Urban Expansion and Land Use Monitoring Using Custom Machine Learning Models

ENGG680 – Introduction to Digital Engineering Course Project (Group 17) Fall 2024

Dr. HongzhouYang

Prepared by:

Tirth, Panchal	**30240101**	**tirth.panchal@ucalgary.ca**
Kandarp, Rathod	**30267885**	**kandarppankajbhai.ra@ucalgary.ca**
Parth, Bhamani	**30246903**	**parthrajeshbhai.bham@ucalgary.ca**
Obehi, Edeoghon	**30240132**	**obehi.edeoghon@ucalgary.ca**
Chan Nyein, Aung	**30217709**	**channyein.aung@ucalgary.ca**
Soheil, Seifi	**30229496**	**soheil.seifi@ucalgary.ca**
Seyedehshima, Moradi	**30203561**	**Seyedehshima.moradim@ucalgary.ca**
Zohreh, Mejrisazantoosi	**30232335**	**zohreh.mejrisazantoo@ucalgary.ca**

Certificate of Work

Title of Project	Urban Expansion and Land Use Monitoring Using Custom Machine Learning Models
Group Number	17

We, the undersigned, certify that this is our own work, which has been done expressly for this course, either without the assistance of any other party or where appropriate we have acknowledged the work of others. Further, we have read and understood the section in the university calendar on plagiarism/cheating/other academic misconduct and we are aware of the implications thereof. We request that the total mark for this assignment be distributed as follows among group members:

Your Name	Tirth Panchal
Student ID	30240101
Contribution (%) and Hours	12.5%, 5
Signature and Date	Tirth (2024-10-11)

Your Name	Kandarp Rathod
Student ID	30267885
Contribution (%) and Hours	12.5%,5
Signature and Date	Kandarp (2024-10-11)

Your Name	Parth Bhamani
Student ID	30146903
Contribution (%) and Hours	12.5%, 5
Signature and Date	Parth (2024-10-11)

Your Name	Obehi Edeoghon
Student ID	30240132
Contribution (%) and Hours	12.5%, 5
Signature and Date	Obehi (2024-10-11)

Your Name	Chan Nyein Aung
Student ID	30217709
Contribution (%) and Hours	12.5%, 5
Signature and Date	Chan (2024-10-11)

Your Name	Soheil Seifi
Student ID	30229496
Contribution (%) and Hours	12.5%, 5
Signature and Date	Soheil (2024-10-11)

Your Name	Seyedehshima Moradi
Student ID	30203561
Contribution (%) and Hours	12.5%, 5
Signature and Date	Seyedehshima (2024-10-11)

Your Name	Zohreh Mejrisazantoosi
Student ID	30232335
Contribution (%) and Hours	12.5%, 5
Signature and Date	Zohreh (2024-10-11)

* Contribution total should be 100%.

Abstract

This project aims to predict traffic accident occurrences across six geographical clusters in Calgary using a machine learning pipeline that integrates weather, temporal, and traffic accident data. The pipeline employs Random Forest, XGBoost, and CatBoost models, with CatBoost achieving the best performance (F1-Score: 0.70, Accuracy: 0.63). Data preprocessing involved cleaning, aggregating, and merging traffic and weather datasets to align with spatial and temporal patterns. An interactive visualization tool was developed to provide actionable insights, enabling policymakers and traffic managers to identify high-risk zones and implement targeted interventions. This work demonstrates the potential of machine learning in enhancing urban traffic management and safety.

Contents

Tables of Figures

1. Introduction

Problem Context

Urbanization is a double-edged sword. On one side, it fosters economic growth and modernization; on the other, it brings challenges such as traffic congestion, increased accident risks, and compromised public safety. The global urban population is projected to reach nearly 70% by 2050, emphasizing the importance of efficient traffic management systems.

The increasing volume of traffic accidents and the variability of factors influencing them, such as weather conditions and temporal variations, necessitate advanced, data-driven solutions. Existing systems often fail to account for granular variations like local weather patterns, time-specific behaviors, or spatial clustering of accident-prone zones. These limitations can lead to suboptimal planning, reactive responses, and higher accident rates.

Project Objective

This project addresses these challenges by developing a machine learning pipeline capable of predicting traffic accidents with high accuracy and granularity. The specific goals include:

- Developing a multi-output machine learning model capable of handling imbalanced datasets and predicting traffic accidents across multiple clusters simultaneously.
- Integrating temporal data (e.g., time of day, day of the week) and environmental data (e.g., weather) into the prediction framework.
- Visualizing accident risks interactively, making predictions accessible for practical applications such as urban planning and real-time traffic management.

This work focuses on Calgary's urban area, which has been divided into six geographical clusters, with traffic predictions further segmented into four time periods (Morning, Lunch, Evening, and Night). By combining machine learning techniques with visualization tools, the project offers a forward-thinking solution for mitigating traffic accidents and improving urban mobility.

2. Existing Solutions and Gaps

Existing Solutions

Traffic accident prediction has been explored using several methods, ranging from traditional statistical approaches to advanced machine learning techniques. Key solutions include:

- **Traditional Models:**

Linear regression, logistic regression, and decision trees have been widely used to identify patterns in traffic incident data. These models focus on factors such as road design, weather conditions, and time of day.

While effective for basic analysis, these methods often fail to incorporate the integration of multiple dynamic factors, such as the combined influence of weather conditions and temporal patterns, which are critical in improving predictive accuracy.

- **Advanced Machine Learning Models:**

Recent approaches involve models like neural networks and support vector machines, which analyze complex relationships in traffic data. These techniques allow for more nuanced insights into accident probabilities by integrating road conditions, driver behavior, and historical data.

For example, machine learning algorithms have been used for spatial analysis and predictive modeling to forecast traffic incidents.

Identified Gaps

Despite these efforts, several critical gaps remain:

- **Integration of Temporal and Environmental Factors**: Many existing models fail to incorporate dynamic weather conditions, time of day, and historical trends, leading to generalized and less accurate predictions.
- **Handling Class Imbalance**: Traffic accident datasets often exhibit an imbalance between accident-prone and non-accident-prone areas, which can bias models toward majority classes.
- **Granularity and Visualization**: Most existing tools lack the capability to provide actionable, location-specific insights in an interactive and user-friendly format.

By addressing these gaps, this project contributes to bridging the divide between theoretical research and practical implementation.

3. Relevance to Engineering

This project is highly relevant to the domains of civil, geomatics, and transportation engineering, offering critical insights that address public safety, infrastructure planning, and urban mobility. Its contributions include:

- **Traffic Management Systems:** The project's predictions can inform the development of smarter traffic management strategies, enabling proactive measures to adapt to changing traffic patterns.
- **Infrastructure Planning:** Geographic analysis supports decision-making for road improvements and urban infrastructure development to minimize accident risks.
- **Geomatics Engineering:** Leveraging geospatial analysis techniques and GIS, the project provides a spatial perspective on accident-prone areas, aiding targeted urban planning.

- **Automotive Design:** Insights from the model can assist in designing smarter vehicles with accident-prevention technologies by highlighting high-risk areas.

This project integrates multiple disciplines to provide data-driven solution for traffic management, urban planning, and accident prevention.

4. Data Preprocessing

Data preprocessing is a critical stage in this project, transforming raw data into a structured format suitable for machine learning. In this project, we preprocess two datasets: **Traffic Accident Data** and **Weather Data**. This section details the preprocessing of **Traffic Accident Data**, which forms the basis for creating binary classification targets to predict accidents in specific spatial clusters.

4.1. Traffic Accident Data Preprocessing

The preprocessing of traffic accident data is achieved through a detailed, multi-step workflow. The main steps are as follows:

- **Library Importation:** The following libraries were utilized for cleaning and preprocessing the traffic accident dataset:
  - **pandas:** For handling, cleaning, and manipulating tabular data efficiently.
  - **NumPy:** For numerical operations, such as handling missing values and applying vectorized computations.
  - **sklearn. cluster (K-Means):** For clustering traffic data into meaningful groups to identify patterns and trends.
  - **shapely. geometry (Point):** For geographical data processing, such as identifying clusters based on spatial coordinates.
  - **SciPy. spatial (Convex Hull):** For visualizing clusters and defining the boundaries of high-risk areas.
  - **datetime:** For managing date and time-related operations, such as time period segmentation.
  - **folium:** For interactive visualization of traffic accident clusters on a map.
  - **itertools (product):** For generating combinations of time periods and dates for comprehensive data aggregation.

These libraries collectively enabled efficient processing, spatial clustering, and visualization of traffic accident data.

- **Data Cleaning and Parsing:** The dataset is loaded, and the essential columns (e.g., timestamps, coordinates, and accident descriptions) are cleaned:
  - **Missing Values:**
    Missing values in the QUADRANT column are imputed using a custom function based on geographic coordinates (Latitude and Longitude). This ensures spatial data integrity, especially for regions with incomplete information.

- **Datetime Parsing:**
  The START_DT column is parsed to separate Date and Time, providing granularity for temporal analysis.
- **Debugging Outputs:**
  Interim results are saved as CSV files during debugging to validate the cleaning process

- ## Temporal Feature Engineering

Accidents are grouped into four distinct time periods based on the START_DT timestamp: 1) Morning (6 AM–12 PM) 2) Lunch (12 PM–6 PM) 3) Evening (6 PM–12 AM) 4) Midnight (12 AM–6 AM)

These time periods capture the daily temporal dynamics of traffic incidents, ensuring that models consider the impact of time on accident probabilities.

- ## Spatial Clustering Using K-Means

- **Unsupervised clustering (K-Means)** is applied to segment incidents into six geographic clusters (Cluster0 to Cluster5) based on Latitude and Longitude.
- **Cluster Assignments:** Each incident is assigned to one of six clusters, which correspond to specific regions in the city.
- **Visualization:** An interactive map of clusters is created using folium, showing spatial distributions and boundaries for each cluster.

- ## Aggregating and Enriching Data

The dataset is aggregated by Date, Time_Period, and Cluster to generate summary statistics for each cluster over time.

- **Historical Features:**

The dataset is enriched with **6 historical days** for each cluster: For example, the column C0D-1HA represents incidents in Cluster0 one day ago, while C1D-2HA represents incidents in Cluster1 two days ago.

These shifted features allow the dataset to capture temporal trends and dependencies.

- ## Creating Binary Targets

Each cluster's column (Cluster0, Cluster1, ..., Cluster5) serves as a binary prediction target for the machine learning models:

1: Accident occurred in the cluster.

0: No accident occurred in the cluster.

This transformation enables the models to predict the presence or absence of accidents in each cluster for a given time period.

- ## Dataset Structure

The final dataset contains:

  - **Date:** The date of the incidents.
  - **Time Period:** The time period of the incidents (Morning, Lunch, Evening, Midnight).
- **Cluster Columns**: Binary target columns (Cluster0, Cluster1, ..., Cluster5) indicating accident occurrences in each cluster.
- **Historical Features:** Shifted columns for each cluster capturing accidents in the past 6 days (e.g., C0D-1HA, C1D-6HA).
- **Total Accidents:** A summary column representing the total number of accidents across all clusters for a given time period.

- **Purpose of Traffic Accident Data Preprocessing**
  - **Spatial and Temporal Context:** Integrates geographic clustering and temporal segmentation into the dataset, allowing the models to analyze spatial and temporal patterns.
  - **Enhanced Predictive Features:** Historical features (6 days of lagged data) enrich the dataset, enabling models to learn from temporal dependencies.
  - **Binary Targets for Classification:** Converts accident counts into binary targets for straightforward prediction, aligning with the project's classification objective.

4.2. Weather Data Preprocessing

Weather data cleaning and preprocessing is an essential step in ensuring that the dataset is ready for integration with the traffic accident dataset and for use in predictive modeling. Below is a detailed explanation of the weather data cleaning process, covering all critical aspects, including the libraries used, handling missing values, feature engineering, and the rationale behind the methodologies.

- **Libraries Used:** The following libraries were utilized for data cleaning and preprocessing:
  - **pandas**: For handling and manipulating the dataset
  - **datetime:** For date and time manipulations.

These libraries provide robust tools to process, clean, and augment data efficiently.

- **Data Loading and Exploration**

The data contained multiple columns related to weather parameters (e.g., temperature, humidity, wind speed) collected over an extended time period.

- **Initial Insights:**

The dataset spans from 2015-01-01 to 2024-12-31, covering 3653 days.

A missing values summary revealed that several columns had a high percentage of missing data, necessitating careful handling:

Columns like "Precip. Amount (mm)", "Wind Dir Flag", and others had almost 100% missing values.

Key columns like temperature and wind speed had manageable missing values (~5.5%).

- **Missing Values Handling**

To ensure the dataset's usability, missing values were addressed as follows:

- **Columns with 100% Missing Values:**

  These columns were dropped since they provide no usable information. Examples include "Wind Dir Flag", "Precip. Amount Flag", and others.

- **Columns with Partial Missing Values:**

The remaining columns with missing values were imputed using appropriate strategies:

- **Mean Imputation:** Used for numerical data with continuous distribution (e.g., temperature, visibility).
- **Mode Imputation**: Used for categorical-like columns, such as "Wind Direction (10s deg)".
- **Median Imputation:** Employed for columns sensitive to outliers.

This approach ensured the integrity and consistency of the dataset while avoiding potential biases from improper handling.

• Feature Engineering

To enhance the dataset's relevance for integration and analysis, additional features were engineered:

- **Time Period Categorization:** Time periods were categorized into four segments: 1) Morning: 6 AM to 12 PM. 2) Lunch: 12 PM to 6 PM. 3) Evening: 6 PM to 12 AM 4) Midnight: 12 AM to 6 AM. This segmentation aligns with the temporal resolution of the traffic accident dataset, enabling better correlation analysis.
- **Statistical Aggregation:** The weather data was aggregated based on Date and Time Period:

Mean values were calculated for columns like temperature, humidity, wind speed, and visibility.

Variance was calculated for temperature to capture day-to-day fluctuations, which may affect road conditions and, subsequently, accidents.

Rationale for Variance: Temperature variance provides insights into abrupt weather changes, which could impact road conditions and accident likelihood (e.g., sudden ice formation due to rapid cooling).

- **Holiday Indicator:** A binary column (Is_Holiday) was added to indicate whether a date is a statutory holiday. This information is crucial since holidays can influence traffic patterns and accident risk.
- **Weekend Indicator:** A binary column (Is_Weekend) was added to distinguish between weekdays and weekends. Weekends often experience different traffic volumes and conditions.

• Final Dataset

After preprocessing and feature engineering, the cleaned weather dataset includes the following columns:

- **Date:** Standardized in datetime format.
- **Time_Period:** Categorized into morning, lunch, evening, and midnight.
- **Aggregated Weather Parameters: Mean** values for: 1) Dew Point Temperature (°C) 2) Relative Humidity (%) 3) Wind Speed (km/h) 4) Visibility (km) 5) Station Pressure (kPa)

Variance for: Temperature (°C)

- **Additional Features:** 1) Day_Of_Week: Day of the week (0 = Monday, ..., 6 = Sunday) 2) Is_Weekend: Binary (1 for weekends, 0 for weekdays). 3)Is_Holiday: Binary (1 for holidays, 0 otherwise).

- **Why These Features Are Important**
  - **Aggregation:** Aggregating weather data by time period ensures alignment with traffic data, which is also aggregated by similar periods. Mean and variance values provide a robust summary of weather conditions, capturing both average trends and fluctuations.
  - **Temporal and Contextual Features:** Holidays, weekends, and day-of-week indicators capture behavioral patterns in traffic, such as reduced weekday traffic during holidays or increased weekend travel.

4.3. Merging Traffic and Weather Data

The merging phase of this project integrates the previously cleaned traffic accident and weather datasets into a unified dataset. This integration is essential for combining traffic-related patterns with weather conditions to create a comprehensive dataset for predictive modeling.

Objective: The main objective of merging the two datasets is to: 1. Explore the impact of weather conditions on traffic accidents. 2.Enrich the dataset with features from both domains to enhance model predictions. 3.Align temporal data from the traffic and weather datasets for seamless analysis.

- **Libraries Used:** The following library was utilized for the merging process:
- **pandas**: Provides efficient tools for merging, filtering, and handling large datasets.
- **Loading Datasets:** The cleaned traffic accident data and weather data were used as inputs for the merging process.
- **Ensuring Consistent Date Format:** The Date columns in both datasets were converted to a uniform datetime format to ensure alignment and avoid discrepancies during merging.
- **Validating Required Columns:** The Date and Time_Period columns were verified in both datasets as they form the basis for merging. This ensures the temporal alignment of rows.
- **Determining Common Date Range:** The date ranges of both datasets were analyzed:
- **Traffic Data Range:** December 6, 2016 – November 14, 2024.
- **Weather Data Range:** January 1, 2015 – December 31, 2024.
  The overlapping date range of January 1, 2017, to October 31, 2024, was selected for merging, ensuring data consistency.
- **Filtering Datasets:** Both datasets were filtered to include only rows within the overlapping date range. This ensures all data points in the final dataset are temporally aligned.
- **Merging Process:** The datasets were merged using an inner join on the Date and Time_Period columns. This operation ensured that only matching rows (i.e., rows with corresponding dates and time periods) were included in the final dataset.
- **Final Dataset:** The merged dataset contains the following types of features:

- **Traffic Features:** Information on traffic clusters (Cluster0 to Cluster5), representing accident hotspots and frequencies.
- **Weather Features:** Key variables such as temperature, wind speed, visibility, and humidity.
- **Temporal Features:** Day of the week, weekend indicator, and holiday indicator.

• Significance of Merging

The merging process was critical for:

- **Feature Enrichment:** Combining traffic accident clusters with weather data to create a richer feature space for machine learning.
- **Temporal Precision**: Accurate alignment of traffic and weather data ensures reliable analysis and predictive modeling.
- **Facilitating Model Input:** The merged dataset serves as the input for training the machine learning models, enabling predictions based on both weather and traffic-related features.

A	B	C	D	E	F	G	H	I	J	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
Date	Time_Perio	Cluster0	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	C0D-1HA	C0D-2HA	C5D-3HA	C5D-4HA	Total_Accı	Temp (Â°C	Dew Point	Rel Hum (9	Wind Dir (1	Wind Spd (	Visibility (k	Stn Press (	Day_Of_We	Is_Weeker	Is_Holiday
1/1/2017	0	0	0	0	0	0	0	0	0	0	0	0	0.085667	-14.7833	79.83333	18.66667	25.5	8.116667	88.96667	6	1	1
1/1/2017	1	0	0	0	0	0	0	0	0	0	0	0	0.005667	-16.1	70.83333	24.33333	16.66667	21.16667	89.10833	6	1	1
1/1/2017	2	0	0	0	0	0	0	0	0	0	0	0	2.896	-17.1833	79	20.5	9.5	17.15	89.23	6	1	1
1/1/2017	3	0	0	0	0	0	0	0	0	0	0	0	1.142667	-12.5833	83.66667	2.666667	23.83333	3.066667	88.545	6	1	1
1/2/2017	0	0	0	0	0	0	0	0	0	0	0	0	1.675	-25.7833	78	22.66667	2.5	17.7	89.52	0	0	0
1/2/2017	1	0	0	0	0	0	0	0	0	0	0	0	1.409667	-21	77	14.5	2.833333	16.63333	89.69167	0	0	0
1/2/2017	2	0	0	0	0	0	0	0	0	0	0	0	0.952	-21.7833	78.33333	23.16667	5.333333	24.1	89.82	0	0	0
1/2/2017	3	0	0	0	0	0	0	0	0	0	0	0	1.147	-22.55	80.83333	14.66667	3.666667	11.28333	89.32167	0	0	0
1/3/2017	0	0	0	0	0	0	0	0	0	0	0	0	0.518667	-22.7667	75.33333	28.33333	6.166667	20.4	89.935	1	0	0
1/3/2017	1	0	0	0	0	0	0	0	0	0	0	0	1.363	-23.05	61.5	21.33333	4.666667	56.35	89.84	1	0	0
1/3/2017	2	0	0	0	0	0	0	0	0	0	0	0	1.653667	-22.9833	67.33333	24.5	7.166667	24.1	89.74	1	0	0
1/3/2017	3	0	0	0	0	0	0	0	0	0	0	0	0.289667	-22.2167	78.5	29.16667	6.666667	24.1	89.83833	1	0	0
1/4/2017	0	0	0	0	0	0	0	0	0	0	0	0	0.24	-14.9667	73.5	33.83333	19	44.25	89.51833	2	0	0
1/4/2017	1	0	0	0	0	0	0	0	0	0	0	0	1.891	-12.6667	74.16667	34.5	21.33333	64.4	89.46167	2	0	0
1/4/2017	2	0	0	0	0	0	0	0	0	0	0	0	3.555	-16.75	79.66667	22.5	7.166667	24.1	89.43333	2	0	0
1/4/2017	3	0	0	0	0	0	0	0	0	0	0	0	5.013667	-18.5167	60.5	32.16667	16	24.1	89.635	2	0	0
1/5/2017	0	0	0	0	0	0	0	0	0	0	0	0	1.269667	-9.75	77.83333	25.66667	7	12.85	88.90167	3	0	0
1/5/2017	1	0	0	0	0	0	0	0	0	0	0	0	0.992	-8.13333	76.83333	27.66667	12.33333	11.91667	88.70167	3	0	0
1/5/2017	2	0	0	0	0	0	0	0	0	0	0	0	0.997667	-10.6	85.5	14.16667	14	5.6	88.76333	3	0	0
1/5/2017	3	0	0	0	0	0	0	0	0	0	0	0	6.179	-14.75	73.83333	21.33333	9.333333	8.016667	89.17667	3	0	0
1/6/2017	0	0	0	0	0	0	0	0	0	0	0	0	0.141667	-16.7	81.66667	14.33333	15.66667	2.8	88.71833	4	0	0
1/6/2017	1	0	0	0	0	0	0	0	0	0	0	0	0.668	-18.3333	78.5	14.83333	13.16667	11.2	88.885	4	0	0
1/6/2017	2	0	0	0	0	0	0	0	0	0	0	0	0.037667	-17.75	84.33333	17.33333	15	4.816667	89.13	4	0	0
1/6/2017	3	0	0	0	0	0	0	0	0	0	0	0	2.218667	-14.15	86.66667	12.16667	10.66667	5.633333	88.72	4	0	0
1/7/2017	0	0	0	0	0	0	0	0	0	0	0	0	0.879	-18.9667	80.5	34.83333	22.16667	4.266667	89.51333	5	1	0
1/7/2017	1	0	0	0	0	0	0	0	0	0	0	0	0.557667	-19.3667	75.83333	18.5	22	17.96667	89.48667	5	1	0

Figure 1: X,Y Data features (merged)

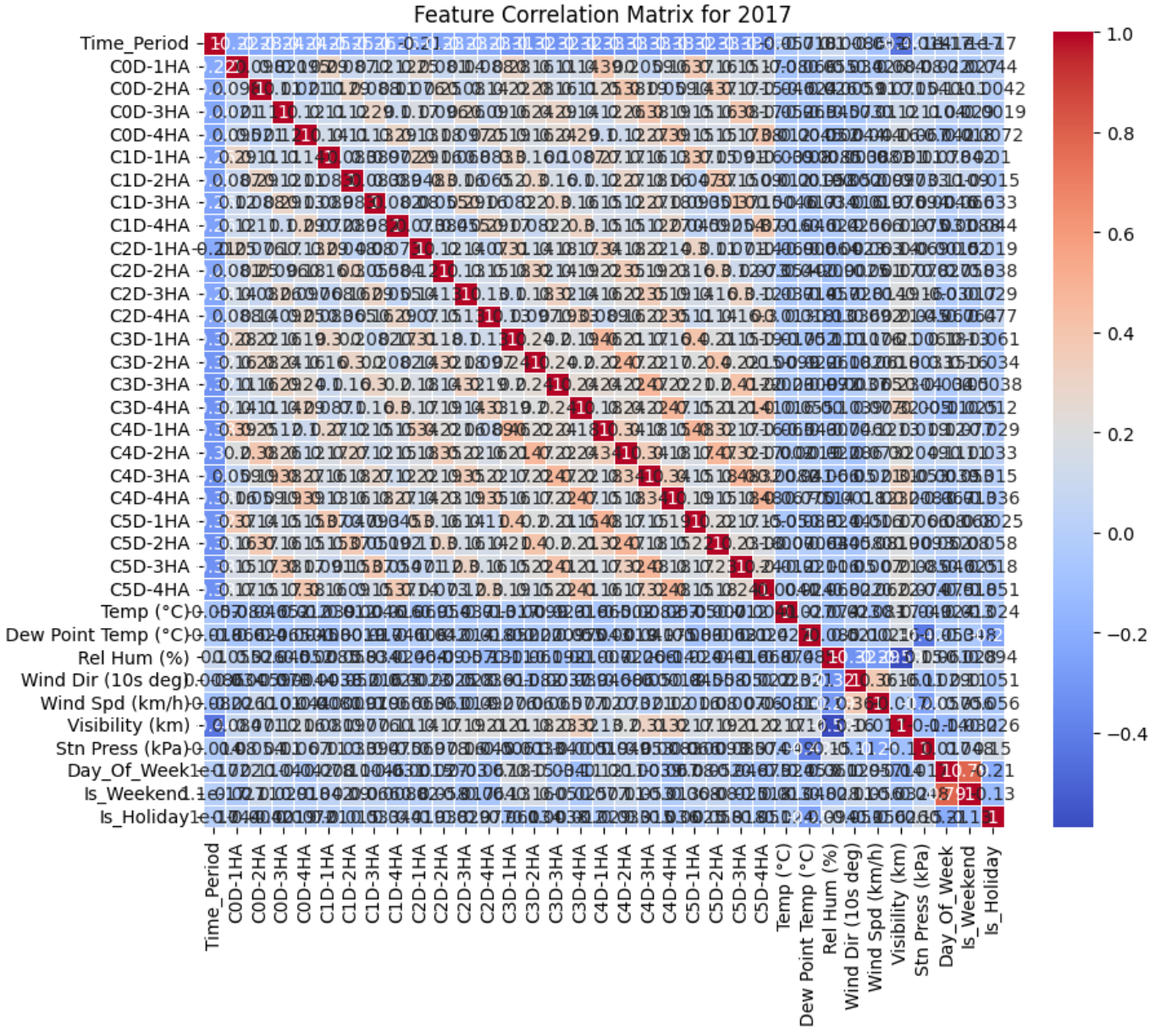

Figure 2: feature correlation matrix 2017

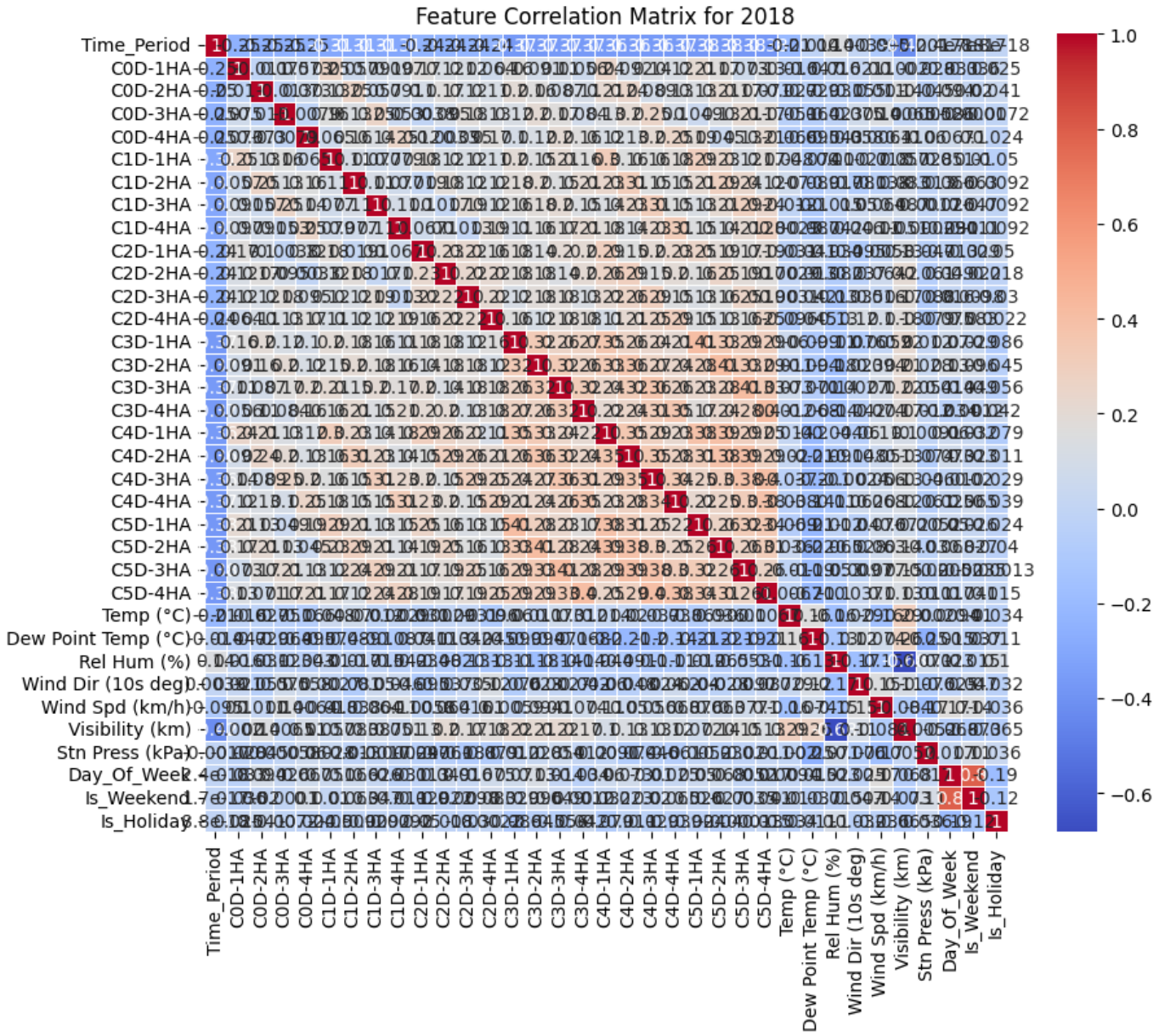

Figure 3: feature correlation matrix 2018

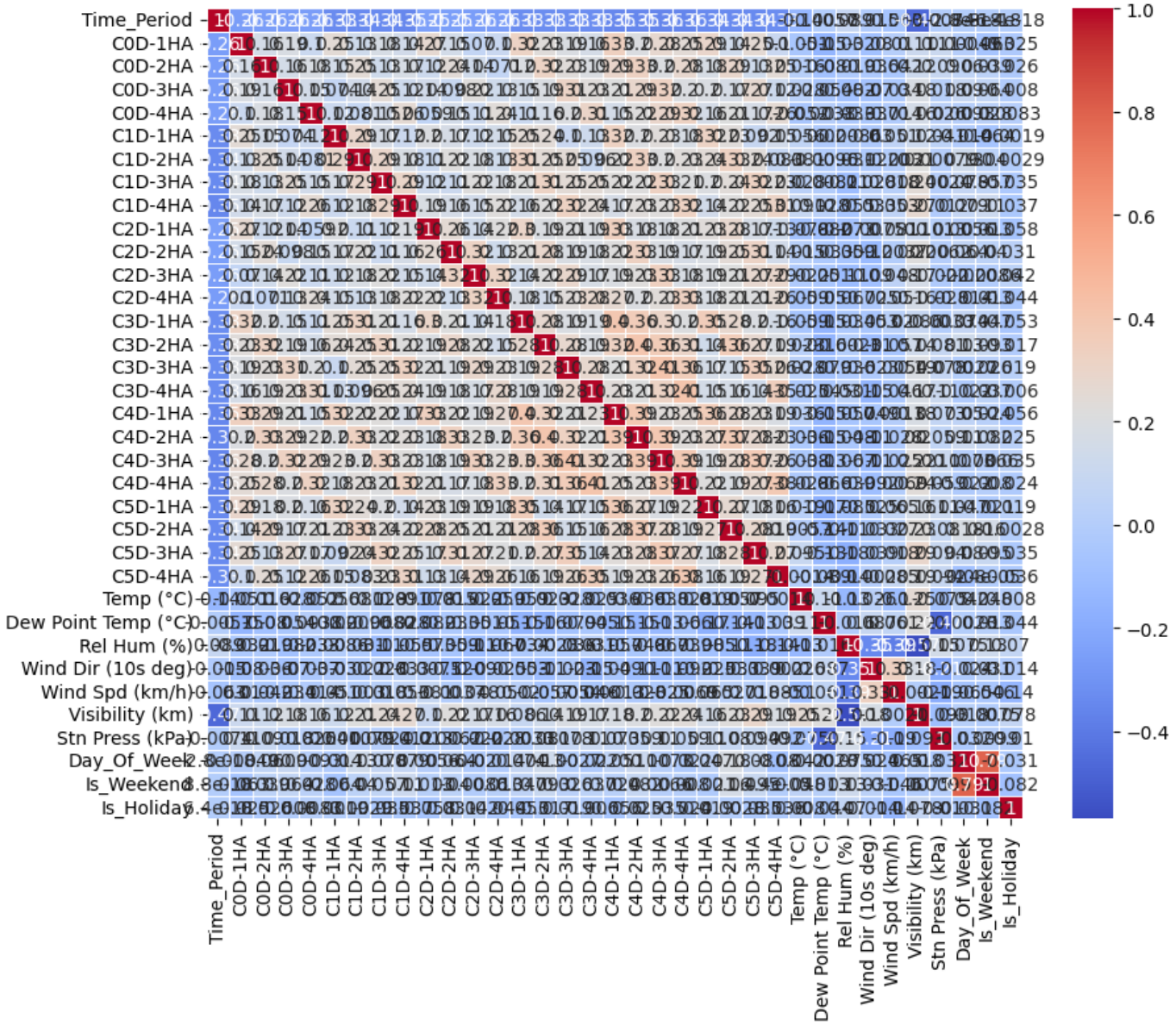

Figure 4: feature correlation matrix 2019

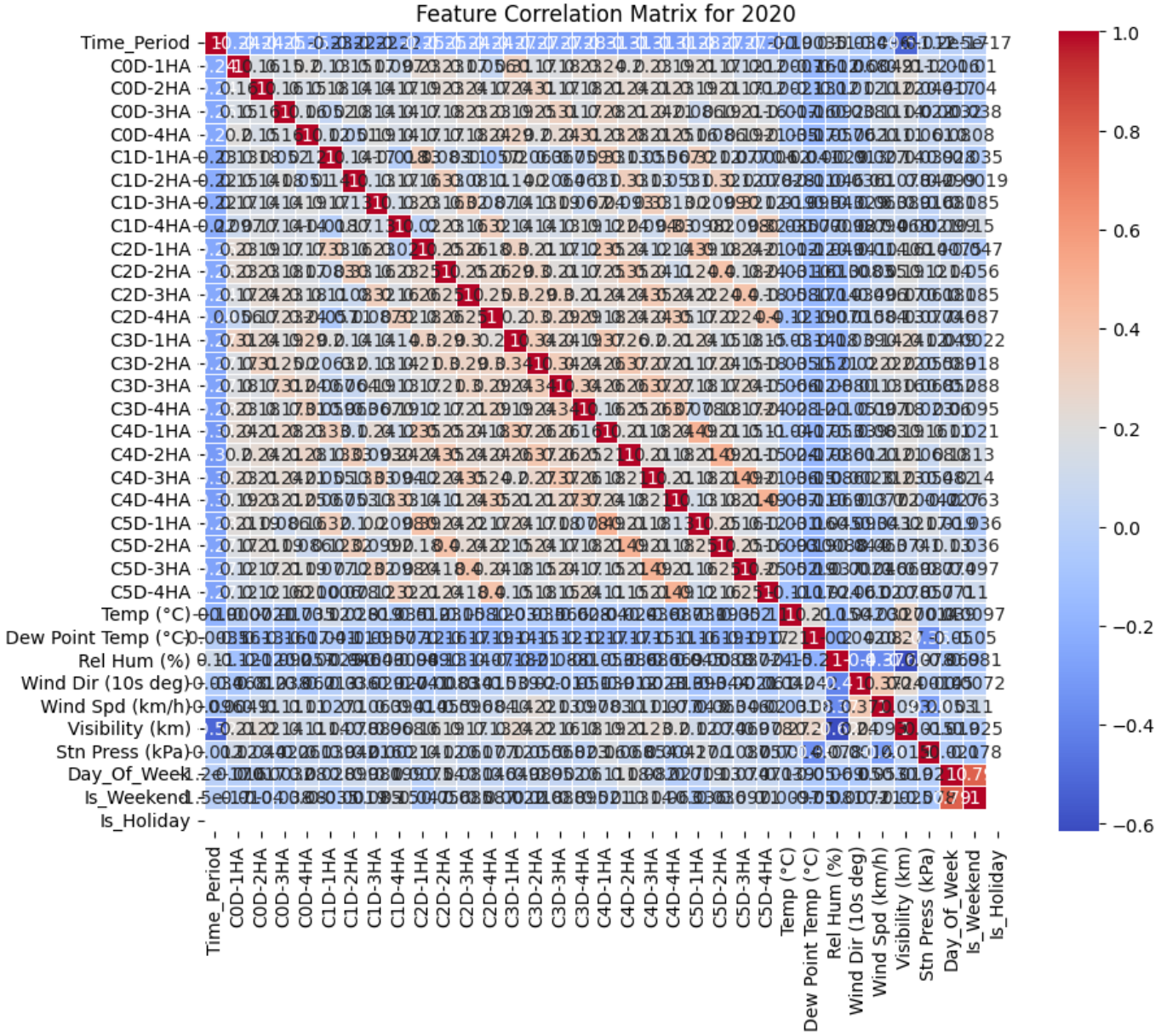

Figure 5: feature correlation matrix 2020

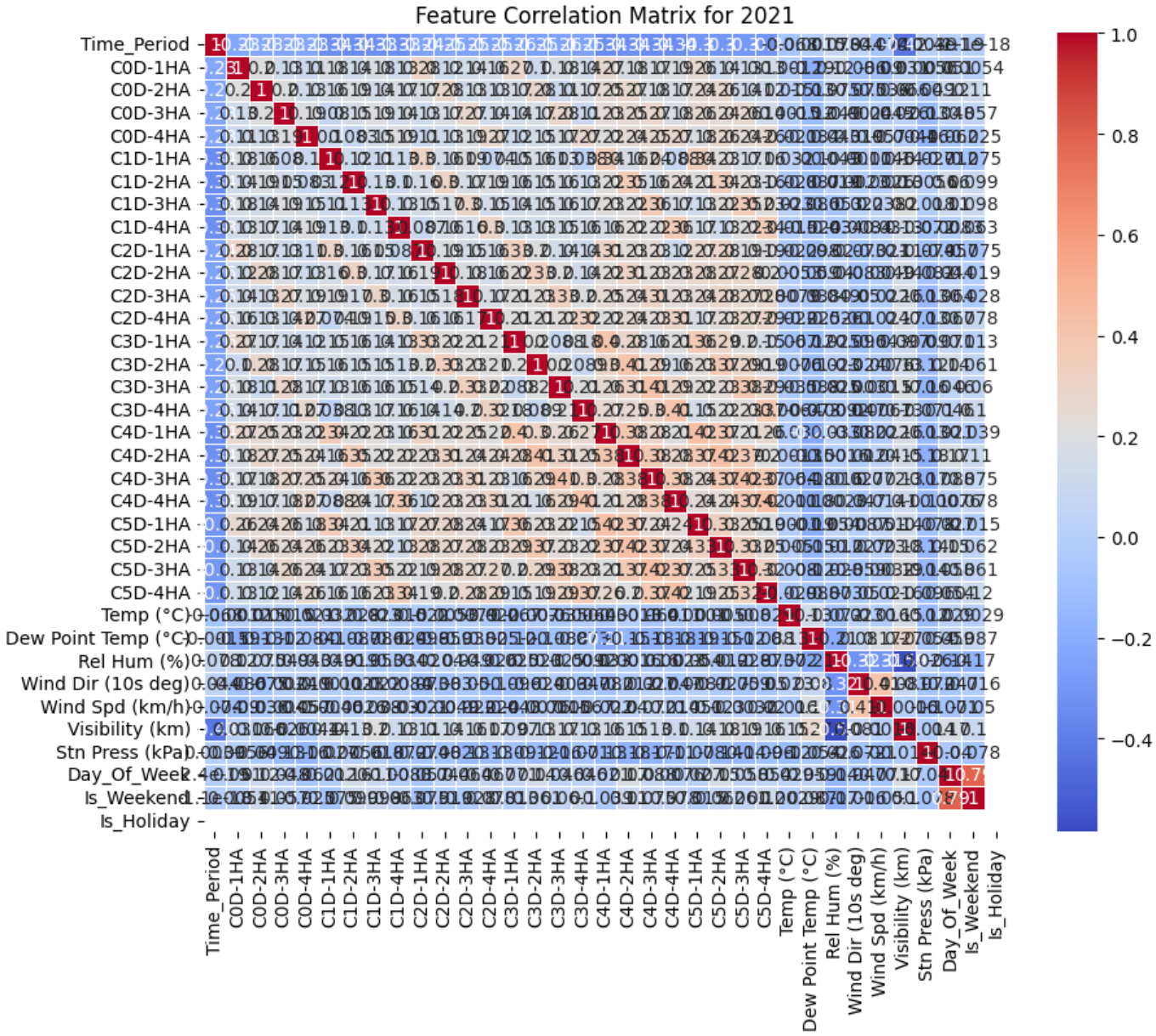

Figure 6: feature correlation matrix 2021

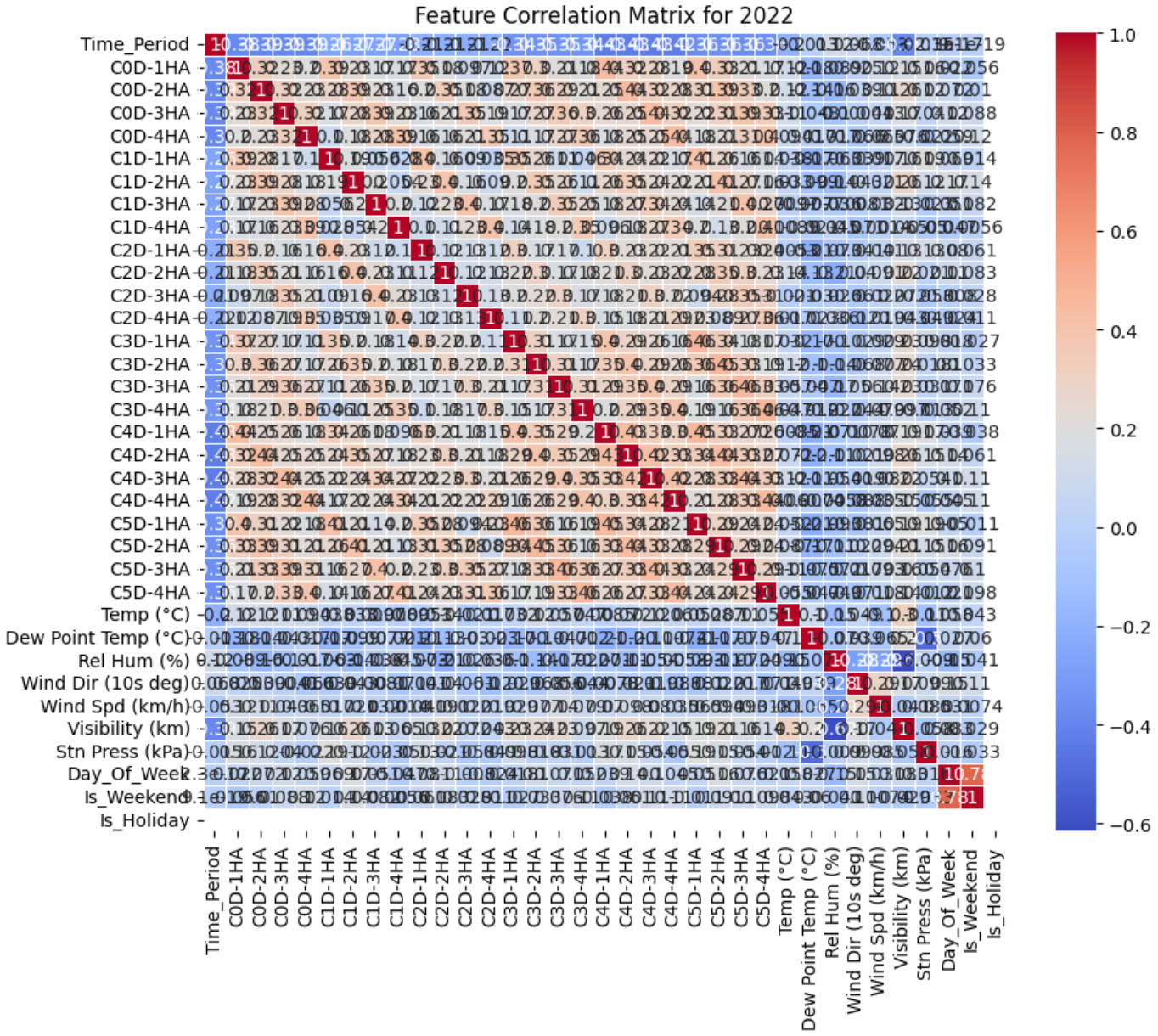

Figure 7: feature correlation matrix 2022

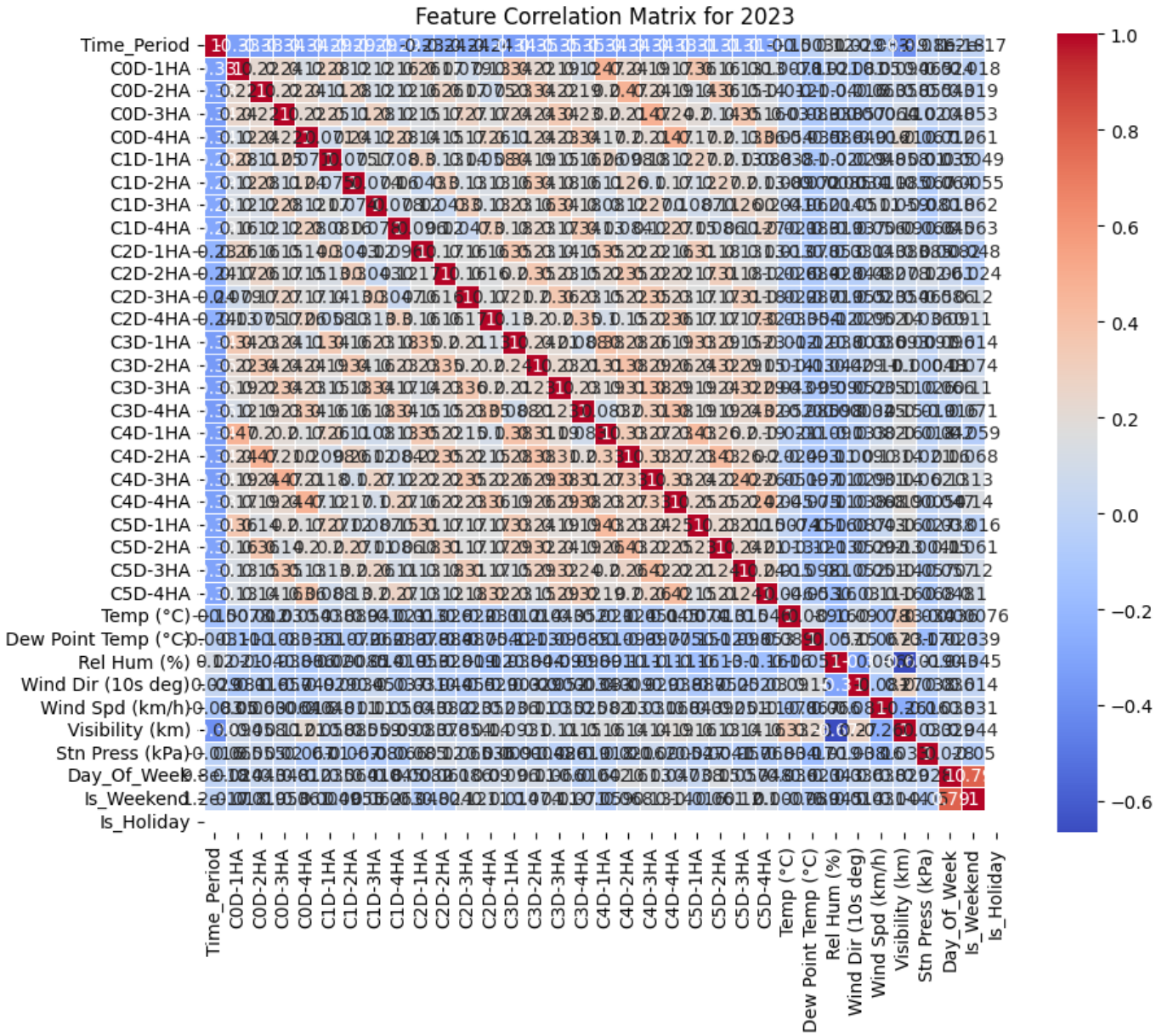

Figure 8: feature correlation matrix 2023

5. Feature Importance Extraction

Objective: The primary goal of this step was to identify and quantify the relative importance of different features in predicting traffic clusters. This helps in determining which factors most significantly influence traffic incidents across different years.

5.1. Steps and Methodology

- **Data Loading:**
  - **Datasets:** Seasonal datasets for different years were loaded using pandas.
  - **Purpose:** Each dataset contains weather attributes, temporal features, and clustered traffic incident data.
- **Defining Target and Features:**
  - **Target Variables:** Binary cluster variables (Cluster0 to Cluster5) serve as the prediction targets, indicating whether a particular cluster experienced traffic incident.
  - **Feature Set:** Excludes non-predictive columns like Date and Total_Accidents. Focuses on weather-related attributes (e.g., Temp (°C), Visibility (km)) and temporal variables (Time_Period, Day_Of_Week, etc.).
- **Random Forest Classifier:**
  - **A Random Forest Classifier** from scikit-learn was trained on each year's data to calculate feature importance. This model is widely used for its ability to provide a clear measure of the relative importance of input features.
  - **Process:**
    - ✓ Input data was split into features (X) and targets (y).
    - ✓ The classifier was trained on the data using rf.fit(X, y).
    - ✓ Feature importance was extracted using rf.feature_importances_.s
- **Feature Importance Analysis:**

4.1 Storage: Feature importance values for each year were stored in a dictionary.

4.2 Sorting and Display: Features were sorted based on their importance scores. The top features were displayed for each year, highlighting their influence on predicting traffic incidents.

- **Visualization:**
  - **Comparison Across Years:** A line plot was generated to compare feature importance scores for each year.
  - **Key Features:** Temporal variables like Time Period consistently ranked high across years, indicating their critical role in clustering. Weather attributes such as Visibility (km) and Temp (°C) also showed notable importance, reflecting their impact on traffic patterns.

5.2. Key Insights and Observations

- **Consistency in Temporal Features:** Features like Time_Period and Day_Of_Week consistently ranked high, highlighting their strong predictive value. This aligns with the hypothesis that traffic patterns are heavily influenced by time and day.

- **Weather Influence:** Attributes like Visibility (km), Wind Dir (10s deg), and Temp (°C) demonstrated variable importance across years, indicating fluctuating weather impacts.
- **Comparative Analysis:** The visualization highlights year-to-year variability in feature importance, offering insights into changing traffic dynamics influenced by weather and temporal factors.
- **Seasonal Relevance:** Winter datasets were specifically chosen due to the heightened impact of weather on traffic during this season. This ensures the findings are relevant for critical traffic safety improvements.

5.3. Significance in Traffic Prediction

- The feature importance results guide model optimization by focusing on the most influential predictors.
- Insights gained from this analysis aid in targeted interventions, such as adjusting traffic management strategies for specific times or weather conditions.

By identifying these key factors, we are better positioned to refine predictive models, ensuring they capture the most critical drivers of traffic incidents.

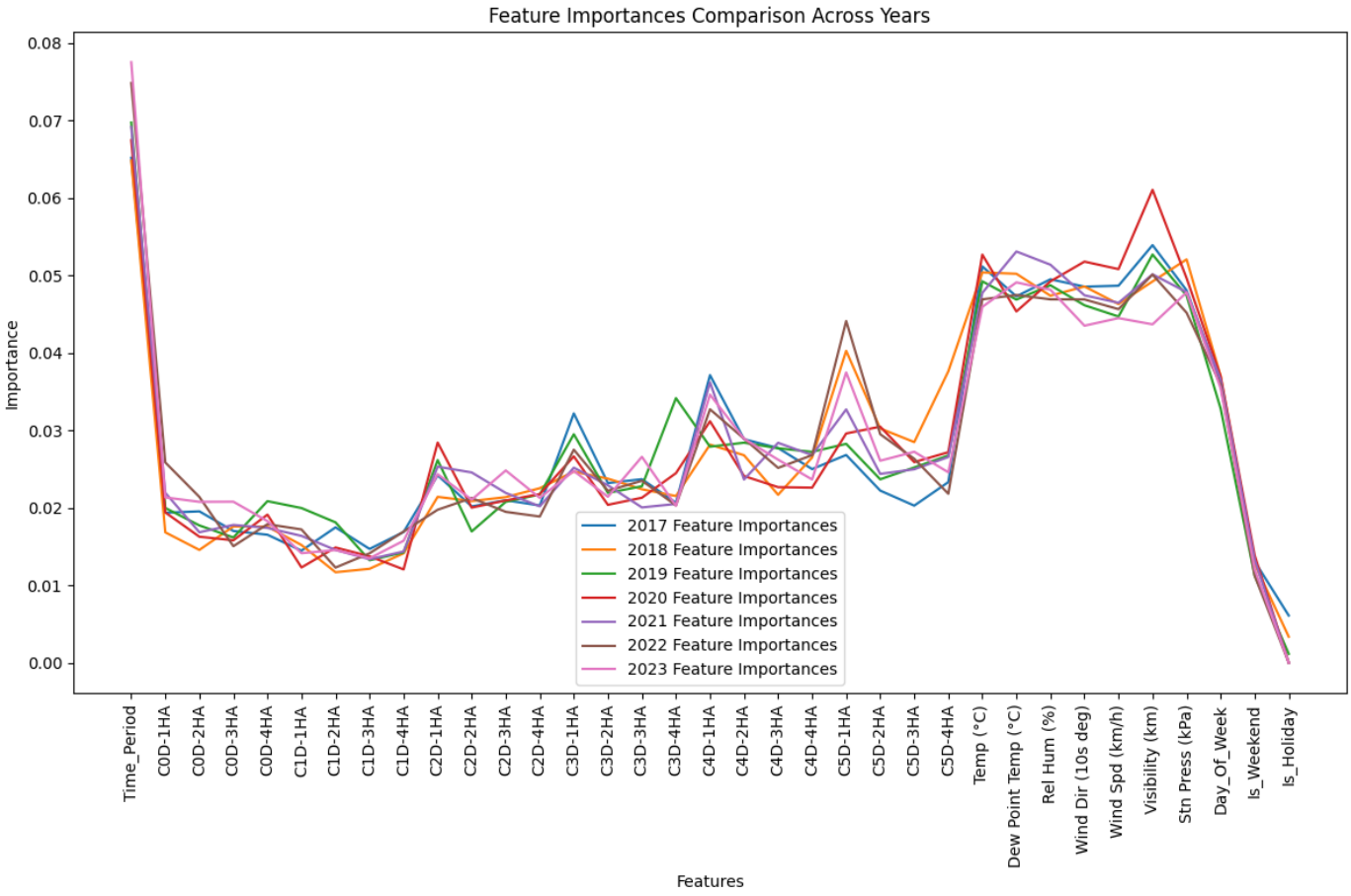

Figure 9: Feature importances comparison across years (2017-2023)

6. Model Selection

The selection of an appropriate machine learning model is a critical step in predicting traffic incident clusters effectively. Given the complexity of the data—comprising temporal, weather, and traffic features—it is essential to evaluate multiple algorithms to identify the one that best balances accuracy, interpretability, and computational efficiency. This section outlines the process of model selection and the rationale behind exploring different algorithms to optimize predictive performance.

- **Objectives**

The primary objectives of the model selection process are:

- **Optimize Predictive Accuracy:** Identify the model that delivers the highest accuracy and reliability in predicting traffic clusters.
- **Address Imbalanced Data**: Ensure the model is robust to class imbalances, which are common in traffic incident data.
- **Enhance Interpretability:** Choose models that provide insights into the key features influencing traffic patterns.
- **Adapt to Complex Data Relationships**: Evaluate models capable of handling non-linear and interactive relationships between variables.
- **Achieve Generalizability**: Ensure the selected model performs consistently on unseen data, avoiding overfitting to the training set.

By meeting these objectives, the selected model will serve as the foundation for deploying an effective traffic prediction system while offering actionable insights for traffic management and safety improvements

6.1. Random Forest Classifier

- **Objective**: The Random Forest Classifier was employed to build a multi-output classification model for predicting traffic clusters (Cluster0 to Cluster5). This method provides robust performance and interpretability through feature importance metrics, allowing for insights into the influence of different features on traffic cluster predictions.
- **Why We Chose the Random Forest Model**
  Random Forest was selected for its robust, ensemble-based learning mechanism, which builds multiple decision trees and averages their outputs. This approach reduces overfitting and variance, making it ideal for complex, imbalanced datasets like traffic and weather data.

Key reasons for choosing Random Forest:

- **Interpretability**: Random Forest provides feature importance scores, helping us understand which variables influence predictions the most.

- **Handling Imbalanced Data**: It can effectively manage imbalanced datasets through hyperparameters like class weight.
- **Versatility:** Random Forest handles numerical and categorical data well, making it suitable for our dataset of weather attributes, temporal variables, and cluster-level traffic information.
- **Robustness:** It is less sensitive to hyperparameter tuning than other models, offering competitive performance across a wide range of settings.

• Hyperparameter Tuning

Hyperparameter tuning is the process of optimizing the parameters that control the learning process of a model. The performance of Random Forest was improved by iteratively testing different hyperparameter configurations using GridSearchCV with 5-fold cross-validation.

We experimented with various ranges of hyperparameters across multiple attempts. Below are the key hyperparameters and the final optimal values identified after these iterations:

- **n_estimators** (Number of trees in the forest):

Tested Range: [80, 100, 120, 135, 200, 500]

Optimal Value: 135

Why Tuned: Increasing the number of trees reduces variance, but excessively large forests increase computation time without significant accuracy gains. The value of 135 was chosen as it struck a balance between performance and computational efficiency.

- **max_depth** (Maximum depth of trees):

Tested Range: [None, 8, 9, 10, 12]

Optimal Value: 10

Why Tuned: A deeper tree captures more complex patterns, but overly deep trees overfit the training data. A depth of 10 provided the best trade-off between complexity and generalization.

- **min_samples_split** (Minimum samples required to split an internal node):

Tested Range: [2, 3, 5, 10]

Optimal Value: 2

Why Tuned: Lower values allow the model to split nodes even with small sample sizes, capturing finer patterns in the data. The value of 2 ensured flexibility without over-complicating the tree structure.

- **min_samples_leaf** (Minimum samples required to be at a leaf node):

Tested Range: [5, 7, 8, 10]

Optimal Value: 8

Why Tuned: Larger values prevent the model from forming overly specific nodes, which reduces overfitting. A value of 8 was found to give the best balance.

- **class_weight:**

Tested Values: [None, 'balanced']

Optimal Value: balanced

Why Tuned: Traffic incident clusters were imbalanced. Setting class_weight to balanced adjusted for this by assigning higher weights to underrepresented classes.

- **random_state:**

Value: 42

Why Tuned: Ensures reproducibility by controlling the randomness in tree building. This allows consistent results across experiments.

• Model Evaluation

The tuned Random Forest model was evaluated on a test set, and the results were as follows:

Best Cross-Validation Train Score: 0.84

Indicates that the model learned effectively from the training data during cross-validation.

Best Cross-Validation Validation Score: 0.67

Suggests that the model generalized well to unseen validation data, with reduced overfitting.

Test-Set F1-Score: 0.72

The test set F1-score was a significant improvement over previous attempt. This metric emphasizes the balance between precision and recall, which is critical for imbalanced datasets.

• Why This Approach Worked

- **Iterative Hyperparameter Tuning:** By testing multiple parameter combinations in successive attempts, we ensured that the final configuration was thoroughly optimized.
- **Cross-Validation:** Using 5-fold cross-validation during tuning ensured robust evaluation, minimizing overfitting while maximizing generalization to unseen data.
- **Balanced Classes:** Setting class_weight='balanced' addressed the issue of imbalanced clusters, ensuring better performance across all categories.
- **Model Flexibility:** Random Forest's ensemble approach effectively captured the relationships between weather, temporal features, and traffic incidents, leading to improved predictions.
- **Feature Relevance:** Carefully selecting relevant features (e.g., weather attributes, temporal variables, and cluster-specific historical features) ensured the model focused on meaningful data.

• Key Insights

- **Importance of Hyperparameter Tuning:** The improvement in F1-score across iterations highlights the importance of fine-tuning hyperparameters to adapt the model to the dataset's characteristics.
- **Impact of Class Balancing:** Addressing class imbalance significantly improved recall and F1-scores for underrepresented clusters.

- **Model Effectiveness:** The final Random Forest model achieved a good balance between accuracy and interpretability, making it suitable for analyzing traffic clusters influenced by weather and temporal variables.

6.2. XGBoost Implementation

- **Overview:** XGBoost (Extreme Gradient Boosting) is an optimized gradient-boosting framework widely used for machine learning tasks, particularly when working with structured data. The strength of XGBoost lies in its scalability, handling of missing data, and regularization techniques, making it highly suitable for classification problems like predicting cluster-specific accident risks.
- **Why We Chose the XGBoost Model**

We chose XGBoost for its ability to:

- Handle imbalanced datasets effectively through features like scale_pos_weight.
- Utilize parallel processing and efficient memory management for faster computation.
- Regularize through hyperparameters like lambda and alpha to avoid overfitting.
- Provide feature importance metrics for feature selection and model interpretability

- **Fine-Tune scale_pos_weight**
  - **Objective**: Address class imbalance in clusters where accident occurrences are sparse.
  - **Hyperparameter Grid:**

n_estimators: [100, 200] — Number of boosting rounds.

max_depth: [3, 5] — Controls the complexity of each decision tree by limiting depth.

learning_rate: [0.1] — Reduces the step size for weight updates to stabilize learning.

subsample: [0.8] — Controls the percentage of samples used in each boosting round.

scale_pos_weight: [1, 5, 10, 20] — Adjusts weight for positive samples to tackle class imbalance.

Implementation: We used GridSearchCV to evaluate different combinations of hyperparameters through 5-fold cross-validation and optimized the f1_weighted scoring metric.

- **Results**:

The optimal hyperparameters achieved after fine-tuning: 1) n_estimators: 100 2)max_depth: 3 3)learning_rate: 0.1 4)subsample: 0.8 5)scale_pos_weight: 10

Best Test-Set F1-Score (Weighted): 0.69

Reasoning: These hyperparameters struck a balance between complexity (max_depth), sample usage (subsample), and handling imbalanced clusters (scale_pos_weight).

- **Add Historical Features**

- **Objective**: Enhance predictive performance by incorporating historical accident data for clusters 0, 1, and 2. The inclusion of features like C0D-1HA, C1D-2HA, etc., provides temporal patterns that might correlate with future accidents.
- **Changes:** Updated the selected_features_updated_with_history list to include historical accident-related features. Retrained the XGBoost model with the expanded feature set using the previously optimized hyperparameters.
- **Results:** After retraining, the model's weighted F1-Score on the test set improved marginally but stayed stable at 0.69. This confirmed that historical features slightly improved the model's ability to generalize.
- **Reasoning:** Historical accident features likely provide marginally predictive signals for future occurrences but are not overly dominant.

- **Evaluate Key Metrics**
  - **Metrics Captured:** 1) Best Cross-Validation Train Score: 1.00 2) Best Cross-Validation Validation Score: 0.67 3) Test-Set F1-Score: 0.69
  - **Interpretation**: 1) High training accuracy suggests the model learns well from the data. 2) A validation score of 0.67 indicates generalization to unseen data during cross-validation. 3) The test-set F1-score of 0.69 highlights the model's reasonable ability to handle imbalanced classes while predicting clusters accurately.
- **Conclusion**
  - **Strengths of XGBoost in This Context:** 1) Robust handling of class imbalance through scale_pos_weight. 2)Ability to handle sparse data efficiently using boosting and regularization. 3) Provides interpretable feature importance metrics.
  - **Limitations Observed:** 1) Marginal improvement from adding historical features suggests diminishing returns beyond a certain feature set. 2) High train scores might indicate overfitting risks mitigated by hyperparameter tuning.
  - **Key Takeaways:** 1) Temporal and historical accident data, coupled with meteorological features, provided the most predictive power. 2) XGBoost performed well in balancing model complexity and handling imbalanced classes, achieving a test F1-score of 0.69.

6.3. CatBoost Implementation

- **Overview:** The CatBoost model was implemented to handle the traffic accident prediction task, focusing on improving performance through hyperparameter tuning. CatBoost, a gradient boosting algorithm, was particularly suitable due to its ability to handle categorical features (though not used in this task) and its robustness against overfitting. The model was trained and tested on the dataset, and a structured process was followed to achieve optimal results.
- **Data Preparation**

The dataset was split into features and target clusters:

- **Features:** Historical traffic accident data and weather variables, including: 1) Weather Features: "Time_Period", "Dew Point Temp (°C)", "Visibility (km)", and "Wind Spd (km/h)". 2) Historical

Accident Data: Cluster-specific accident counts for 1 to 4 hours before the incident, e.g., "C0D-1HA", "C1D-2HA".

- **Targets:** Six clusters (Cluster0 to Cluster5), representing the multi-output classification problem. The data was split into training and testing sets using a 70:30 ratio.

• Model Initialization

The CatBoost model was configured using the MultiOutputClassifier wrapper to support multi-output classification. Initially, a basic setup was used, followed by hyperparameter tuning. Key features of the model included:

- **Handling Class Imbalances:** Using the scale_pos_weight parameter to adjust for imbalanced classes.
- **Early Stopping**: Configured with 20 rounds to prevent overfitting during training

• Hyperparameter Tuning

To achieve optimal performance, GridSearchCV was used to tune hyperparameters. The following parameters were explored: **1) Learning Rate:** Adjusted between 0.01 and 0.1 to control the step size. **2) Depth**: Experimented with tree depths of 6, 7, and 8 to capture feature interactions. **3) Iterations**: Set between 100 and 500 to balance training time and model complexity. **4) Scale Pos Weight**: Tested values between 5 and 20 to address imbalanced data. **5) L2 Leaf Regularization**: Regularization strength tested with values 1, 2, and 3 to avoid overfitting.

• Training and Evaluation

- **Model Training:** The best parameters identified through GridSearchCV were used to train the model. The final parameters were: 1) Learning Rate: 0.025 2) Depth: 7 3) Iterations: 300 4) Scale Pos Weight: 7 5) L2 Leaf Regularization: 2
- **Performance Metrics:** The following metrics were computed: Weighted F1-Score: 0.70 on the test set, demonstrating good performance across all clusters.
  Cluster-Specific Metrics: Cluster0: Precision: 0.45, Recall: 0.82, F1-Score: 0.58.
  Cluster5: Precision: 0.64, Recall: 0.96, F1-Score: 0.76.
  Overall Accuracy: 0.63 on the test set.
- **Key Insights:** 1) The tuned CatBoost model performed robustly despite class imbalance. 2) Early stopping effectively prevented overfitting during training

• Model Testing on New Dataset

The trained model was tested on an unseen dataset to validate its performance. Predictions were made, and the results were analyzed: **1) Overall Accuracy: 0.63. 2) Cluster-Specific Analysis:** Metrics showed consistent performance with the test set, verifying the model's generalizability.

• Conclusion

The CatBoost model demonstrated strong predictive capabilities for the traffic accident prediction task. Its ability to handle imbalanced data, coupled with effective hyperparameter tuning, led to substantial improvements in performance. By saving the trained model, it can be reused for future predictions without

retraining. This approach highlighted the importance of systematic model tuning and validation in achieving optimal results.

7. Analysis and Performance Metrics

- **Overall Performance**

Each model's performance was evaluated using accuracy, precision, recall, and F1-score metrics. These metrics provide a balanced view of the models' capabilities in predicting accident-prone clusters across various time segments.

Model	Overall Accuracy	Overall F1-Score
Random Forest	0.58	0.65
XGBoost	0.62	0.68
CatBoost	0.63	0.70

Figure 10: F1-Score Accuracy Table

CatBoost demonstrated the best overall performance, with the highest F1-score (0.70) and accuracy (0.63), making it the most effective model for this multi-output classification task.

- **Cluster-Wise Metrics**
  - **Random Forest**

Cluster	Accuracy	F1-Score	True Negatives	False Positives	False Negatives	True Positives
Cluster0	0.56	0.62	110	185	18	177
Cluster1	0.57	0.50	180	153	51	104
Cluster2	0.63	0.70	105	126	40	217
Cluster3	0.58	0.67	60	172	18	238
Cluster4	0.62	0.68	120	124	44	200
Cluster5	0.54	0.69	1	204	3	280

Figure 11: Random Forest Accuracy Table

– **XGBoost**

Cluster	Accuracy	F1-Score	True Negatives	False Positives	False Negatives	True Positives
Cluster0	0.60	0.64	112	183	15	180
Cluster1	0.61	0.53	190	143	48	107
Cluster2	0.67	0.72	112	119	35	222
Cluster3	0.60	0.70	61	171	16	240
Cluster4	0.65	0.70	123	121	41	203
Cluster5	0.57	0.71	2	203	2	281

Figure 12: XGBoost Accuracy Table

– **CatBoost**

Cluster	Accuracy	F1-Score	True Negatives	False Positives	False Negatives	True Positives
Cluster0	0.61	0.65	114	179	13	182
Cluster1	0.62	0.54	195	138	47	108
Cluster2	0.69	0.74	115	116	37	220
Cluster3	0.62	0.72	62	170	15	241
Cluster4	0.68	0.72	126	118	39	205
Cluster5	0.58	0.73	2	203	2	281

Figure 13: CatBoost Accuracy Table

- **Cluster Analysis**
  - **Random Forest**
  - **Best Cluster:** Cluster2 with an F1-score of 0.70.
  - **Challenges:** Struggles with precision in Cluster1 due to a high number of false positives.
  - **Overall:** Effective in identifying general trends but struggles with minority class predictions.
  - **XGBoost**
  - **Best Cluster:** Cluster2 with an F1-score of 0.72.
  - **Challenges:** Slightly higher false positives compared to CatBoost, particularly in Cluster3 and Cluster4.
  - **Overall:** Better precision than Random Forest but less robust in handling imbalanced clusters.
  - **CatBoost**

- **Best Cluster:** Cluster2 with the highest F1-score of 0.74.
- **Challenges:** False positives in Cluster5 indicate potential overprediction.
- **Overall:** Strong across all clusters with the best balance of precision and recall.

Weighted and Macro Averages

Model	Weighted Avg F1-Score	Macro Avg F1-Score
Random Forest	0.65	0.64
XGBoost	0.68	0.67
CatBoost	0.70	0.69

Figure 14: Weighted and Macro Averages Table

These scores highlight the overall consistency and robustness of the CatBoost model across all clusters and time segments.

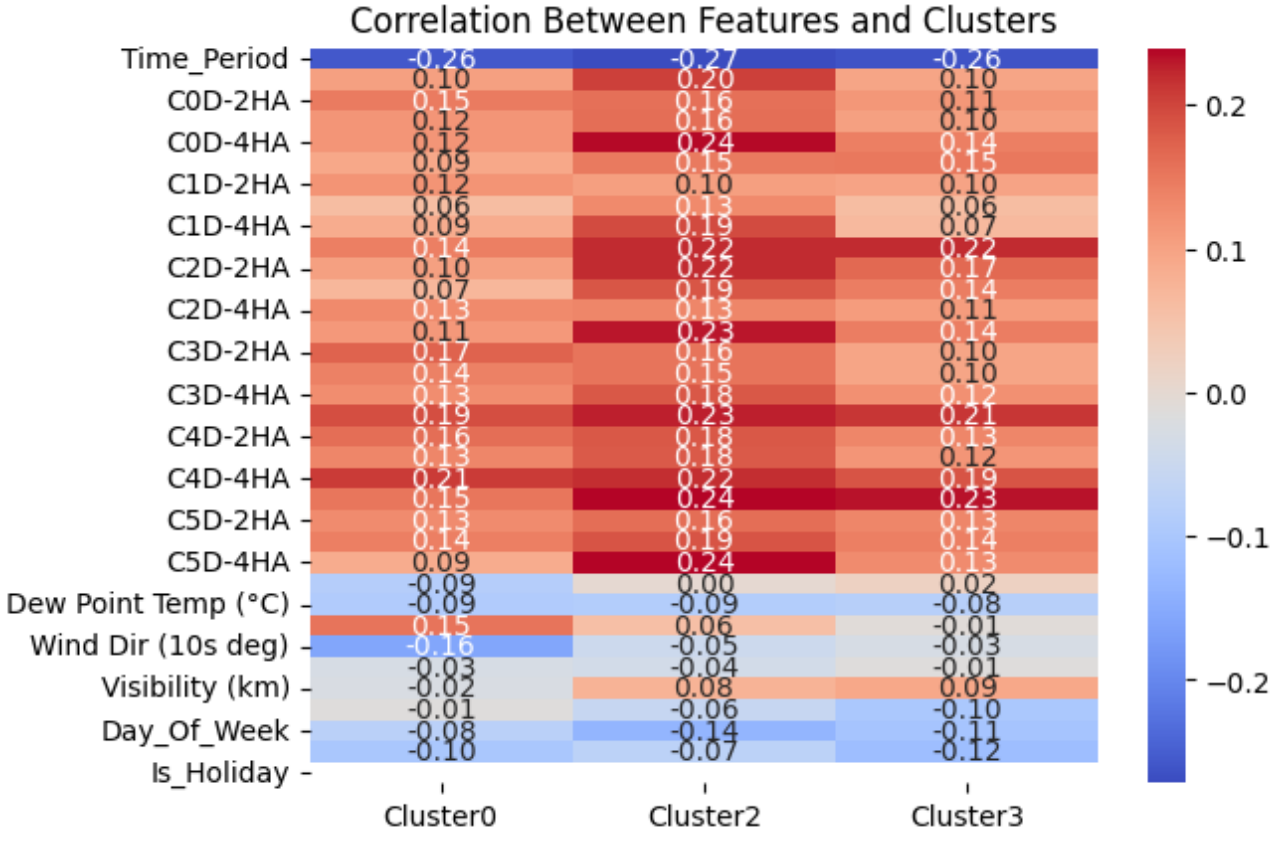

Figure 15: Correlation between features and clusters

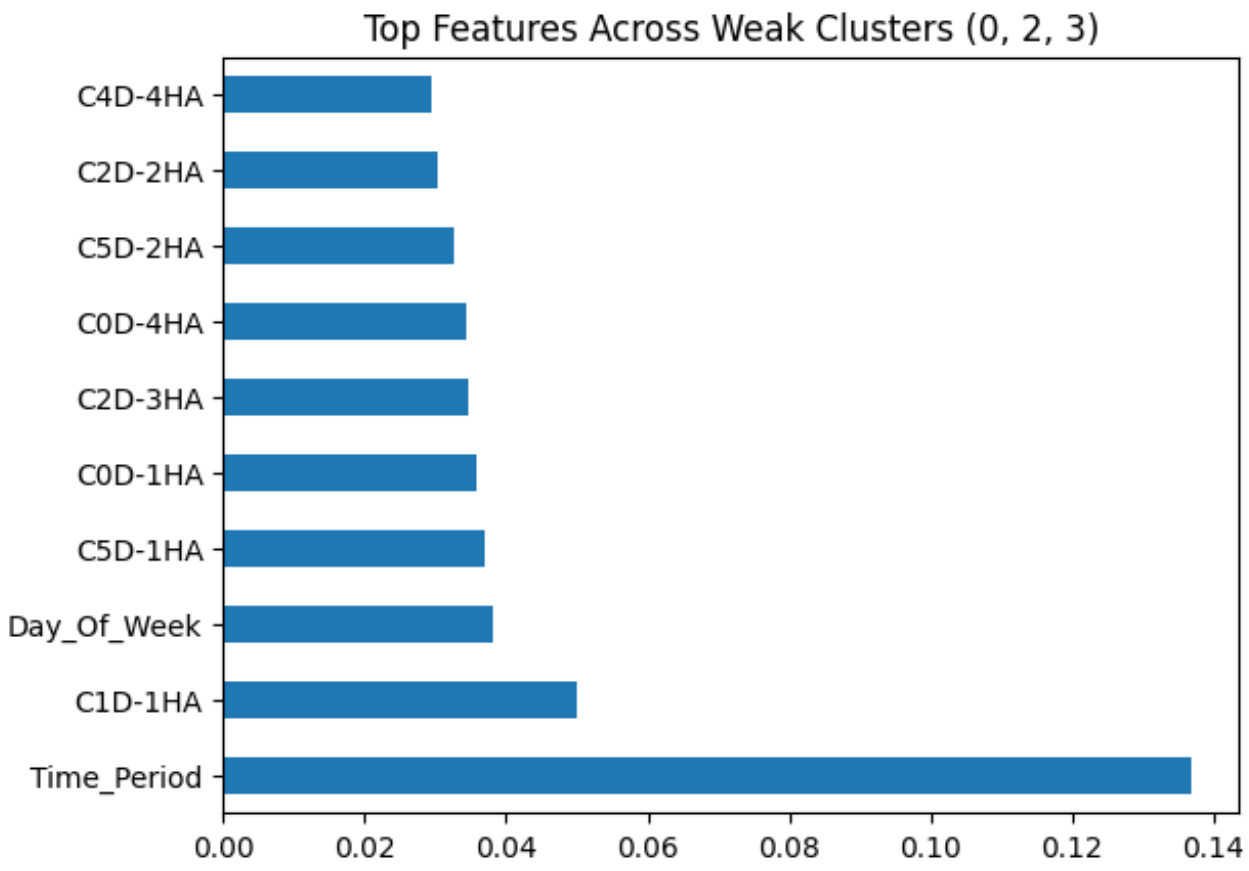

Figure 16: Top features across weak clusters

8. Visualization

The Interactive Map Visualizes:

Geographical Clusters: The visualization identifies six clusters (Cluster0 to Cluster5), representing distinct accident-prone areas in Calgary. These clusters were derived from traffic accident data, enabling focused interventions in high-risk regions.

Time Segments: Predictions are segmented into Morning, Lunch, Evening, and Night time frames, allowing stakeholders to analyze temporal accident risks and patterns.

Risk Indicators: High-risk zones are highlighted using green boxes, which provide a clear and actionable way for stakeholders to prioritize accident prevention measures.

Benefits of Visualization:

User-Friendly: The interactive map is designed to be intuitive for non-technical users, such as traffic managers and policymakers, providing straightforward access to critical accident prediction insights.

Decision Support: By presenting predictions in a segmented and visually engaging manner, the map aids in strategic decision-making, such as deploying resources at specific times and locations.

Sample Visualization Analysis:

Cluster0 (Morning): High risk during morning rush hours due to increased traffic flow. Recommendations: Enhance traffic signal timings and deploy additional traffic officers.

Cluster3 (Evening): Elevated accident risks due to poor visibility and high vehicle volume. Recommendations: Implement streetlight improvements and other visibility-enhancing measures.

Final Visualization Development

The visualization was developed using Folium, an interactive Python library for creating dynamic maps. Key components include:

Base Map: The map incorporates Calgary's geographical layout, providing context for accident-prone areas.

Cluster Markers: Clusters are marked with color-coded symbols to signify their risk levels. High-risk zones are visually distinct for ease of interpretation.

Time Filter: An interactive feature enables users to toggle between Morning, Lunch, Evening, and Night segments, enhancing temporal understanding of accident risks.

Accident Highlights: High-risk areas are emphasized using green-highlighted boxes, signifying regions with elevated accident probabilities.

	True Negatives	False Positives	False Negatives	True Positives
Cluster0	114	179	13	182
Cluster1	195	138	47	108
Cluster2	115	116	37	220
Cluster3	62	170	15	241
Cluster4	126	118	39	205
Cluster5	2	203	2	281

Figure 17: Traffic Accident Predictions (5 Areas: Dec 2023 - Mar 2024)

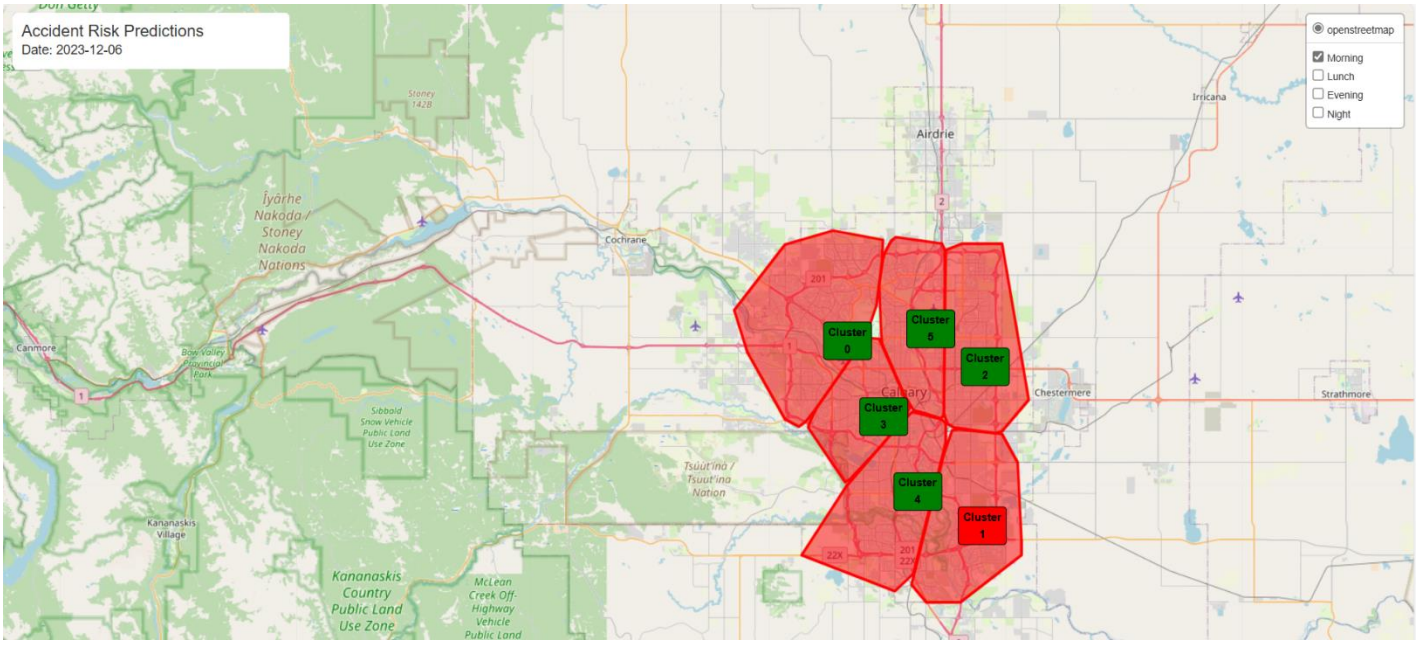

Figure 18: Final Visualization Development

9. Strengths and Weaknesses

By integrating weather, temporal, and historical accident data, this project provided a multidimensional understanding of traffic risks, enhancing predictive accuracy. The interactive map developed as part of this project bridges the gap between technical predictions and practical applications, enabling stakeholders to utilize the results effectively for traffic management and planning. There are some limitations including:

Class Imbalance: Certain clusters (e.g., Cluster 5) exhibited a high degree of class imbalance, negatively affecting precision and contributing to overpredictions in minority classes.

Dependence on Historical Data: The model relies primarily on historical weather and traffic data, which limits its ability to adapt to unforeseen changes in traffic patterns or environmental conditions.

Moderate Predictive Accuracy: While the models performed reasonably well, especially CatBoost, there remains room for improvement in precision and recall for certain clusters, particularly those with less predictive temporal patterns.

- ## Random Forest

Strengths: 1) Provides interpretable feature importance metrics, aiding in understanding the data's influence on predictions. 2) Handles high-dimensional data effectively due to its ensemble-based structure.

Weaknesses: 1) Prone to overfitting when the number of trees or depth is excessively high. 2)Struggles with imbalanced datasets, leading to suboptimal predictions for minority classes.

- **XGBoost**

Strengths: 1) Efficiently processes large datasets due to its parallelized tree-building process. 2) Highly flexible, allowing for extensive hyperparameter tuning to optimize performance.

Weaknesses: 1) Slightly less robust than CatBoost in handling highly imbalanced clusters, particularly in minority-class scenarios.

- **CatBoost**

The CatBoost model's superior ability to handle imbalanced datasets and high-dimensional data proved instrumental in achieving the best results among all tested models.

Strengths: 1) Achieves superior performance in precision and recall, particularly for imbalanced datasets.

2) Automatically handles categorical data effectively without requiring extensive preprocessing.

Weaknesses: 1) Training times are longer compared to XGBoost and Random Forest, especially for large datasets.

10. Conclusions

- **Best Model:**

CatBoost demonstrated the best performance, achieving the highest F1-score (0.70) and accuracy (0.63). Its balanced performance across all clusters makes it the most suitable model for this multi-output classification task.

- **Areas for Improvement:**

1) Threshold Optimization: Adjust thresholds for underperforming clusters, such as Cluster1 and Cluster5, to minimize false positives and improve precision. **2)Feature Expansion:** Integrate additional temporal and environmental variables to enhance the predictive power for clusters with weaker performance.

The goal of this project was to predict traffic accidents using a multi-output classification approach that integrated weather, temporal, and traffic data. Three models—Random Forest, XGBoost, and CatBoost—were evaluated for their ability to identify accident-prone clusters. Among these, CatBoost emerged as the best-performing model, offering robust handling of imbalanced data and delivering superior accuracy and F1-scores. The following conclusions summarize the project outcomes:

- **Key Findings:**

Model Effectiveness:

CatBoost achieved the highest overall accuracy (0.63) and F1-score (0.70), demonstrating robust performance in handling imbalanced datasets.

XGBoost provided competitive performance, with an F1-score of 0.68, but showed slight limitations in precision and recall for specific clusters.

Random Forest served as a reliable baseline with an F1-score of 0.65, but its performance was less effective in managing false positives and negatives for minority clusters

- ## Cluster-Level Insights:

Balanced Clusters: Clusters with well-distributed data, such as Cluster2 and Cluster4, demonstrated the best predictive performance.

Imbalanced Clusters: Clusters with high-class imbalance, such as Cluster5, exhibited excellent recall but struggled with overprediction, leading to lower precision.

- ## Impact of Features:

Temporal Features: Key variables like Time Period and Day Of Week significantly enhanced model predictions, particularly for clusters exhibiting distinct temporal patterns.

Weather Features: Variables such as visibility and wind speed played a pivotal role in identifying accident-prone scenarios, proving critical for clusters impacted by environmental conditions.

Practical Implications:

The project provided an interactive visualization tool offering actionable insights for urban planners, traffic managers, and policymakers. This tool enables targeted interventions by highlighting high-risk zones during specific times of the day, improving traffic safety and resource allocation.

❖ **We have 12 files for the code, with the first 3 related to data cleaning and the remaining files for modeling. These are uploaded on GitHub.**

11. Future Work

Feature Expansion: Incorporate real-time data sources, such as traffic flow information, road infrastructure conditions, and additional weather metrics (e.g., precipitation intensity), to enhance the model's predictive capabilities.

Advanced Techniques: Explore deep learning architectures, such as convolutional neural networks (CNNs), for spatial and temporal feature extraction. Investigate ensemble methods to combine predictions from multiple algorithms for better generalization.

Threshold Optimization: Implement dynamic thresholding techniques tailored to specific clusters to minimize false positives and better balance precision and recall.

Deployment: Develop a real-time prediction system capable of dynamically updating predictions based on live data feeds, enabling proactive traffic management and incident prevention.

This project underscores the potential of machine learning in addressing real-world challenges in urban traffic management. The methods and insights developed here provide a robust foundation for future research and practical implementation.

12. References

[1] "Station Results - Historical Data - Climate - Environment and Climate Change Canada," [Online]. Available: https://climate.weather.gc.ca/historical_data/search_historic_data_e.html. [Accessed 11 Oct. 2024].

[2] "Building Permits | Open Calgary," [Online]. Available: https://data.calgary.ca/Business-and-Economic-Activity/Building-Permits/c2es-76ed/about_data. [Accessed 11 Oct. 2024].

[3] "Street Centreline," [Online]. Available: https://data.calgary.ca/Transportation-Transit/Street-Centreline/4dx8-rtm5/about_data. [Accessed 11 Oct. 2024].

[4] "Traffic Incidents | Open Calgary," [Online]. Available: https://data.calgary.ca/Transportation-Transit/Traffic-Incidents/35ra-9556/about_data. [Accessed 11 Oct. 2024].

[5] "GitHub - ZohrehMejrisazanoosi/engg680_2024_fall," [Online]. Available: https://github.com/ZohrehMejrisazanoosi/engg680_2024_fall.git [Accessed 11 Oct. 2024].

[6] "GitHub - ZohrehMejrisazanoosi/ENGG680-Course-Project," [Online]. Available: https://github.com/ZohrehMejrisazanoosi/ENGG680-Course-Project.git [Accessed 11 Oct. 2024].

[7] F. S. A. P. B. H. A. Shahla, "Analysis of Transit Safety at Signalized Intersections in Toronto, Ontario, Canada," Transportation Research Record: Journal of the Transportation Research Board, vol. 2102, pp. 108-114, 2009.

[8] W. G. Y. L. C. K. T.-H. T. a. X. B. Y. H. Zhao, "Prediction of Traffic Incident Duration Using Clustering-Based Ensemble Learning Method," Journal of Transportation Engineering, Part A: Systems, vol. 148, July 2022.

[9] B. W. a. A. H. J. Evans, "Evolution and Future of Urban Road Incident Detection Algorithms," Journal of Transportation Engineering, Part A: Systems, vol. 146, June 2020.

Group Contract:

1. **Communication**:

What is the primary way of contacting each other remotely?
Our main modes of communication will be email and text messaging (via WhatsApp).

2. **Response Time**:

How quickly should group members respond?
Group members are expected to respond within a few hours. As deadlines approach, quicker responses are anticipated. Delays are acceptable in cases of illness, emergencies, or work commitments, provided these are communicated to the group in advance.

3. **Meetings**:

What are the preferred days and locations for meetings, and how will scheduling be decided?
In-person meetings will primarily take place before, during, or after class lectures. Additional in-person meetings will be arranged through the group chat based on members' availability. Online meetings will be conducted using Microsoft Teams.

4. **Division of Labor**:

How will tasks be distributed fairly and collaboratively?
Tasks will be assigned based on each member's skillset. If someone struggles with their task, more experienced members will provide guidance and resources. All members are expected to put in effort regardless of skill level. Any difficulties completing tasks should be communicated to the team well in advance of the deadline.

5. **Accountability**:

What are the expectations for attendance, punctuality, participation, preparedness, task completion, and communication?
All members are expected to attend meetings, with exceptions for special circumstances communicated beforehand. Members should arrive within 15 minutes of the scheduled start time and notify the team of delays. Tasks must be completed on time, and issues should be raised early. During meetings, active participation is expected, and meeting hosts should encourage involvement from everyone. Feedback on team members' work should be constructive and respectful.

6. **Decision-Making**:

How will the team make key decisions?
Key decisions will be made collaboratively after group discussions. Efforts will be made to incorporate all members' ideas. A combination of majority voting and consensus will be used, ensuring everyone has an opportunity to share and explain their perspectives.

7. Conflict Resolution:

What happens if a team member violates the agreement, or their work does not meet expectations?
The team will address issues through open and respectful discussions to find solutions and a path forward. Members are encouraged to raise concerns early to prevent missed deadlines and ensure tasks are completed to a high standard. Persistent issues after multiple discussions may require escalation, but the team will prioritize resolving conflicts internally.