 complaints with identified location keywords.

The extracted location keywords were then geocoded using the Azure Maps API. This process involved converting the keywords into precise geographic coordinates for plotting on a map. The resulting coordinates were visually analyzed and plotted using the Folium library. While the geocoding process achieved satisfactory results overall, there were instances where certain locations were not identified accurately.

The location identification process provided valuable insights into the geographic distribution of complaints and enabled the identification of complaint hotspots based on category and location. By visualizing the data on a map, patterns and trends in complaint occurrence could be analyzed, facilitating targeted interventions and informed decision-making.

Overall, the results of the location identification task demonstrated the effectiveness of the iterative approach in extracting location-related keywords from the complaint descriptions. The combination of techniques allowed for a more comprehensive understanding of the spatial distribution of complaints and provided a foundation for further analysis and decision-making.

2.5 Conclusion

In this project, we tackled the challenges of complaint categorization and location identification using a combination of natural language processing (NLP) techniques and machine learning algorithms. The objective was to develop models that could accurately classify complaints into different categories and extract location information from the complaint descriptions.

For complaint categorization, we trained several machine learning models, including Support Vector Machine (SVM), Logistic Regression (LR), Random Forest Classifier (RFC), and Multinomial Naive Bayes (M-NB). The models exhibited high accuracy in categorizing complaints into their respective categories. Additionally, the implementation of upsampling using the SMOTE technique helped address the issue of imbalanced data, resulting in improved performance across various categories.

In the task of location identification, we employed a multi-step approach that involved named entity recognition (NER), pattern matching, and heuristics. Despite the challenges posed by the nature of Indian locality names and the presence of non-standardized language usage in complaints, the iterative approach significantly increased the extraction of location-related keywords. By geocoding these keywords using the Azure Maps API, we obtained geographic coordinates that facilitated visual analysis and the identification of complaint hotspots.

The successful completion of both tasks provides valuable insights for organizations and policymakers. The complaint categorization models can effectively categorize complaints, enabling better understanding of the issues faced by individuals and communities. This information can guide resource allocation, policy formulation, and targeted interventions to address specific categories of complaints.

Furthermore, the location identification process helps in visualizing complaint hotspots on a map, providing spatial context to the issues. This spatial analysis enhances decision-making by identifying areas with higher complaint density and enabling policymakers to prioritize interventions and allocate resources accordingly.

Overall, this project demonstrates the importance of leveraging NLP techniques and machine learning algorithms to analyze customer complaints and extract valuable insights. By automating the categorization and location identification processes, organizations can streamline complaint management, enhance customer satisfaction, and facilitate targeted interventions for a better overall user experience.

Chapter 3

A Journey of Growth and Discovery - My Internship Experience

Reflecting upon my internship journey, I realize it has been a dynamic and transformative experience. At the outset, I faced numerous challenges, grappling with the uncertainty of securing an internship and harboring self-doubt about my ability to meet the demands of the position. However, against all odds, I found myself in the midst of an exhilarating introduction that rapidly transitioned into a technical interview, ultimately leading to the realization of my internship aspirations. The projects I was entrusted with proved to be far more significant than I initially perceived, carrying the hopes and expectations of my supervisor, Sharandeep Sir.

Engaging in in-depth discussions and technical exchanges with my supervisor and mentors - Anurag Sir and Sharandeep Sir have been immensely rewarding, enhancing my understanding of the subject matter and enabling me to deliver exemplary work. The internship experience took me beyond the confines of my hometown in Chandigarh, as I ventured into the vibrant city of Delhi and experienced the feel of local transport and bustling markets. Embracing a hybrid work model, which seamlessly blended the office environment with the comfort of my own home, afforded me the luxury of dedicated work hours and opportunities for relaxation. The corporate office ambiance, with its energetic vibes, added a touch of professionalism and inspiration to my daily routine.

This internship has served as a catalyst for personal and professional growth, pushing the boundaries of my knowledge and introducing me to new possibilities. The unwavering support and guidance provided by my supervisors and mentors have been instrumental in shaping my experience and fostering a nurturing learning environment. This being my first internship, these cherished memories will forever hold a special place in my heart, leaving an ever lasting imprint on my future endeavors.

Harsh Bansal
Amity University Punjab