When Roads Speak : Unveiling the Pattern of Road Accidents in India

Madhurima Chatterjee (University of Calcutta)
Nakshatra Pal (Maulana Abul Kalam Azad University of Technology)
Liton Bhattacharjee (University of Calcutta)
Urmi Kanrar (University of Calcutta)
Suman Bag (University of Calcutta)

Project Guide / Mentor Name: Surojit Mukherjee

Period of Internship: 19th May 2025 – 15th July 2025

Report submitted to: **IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata**

Contents

CERTIFICATE

This is to certify that the dissertation report entitled **'When Roads Speak : Unveiling the Pattern of Road Accidents in India'**, submitted by Madhurima Chatterjee, Nakshatra Pal, Liton Bhattacharjee, Urmi Kanrar, Suman Bag, is an account of the project work completed by IDEAS - TIH, ISI KOLKATA, under my supervision and direction, and it is deserving of consideration for the institute's award of their internship.

14/7/25

Dr. Surojit Mukherjee
Supervisor
Visiting Professor
VGSoM, IIT Kharagpur

Visiting Professor
Vinod Gupta School of Management
Indian Institute of Technology Kharagpur
Kharagpur- 721302, West Bengal

Dr. Agnimitra Biswas
CEO
IDEAS - TIH, ISI KOLKATA

Acknowledgement

Our sincere appreciation is out to **Dr. Agnimitra Biswas, CEO, IDEAS - TIH, ISI KOLKATA** for his invaluable assistance in the execution of our research, **"When Roads Speak : Unveiling the Pattern of Road Accidents in India"**.

Our mentor, **Dr. Surojit Mukherjee**, deserves special recognition for the time and work he put in during the semester. We greatly benefited from his insightful counsel and recommendations as we completed the project. We will always be thankful to him for this.

Abstract

Road traffic accidents remain one of the most pressing yet under-addressed public health emergencies of the modern age. Their severity often culminating in fatalities or lifelong disabilities, not only devastates families but also imposes immense economic and infrastructural strain. In India, the staggering annual toll of over 150,000 deaths from road accidents highlights a systemic crisis that demands urgent, data-driven intervention. The study of accident severity is thus not merely statistical analysis, it is a societal imperative with direct implications for policymaking, emergency response, and sustainable mobility.

This project is of critical significance as it attempts to unravel the complex interplay of environmental, vehicular, temporal, and human factors that contribute to road accident severity across Indian states. By synthesizing heterogeneous datasets spanning from 2018 to 2023, sourced from government portals and state authorities, we construct a high-fidelity analytical pipeline that integrates both traditional machine learning models (Logistic Regression, SVM, Random Forest, XGBoost) and advanced deep learning architectures (MLP, GRU). Moreover, we bridge the gap between classification and time-series forecasting through SARIMA and neural network-based temporal models, enabling predictive insights at both granular (daily) and aggregate (monthly) levels.

The outcomes underscore the disproportionate risk associated with poor weather conditions, unsafe road infrastructure, and alcohol involvement, while also exposing the limitations of standard classifiers in handling class imbalance and temporal volatility. Our study not only advances the methodological frontier in accident analytics but also provides a strategic foundation for actionable, evidence-based road safety reforms.

1 Introduction

Road accidents are not merely unfortunate events, they are a persistent and preventable global crisis that inflicts profound human, social, and economic costs. In India, the alarming frequency and severity of such incidents have positioned road safety as a critical public health and policy concern. With over 150,000 fatalities recorded annually, the nation faces an urgent need for intelligent, data-driven strategies to understand and mitigate the factors that contribute to accident severity.

To address this, our study employs a dual-layered analytical framework integrating supervised and deep learning and time series forecasting. We begin with extensive preprocessing and feature engineering on a **multi-source dataset (2018–2023)**, capturing granular accident-level attributes across temporal, geographic, infrastructural, vehicular, and human dimensions. For predictive classification, we have implemented **Logistic Regression, Random Forest, Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGBoost)** to model accident severity, rigorously evaluated via metrics such as **F1-score, confusion matrices, and ROC-AUC**. Recognizing the volatility in daily data, we have extended our analysis to monthly aggregation and applied advanced forecasting models including **Seasonal ARIMA (SARIMA), Multi-Layer Perceptron (MLP), and Gated Recurrent Unit (GRU)** networks to capture temporal dependencies and generate forward-looking accident trends. This methodological synthesis not only enhances predictive robustness but also facilitates deeper insight into the dynamic nature of road safety in India.

About Internship :

As part of the **IDEAS-TIH, ISI Kolkata Summer Internship Program 2025**, we were fortunate to receive an intensive series of technical training sessions in the early weeks of the program. The carefully structured modules covered a wide array of foundational and advanced concepts, including **python programming, data visualization with Power BI, seaborn and matplotlib, exploratory data analysis, hypothesis testing, supervised classification methods, sentiment analysis, generative artificial intelligence, large language model and as well as survey questionnaire designing**. Additionally, **expert sessions on industry trends, model evaluation metrics, and effective communication of insights** provided us with a well-rounded perspective on practical data science workflows.

These sessions played a pivotal role in equipping us with the technical fluency and analytical thinking required to undertake our project titled **"When Roads Speak: Unveiling the Pattern of Road Accidents in India"**. We applied our knowledge of Python for data preprocessing and wrangling, utilized visualization tools to explore accident trends, and implemented machine learning algorithms for predictive modelling. Furthermore, the training in statistical testing and time series forecasting enabled us to explore accident frequency trends using time series and deep learning models. The foundational instruction we received empowered us not only to execute technically sound methodologies but also to interpret results meaningfully thereby ensuring our project aligned with real-world impact and analytical rigor.

2 Literature Survey

- Dombalyan et al.[1] (2017) developed a traffic forecasting model for Russia's M-4 "Don" highway, integrating AIMSUN-based simulations with entropy models to predict workplace, business, recreational, and freight trips. The model emphasizes realistic driver behavior and road network dynamics to assess toll impacts, guide infrastructure planning, and reduce economic risks. It stands out for its structured methodology, high accuracy, and effectiveness in complex transport systems.
- Cuenca et al.[2] presented a case study of traffic accidents classification and severity prediction in Spain (2011-2015). The authors used different statistical and predictive machine learning algorithms through comparisons on the different datasets to classify, evaluate and predict the crash severity of road traffic accidents. To this end, are compared three different machine learning classification techniques, such as Gradient Boosting Trees, Deep Learning and Naïve Bayes.
- Khanum et al.[3] (2023) proposed a random forest-based model to predict traffic accident severity on Indian highways. They used data from NHAI Concessionaires for two major projects: Pune-Solapur and the Barwa-Adda-Panagarh Section of NH-2 (km 398.240 to km 521.120), including the Panagarh Bypass in Jharkhand and West Bengal. The study followed a multi-step process involving data collection, feature selection, model training, parameter tuning, and evaluation using accuracy and F1 score.
- Fang Zong et al.[4] (2013) predicted road accident severity and duration using real data from Jilin province, China. They applied the Ordered Probit model for severity (deaths, injuries, property damage) and the Hazard/AFT model for duration. Key influencing factors included vehicle type, accident time, weather, road conditions, and emergency response. The Ordered Probit model outperformed SVM in accuracy. Findings aid emergency response and traffic planning, supporting quicker decisions and improved safety policies.
- Hayakawa et al.[5] (2000) conducted a comparative study on traffic accident risk perception in Japan and the U.S. They found Japanese pedestrians and cyclists face higher fatality rates, making Japanese drivers more fearful and inclined to buy insurance out of social responsibility. In contrast, Americans, mostly car users, focus on financial protection. The study highlights how cultural values and real accident conditions influence how people perceive and respond to traffic risks.
- Rahim et al.[6] (2021) proposed a novel deep learning approach with a customized F1-loss function to predict traffic crash severity. Using Louisiana work zone crash data (2014–2018), it transforms road, vehicle, and human-related features into images via t-SNE and convex hull algorithms. A CNN model is then trained to optimize precision and recall directly. Compared to traditional machine learning models, the deep learning method showed improved performance in predicting fatal and injury crashes, supporting better traffic safety and congestion management.

3 Uniquness of the Paper

The primary objective of this project is to comprehensively analyze and predict the patterns and severity of road accidents across India using advanced data science techniques,as road accidents increasing now-a-days for not to maintain the rules and regulations and other risk factors. Predicting road accidents is crucial as it enables proactive safety measures, faster emergency response, and data-driven policymaking. It raises public awareness, promotes safer driving, and reduces economic losses. This project supports a safer transport system and helps minimize the impact of road accidents in India.

By integrating heterogeneous datasets spanning 2018 to 2023, the study aims to: **Identify Key Factors**: Uncover the critical environmental, vehicular, temporal, and human factors, time that contribute to the severity of road accidents in various Indian states.

Predict Accident Severity: Develop and evaluate predictive models—including Logistic Regression, Random Forest, Support Vector Machine (SVM), and XGBoost—to accurately classify the severity of road accidents.

Forecast Accident Trends: Employ time series forecasting methods, such as SARIMA, Multi-Layer Perceptron (MLP), and Gated Recurrent Unit (GRU) networks, to model and predict future accident trends at monthly level.

Enable Evidence-Based Policy: Provide actionable insights and a methodological foundation for policymakers and stakeholders to design data-driven road safety interventions and reforms.

Through this multi-layered analytical approach, the project seeks to advance the methodological frontier in accident analytics and contribute to the reduction of accident-related fatalities and injuries in India.

4 Methodology

4.1 Data Description

The India Road Accident Dataset provides a comprehensive view of road accidents across various states in India. The dataset inludes 3000 accident records spanning from 2018 to 2023, with detailed attributes such as accident severity, weather conditions, road type, alcohol involvement, causalties, and more. This dataset seeks to shed light on the road accidents across all states.

- **Source :** Synthesized from multiple real world datasets, including the Ministry of Road Transport and Highways (MoRTH) reports, Open Government Data (OGD) India, and accident statistics from various Indian states.

- **Link :-**
  (a) Daily database :`https://docs.google.com/spreadsheets/d/1jIaPA7Ah79oEC9AF593RBI`
  `editusp=drive_link&ouid=100772017413046891413&rtpof=true&sd=true`
  (b) Monthly database :`https://drive.google.com/file/d/1TZDyal-DgCyKDjjHBRZfoNXq8X`
  `view?usp=drive_link`

- **Columns :-**
  (a) Daily database : (i) State Name – Name of the Indian state where the accident occurred (e.g., West Bengal, Sikkim, Goa etc)

(ii) Year – Year of the accident occurred (2018-2023)
(iii) Month – Month when the accident happened (January - December)
(iv) Day of the week – Different days of week (ex: Monday, Tuesday, Wednesday, etc)
(v) Time of the day – Exact time of the accident (HH:MM format where H means hours , M means minute)
(vi) Accident Severity – Categorized as fatal, serious or minor
(vii) Number of Vehicles involved – Total number of vehicles in the accident
(viii) Vehicle type involved – Different types of vehicles involved in the accident (ex: Car, Truck, Two wheeler, etc)
(ix) Number of Casualties – Number of persons died/injured
(x) Number of Fatalities – Number of persons died
(xi) Weather Conditions – Categorized into Rainy, Stormy, Hazy, Foggy, Clear
(xii) Road type – Categorized into National Highway, Urban Road, State Highway, Village Road
(xiii) Road conditions – Categorized into Wet, Dry, Under Construction, Damaged
(xiv) Lighting Conditions – Categorized into Dark, Dusk, Dawn, Daylight
(xv) Traffic Control present – Categorized into Signs, Signals, Police Check-post, None
(xvi) Speed Limit(km/h) – Speed limit at accident locations
(xvii) Driver Age – age of the driver involving in accident
(xviii) Driver Gender – Categorized into Male, Female
(xix) Driver License Status – Categorized into Valid, Expired, None
(xx) Alcohol involvement – Categorized into yes, No
(xxi) Accident Location Details – Categorized into Curve, Straight Road, Bridge, Intersection
float

Table I: Data head of daily accident data

State Name	Year	Month	Day of Week	Time_of_Day	Accident Severity	...	Driver License Status	Alcohol Involvement	Location Details
Jammu and Kashmir	2018	January	Monday	16.0432	Minor	...	Expired	Yes	Straight Road
Gujarat	2018	January	Friday	8.9108	Minor	...	None	Yes	Intersection
Uttarakhand	2018	January	Wednesday	5.5727	Minor	...	Expired	Yes	Intersection
Nagaland	2018	January	Tuesday	13.1452	Fatal	...	Expired	Yes	Curve
Meghalaya	2018	January	Thursday	0.6961	Serious	...	None	No	Curve
					...				
Maharashtra	2023	December	Sunday	6.909	Fatal	...	Expired	Yes	Intersection
Tamil Nadu	2023	December	Sunday	23.7569	Serious	...	Valid	No	Straight Road
Nagaland	2023	December	Wednesday	6:41	Minor	...	Expired	Yes	Intersection

(b) Monthly Database : (i) State Name – Name of the Indian state where the accident occurred (e.g., West Bengal, Sikkim, Goa etc)
(ii) Year – Year of the accident occurred (2018-2023)
(iii) Month – Month when the accident happened (January - December)
(iv) Timestamp – Year, date, time combined together (ex: 2018-04-07023:35:45.240')
(v) Day of the week – Different days of week (ex: Monday, Tuesday, Wednesday, etc)
(vi) Time of the day – Exact time of the accident (HH:MM format where H means hours , M means minute)
(vii) Accident Severity – Categorized as fatal, serious or minor
(viii) Number of Vehicles involved – Total number of vehicles in the accident
(ix) Vehicle type involved – Different types of vehicles involved in the accident (ex:

Car, Truck, Two wheeler, etc)
(x) Number of Casualties – Number of vehicles included.
(xi) Number of Fatalities – Number of persons died
(xii) Weather Conditions – Categorized into Rainy, Stormy, Hazy, Foggy, Clear
(xii) Road type – Categorized into National Highway, Urban Road, State Highway, Village Road
(xiii) Road conditions – Categorized into Wet, Dry, Under Construction, Damaged
(xiv) Lighting Conditions – Categorized into Dark, Dusk, Dawn, Daylight
(xv) Traffic Control present – Categorized into Signs, Signals, Police Check-post, None
(xvi) Speed Limit(km/h) – Speed limit at accident locations
(xvii) Driver Age – age of the driver involving in accident
(xviii) Driver Gender – Categorized into Male, Female
(xix) Driver License Status – Categorized into Valid, Expired, None
(xx) Alcohol involvement – Categorized into yes, No
(xxi) Accident Location Details – Categorized into Curve, Straight Road, Bridge, Intersection
(xxii) Month-num – set of numbers from 1-12 where 1: January, 2: February, ..., 12: December symbolizing the month of the accident (ex: [2,2] means two accidents in February)
(xxiii) Weekday-num – set Numbers from 0-6 where 0: Sunday, 1: Monday,.....,6: Saturday symbolizing the weekday the accident occurred (ex: [0,0,2] means two accidents on Sunday and one accident on Tuesday)
(xxiv) Time in Seconds – time of accident converted into seconds(originally was in hours and minutes)
(xxv) Total Accidents – total accidents in a month corresponding to each state

Table II: Data head of monthly accident data

State Name	Year	Month	timestamp	...	Time_in_seconds	Total_Accidents
Andhra Pradesh	2018	April	['2018-04-07 23:35:45.240']	...	[84945.24]	1
Andhra Pradesh	2018	December	['2018-12-01 06:43:26.400', ...]	...	[24206.4, 57453.48, 2909.16]	3
Andhra Pradesh	2018	February	['2018-02-02 15:25:08.040', ...]	...	[55508.04, 64351.08]	2
Andhra Pradesh	2018	January	['2018-01-03 12:12:32.400']	...	[43952.4]	1
Andhra Pradesh	2018	June	['2018-06-04 16:43:31.800', ...]	...	[60211.8, 62665.92, 1786.32]	3
...						
West Bengal	2023	August	['2023-03-01 09:08:59.280']	...	[32939.28]	1
West Bengal	2023	May	['2023-05-02 10:22:11.640', ...]	...	[37331.64, 44283.24, 64189.44]	3
West Bengal	2023	November	['2023-11-02 05:00:26.280', ...]	...	[18026.28, 59673.6]	2

4.2 Flowcharts of Analytical Works

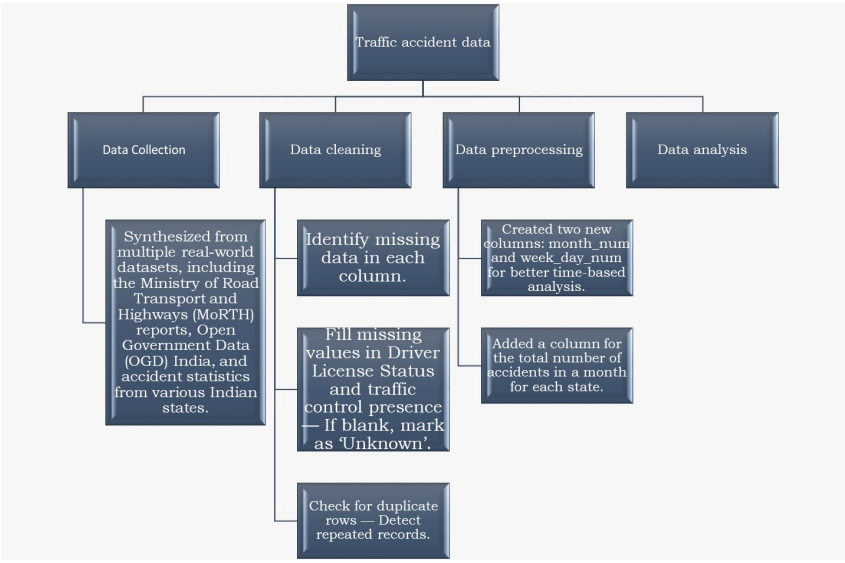


Figure 1: Flowchart of data collection and analysis overview

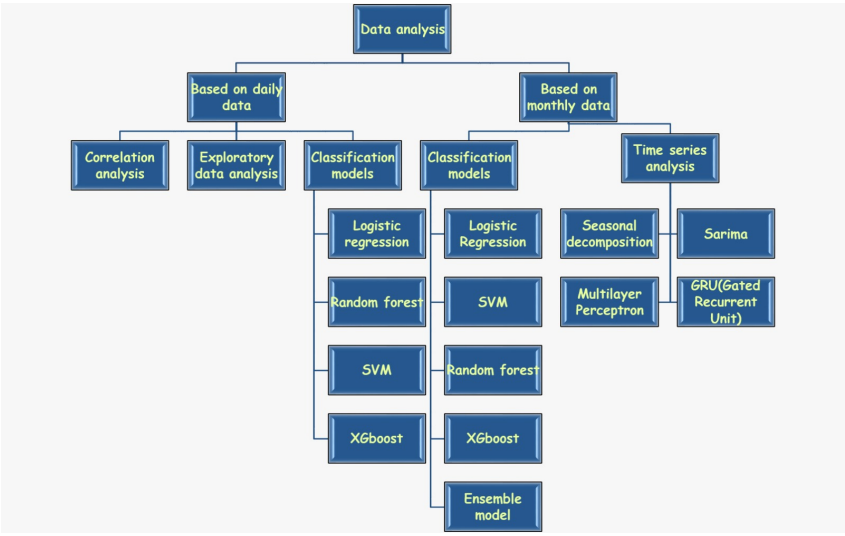


Figure 2: Flowchart of deeper data analysis

4.3 Predictive Modelling

Predictive modelling (by supervised learning algorithms) is the process of building a model that learns from labeled previous data to predict outcomes for new and unseen data. In this approach, the model is trained on input-output pairs, where the input features are used to predict a known target variable.

4.3.1 Logistic Regression

Logistic Regression is a supervised machine learning algorithm used for classification problems. It predicts the likelihood that an input belongs to a particular class, as opposed to

linear regression, which predicts continuous values. When it comes to binary or multiclass categorization, the output can fall into one of two or more groups. It transforms inputs into a probability value between 0 and 1 using the sigmoid function.

Logistic regression can be classified into three main types based on the nature of the dependent variable. Types are binomial, multinomial and ordinal logistic regression.

- **Assumptions :** Understanding the assumptions behind logistic regression is important to ensure the model is applied correctly, main assumptions are —

  i) Independent observations: Each data point is assumed to be independent of the others means there should be no correlation or dependence between the input samples.
  ii) Binary dependent variables: It takes the assumption that the dependent variable must be binary, means it can take only two values. For more than two categories SoftMax functions are used.
  iii) Linearity relationship between independent variables and log odds: The model assumes a linear relationship between the independent variables and the log odds of the dependent variable which means the predictors affect the log odds in a linear way.
  iv) No outliers: The dataset should not contain extreme outliers as they can distort the estimation of the logistic regression coefficients.
  v) Large sample size: It requires a sufficiently large sample size to produce reliable and stable results.

- **Sigmoid function :** A crucial component of logistic regression, the sigmoid function transforms the model's raw output into a probability value between 0 and 1. This function creates a "S"-shaped curve known as the logistic curve or sigmoid curve by mapping any real integer into the range 0 to 1. The sigmoid function is ideal for this since probabilities must fall between 0 and 1. To determine the class label in logistic regression, we often use a threshold value of 0.5. The input is categorized as Class 1 if the sigmoid output is equal to or greater than the threshold. The input is categorized as Class 0 if it is below the threshold.

- **Working procedure :** Suppose we have input features represented as a matrix,

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

  and the dependent variable is **Y** having only binary value i.e. 0 or 1.

$$Y = \begin{cases} 0 & \text{if } \textit{Class 1} \\ 1 & \text{if } \textit{Class 2} \end{cases}$$

  then, apply the multi-linear function to the input variables X.

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

Here x_i is the i^{th} observation of X, $w_i = [w_1, w_2, w_3, \cdots, w_m]$ is the weights or Coefficient and b is the bias term also known as intercept. Simply this can be represented as the dot product of weight and bias.

$$z = \mathbf{w} \cdot \mathbf{X} + b$$

At this stage, z is a continuous value from the linear regression. Logistic regression then applies the sigmoid function to z to convert it into a probability between 0 and 1 which can be used to predict the class.
Now, we use the sigmoid function where the input will be z and we find the probability between 0 and 1. i.e. predicted y.
The sigmoid function is given by,

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Here, $\sigma(z)$ tends towards 1 as $z \to \infty$ and tends towards 0 $z \to -\infty$
where the probability of being a class can be measured as:
$P(y = 1) = \sigma(z)$ and $P(y = 0) = 1 - \sigma(z)$

4.3.2 Random Forest

Random Forest is an ensemble learning method used for classification and regression. It builds multiple decision trees during training and outputs the majority class (for classification) or the average prediction (for regression). It is reliable, effective with big datasets, and compatible with both categorical and numerical data.

- **Mathematical Representation :**

  Training Dataset
  Let the training data be:

  $$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

  Where,
  $x_i \in \mathbb{R}^d$ = input feature vector
  $y_i \in \{1, 2, \ldots, K\}$ = class label (for K classes)
  n = number of training examples

  Bootstrap Sampling
  We construct T decision trees, each trained on a bootstrap sample:

  $$\mathcal{D}_t = \left\{ \left(x_1^{(t)}, y_1^{(t)} \right), \ldots, \left(x_n^{(t)}, y_n^{(t)} \right) \right\}, \quad t = 1, \ldots, T$$

  Each $\mathcal{D}_t$ is formed by sampling n instances with replacement from $\mathcal{D}$.

  Tree Construction (Random Feature Selection)

  For each bootstrap sample $\mathcal{D}_t$:
  1) A decision tree $h_t(x)$ is grown recursively.
  2) At each internal node, randomly choose a subset of features $\mathcal{F}_t \subset \{1, \ldots, d\}$, of size $m \ll d$. Then choose the best split over $\mathcal{F}_t$ based on impurity criteria like:
  a) Gini impurity:

  $$G(S) = 1 - \sum_{k=1}^{K} p_k^2$$

b) Entropy:

$$H(S) = -\sum_{k=1}^{K} p_k \log p_k$$

The split is chosen to minimize the impurity of child nodes.

Output of Each Tree
Each tree gives a prediction function:

$$h_t(x) : \mathbb{R}^d \to \{1, \ldots, K\}$$

This is a hard classification (class label as output).

Aggregating Tree Outputs (Majority Voting)
The final prediction $\hat{y}$ for input x is based on the majority vote among T trees:

$$\hat{y} = \arg\max_{k \in \{1,\ldots,K\}} \sum_{t=1}^{T} \mathbf{1}(h_t(x) = k)$$

Where,
$\mathbf{1}(\cdot)$ is the indicator function:

$$\mathbf{1}(h_t(x) = k) = \begin{cases} 1, & \text{if } h_t(x) = k \\ 0, & \text{otherwise} \end{cases}$$

4.3.3 Support Vector Machine (SVM)

SVM is a classification and regression supervised learning technique. By maximizing the margin between classes, it determines the optimal hyperplane for class separation. Support vectors are the points that are closest to the hyperplane. SVM can handle non-linear data using kernel functions and performs well for binary classification (such as spam vs. non-spam).

- **Mathematical formulation :**

  Problem Setup
  Given a dataset

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, +1\}$$

$$w^T x + b = 0$$

  Functional Margin
  For a data point (x_i, y_i), the functional margin is,

$$\gamma_i = y_i(w^T x_i + b)$$

  To correctly classify all points:

$$y_i(w^T x_i + b) \geq 1 \quad \forall i$$

Optimization problem (Primal form)
We aim to maximize the margin, which is equivalent to minimizing $\|w\|^2$. So, the primal optimization problem is,

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1 \quad \forall i$$

This is a convex quadratic optimization problem with linear constraints.

Lagrangian formulation
We form the Lagrangian with multipliers $\alpha_i \geq 0$:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$$

Karush-Kuhn-Tucker (KKT) conditions
We minimize $\mathcal{L}$ with respect to w and b, and maximize with respect to $\alpha_i \geq 0$.
Set gradients to 0 :
$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i = 0$
$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$
Dual Problem
Substitute w back into the Lagrangian to get the dual:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)$$

subject to:

$$\alpha_i \geq 0 \quad \forall i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

This is a quadratic programming (QP) problem.

Solution (Support vectors)
Solve the dual to get α_i^*.
Only those with $\alpha_i^* > 0$ are support vectors.
From this, compute, $w = \sum_{i=1}^{n} \alpha_i^* y_i x_i$. Then choose any support vector (x_s, y_s), then compute:

$$b = y_s - w^T x_s$$

Final decision function

$$f(x) = \text{sign}(w^T x + b) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i x_i^T x + b\right)$$

4.3.4 eXtreme Gradient Boosting (XGBoost) Classifier

XGBoost, which stands for eXtreme Gradient Boosting, is a sophisticated machine learning technique that is designed for effectiveness, speed, and high performance. It is a sort of ensemble learning technique that combines several weak models to create a stronger model, and it is an optimal version of gradient boosting.

- **Mathematical representation :** We begin with an initial forecast, which is frequently set to zero, in what might be seen as an iterative process. Each tree is then added to minimize mistakes. Mathematically the model can be represented as,

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Where,
$\hat{y}_i$ is the final predicted value for the i^{th} data point
K is the number of trees in the ensemble
$f_k(x_i)$ represents the prediction of the k^{th} tree for the i^{th} data point.

A regularization term and a loss function make up the two components of the XGBoost objective function. The regularization term streamlines complicated trees, while the loss function gauges how well the model matches the data.
The general form of the loss function is:

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where,
$l(y_i, \hat{y}_i)$ is the loss function which computes the difference between the true value y_i and the predicted value $\hat{y}_i$
$\Omega(f_k)$ is the regularization term which discourages overly complex trees.

Now instead of fitting the model all at once we optimize the model iteratively. We start with an initial prediction $\hat{y}_i^{(0)} = 0$ and at each step we add a new tree to improve the model.
The updated predictions after adding the t^{th} tree can be written as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Where,
$\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration
$f_t(x_i)$ is the prediction of the t^{th} tree for the i^{th} data point.
The regularization term $\Omega(f_t)$ simplifies complex trees by penalizing the number of leaves in the tree and the size of the leaf. It is defined as:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Where,
T is the number of leaves in the tree
γ is a regularization parameter that controls the complexity of the tree
λ is a parameter that penalizes the squared weight of the leaves w_j

Finally, when deciding how to split the nodes in the tree we compute the information gain for every possible split.
The information gain for a split is calculated as:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

Where,
G_L, G_R are the sums of gradients in the left and right child nodes
H_L, H_R are the sums of Hessians in the left and right child nodes

XGBoost chooses the split that has the highest gain by computing the information gain for each potential split at each node. This successfully lowers mistakes and enhances the model's performance.

4.3.5 Ensemble Model (Stacking)

Stacking, or Stacked Generalization, is an ensemble learning technique that combines multiple classification or regression models (called base learners) to improve predictive performance. Instead of relying on a single model, stacking trains a meta-model (also called a level-1 model) to learn how to best combine the predictions of the base learners.

- **Mathematical Representation :** Let $\mathbf{X} \in \mathbb{R}^{n \times d}$: input feature matrix (with n samples and d features) and $\mathbf{y} \in \{0, 1, \ldots, K-1\}^n$: target labels (for K classes) **Level-0 (Base Learners)** Assume there are M base models:

$$h_1(\mathbf{X}),\ h_2(\mathbf{X}),\ \ldots,\ h_M(\mathbf{X})$$

Each base model outputs predictions :
Class labels: $h_m(\mathbf{X}) \in \{0, 1, \ldots, K-1\}^n$
Class probabilities: $h_m(\mathbf{X}) \in [0, 1]^{n \times K}$
Stack these predictions to form a new feature matrix $\mathbf{Z}$:

$$\mathbf{Z} = \begin{bmatrix} h_1(\mathbf{X}) & h_2(\mathbf{X}) & \ldots & h_M(\mathbf{X}) \end{bmatrix} \in \mathbb{R}^{n \times (M \cdot K)}$$

Level-1 (Meta-Learner) : A meta-model H is trained on the stacked predictions $\mathbf{Z}$ to produce the final prediction:

$$\hat{y} = H(\mathbf{Z}) = H(h_1(\mathbf{X}),\ h_2(\mathbf{X}),\ \ldots,\ h_M(\mathbf{X}))$$

Summary Equation :

$$\hat{y} = H(h_1(\mathbf{X}),\ h_2(\mathbf{X}),\ \ldots,\ h_M(\mathbf{X}))$$

This meta-model H could be Logistic Regression, SVM, etc., trained on the predictions of the base models.

4.4 Time Series Analysis

Time series analysis is a statistical method used to analyze data points collected or recorded at successive, evenly spaced points in time, with the goal of understanding underlying patterns (such as trend, seasonality, and cycles) and making forecasts or inferences about future values.

4.4.1 Augmented Dickey-Fuller test(ADF Test)

The ADF test is a statistical test used to determine whether a given time series is stationary or contains a unit root. In simple terms, the test checks for the presence of a unit root, which indicates non-stationarity in the data. Stationary data is critical for many time series models, as non-stationary data can lead to unreliable results in forecasting and other applications.

- **Null hypothesis (H_0)**:The time series contains a unit root (i.e., it is non-stationary).
- **Alternative hypothesis (H_1)**: The time series does not contain a unit root (i.e., it is stationary)
- **Mathematical model**:

  The general model for the ADF test can be written as :
  $\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta Y_{t-i} + \epsilon_t$
  where,
  Y_t is the value of the time series at time t.
  $\Delta Y_t = Y_t - Y_{t-1}$ represents the first difference of Y_t, i.e, how much the series changes from one period to the next.
  α is a constant (drift term).
  βt is the time trend (optional).
  γ is the coefficient on the lagged level of the time series, Y_{t-1}, and is the key parameter being tested.
  $\sum_{i=1}^{p} \delta_i \Delta Y_{t-i}$ represents the sum of lagged first differences of lag p, which accounts for higher order auto correlation in the time series.
  ϵ_t is a error term.

- **Test statistic**: The test statistic for ADF test is calculated as:

  $$ADF = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$$

  where,
  $\hat{\gamma}$ is the estimated value of γ.
  $SE(\hat{\gamma})$ is the standard error of $\hat{\gamma}$.

- **Hypothesis testing**: The ADF test focuses on the value of γ. Specifically, it tests the null hypothesis that $\gamma = 0$(i.e.,the presence of a unit root) against the alternative hypothesis that $\gamma < 0$(i.e., stationarity).
- **Decision rule**: Reject the null hypothesis, which states that the series is stationary, if the test statistic is smaller than the critical value. The null hypothesis cannot be rejected if the test statistic exceeds the crucial value.

4.4.2 ACF and PACF Plot

ACF and **PACF** plots are essential tools for identifying patterns in time series data, particularly when building models like ARIMA. These plots help understand the correlation between a time series and its lagged values, providing insight into which lags to include in a model.

1. **Autocorrelation Function (ACF) plot**: The ACF plot gives a comprehensive picture of how current values connect to past ones across a number of time steps by displaying the correlation between the time series and its lagged values.

- **Purpose**: The ACF plot helps identify the degree of correlation between observations at different lags. It is useful for determining the MA (Moving Average) terms in ARIMA models.If the ACF plot shows significant correlation at lag k and then cuts off, this suggests that $q = k$.

- **Interpretation**:
  The x-axis shows the number of lags.
  The y-axis shows the autocorrelation values (ranging from -1 to 1).
  Significant spikes outside the confidence interval (dashed lines) suggest a strong correlation at that particular lag.
  For example, if a spike at lag 1 is significant but the correlation quickly drops to near zero for later lags, it suggests that only the immediate previous value (lag 1) is important for predicting the current value.

2. **Partial Autocorrelation Function (PACF) Plot**: The PACF plot eliminates the impact of lesser lags and displays the correlation between a time series and its lagged values.

- **Purpose**: PACF helps in identifying the AR (AutoRegressive) terms in ARIMA models.If the PACF plot shows significant correlation at lag k and then cuts off, this suggests that $p = k$. It explains the direct relationship between a time series and its lag, removing the influence of any intermediate lags.

- **Interpretation**:
  Like the ACF plot, the x-axis represents the lags and the y-axis represents the partial autocorrelations.
  A spike at a given lag indicates a significant direct relationship between the time series and its lagged version, after accounting for the influence of previous lags.
  The first significant spike in the PACF plot can help determine the number of AR terms.

4.5 Forecasting Model

A forecasting model is a mathematical or statistical tool used to predict future values or trends based on historical data. It identifies patterns in past data and uses them to estimate future outcomes.

4.5.1 Seasonal AutoRegressive Integrated Moving Average (SARIMA) :

SARIMA, which stands for Seasonal Autoregressive Integrated Moving Average, is a versatile and widely used time series forecasting model. It is an expansion of the non-seasonal ARIMA model, which was created to deal with seasonal data. Since SARIMA can identify both short-term and long-term connections in the data, it is a powerful forecasting tool. With seasonal components, it blends the ideas of moving average (MA), integrated (I), and autoregressive (AR) models.

- **Notation :** SARIMA model is denoted by, SARIMA(p, d, q)(P, D, Q, s) where, AR(p) = Autoregressive component of order p, MA(q) = Moving average component of order q, I(d) = Integrated component of order d, Seasonal AR(P) = Seasonal autoregressive component of order P, MA(Q) = Seasonal moving average component of order Q, Seasonal I(D) = Seasonal integrated component of order D, s = Seasonal period.

- **Mathematical representation :** The mathematical representation of SARIMA is as follows,
  $(1 - \phi_1 B)(1 - \Phi_1 B^s)(1 - B)(1 - B^s)y_t = (1 + \theta_1 B)(1 + \Theta_1 B^s)\epsilon_t$
  where,
  y_t = observed time series at time t
  B = backward shift operator, representing the lag operator (i.e., $By_t = y_{t-1}$)
  ϕ_1 = non - seasonal autoregressive coefficient
  Φ_1 = seasonal autoregressive coefficient
  θ_1 = non - seasonal moving average coefficient
  Θ_1 = seasonal moving average coefficient
  s = seasonal period
  ϵ_t = white noise error term at time t

  Let's go through each component of the equation :

- **Autoregressive (AR) component :** The relationship between a specific number of lagged observations (prior values in the time series) and the present observation is captured by the autoregressive non-seasonal component $(1-\phi_1 B)$. The backshift operator, represented by the B term, is frequently employed in time series analysis. It is a representation of the lag operator, which moves the time series backward by a predetermined amount of time. How many previous values are taken into account in the model depends on the autoregressive component's order, indicated by (p).

- **Seasonal Autoregressive (SAR) component :** The sesonal autoregressive component is denoted by $(1 - \Phi_1 B^s)$. This element depicts the correlation between the current observation and a predetermined number of seasonal lag observations. Applying the backshift operator to the seasonal lag observations is represented by the B^s term.

- **Non-Seasonal differencing component :** With d being the order of non-seasonal differencing (1 - B) represents the non-seasonal differencing component. Through a specific number of differencings, this component makes the time series stationary.

- **Seasonal differencing component :** The seasonal differencing component is represented by (1 - B^s), where D is the order of seasonal differencing and. This component is used to make the time series stationary by differencing it at seasonal intervals.

- **Observed time series :** The acronym y_t represents the observed time series. It is a representation of the previous data we wish to forecast.

- **Moving average component :** The moving average component is represented by $(1 + \theta_1 B)$. This component captures the relationship between the current ob-

servation and the residual errors from a moving average model applied to lagged observations.

- **Seasonal moving average component :** The seasonal moving average component is represented by $(1 + \Theta_1 B^s)$. The correlation between the present observation and the residual errors from a moving average model applied to lagged observations at seasonal intervals is captured by this component.

- **Error term :** The error term is denoted by ϵ_t. It represents the random noise or unexplained variation in the time series.

4.5.2 Multi-Layer Perceptron (MLP):

MLP is a type of artificial neural network where each layer of neurons is fully connected to the next layer. It is one of the foundational models in deep learning and is typically used for classification and regression tasks.

- **Input layer:** The first layer that receives the input features. Each input corresponds to a node in this layer. The number of nodes equals the number of features in the dataset.
  Let the input vector at time t be $X(t) = [x_1(t), x_2(t), ..., x_p(t)]^T$, where $x_i(t)$ represents the lagged values of the time series.

- **Hidden layers:** An MLP with one hidden layer and neurons computes a transformation of the input vector as:

$$Z_i(t) = \phi(\sum_{i=1}^{p} w_{ij}^{(1)} x_i(t) + b_j^{(1)}), \qquad j = 1, 2, 3, ..., m$$

  Where:
  $w_{ij}^{(1)}$ are the weights from the input layer to the hidden layer.
  $b_j^{(1)}$ are the biases for the hidden neurons.
  $\phi(.)$ is an activation function (commonly ReLU, sigmoid, or tanh).

- **Output layer:** The output of the MLP, representing the forecast $\hat{y}(t+1)$, is computed as:

$$\hat{y}(t+1) = \sum_{j=1}^{m} w_j^{(2)} z_j(t) + b^{(2)}$$

  where,
  w_j^2 are the weights from the hidden layer to the output layer.
  $b^{(2)}$ is the bias for the output neuron.
  Thus, the output layer linearly combines the activations from the hidden layer to produce the forecast.

- **Activation functions:** These help the model to learn more intricate patterns by adding non-linearity.
  The following are typical MLP activation functions.
  ReLU (Rectified Linear Unit): $f(x) = max(0, x)$,commonly used in hidden

layers.
Sigmoid: $\frac{1}{1+e^{-x}}$,often used in binary classification problems.
Softmax: Used in the output layer for multi-class classification providing a probability distribution over the classes.
Tanh: A smoother activation function than ReLU, $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

- **Error function:** The MLP is trained by minimizing the error between the predicted output $\hat{y}(t+1)$ and the actual value $y(t+1)$. The most common error function used is **the Mean Squared Error (MSE)**:

$$L(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y(t+1) - \hat{y}(t+1))^2$$

Where:
N is the number of observations.
θ represents the parameters (weights and biases) of the MLP.
The weights $w_{ij}^{(1)}$, $w_j^{(2)}$ and biases $b_j^{(1)}$, $b^{(2)}$ are updated using back propagation and an optimization algorithm such as **stochastic Gradient Descent (SGD)** or **Adam optimizer**.

- **Optimization:** The network uses optimization algorithms to minimize the loss by adjusting weights and biases. Common algorithms include:
Gradient descent: A method to minimize the loss by updating weights in the direction of the negative gradient.
Stochastic Gradient Descent (SGD): An approximation of gradient descent, which updates weights more frequently by using a single data point (or a batch) at a time.
Adam (Adaptive Moment Estimation): A popular optimizer that adjusts learning rates based on moving averages of the gradients, often leading to faster convergence.

- **Backpropagation:**
(a) Forward pass: During the forward pass, the inputs are passed through the network, and the output is calculated using the current weights and biases.
(b) Loss calculation: The error (loss) is computed using the MSE function defined above.
(c) Backward pass (Gradient calculation): Using the chain rule, the gradients of the loss function with respect to the network's parameters are calculated.
For each parameter θ :

$$\frac{\delta L}{\delta \theta} = \frac{\delta L}{\delta \hat{y}(t+1)}.\frac{\delta \hat{y}(t+1)}{\delta \theta}$$

For the output layer, the gradient with respect to the weights $w_j^{(2)}$ is:

$$\frac{\delta L}{\delta w_j^{(2)}} = (\hat{y}(t+1) - y(t+1)).z_j(t)$$

For the hidden layer weights $w_{ij}^{(1)}$, the gradient involves backpropagating the error through the activation function $\phi(.)$:

$$\frac{\delta L}{\delta w_{ij}^{(1)}} = (\hat{y}(t+1) - y(t+1)).\phi'(\sum_{i=1}^{p} w_{ij}^{(1)} x_i(t) + b_j^{(1)}).x_i(t)$$

(d) Parameter update: Using gradient descent, the parameters are updated as:

$$\theta \leftarrow \theta - \eta . \frac{\delta L}{\delta \theta}$$

where η is the learning rate.

- **Training:** The MLP regressor is trained on a dataset by iteratively performing forward propagation, calculating the loss, and updating the weights through back-propagation until convergence.

- **Forecasting:** Once the MLP model is trained, we can use it to predict future values of the time series. The prediction at each time step is obtained by passing the lagged values of the series into the network:

  $$\hat{y}(t+1) = f(X(t); \theta)$$

  For multi-step forecasting (predicting more than one step ahead), predictions are made recursively, where previous predictions are used as inputs for future predictions.

4.5.3 Gated Recurrent Unit (GRU)

A GRU is a type of recurrent neural network architecture designed to handle sequential data (e.g., time series, text) while overcoming the limitations of traditional RNNs (like vanishing gradients). The core idea behind GRUs is to use gating mechanisms to selectively update the hidden state at each time step allowing them to remember important information while discarding irrelevant details.

- **Cell architecture :** The GRU consists of two gates :
  i) Update gate (z_t) : This gate decides how much information from previous hidden state should be retained for the next time step.
  ii) Reset gate (r_t) : This gate determines how much of the past hidden state should be forgotten.
  Compared to conventional RNNs that just use hidden state, these gates enable GRU to regulate the information flow more effectively.

- **Mathematical formulation :** Let,
  x_t = input at time t
  h_t = hidden state at time t
  h_{t-1} = previous hidden state
  σ: sigmoid activation
  $\odot$: element-wise multiplication
  W, U: learnable weight matrices

  Update gate:

  $$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

  Reset gate:

  $$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

  Candidate hidden state:

  $$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}))$$

Final hidden state:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

- **Working intuition :** The reset gate decides how much past memory to forget. The update gate decides how much new memory to incorporate. Then GRU combines forget and input gates into a single update gate, and it does not maintain a separate cell state.

4.6 Evaluation Metric :

An evaluation metric is a measure used to assess the performance of a model by comparing its predictions to the actual outcomes. It helps determine how well the model is performing and guides model selection and improvement.

4.6.1 Accuracy, Precision, Recall and F-1 Score

- **Accuracy :** Proportion of total correct predictions out of all predictions made.
  Formula :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision :** Proportion of correctly predicted positive instances out of all instances predicted as positive.
  Formula :

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall :** Proportion of correctly predicted positive instances out of all actual positive instances.
  Formula :

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score :** Harmonic mean of precision, recall, balancing both metrices.
  Formula :

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.6.2 Confusion Matrix

Confusion matrix is a simple table that is used to gauge the effectiveness of a classification model. By contrasting the model's predictions with the actual outcomes, it demonstrates whether the model was accurate or not. This aids in identifying the model's areas of error so that it can be improved.

Four categories are used to categorize the predictions:

- **True Positive (TP) :** The model correctly predicted a positive outcome i.e the actual outcome was positive.

- **True Negative (TN) :** The model correctly predicted a negative outcome i.e the actual outcome was negative.

- **False Positive (FP) :** The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- **False Negative (TP) :** The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.

4.6.3 AUC - ROC Curve

ROC, stands for Receiver Operating Characteristics plots true positive rate versus false positive rate at different thresholds. It represents the trade-off between the sensitivity (recall) and specificity (proportion of actual negatives that the model correctly identifies) of a classifier.

AUC, stands for Area Under the Curve measures the area under the ROC curve. A higher AUC value indicates better model performance as it suggests a greater ability to distinguish between classes. An AUC value of 1.0 indicates perfect performance while 0.5 suggests it is random guessing.

4.6.4 Root Mean Squared Error (RMSE)

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

To compute RMSE, calculate the residual (difference between prediction and truth) for each data point, compute the norm of residual for each data point, compute the mean of residuals and take the square root of that mean.

- **Formula :** Root mean square error can be expressed as :

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

where,
N is the number of data points
$y(i)$ is the i^{th} measurement
$\hat{y}(i)$ is its corresponding prediction.

4.6.5 Mean Absolute Error (MAE)

Mean Absolute Error is a popular measure which calculates the average difference between the calculated values and actual values. It is also known as scale-dependent accuracy as it calculates error in observations taken on the same scale used to predict the accuracy of the machine learning model.

- **Formula :** Mean absolute error can be expressed as :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Where,
y_i = Actual value for the i^{th} observation
$\hat{y}_i$ = Calculated value for the i^{th} observation
n: Total number of observations.

5 Result Analysis and Discussion

In the previous section, we have discussed about methodologies which have been implemented in our project. Here, we are discussing about the results which we have gotten after applying that methods.

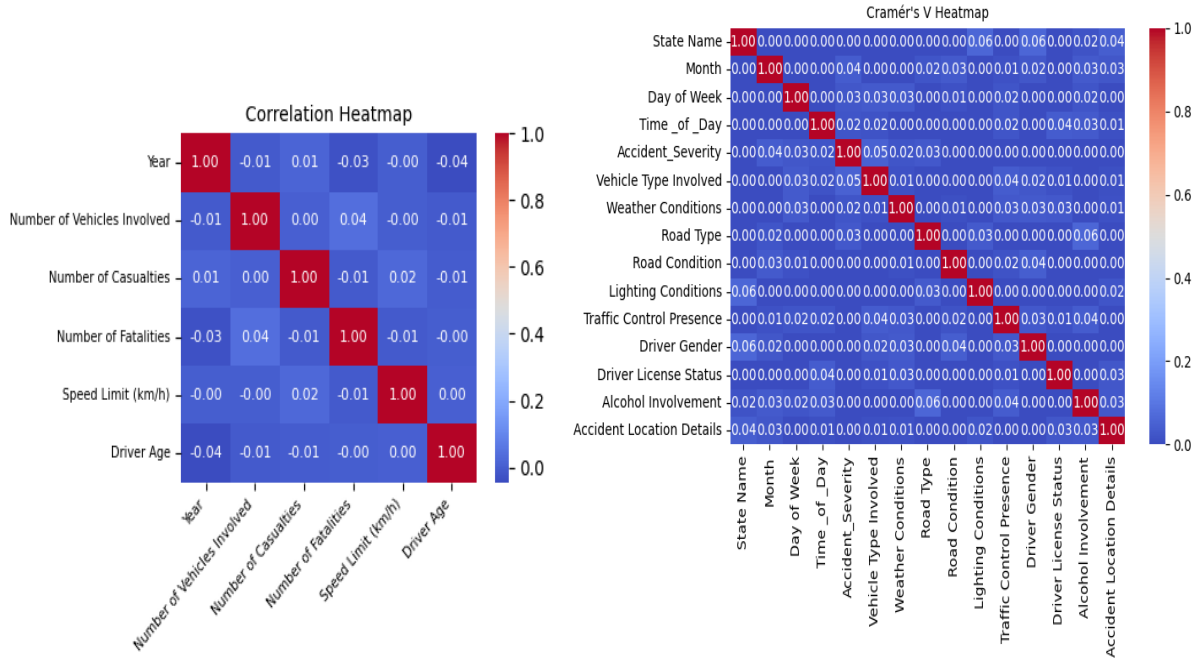
We have done our project on the basis of two different perspectives, former was based on daily database and later was based on monthly database.

In our daily dataset, there were some missing values in two columns named "Driving License Status" and "Traffic Control Presence". We have imputed those missing values by putting "Unknown". It was our data preprocessing phase.

5.1 Analysis based on daily data

5.1.1 Correlation Analysis

At first we have visualized the correlation heatmap. For numerical variables we have used Pearson's measure and for categorical variables we have used Cramer's V method for calculating correlation between the features and then we have visualized their correlations by correlation heatmap.



(a) (b)

Figure 3: Correlation heatmap of (a) Numerical Variables and (b) Categorical Variables.

- (a) The correlation heatmap illustrates a near-zero linear association among the numerical variables, with the highest absolute correlation being just 0.04. This indicates that the features are statistically independent, exhibiting minimal interdependence or redundancy.
- (b) The Cramér's V heatmap reveals a uniformly low level of association among the categorical variables, with most values hovering near zero. The highest observed

association is only 0.11 between Weather Conditions and Road Condition, indicating a very weak relationship. All other variable pairs exhibit negligible dependency, affirming that the categorical features are largely independent of one another.

5.1.2 Exploratory Data Analysis (EDA)

In this part, we have plotted a line diagram for visualizing number of accidents in a calender year across all states. Afterthat, we have visualized the relationship between predictors and response variable. There, our response variable was "Accident Severity" and we have checked how severity of accidents was differing with other factors.

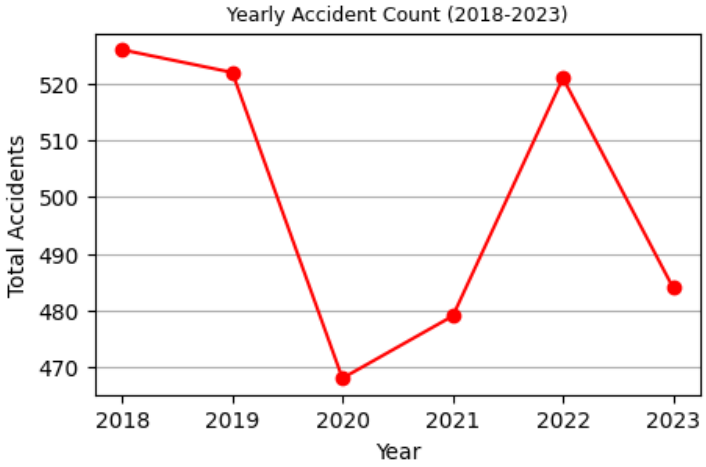
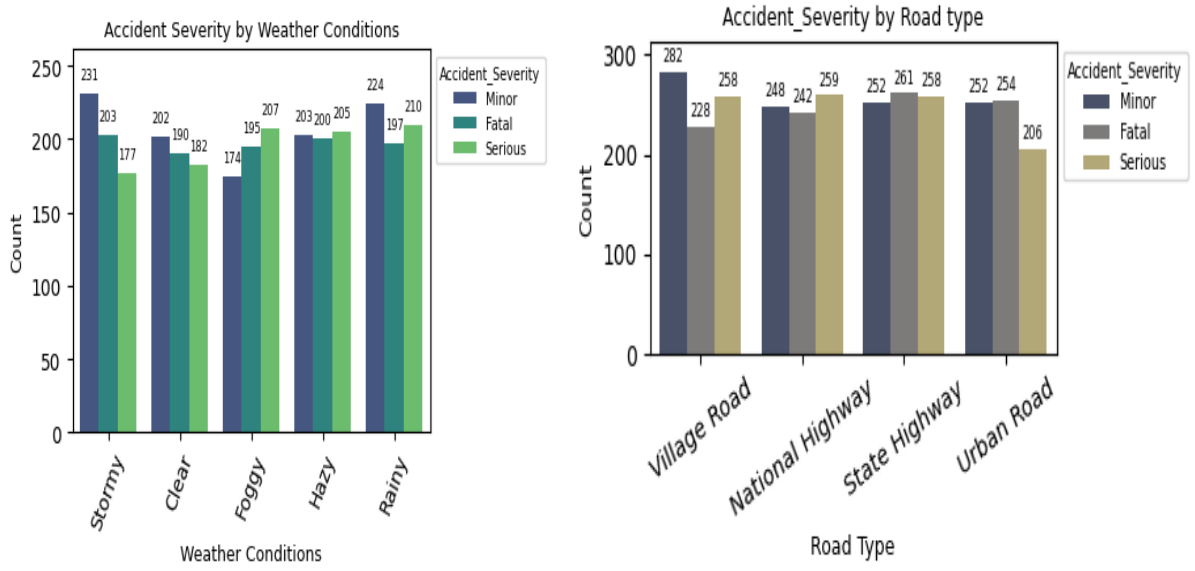


Figure 2 : Yearly Accident Count (2018–2023)

The line graph, which shows significant variations, shows the annual accident numbers from 2018 to 2023. Accidents decreased precipitously between 2019 and 2020, then gradually recovered to reach a peak in 2022. In 2023, a following decline is noted. This pattern points to possible outside factors, such as the 2020 decline that most likely reflected movement restrictions brought on by the pandemic. Then, we have analyzed relationships with other predictor variables one by one.

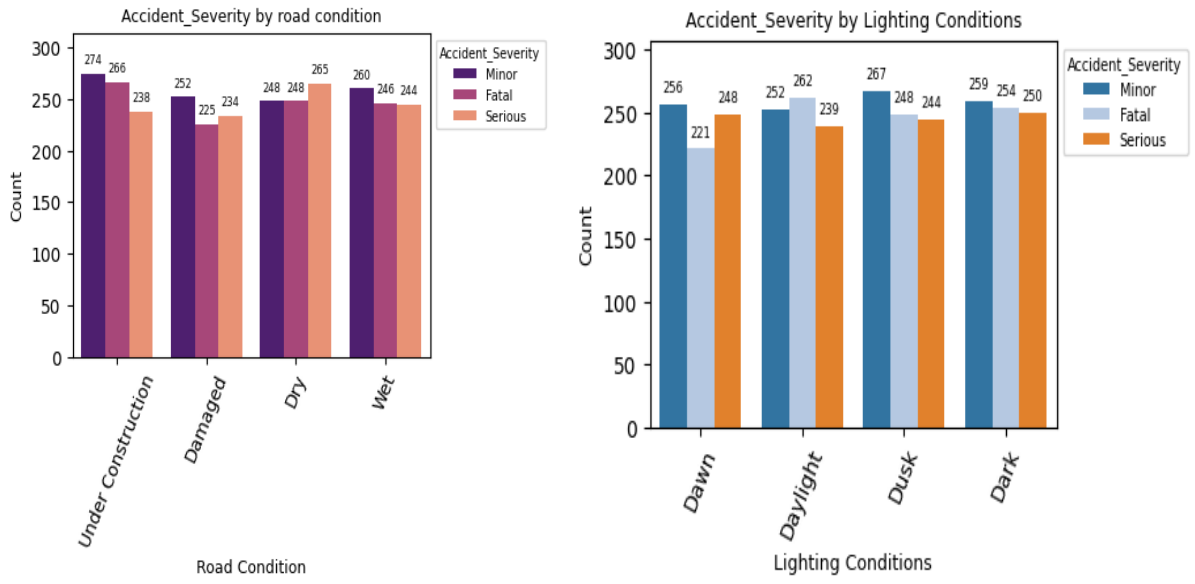


(a) (b)

Figure 4: Accident severity by (a) Weather conditions and (b) Road type.

- (a) The bar chart illustrates the distribution of accident severity across various weather conditions. Minor accidents outnumber serious and fatal ones under some conditions (stormy, clear and rainy), with the highest count observed during stormy weather (231). Serious accidents are notably elevated in foggy, hazy and rainy conditions (207, 205 and 210 respectively), while fatal accidents remain relatively lower across almost all categories. Rainy and hazzy conditions also show a high volume of incidents across all severities. Overall, adverse weather especially stormy, rainy, hazzy and foggy correlates with elevated accident counts, underscoring its impact on road safety.

- (b) The severity of accidents on various road types is shown in the bar chart. Village Roads have the highest number of fatalities (228) and minor accidents (282). The most significant accidents (261), with almost equal numbers of minor (252) and fatal (258) incidents, are reported on State Highways. The numbers on National Highways are fairly evenly distributed throughout all severities, with major accidents (259) just surpassing minor and fatal incidents. Urban Roads had the fewest serious accidents (206), but the highest number of minor (252) and fatal (254) incidents. State and village highways have the most severe accident traits overall.

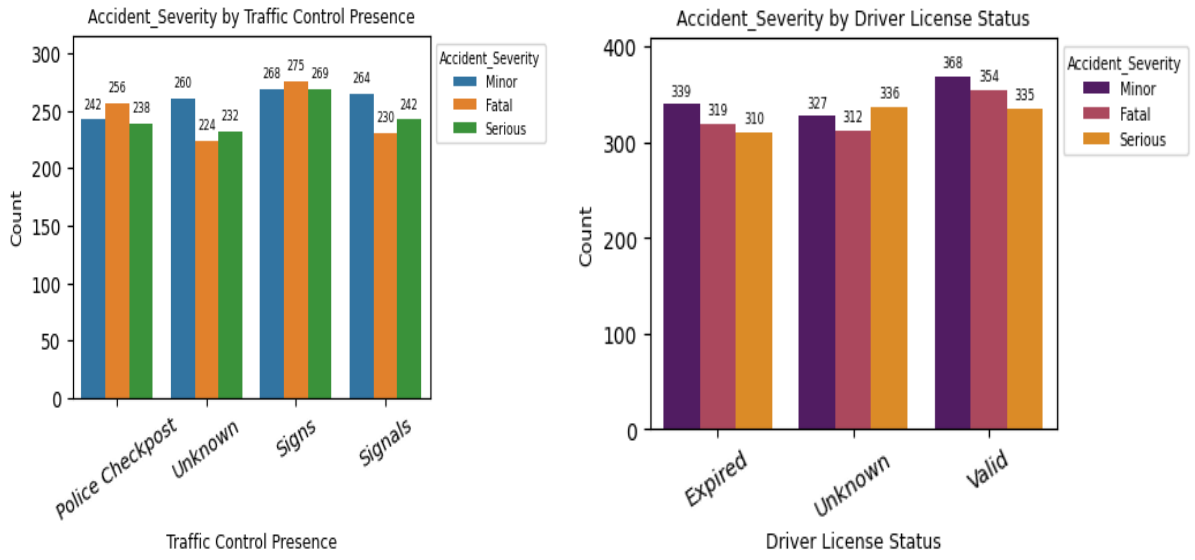


(a) (b)

Figure 5: Accident severity by (a) Road condition and (b) Lighting conditions.

- (a) The bar chart shows accident severity across different road conditions. Under construction roads record the highest minor (274) and fatal (266) accidents but the lowest serious cases (238). Wet roads follow with high fatal (260) and serious (244) accidents. Dry roads display equal fatal and serious counts (248 each), with slightly lower minor cases (265). Damaged roads report the lowest fatal (225) and minor (252) accidents, with 234 serious cases. Overall, roads under construction exhibit the highest accident severity across all types, emphasizing their hazardous nature.

- (b) The severity of accidents under various illumination situations is depicted in the bar chart. The highest number of minor accidents occurs at dusk (267), with dark (259) and daylight (262) coming in close succession. Daylight has the most fatal collisions (254), while Dawn has the fewest fatalities (221). Serious accidents range from 239 (Daylight) to 250 (Dark), which is comparatively constant across all conditions. In general, there are more accidents between dusk and nighttime hours, especially those that are minor or fatal.

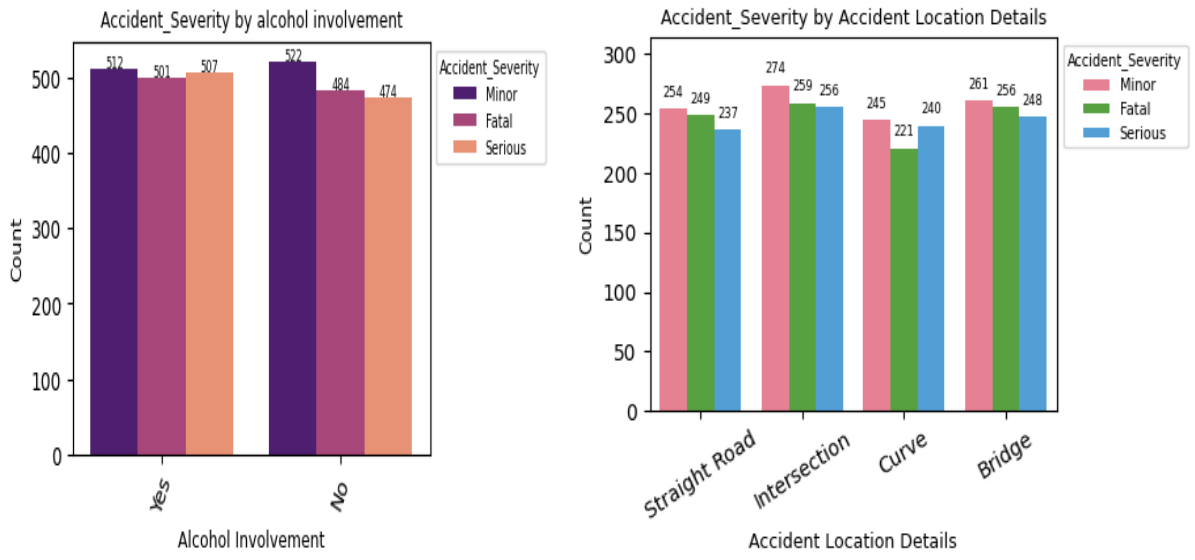


(a) (b)

Figure 6: Accident severity by (a) Traffic control presence and (b) Driver license status.

- (a) The bar chart shows accident severity by traffic control presence. Signs have the highest counts across all severity levels—minor (268), fatal (275), and serious (269), indicating the most accidents occur in their presence. Signals follow with high minor (264) and serious (242) counts, and 230 fatal cases. Police Checkposts show 256 fatal, 242 minor, and 238 serious accidents. The Unknown category has the lowest values in particularly fatal (224) and serious (232). Overall, signs and signals are associated with the highest accident severity counts.

- (b) The severity of the accidents is displayed in the bar chart according to the driver's license status. The majority of accidents, regardless of severity, are caused by drivers with valid licenses, including minor (368), fatal (354), and serious (335). 339 minor, 319 fatal, and 310 serious accidents are followed by expired licenses. Among non-valid categories, the unknown category has the highest number of major accidents (336), but the lowest number of fatalities (312) and minor accidents (327). Since they are more likely to be on the road, people with valid licenses are generally involved in the most accidents.



(a) (b)

Figure 7: Accident severity by (a) Alcohol involvement and (b) Accident location details.

- (a) Based on alcohol participation, the bar chart analyzes the severity of accidents. Accidents with alcohol are more common than those without; there have been reports of minor (512), fatal (501), and serious (507) incidents. The No Alcohol category, on the other hand, has fewer fatal (484) and serious (474) incidents but somewhat more minor accidents (522). Therefore, there is a larger severity risk associated with alcohol consumption, as evidenced by the fact that fatal and serious accidents are more often when alcohol is present, although minor accidents are slightly more common when alcohol is not present.

- (b) The accident severity for each type of place is represented in a bar chart. The majority of minor accidents (274), fatalities (259), and serious cases (256) occur at intersections. All severity levels—minor (245), fatal (221), and serious (240)—have the lowest counts for curves. The distribution of fatal (256) and serious (248) incidents on bridges is balanced. Values for all categories are moderate on straight highways. Intersections are generally the most accident-prone sites, especially for small incidents, most likely as a result of increased vehicle interaction and crossing conflicts.

5.1.3 Classification Model Building and Evalution

In this section, we have done our classification model building and evaluation work. We have interested to predict the accident severity based on different predictor variables and that is why we have taken "Accident Severity" as our response variable and remaining variables are as predictor variables.

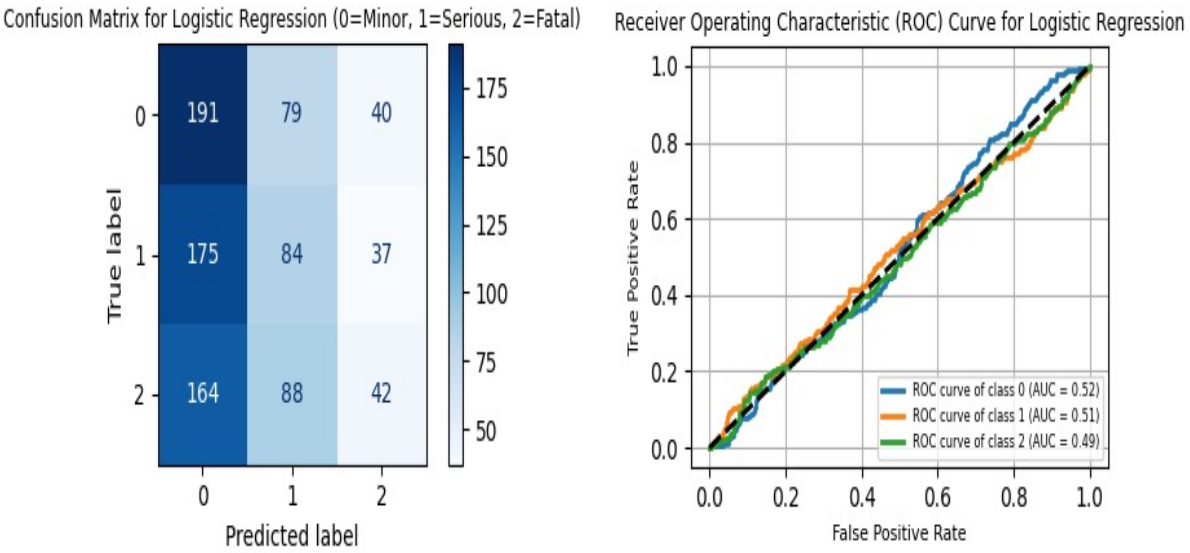
We have applied logistic regression, random forest classifier, support vector machine classifier and xgboost classifier for doing our classification staff.

Classification Report (Logistic regression):

	precision	recall	f1-score	support
0	0.36	0.62	0.45	310
1	0.33	0.28	0.31	296
2	0.35	0.14	0.20	294
accuracy			0.35	900
macro avg	0.35	0.35	0.32	900
weighted avg	0.35	0.35	0.32	900

Figure 8: Classification report of logistic regression

The classification report reflects suboptimal performance of the logistic regression model, with overall accuracy score at just 0.35. Class 0's recall (0.62) is moderate, but its precision (0.36) and F1-score (0.45) are still low. Both Classes 1 and 2 exhibit extremely low performance, with Class 2 showing the severe misclassification with an F1-score of just 0.20. Low precision, recall, and F1-scores that are consistent across classes indicate that the model is not very good at generalizing.



(a) (b)

Figure 9: (a)Confusion Matrix and (b) ROC Curve.

- (a) This logistic regression model predominantly misclassifies classes, particularly Class 1 (Serious) and Class 2 (Fatal), which are commonly forecasted as Class 0 (Minor), according to the confusion matrix. Poor class separation was demonstrated by the fact that only 42 out of 294 dead cases and 84 out of 296 critical cases were accurately identified. The model has a substantial bias toward the majority class, as evidenced by the large percentage of real critical and fatal cases that are mistakenly classified as minor (175 and 164, respectively).
- (b) The ROC curves presented across three classes exhibit performance close to random guessing, as evidenced by the AUC scores: 0.52 for class 0, 0.51 for class 1,

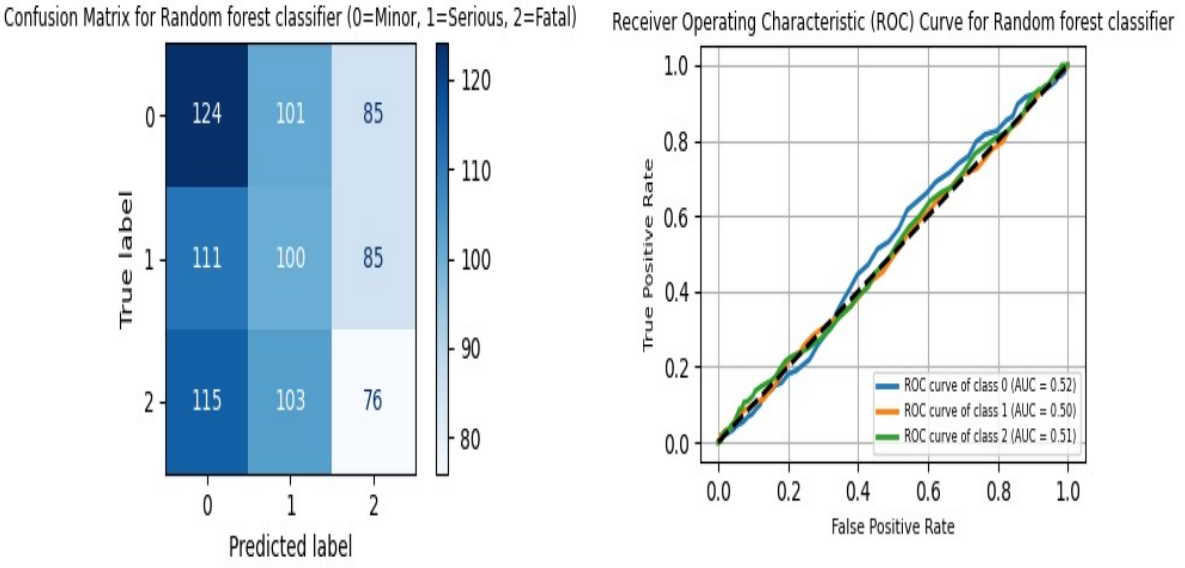
and 0.49 for class 2. These values, all hovering around 0.5, indicate that the model lacks discriminative power in distinguishing between the classes. The nearly diagonal trajectory of the ROC curves further reinforces this interpretation, reflecting minimal separation between true positives and false positives across the decision thresholds. This suggests poor model calibration and inherently inseparable classes.

Classification Report (Random forest) :

	precision	recall	f1-score	support
0	0.35	0.40	0.38	310
1	0.33	0.34	0.33	296
2	0.31	0.26	0.28	294
accuracy			0.33	900
macro avg	0.33	0.33	0.33	900
weighted avg	0.33	0.33	0.33	900

Figure 10: Classification report of random forest

According to the classification report, overall performance was poor, with an accuracy score of only 0.33,roughly the same as speculation in a three-class assignment. All classes had poor precision, recall, and F1-scores, with class 2 having the lowest F1-score at just 0.28. The conclusion that the model is not learning significant patterns is further supported by the weighted and macro averages for all indicators, which both stand at 0.33



(a) (b)

Figure 11: (a) Confusion Matrix and (b) ROC Curve.

- (a) Significant overlap between the three severity classifications is shown in this confusion matrix. The model misclassifies a significant portion of observations in all categories, despite properly classifying 124 cases of Minor, 100 cases of Serious,

and 76 cases of Fatal. For example, 103 fatal cases are incorrectly classified as serious and 115 fatal cases as minor. This pervasive misclassification indicates that the classes are not clearly separated, indicating that the model has difficulty differentiating between the outcomes' severity.

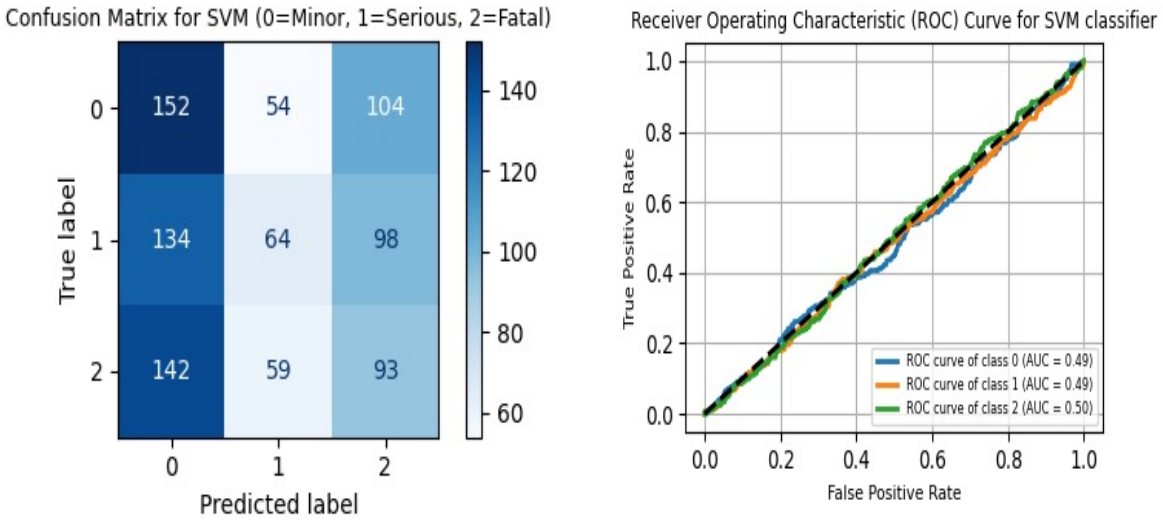
- (b) The poor categorization performance is supported by this ROC curve. The model does no better than random chance in differentiating between the classes, as indicated by the AUC values of 0.52 for class 0 (Minor), 0.50 for class 1 (Serious), and 0.51 for class 2 (Fatal). The model's incapacity to consistently rank true positives above false positives across decision thresholds is shown in the ROC curves' near-diagonal shape.

Classification Report (SVM):

	precision	recall	f1-score	support
0	0.36	0.49	0.41	310
1	0.36	0.22	0.27	296
2	0.32	0.32	0.32	294
accuracy			0.34	900
macro avg	0.34	0.34	0.33	900
weighted avg	0.34	0.34	0.33	900

Figure 12: Classification report of support vector machine

All three classes exhibit less than ideal predictive performance, according to this classification report. Class 1 (Serious) scores the lowest with an F1-score of only 0.27, indicating a poor balance between precision and recall, while Class 0 (Minor) achieves the best F1-score of 0.41. Class 2 (Fatal), which has low values of about 0.32, also has difficulties. The overall accuracy score is just 0.34, and the macro and weighted averages for all important measures (precision, recall, and F1-score) are roughly 0.33–0.34. This clearly implies that the model is not generalizable and does not account for significant class differences.



(*a*) (*b*)

Figure 13: (a) Confusion Matrix and (b) ROC Curve.

- (a) The confusion matrix shows that the three severity classifications are frequently misclassified. A sizable portion of samples from Classes 1 and 2 are incorrectly classified, especially being forecasted as Class 0, even though Class 0 contains the most accurately predicted cases (152). For instance, 142 fatal cases and 134 serious cases are mistakenly classified as minor. These trends point to a systematic tendency to overestimate the Minor category. The comparatively low correct classification counts for Serious (64) and Fatal (93) underscore the model's failure to discriminate across severity categories efficiently.
- (b) The ROC curves confirm its near-random classification behavior. With AUC scores of 0.49 for Class 0 and Class 1, and 0.50 for Class 2, the model demonstrates no discriminative ability in ranking positive versus negative instances across decision thresholds. The curves follow a trajectory close to the diagonal line, indicating that the model is not effectively distinguishing true positives from false positives.

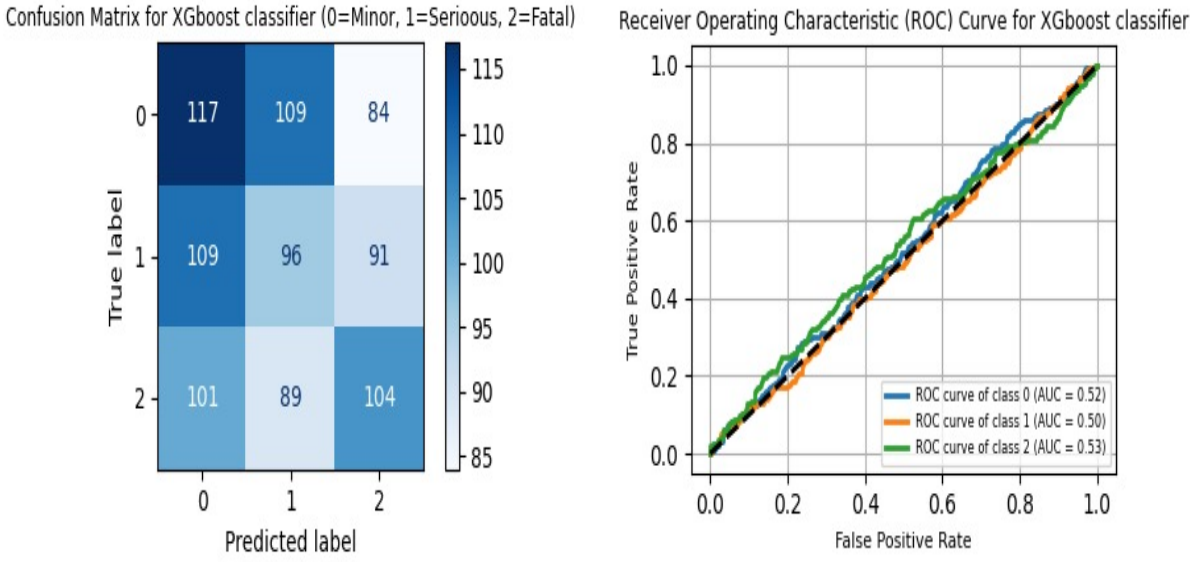
Classification Report (XGBoost):

	precision	recall	f1-score	support
0	0.36	0.38	0.37	310
1	0.33	0.32	0.33	296
2	0.37	0.35	0.36	294
accuracy			0.35	900
macro avg	0.35	0.35	0.35	900
weighted avg	0.35	0.35	0.35	900

Figure 14: Classification report of XGBoost

In comparison to the other assessed models, this classification report shows comparatively better performance. XGBoost exhibits marginally greater consistency

across all classes with an overall accuracy score of 0.35 and balanced macro and weighted averages (precision, recall, and F1-score all at 0.35). Among all investigated classifiers, it produces the greatest F1-score for class 0 (Minor) at 0.37, class 1 (Serious) at 0.33, and class 2 (Fatal) at an impressive 0.36. In comparison to previous classification models, this performance shows that XGBoost catches class-specific patterns more successfully, providing better generalization and less severe bias, even though it is still modest.



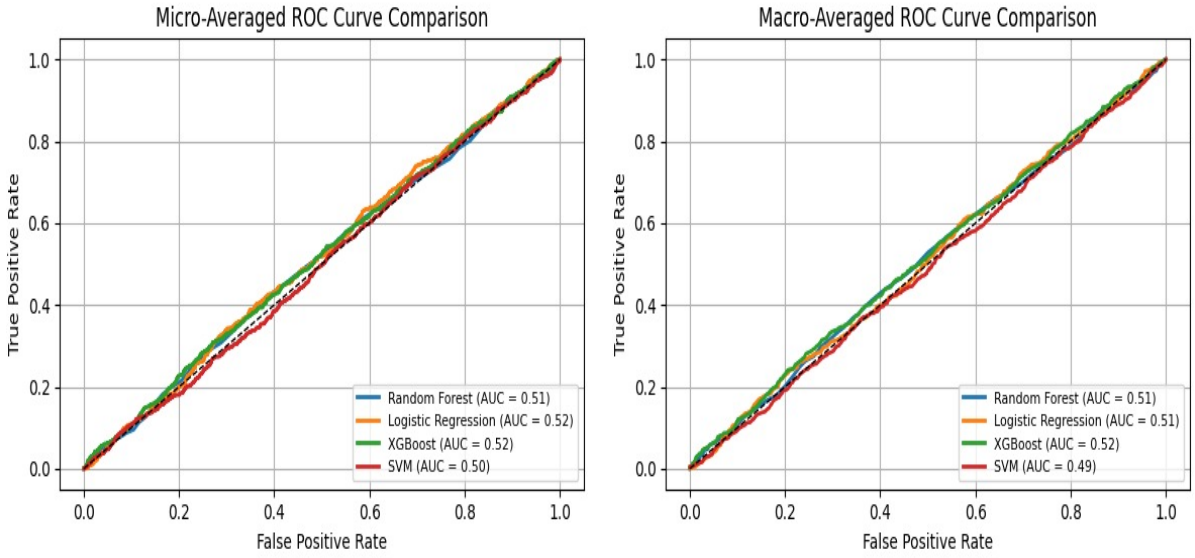
(*a*) (*b*)

Figure 15: (a) Confusion Matrix and (b) ROC Curve.

- (a) Its relative superiority is further supported by the confusion matrix's more balanced distribution of accurate and inaccurate predictions. The classifier outperforms other models, especially on the Fatal class, correctly identifying 117 cases of Minor, 96 of Serious, and 104 of Fatal. Although they still occur, misclassifications are less severe and more uniformly distributed than in earlier classifiers, suggesting a more sophisticated capacity for class distinction. The enhanced true positive rates, particularly for class 2, imply that XGBoost is more adept to manage intricate patterns and ambiguous occurrences in the feature space.

- (b) The ROC curves display AUC scores of 0.52 for class 0, 0.50 for class 1, and a leading 0.53 for class 2, the highest among all models evaluated. Although these scores still reflect limited discriminatory ability, the slight elevation in AUC values implies that XGBoost provides marginally more reliable probabilistic estimates and threshold-based ranking. The curves lie just above the diagonal, suggesting the model is making small but meaningful distinctions between true positives and false positives. This incremental yet consistent improvement across metrics and visual evidence positions XGBoost as the most effective classifier among the models assessed.

5.1.4 Comparison Between Models :



(*a*) (*b*)

Figure 16: ROC curves of (a) Micro average and (b) Macro average.

- **(a) Interpretation :**
  (i) XGBoost and Logistic Regression have the highest AUC values, both at 0.52, indicating a slight improvement over random guessing but still weak predictive performance overall.
  (ii) Random Forest has an AUC of 0.51, showing marginally better-than-random results, but not strong enough to suggest reliable classification.
  (iii) SVM performs the worst with an AUC of 0.50, which is equivalent to random guessing. This implies that the model fails to capture meaningful patterns in the data under the current setup.
  (iv) The ROC curves for all models lie very close to the diagonal reference line, which confirms low discriminative ability across the models.
  (v) The narrow range of AUC values (0.50 to 0.52) suggests that no model is clearly outperforming the others, and all are likely struggling with the same underlying issues.
  (vi) Overall, the micro-averaged ROC analysis shows that even though XGBoost and Logistic Regression are slightly better, all models are underperforming, and significant data-level or modeling-level improvements are needed to achieve meaningful classification performance.

- **(b) Interpretation :**
  (i) XGBoost again performs the best among the four, with an AUC of 0.52, indicating a slight ability to distinguish between classes, though still very weak.
  (ii) Both Random Forest and Logistic Regression show identical AUC scores of 0.51, suggesting only marginally better-than-random performance across classes.
  (iii) SVM has the lowest macro AUC of 0.49, which implies that it performs worse than random guessing when all classes are given equal importance. This is due to

misclassification in minority classes or poor generalization across the board.
(iv) The closeness of all ROC curves to the diagonal line further confirms low discriminative power for all models.
(v) Overall, the macro-averaged evaluation reinforces that while XGBoost slightly leads, none of the models are providing reliable classification performance.

5.2 Analysis based on monthly data

In case of daily database we have only built classification models and have seen that model accuracies are very poor. According to that results and evaluation metrics, it is hard to predict a clear scenario on the basis of daily data since there're uncertainty. For this we have converted our daily dataset on monthly data through programming language and have calculated month-wise number of accidents and then have added a column named "Total Accidents" and then have built same classification models by keeping this newly added as response variable (i.e., we have predicted total number of accidents in a month based on other covariates). Moreover, since in daily dataset response variable was categorical, we were not able to do time series analysis. But on that monthly database we have applied time series modelling for analysis with respect to time.

5.2.1 Classification Model Building and Evaluation

We have started with classification model building and analysis.

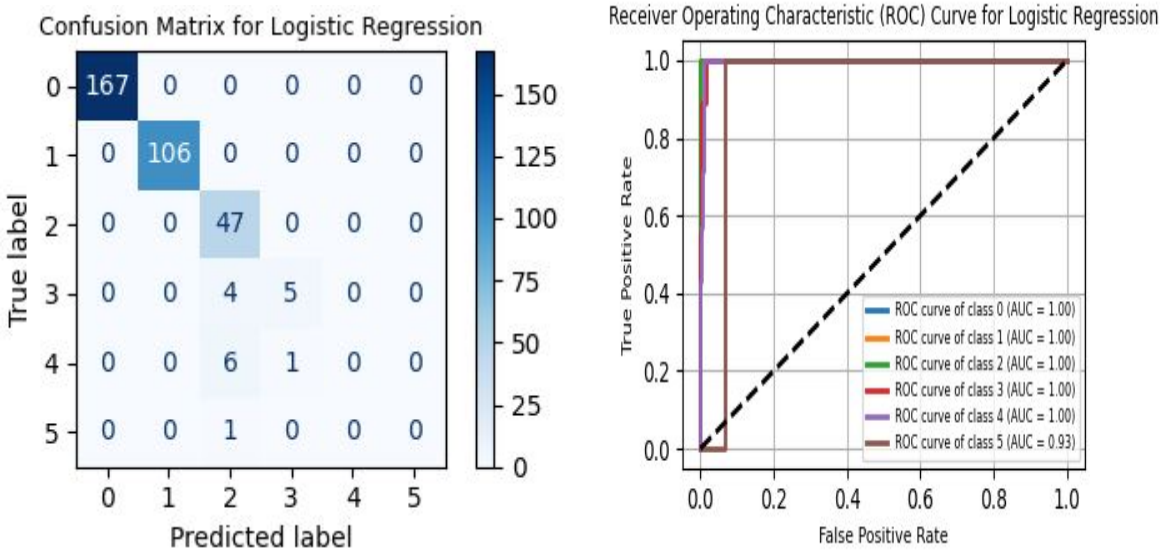
Classification Report (Logistic regression):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	167
1	1.00	1.00	1.00	106
2	0.81	1.00	0.90	47
3	0.83	0.56	0.67	9
4	0.00	0.00	0.00	7
5	0.00	0.00	0.00	1
accuracy			0.96	337
macro avg	0.61	0.59	0.59	337
weighted avg	0.95	0.96	0.95	337

Figure 17: Classification report of Logistic regression

- **Interpretation :**
i) Classes 0 and 1 exhibit perfect scores for precision, recall, and F1-score (all 1.00), indicating the model is highly reliable on these frequent categories.
ii) Class 2 achieves high recall (1.00) and F1-score (0.90), though precision is slightly reduced at 0.81, suggesting occasional false positives.
iii) For class 3, the F1-score drops to 0.67, driven by lower recall (0.56) and precision (0.83), showing partial misclassification.
iv) Classes 4 and 5 have zero precision, recall, and F1-score, indicating the model

failed entirely to detect these underrepresented categories.
v) With a macro average F1-score of only 0.59, all classes exhibit inadequate generalization. On the other hand, the weighted average F1-score is high (0.95), indicating that overall metrics are positively skewed by excellent performance on dominant classes.
vi) The model's outstanding performance on majority classes allows it to retain a strong overall accuracy of 96 percentage.



(*a*) (*b*)

Figure 18: (a) Confusion Matrix and (b) ROC Curve.

- **(a) Interpretation :**
  i) All 167 instances of class 0, 106 of class 1, and 47 of class 2 were classified correctly with zero confusion, demonstrating excellent precision and recall.
  ii) Out of 9 class 3 samples, 5 were correctly predicted, while 4 were misclassified as class 2, indicating some confusion between nearby classes.
  iii) The model's incapacity to capture this uncommon class was demonstrated by the fact that none of the seven class 4 examples were accurately predicted; six were incorrectly classified as class 2, and one as class 3.
  iv) The model completely missed class 5 since the one occurrence was mistakenly classed as class 2.
  v) The model's tendency to misclassify minority classes into dominant classes (especially class 2) underscores a bias introduced by class imbalance.

- **(b) Interpretation :**
  i) For classes 0 through 4, all ROC curves display an AUC of 1.00, signifying flawless discrimination between positive and negative samples.
  ii) Class 5 achieves an AUC of 0.93, still very strong, but slightly lower, due to the extremely limited number of samples.
  iii) Very low false positive rates and high true positive rates are shown in all curves that rise sharply close to the y-axis, indicating confident predictions.
  iv) Although actual classification (as seen in the confusion matrix) has trouble

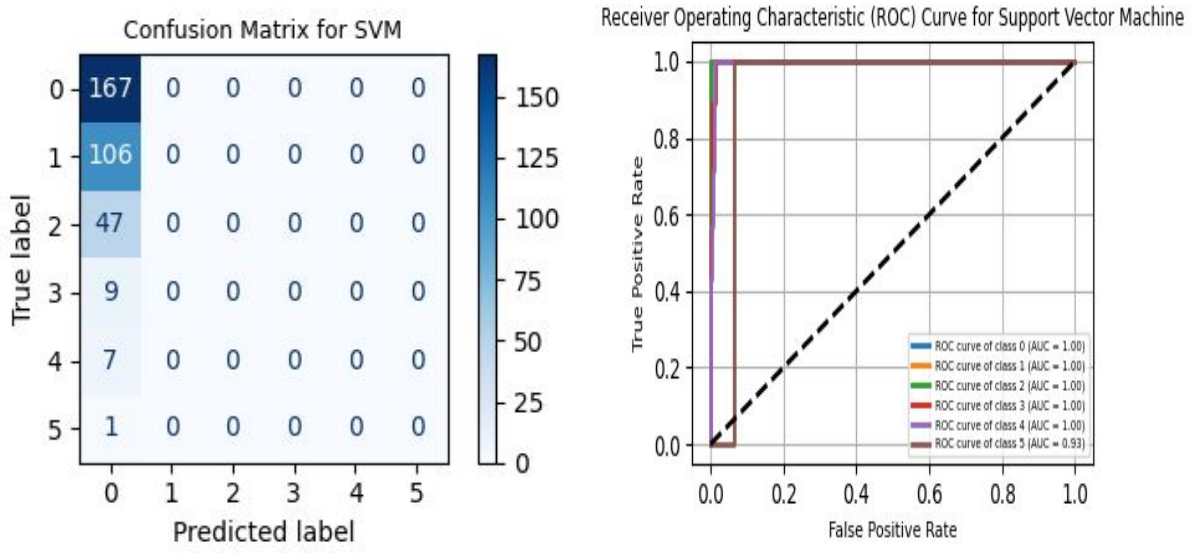
with minority classes, the overall ROC performance indicates that the model's probabilistic outputs are highly discriminative.

Classification Report (SVM):

	precision	recall	f1-score	support
0	0.50	1.00	0.66	167
1	0.00	0.00	0.00	106
2	0.00	0.00	0.00	47
3	0.00	0.00	0.00	9
4	0.00	0.00	0.00	7
5	0.00	0.00	0.00	1
accuracy			0.50	337
macro avg	0.08	0.17	0.11	337
weighted avg	0.25	0.50	0.33	337

Figure 19: Classification report of SVM classifier

- **Interpretation :**
  i) Although the overall accuracy is 50 percentage, this is deceptive because it only includes accurate predictions for the most common class (class 0).
  ii) The model appears to be fully recalling class 0 at the expense of incorrectly classifying all other classes into it, as evidenced by the class 0 precision of 0.50, recall of 1.00, and F1-score of 0.66.
  iii) All remaining classes (1 through 5) have zero precision, recall, and F1-score, confirming the model's total inability to differentiate and detect any class apart from class 0.
  iv) Model bias is evident from the macro average precision (0.08) and F1-score (0.11), which show that performance is critically poor when all classes are treated equally.
  v) Due to class 0's dominance, performance indicators are inflated while minority class performance is ignored, skewing the weighted average metrics.



(*a*) (*b*)

Figure 20: (a) Confusion Matrix and (b) ROC Curve.

- **(a) Interpretation :**
  i) All 167 instances of class 0 were correctly classified with zero error, indicating the model's strong bias toward this dominant class.
  ii) A total inability to recognize any minority class was demonstrated by the complete misclassification of all samples of class 1 (106 instances), class 2 (47 instances), class 3 (9 instances), class 4 (7 instances), and class 5 (1 instance) as class 0.
  iii) No off-diagonal elements exist except in the first column, indicating that the model is overconfidently and universally predicting class 0, regardless of true class.
  iv) A very unbalanced categorization behavior is reflected in this confusion matrix pattern, where prediction is based on class frequency rather than feature discrimination.

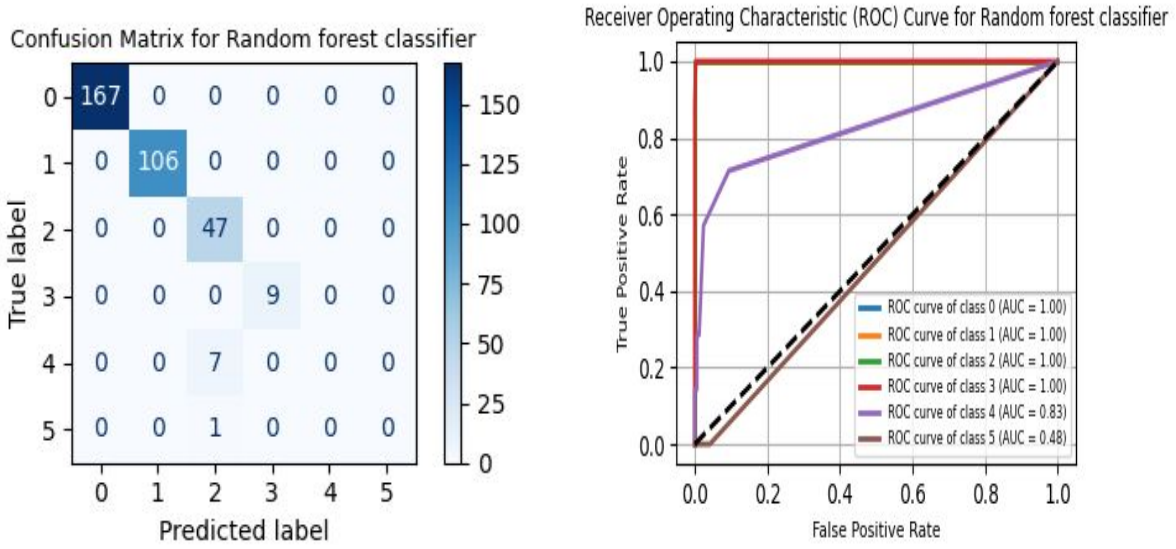
- **(b) Interpretation :**
  i) AUC ratings of 1.00 for classes 0 through 4 and 0.93 for class 5 indicate seemingly flawless separability, and the ROC curve for each class looks visually excellent.
  ii) As actual predictions were significantly biased toward class 0, this runs counter to the classification and confusion matrix results.
  iii) Although the model can provide well-separated probability scores, these high AUC values do not accurately predict the final class.
  iv) The paradox between high ROC-AUC and poor precision (and also recall) highlights the pitfall of using AUC as the sole performance metric, especially in imbalanced classification settings.
  v) The one-vs-rest technique, which is employed in multi-class scenarios and gives the impression of strong performance when one class dominates the predictions, has an impact on the ROC curves.

Classification Report (Random forest):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	167
1	1.00	1.00	1.00	106
2	0.85	1.00	0.92	47
3	1.00	1.00	1.00	9
4	0.00	0.00	0.00	7
5	0.00	0.00	0.00	1
accuracy			0.98	337
macro avg	0.64	0.67	0.65	337
weighted avg	0.96	0.98	0.97	337

Figure 21: Classification report of Random forest classifier

- **Interpretation :**
  i) Perfect precision, recall, and F1-score (1.00) were attained by classes 0 and 1, indicating that the model is very successful in accurately recognizing these majority classes with no false positives or false negatives.
  ii) Class 2 also performed well, with an F1-score of 0.92, precision of 0.85, and recall of 1.00, indicating that while the model properly captures all real class 2 instances, it occasionally mistakenly predicts other classes as class 2 (false positives).
  iii) Class 3 demonstrated exceptional identification of this small class, as seen by its perfect precision and recall (1.00) despite having only 9 instances.
  iv) Class 4 and Class 5 scored 0.00 in all metrics, showing the model failed completely to identify any of these minority class instances, due to severe class imbalance.
  v) Despite the excellent overall accuracy (0.98), the macro average (Precision: 0.64, Recall: 0.67, F1: 0.65) is much lower than the weighted average, indicating that minority classes perform poorly.



(*a*) (*b*)

Figure 22: (a) Confusion Matrix and (b) ROC Curve.

- **(a) Interpretation :**
  i) Strong diagonal values and zero off-diagonal elements for classes 0, 1, 2, and 3 in the confusion matrix demonstrate that these classes were predicted with complete accuracy.
  ii) All 7 instances of class 4 were misclassified as class 2 (6 instances) or class 3 (1 instance), indicating a clear confusion with neighboring or dominant classes.
  iii) The model's incapacity to identify class 5 was further demonstrated by the lone occurrence of class 5 being mistakenly classified as class 2.
  iv) The class imbalance issue is further illustrated by the matrix, which graphically displays the model's bias toward class 2, which receives inaccurate predictions from classes 4 and 5.

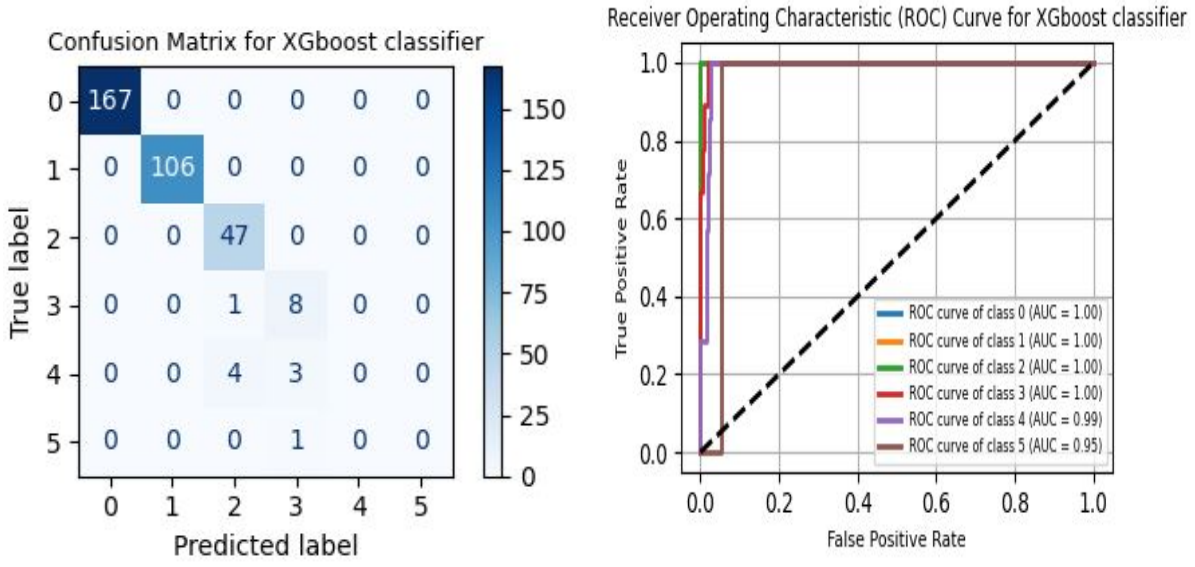
- **(b) Interpretation :**
  i) Class 0, 1, 2, and 3's ROC curves have AUC = 1.00, indicating perfect separability. So, the model can accurately differentiate these classes from others.
  ii) Class 4 indicates moderate separability with an AUC of 0.83. Although it is not ideal, it shows that the model can identify this class to some extent but not very well.
  iii) Class 5 indicates that the model has no discriminatory power because its AUC of 0.48 is poorer than that of random guessing.
  iv) The ROC analysis reinforces the findings from the classification report and confusion matrix, the model performs well on dominant classes but struggles or fails on underrepresented classes, necessitating class-balancing strategies.

Classification Report (XGBoost):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	167
1	1.00	1.00	1.00	106
2	0.90	1.00	0.95	47
3	0.67	0.89	0.76	9
4	0.00	0.00	0.00	7
5	0.00	0.00	0.00	1
accuracy			0.97	337
macro avg	0.60	0.65	0.62	337
weighted avg	0.95	0.97	0.96	337

Figure 23: Classification report of xgboost classifier

- **Interpretation :**
  i) Classes 0 and 1 performed flawlessly on these majority classes, as seen by their excellent precision, recall, and F1-score (1.00).
  ii) Class 2 likewise did very well, with an F1-score of 0.95, recall of 1.00, and precision of 0.90, indicating that although the model occasionally mispredicted class 2, it was able to catch all genuine occurrences.
  iii) With a precision of 0.67, recall of 0.89, and F1-score of 0.76, Class 3 performs somewhat well; while the majority of correct instances were found, there were a few misclassifications.
  iv) Precision, recall, and F1-score for classes 4 and 5 were all 0.00, indicating that the model was completely unable to identify these uncommon classes.
  v) The macro average (Precision: 0.60, Recall: 0.65, F1-score: 0.62) is much lower than the overall accuracy of 0.97, reinforcing the conclusion that minority class performance is weak, despite high accuracy on dominant classes.



(*a*) (*b*)

Figure 24: (a) Confusion Matrix and (b) ROC Curve.

- **(a) Interpretation :**
  i) The model demonstrated exceptional performance on the majority classes by properly classifying all 167 occurrences of class 0, all 106 instances of class 1, and all 47 instances of class 2 without any confusion.
  ii) Only one of the nine class 3 samples was incorrectly classified as class 2, while the other eight were correctly predicted. This illustrates how the model can learn from even tiny class samples as long as they are different and somewhat well-represented.
  iii) The seven class 4 instances were all incorrectly predicted. Instead, three were anticipated to be class 2 and four to be class 3. This implies that class 4's similarity to surrounding, more frequent classes confuses the model, and its rarity causes it to struggle.
  iv) Class 5 had zero recall and precision since its lone case was mistakenly classified as class 2. This demonstrates the model's incapacity to generalize on incredibly uncommon classes, which may be brought about by a lack of training data or the overwhelming dominance of class 2.
  v) The model's bias toward allocating examples to more often occurring or better-learned classes is demonstrated by the fact that all of the misclassified instances from classes 3, 4, and 5 were predicted.
  vi) Class imbalance is seen in the confusion matrix, where minority classes are dominated, leading to flawless performance on classes with high support and zero predictive power on uncommon ones.o be either class 2 or class 3.

- **(b) Interpretation :**
  i) The ROC curves for classes 0, 1, 2, and 3 show AUC = 1.00, reflecting excellent class separability for these categories.
  ii) Despite its failure in final classification, Class 4's AUC of 0.99 shows that the model can distinguish it at the probability level.
  iii) Although Class 5's AUC was lower at 0.95, it still shows a decent underlying probability separation despite the classifier's incorrect prediction. This implies that calibration or class weighting may be improved.
  iv) Overall, the ROC analysis suggests that although training imbalance or prediction limits make it difficult to classify minority classes, the model has learned to distinguish between them probabilistically, providing room for improvement.

5.2.2 Ensemble Model Based on Classification Models

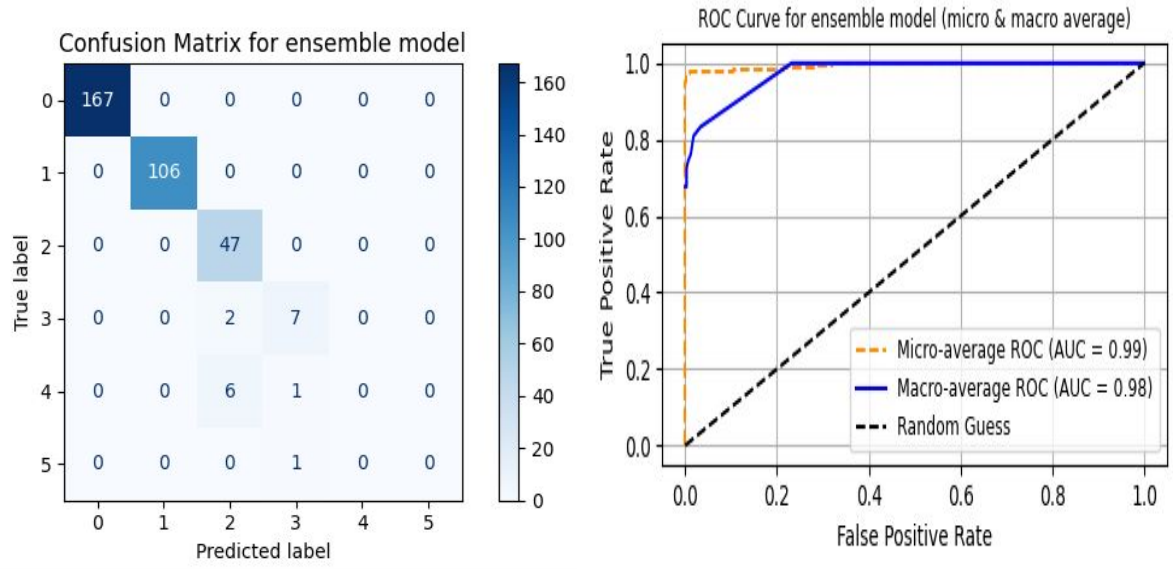
After applying these classification models, we have builded a ensemble model for checking whether it validating our previous models or not based on given models.

Classification Report (Ensemble Model):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	167
1	1.00	1.00	1.00	106
2	0.85	1.00	0.92	47
3	0.78	0.78	0.78	9
4	0.00	0.00	0.00	7
5	0.00	0.00	0.00	1
accuracy			0.97	337
macro avg	0.61	0.63	0.62	337
weighted avg	0.95	0.97	0.96	337

Figure 25: Classification report of ensemble model

- **Interpretation :**
  (i) For both class 0 (167 cases) and class 1 (106 instances), the model's precision, recall, and F1-score were all 1.00, demonstrating perfect prediction.
  (ii) The model detects all true occurrences of class 2 with a precision of 0.85, recall of 1.00, and F1-score of 0.92, but it sometimes incorrectly classifies additional examples as class 2.
  (iii) The model yielded 0.78 precision, recall, and F1-score, correctly predicting 7 out of 9 samples. However, 2 samples were misclassified, showing limited confusion with class 2.
  (iv) Class 4 and class 5 both classes received 0.00 for precision, recall, and F1-score. For class 4 (7 samples), 6 were classified as class 2 and 1 as class 3 and for class 5 (1 sample), the prediction went to class 2. The model failed to identify any true instance of these rare classes.
  (v) The model performs significantly better on majority classes than minority ones, as evidenced by the significantly lower macro average (precision: 0.61, recall: 0.63) compared to the weighted average (precision: 0.95, recall: 0.97).



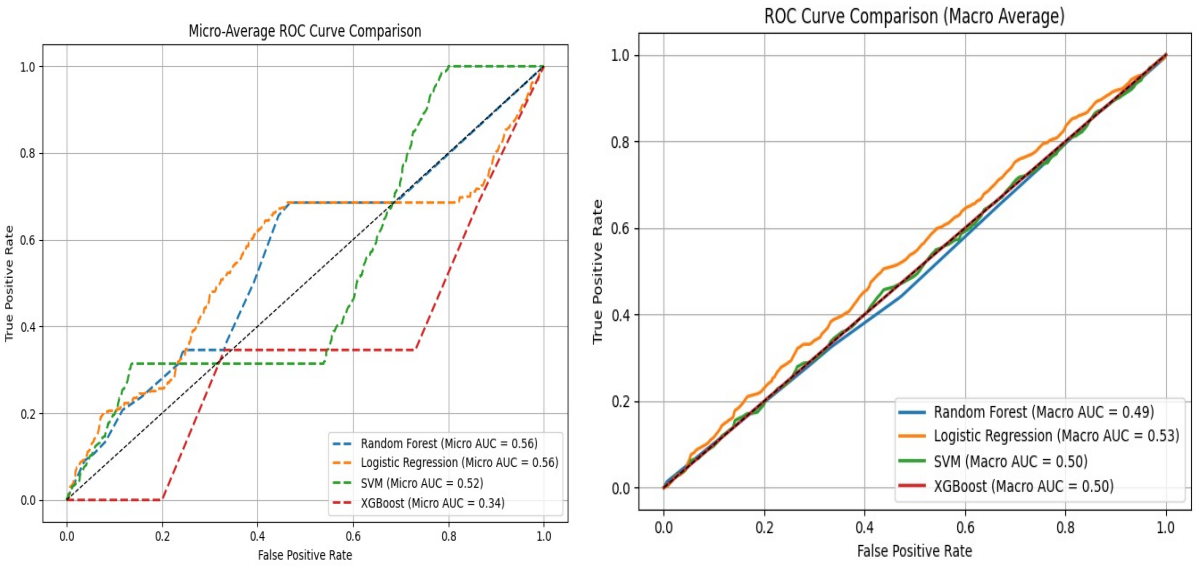
(a) (b)

Figure 26: (a) Confusion matrix and (b) ROC curve.

- **(a) Interpretation :**
  (i) The model demonstrated its ability to handle well-represented and identifiable classes by correctly predicting all samples of class 0 (167), class 1 (106), and class 2 (47).
  (i) Out of the 9 samples, 7 were correctly predicted, and 2 were incorrectly predicted as class 2, revealing mild confusion.
  (iii) Six of the seven class 4 samples were incorrectly classified as class 2, and one as class 3, indicating that the model is unable to distinguish between these classes.
  (iv) Class 2 was predicted for the lone class 5 sample. This suggests that this class has 0 sensitivity, most likely as a result of its low representation in the training set.

- **(b) Interpretation :**
  (i) Even though some individual class scores are low, the Ensemble model performs remarkably well in overall class discrimination, especially on average.
  (ii) This curves indicates that across all classes and instances, the classifier is highly capable of ranking positive instances above negative ones.
  (iii) Across all classes, the macro-average AUC (0.98) demonstrates good but not flawless performance. Even at the probability level, the minor decline from 1.00 indicates that certain classes likely class 4 and class 5 are not well-separated.
  (iv) Despite a high macro AUC, poor precision and recall for classes 4 and 5 suggest that classification thresholds and training data scarcity hinder performance for minority classes.

5.2.3 Comparison Between Classification Models



(a) (b)

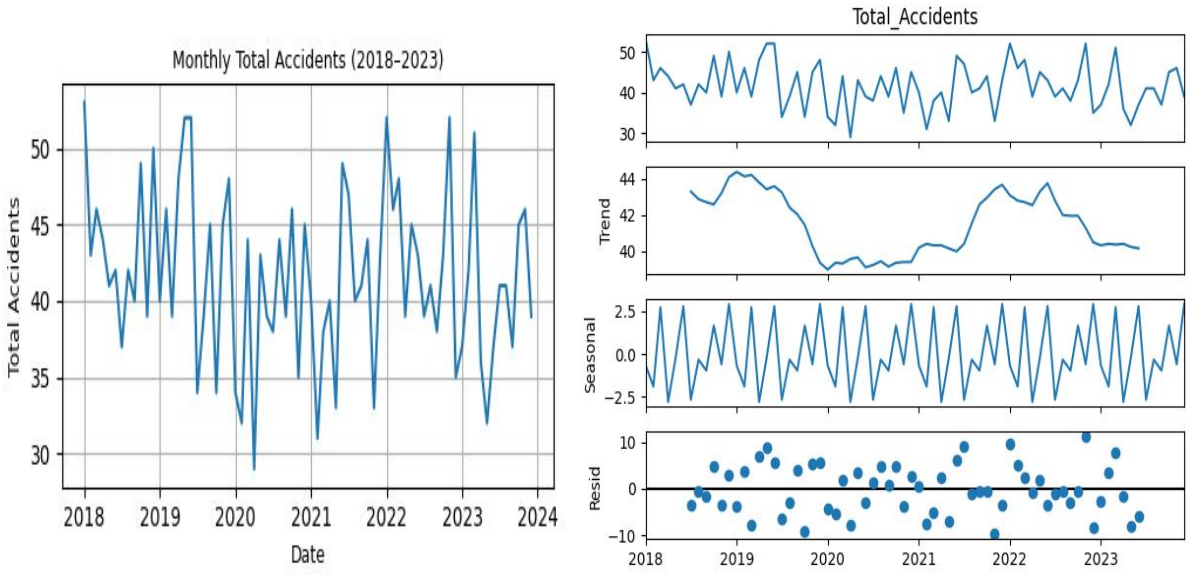
Figure 27: Combined ROC curve of (a) Micro average (b) Macro average.

- **(a) Interpretation :**
  (i) The Random Forest and Logistic Regression models both achieve a Micro AUC of 0.56, indicating slightly better than random performance, but still very weak for practical classification use.
  (ii) The SVM model performs worse, with a Micro AUC of 0.52, reflecting minimal ability to distinguish between classes.
  (iii) The XGBoost model shows the poorest micro-average performance with an AUC of only 0.34, which is worse than random guessing—a strong indicator of model misfit or severe bias in predictions.

- **(b) Interpretation :**
  (i) The Logistic Regression model has the highest Macro AUC of 0.53, indicating slightly better-than-random average performance across all classes.
  (ii) Both the SVM and XGBoost models have a Macro AUC of 0.50, implying no better than random performance on average across the individual class ROC curves.
  (iii) Random Forest, with a Macro AUC of 0.49, actually performs slightly worse than random guessing in the per-class average sense, suggesting it may be highly biased toward dominant classes.

5.2.4 Time Series Modelling and Analysis

In this part, we were interested to predict the total number of accidents in India based on previous data of all states monthly accident counts. That is why we have applied time series model.



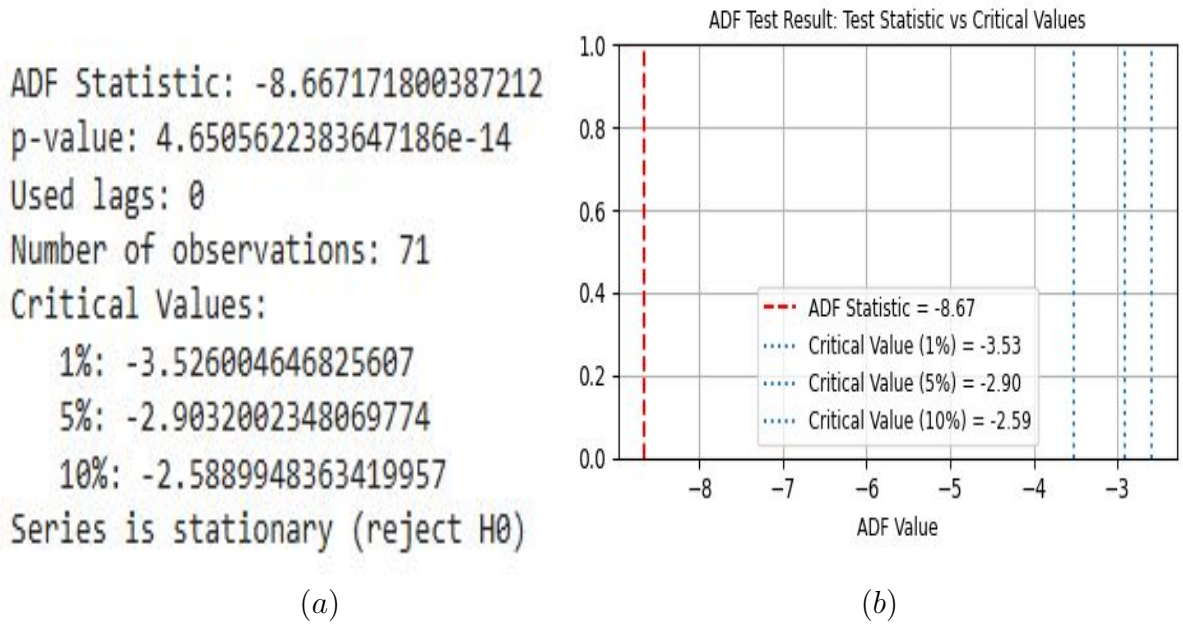
(a) (b)

Figure 28: (a) Monthly accident counts and (b) Seasonal decomposition.

- (a) As the monthly accident series cycles through a clear, repeatable seasonal pattern of peaks and troughs each year, the counts actually conceal a subtle trend, drifting downward from 2018 into early 2020, flattening out at a lower level through 2021, then edging upward during 2022 before settling again in 2023. Therefore, what appears to be random volatility in the raw plot is actually a combination of a strong month-to-month seasonality and a gentle U-shaped multi-year trend, with residual noise uniformly distributed about zero, suggesting that no other systematic factors are at work.

- (b) Important trends are revealed by the time series decomposition of Total Accidents from 2018 to 2023. A strong and regular yearly pattern is shown by the seasonal component, suggesting consistent month-to-month variation across years. The residuals are randomly distributed around zero with no discernible structure, suggesting that the model has captured the majority of the systematic variation through trend and seasonality. The trend component shows a general decline from 2018 to 2020, stabilizing in 2021 and slightly rising in 2022 before leveling off again in 2023.

  Hence, there is a clear existence of seasonality. That was why we have applied SARIMA model. For applying this model at first we have checked stationarity of the series by ADF test because model assume that the series is stationary.



(a) (b)

Figure 29: ADF test (a) Result and (b) Visualization plot.

The monthly total accident series is stationary, according to the results of the Augmented Dickey-Fuller (ADF) test. The null hypothesis(H_0) of non-stationarity is strongly rejected by the ADF statistic of -8.67, which is significantly lower than the critical values (3.53, 2.90, and 2.59, respectively), and the p-value is nearly zero. The basic structure of the time series is statistically stable across time, making it suitable for time series modeling without differencing, even in the face of obvious seasonal and trend components. This validates the earlier decomposition analysis.

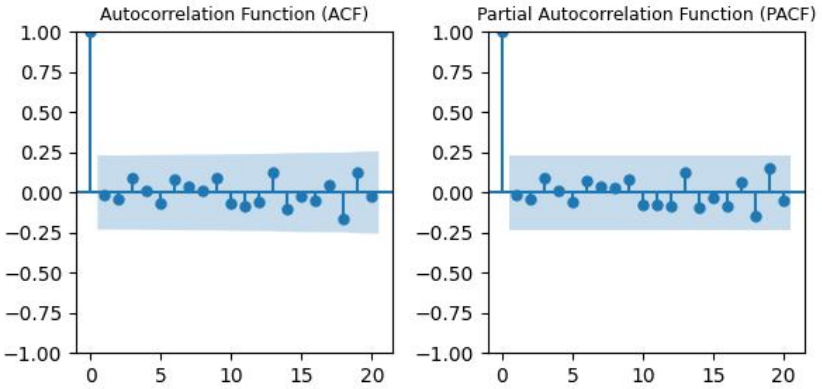


Figure 30: ACF and PACF plot

The ACF and PACF plots show that all lag values fall within the confidence interval, indicating a lack of significant autocorrelation or partial autocorrelation. This suggests the series has no strong dependency on past values, and both AR and MA terms may not be necessary. A simple SARIMA model with minimal non-seasonal components is likely sufficient.

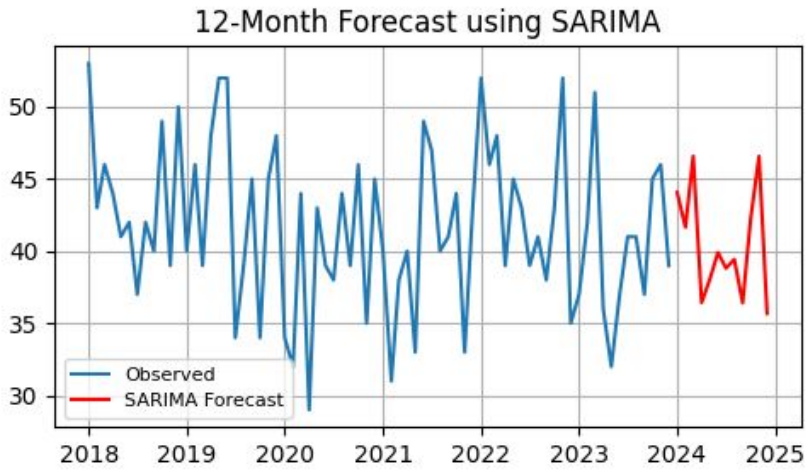
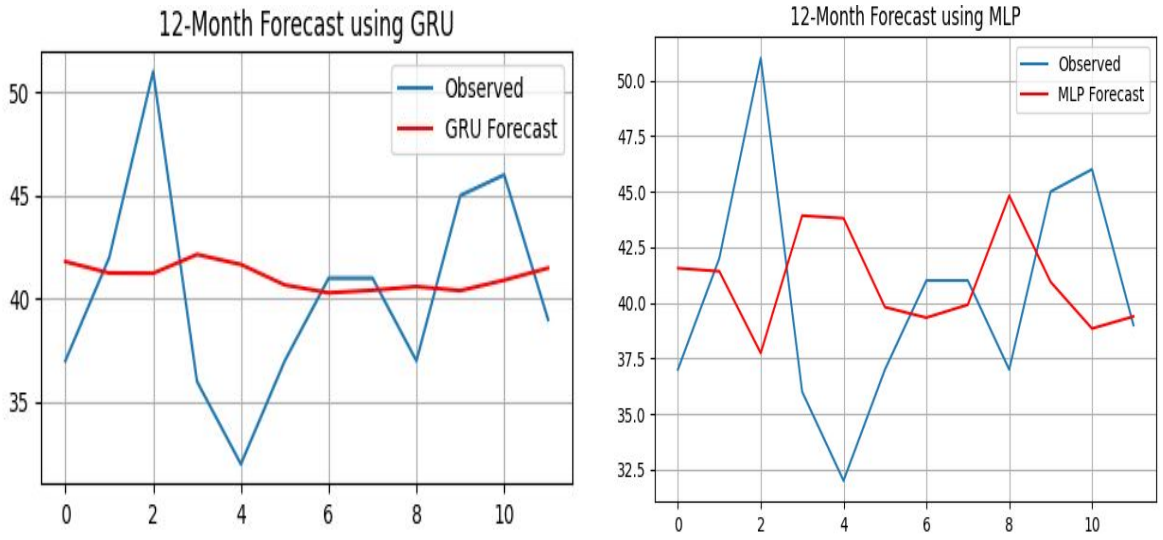


Figure 31: SARIMA forecasting

The smooth shift from observed (blue) to predicted (red) values indicates that the SARIMA model offers a reliable 12-month projection. The forecast maintains the seasonal variations and amplitude range without any sudden changes, which is in good agreement with the historical trend. The model has successfully captured the underlying seasonality and trend, as evidenced by the visual continuity and structural consistency, indicating strong predictive performance and a reliable fit.

5.2.5 Deep Learning Algorithmic Approach

We know that deep learning algorithms are strong in predicting staffs. After applying SARIMA model, we have applied MLP and GRU model, respectively.



(a) (b)

Figure 32: 12-Month forecast of (a) GRU and (b) MLP.

- (a) The GRU model provides a stable and smooth forecast that aligns well with the overall level and long-term trend of the observed data, suggesting good gener-

alization and resistance to noise. This reflects its strength in capturing the central tendency without overfitting. However, the model struggles to reflect short-term variations and seasonal spikes, leading to underestimation of peaks and overestimation of troughs, which indicates a limited ability to model fine-grained temporal dynamics.

- (b) The MLP model captures the general direction and some turning points of the time series, reflecting a moderate understanding of short-term trends and fluctuations. Its forecasts show more variability than GRU, allowing better alignment with observed dynamics. However, it still misses sharper peaks and troughs, and tends to smooth out extreme changes, indicating partial underfitting and limited capacity to capture complex temporal patterns compared to recurrent models.

5.2.6 Model Comparison

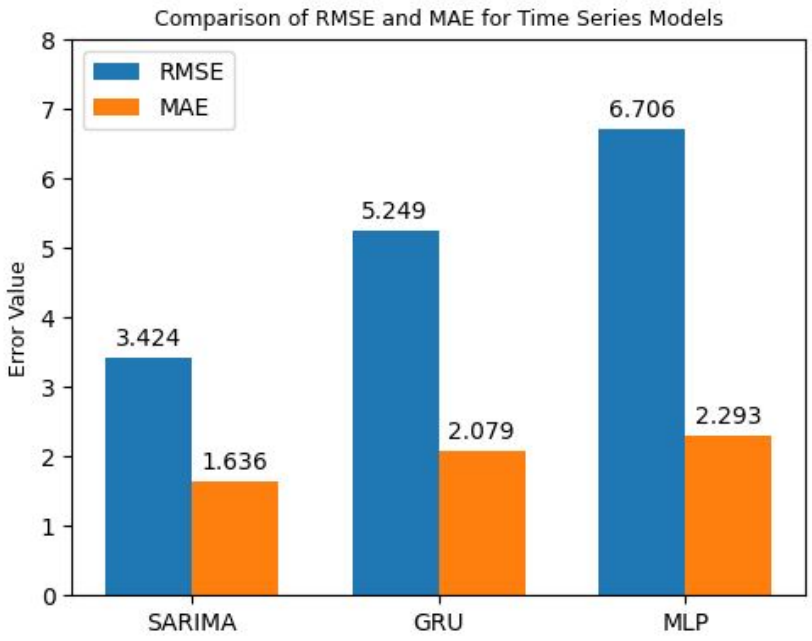


Figure 33: Comparison between forecasting

The SARIMA model shows the best performance with the lowest RMSE and MAE, indicating strong accuracy in capturing trends and seasonality. GRU performs moderately well but has higher errors, while MLP shows the poorest fit with the highest error values, reflecting weaker handling of temporal patterns.

6 Conclusion

Through rigorous data preprocessing, feature engineering, and the application of diverse modeling techniques, the study identifies critical factors—such as adverse weather, unsafe road infrastructure, and alcohol involvement—that disproportionately contribute to accident severity.

Daily data models performed poorly due to high volatility and class imbalance, with most classifiers showing AUCs around 0.50–0.52, implying near-random performance.

XGBoost marginally outperformed others, offering slightly better class separation for fatal and serious accidents.

Key risk factors identified include hazardous road conditions, alcohol consumption, poor lighting, and weather (fog, rain, haze).

Monthly aggregated data provided more stable trends and allowed for effective time series forecasting using SARIMA and neural models like GRU.

The study highlighted the need for balanced data, context-aware modeling, and integration of real-time traffic/weather data for future work.

It serves as a data-driven foundation for road safety reform, aiding emergency response planning, urban traffic design, and public awareness. The findings not only advance methodological approaches in accident analytics but also provide actionable insights for policymakers and stakeholders aiming to implement evidence-based road safety reforms.

This project lays a strategic foundation for data-driven interventions that can ultimately contribute to safer roads and the reduction of accident-related fatalities and injuries across India.

Future Study

- In our monthly dataset, observations for class 4 and 5 (i.e., total number of accidents 5 and 6, respectively) much less for building predictive models. So in future we will try to generate random sample by simulation for getting better reliable and balanced evaluation.
- We will try to apply more deep learning algorithms on our monthly data.
- We will try to fit time series model for each states and will analyze the prediction of total number of accidents per state.
- We will make a comparative study between 12, 6 and 3 months based time series modelling accuracy for both month-wise and state-wise cases.

7 References

1. Dombalyan, Anzhelika, Viktor Kocherga, Elena Semchugova, and Nikolai Negrov. *"Traffic forecasting model for a road section."* Transportation Research Procedia 20 (2017): 159-165.

2. Cuenca, L.G., Puertas, E., Aliane, N. and Andres, J.F., 2018, September. *Traffic accidents classification and injury severity prediction.* In 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE) (pp. 52-57). IEEE.

3. Khanum, H., Garg, A. and Faheem, M.I., 2023. *Accident severity prediction modeling for road safety using random forest algorithm: an analysis of Indian highways.* F1000Research,12,p.494.

4. Zong, Fang, et al. *"Predicting severity and duration of road traffic accident."* Mathematical Problems in Engineering 2013.1(2013):547904.

5. Hayakawa, Hiroshi, Paul S. Fischbeck, and Baruch Fischhoff. *"Traffic accident statistics and risk perceptions in Japan and the United States."* Accident Analysis Prevention 32.6 (2000):827-835.

6. Rahim, Md Adilur, and Hany M. Hassan. *"A deep learning based traffic crash severity prediction framework."* Accident Analysis Prevention 154 (2021): 106090.

7. **GeeksForGeeks** – Taken some definitions related to machine learning algorithms from this website.

8. **Google Colab** – Cloud based environment used for coding, analysis and visualizations.

9. **Github Links of Python Codes :**
(i) `https://github.com/N125-ui/GOOGLE-COLAB/blob/main/ISI_PROJECT.ipynb`
(ii) `https://github.com/N125-ui/GOOGLE-COLAB/blob/main/PROJECT_PART_2.ipynb`