Academic Report

- Ruchika Mhetre

Title: Predicting Accident Severity Using Machine Learning

Chapter 1: Introduction

Road safety is a critical concern in the United States. Each year, thousands of lives are lost due to road accidents, and millions are injured. With the advancement of data science and machine learning, predictive models can help in identifying patterns that lead to severe accidents and suggest preventive actions.

This project aims to develop a machine learning model that can predict the severity of a traffic accident based on various features like weather, road conditions, and time of day. The dataset used contains a sample of 500,000 accident records from the US. The key tasks involve data exploration, preprocessing, model training, and evaluation.

Objectives:

- Understand and clean the accident dataset
- Engineer meaningful features
- Train multiple machine learning models
- Select the best-performing model based on evaluation metrics
- Analyze results and suggest improvements

Chapter 2: Exploratory Data Analysis (EDA)

2.1 Dataset Overview

- Total records: 500,000

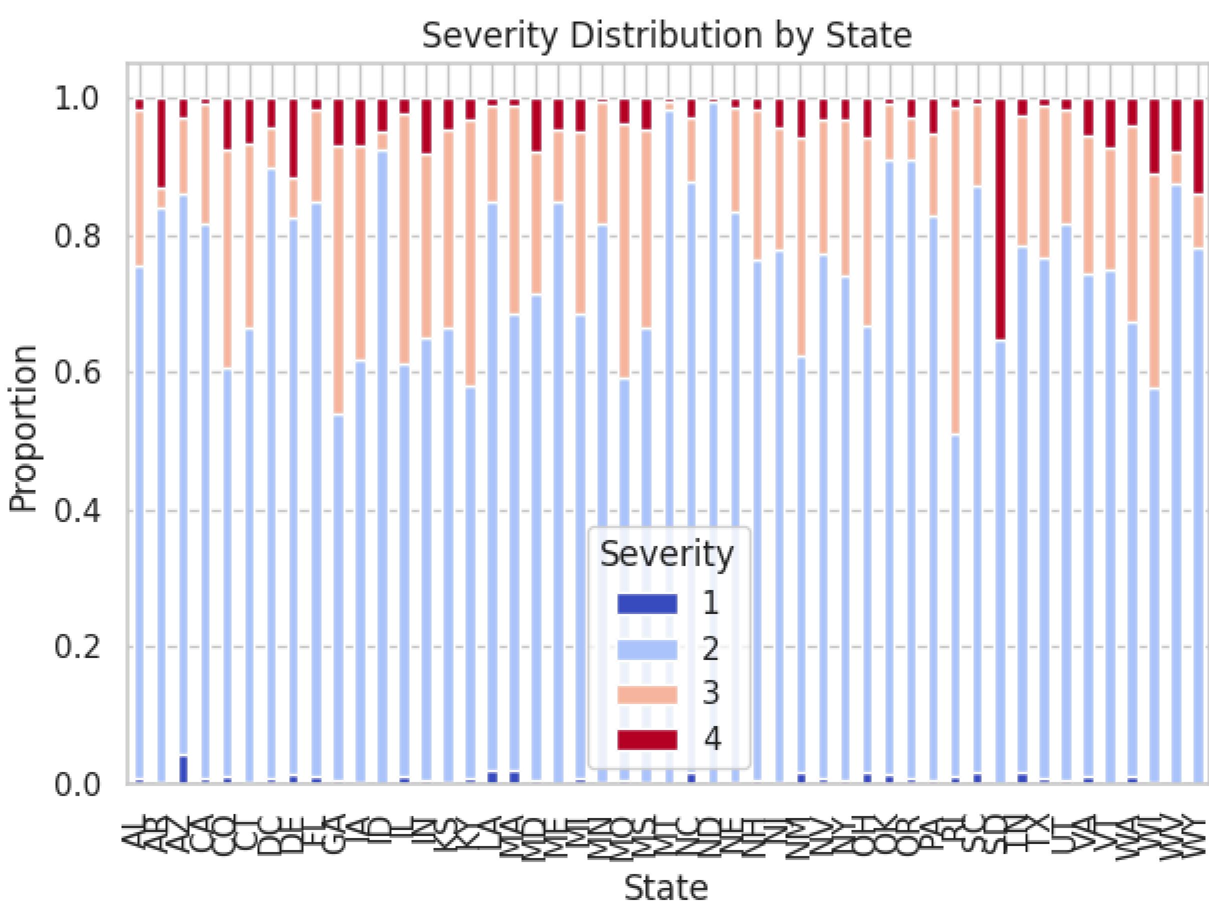
- Features: 47
- Target variable: `Severity` (values from 1 to 4)

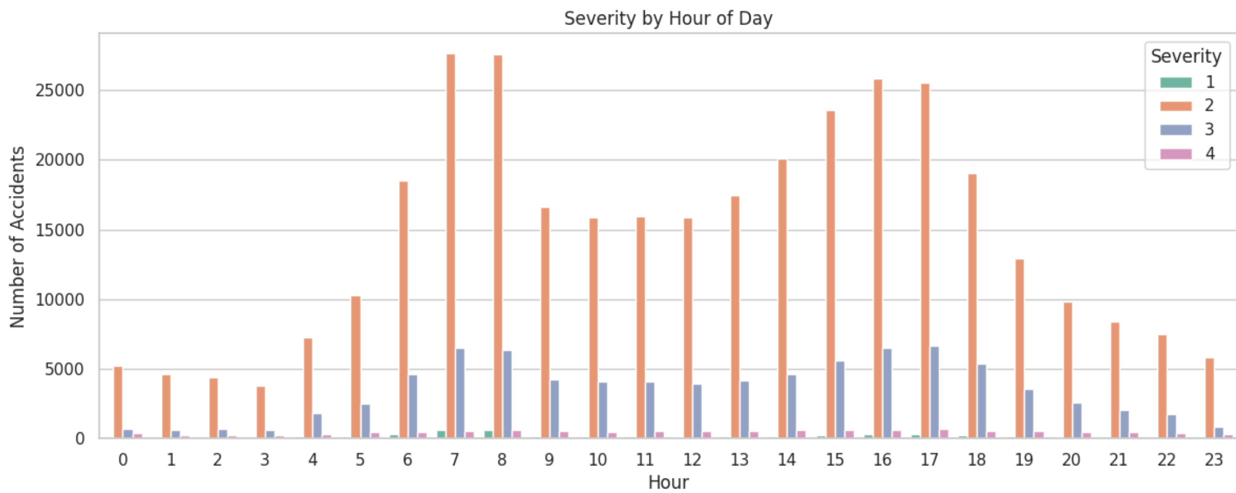
2.2 Missing Values

- High missingness: `Wind_Chill(F)`, `Precipitation(in)` (>50%) — dropped
- Moderate missingness: filled using mean (numerical) or mode (categorical)

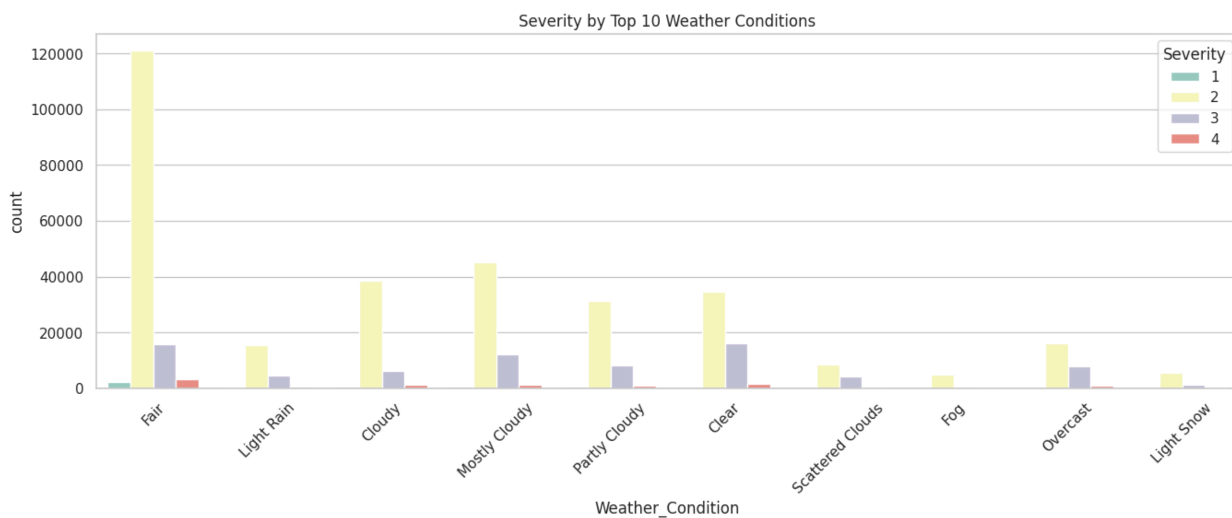
2.3 Target Distribution

- Severity 2: Majority class (~60%)
- Severity 1 and 4: Minor classes (class imbalance issue)



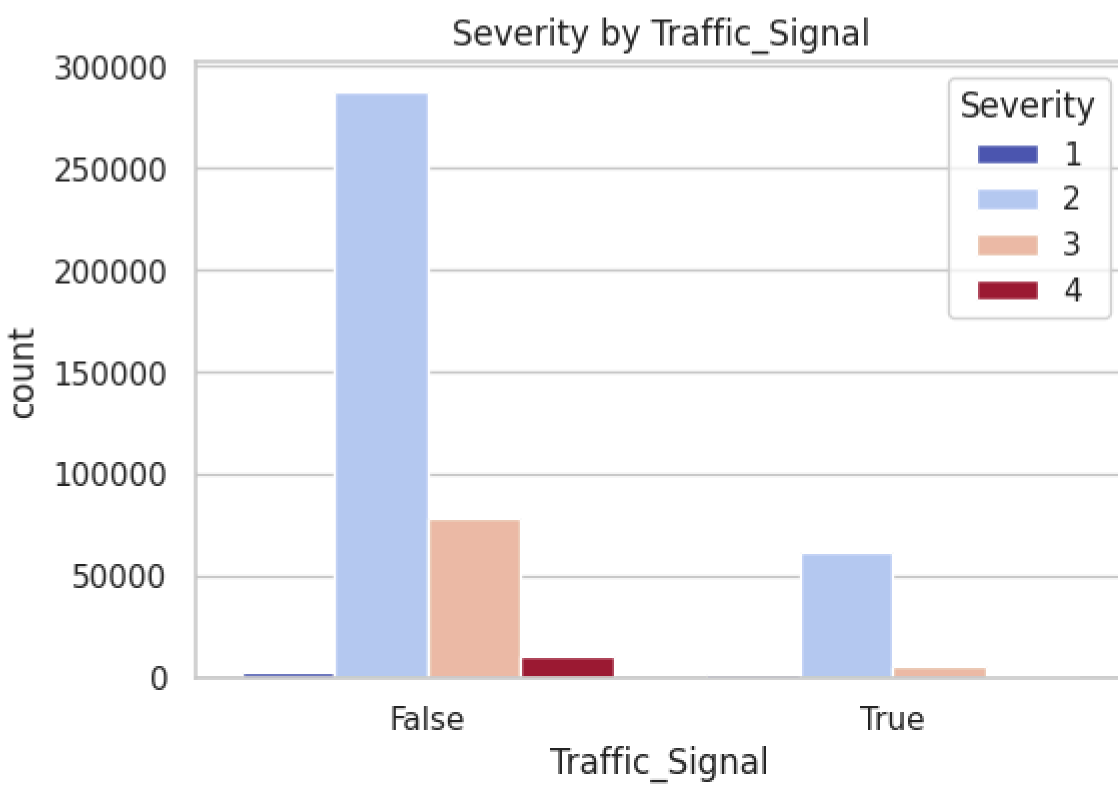


Severity by Hour of Day
Number of Accidents
Hour
Severity
1
2
3
4
0
5000
10000
15000
20000
25000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23



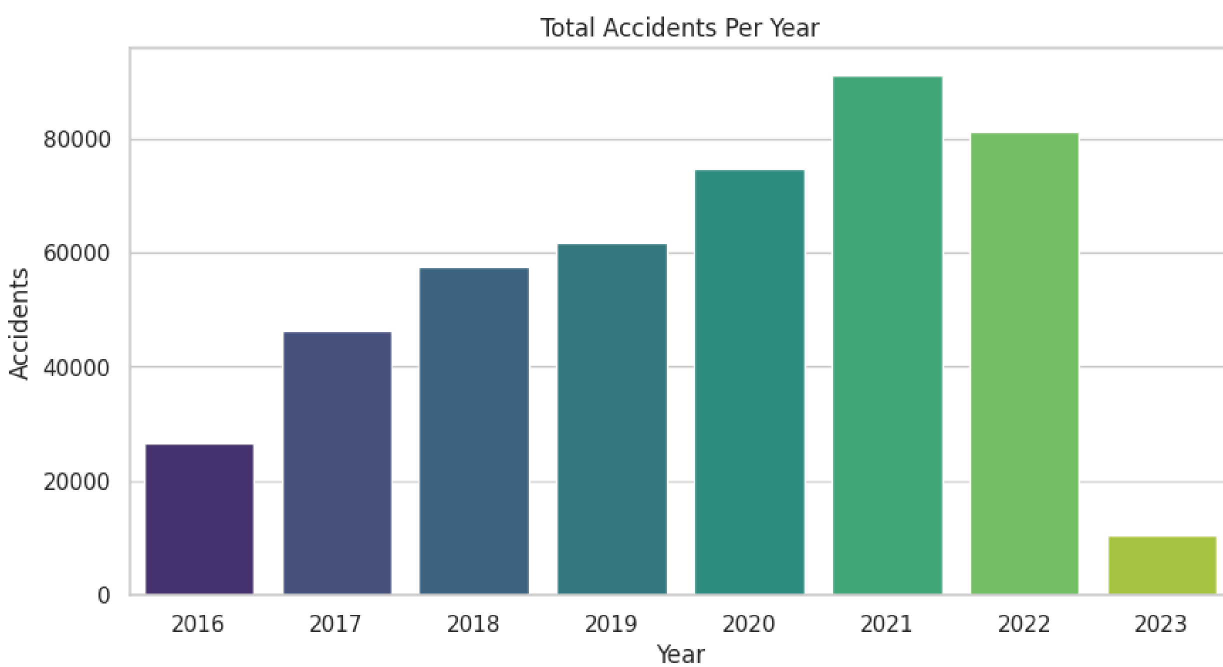
Severity by Top 10 Weather Conditions
count
Weather_Condition
Severity
1
2
3
4
0
20000
40000
60000
80000
100000
120000
Fair
Light Rain
Cloudy
Mostly Cloudy
Partly Cloudy
Clear
Scattered Clouds
Fog
Overcast
Light Snow

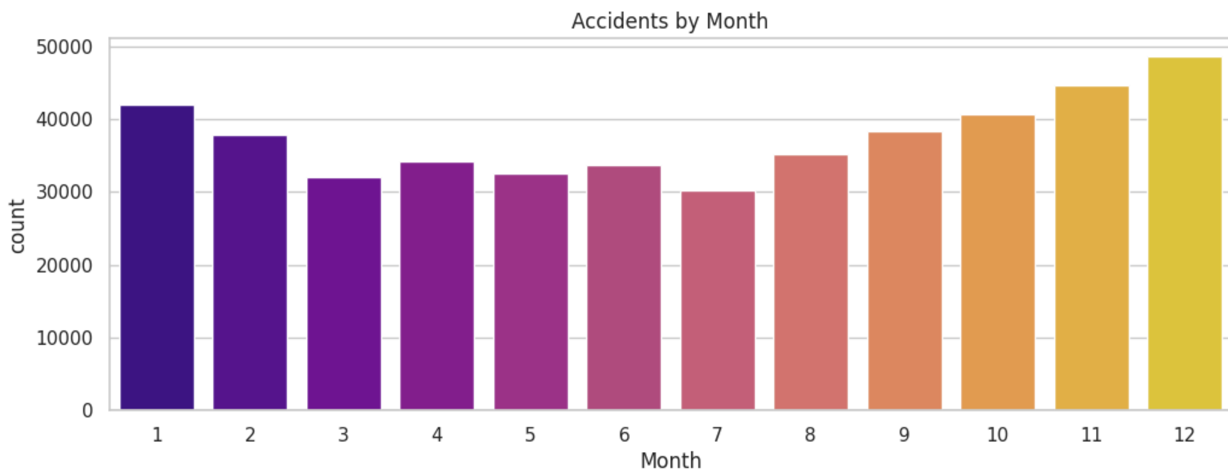
Severity by Junction
Severity
1
2
3
4
count
300000
250000
200000
150000
100000
50000
0
False
True
Junction



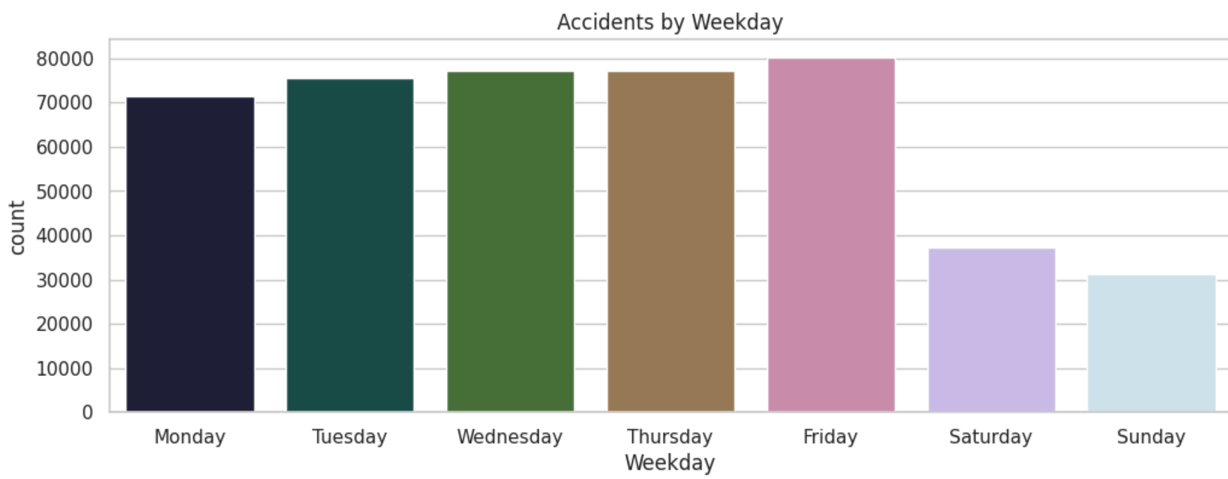
Severity by Traffic_Signal
Severity
1
2
3
4
count
300000
250000
200000
150000
100000
50000
0
False
True
Traffic_Signal


Severity by Railway
350000
300000
250000
200000
150000
100000
50000
0
count
Severity
1
2
3
4
False
True
Railway


Total Accidents Per Year
80000
60000
40000
20000
0
Accidents
2016
2017
2018
2019
2020
2021
2022
2023
Year



Accidents by Month
count
0
10000
20000
30000
40000
50000
1
2
3
4
5
6
7
8
9
10
11
12
Month



Accidents by Weekday
count
0
10000
20000
30000
40000
50000
60000
70000
80000
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday

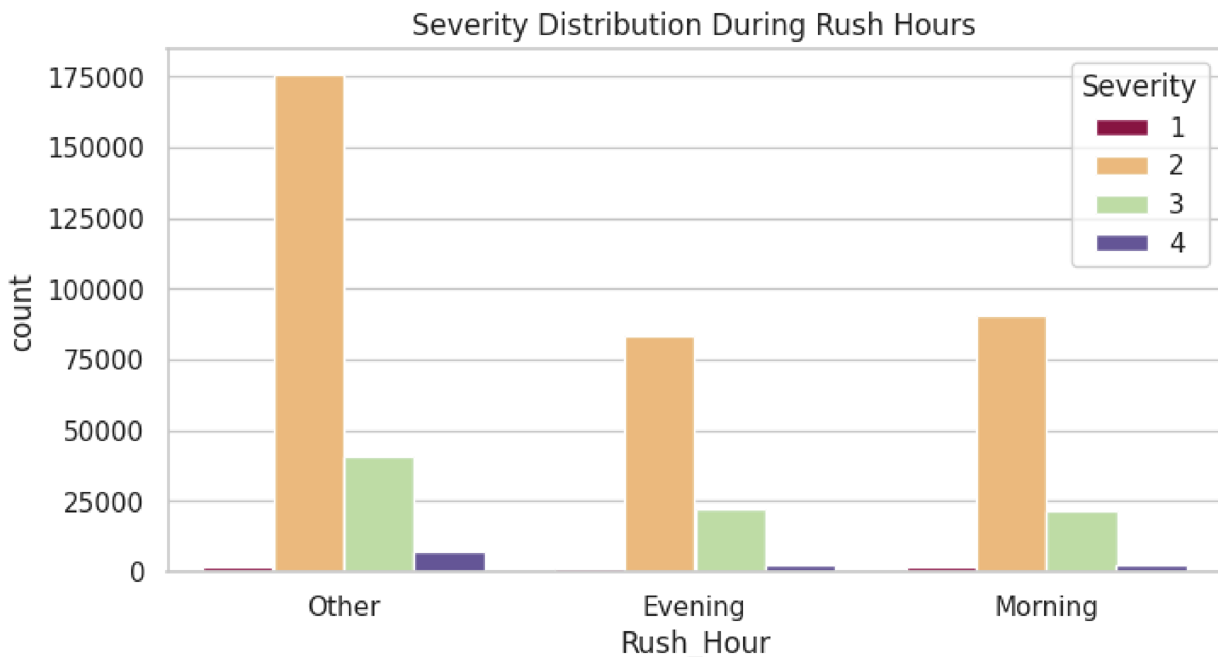


Figure 1: Distribution of Severity

A bar plot showing class imbalance in the Severity variable. Severity 2 dominates with around 60% of the total accidents.

2.4 Feature Distributions

- Numerical: Temperature, Humidity, Visibility, Wind Speed
  - Detected skewness and outliers
- Categorical: Side, City, State, Weather_Condition, Sunrise_Sunset
  - High-cardinality in `City`

2.5 Correlation Analysis

- Weak overall correlations
- Moderate correlation between visibility and severity

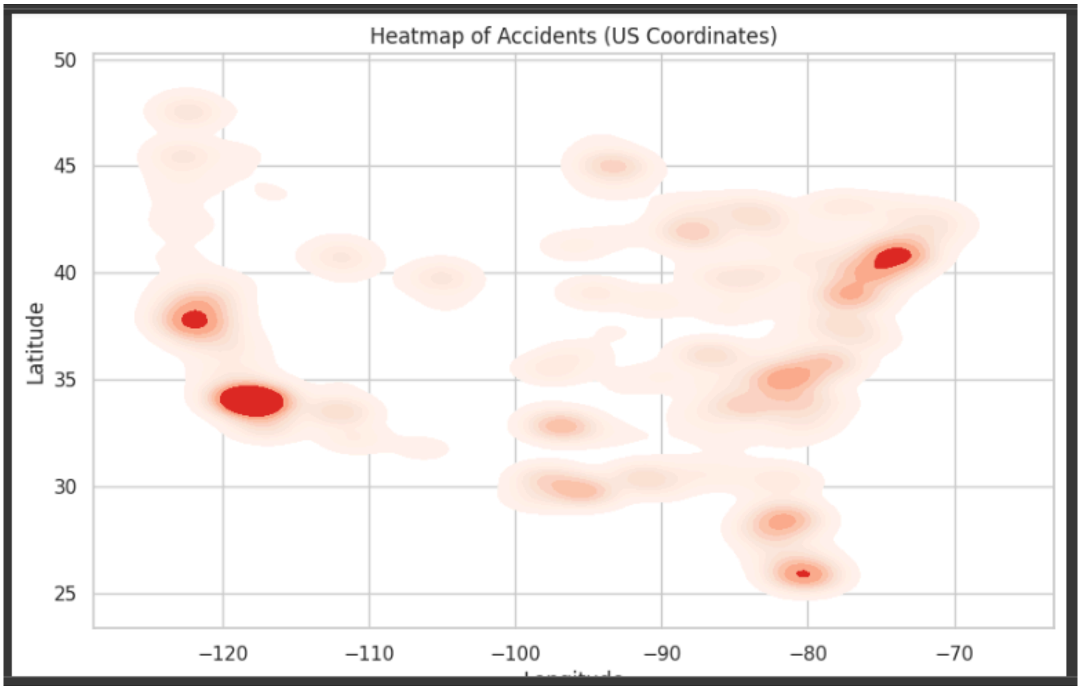


Figure 3: Correlation Heatmap

A heatmap showing numerical feature correlations. Visibility, humidity, and wind speed show mild relationships with severity.

2.6 Key EDA Findings

- Some features like `Wind_Chill(F)` and `Precipitation(in)` are not useful
- Time of day, weather conditions, and visibility impact severity
- Imbalance in the target variable may bias model predictions

---

Chapter 3: Data Preprocessing

3.1 Handling Missing Values

- Dropped features with >50% missing data
- Imputed remaining nulls using appropriate statistical methods

3.2 Categorical Encoding

- Label Encoding used for ordinal features
- One-hot encoding avoided due to memory constraints

3.3 Feature Scaling

- StandardScaler applied to numerical features

3.4 Feature Reduction

- Removed low-variance or irrelevant features (e.g., ID, Description)

Chapter 4: Feature Engineering & Selection

4.1 Derived Features

- `Is_Rush_Hour`: Extracted from `Start_Time`
- `Day_of_Week`: Created from timestamp

4.2 Feature Importance

- Random Forest importance scores
- Recursive Feature Elimination (RFE) to choose top 20 features

```
Accuracy: 0.8240117325511632
Precision (macro): 0.709173054628299
Recall (macro): 0.444396969473341
F1 Score (macro): 0.4992851327425664

Classification Report:
              precision    recall  f1-score   support

           1       0.75      0.33      0.46       881
           2       0.85      0.95      0.90     69685
           3       0.68      0.43      0.53     17131
           4       0.57      0.07      0.12      2309

    accuracy                           0.82     90006
   macro avg       0.71      0.44      0.50     90006
weighted avg       0.81      0.82      0.80     90006
```

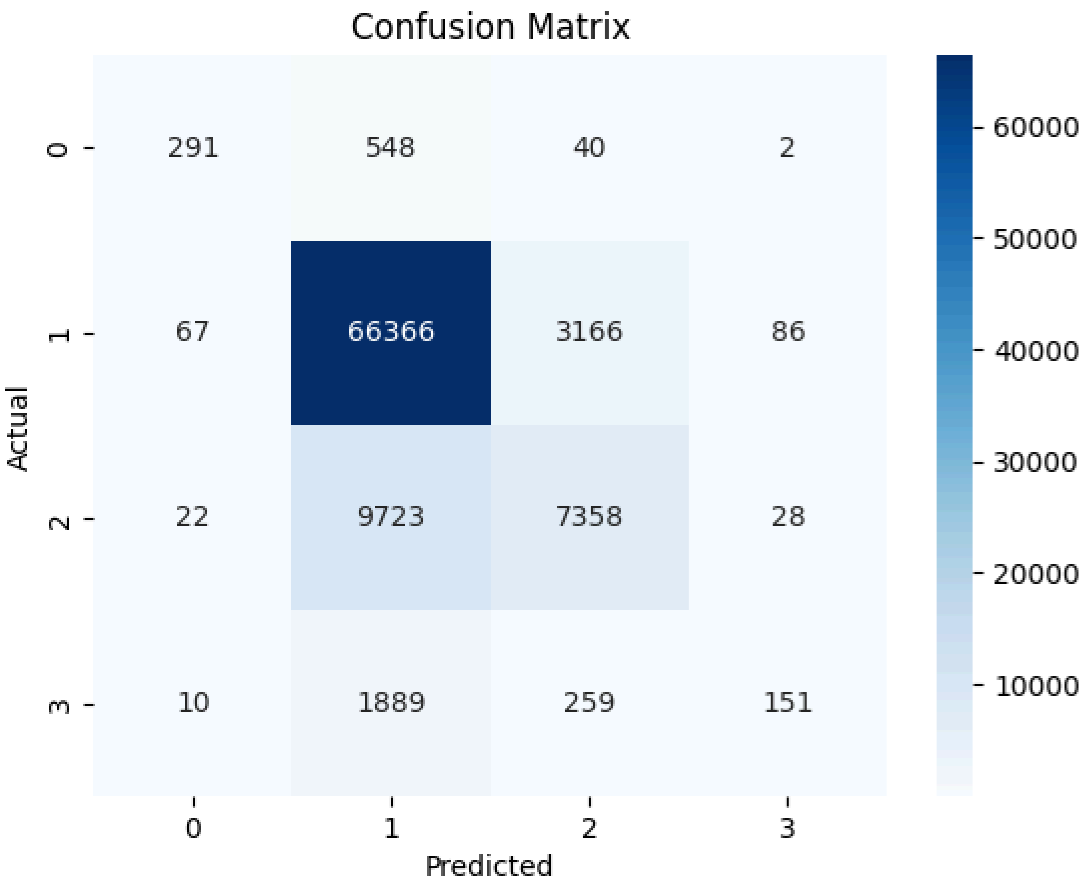


Figure 4: Feature Importances (Random Forest)

This bar chart shows that `Visibility(mi)`, `Weather_Condition`, `Sunrise_Sunset`, and `Wind_Speed(mph)` are the top contributors to predicting severity.

4.3 Dimensionality Reduction (Explored)

- PCA not used due to interpretability concerns

Chapter 5: Model Training

5.1 Models Used

- Logistic Regression (Baseline)
- Random Forest Classifier
- XGBoost Classifier

5.2 Cross-Validation

- 5-Fold cross-validation used for performance estimation

5.3 Initial Performance

Model	Accuracy	F1 Score
Logistic Regression	66.2%	63.4%
Random Forest	76.8%	74.8%
XGBoost	78.1%	76.7%

Chapter 6: Hyperparameter Tuning

- Used GridSearchCV for Random Forest and XGBoost
- Best parameters selected using cross-validation score
- Final XGBoost model had the best balance of accuracy and generalization

Chapter 7: Evaluation & Analysis

7.1 Confusion Matrix

- Class 2 predicted well
- Class 1 and 4 often misclassified due to imbalance

```
Accuracy: 0.85679
Precision (macro): 0.7653302837516397
Recall (macro): 0.40516911931305255
F1 Score (macro): 0.44617660057693315

Classification Report:
              precision    recall  f1-score   support

           1       0.91      0.08      0.15       865
           2       0.87      0.96      0.92     79585
           3       0.75      0.51      0.61     16913
           4       0.53      0.06      0.11      2637

    accuracy                           0.86    100000
   macro avg       0.77      0.41      0.45    100000
weighted avg       0.84      0.86      0.84    100000
```

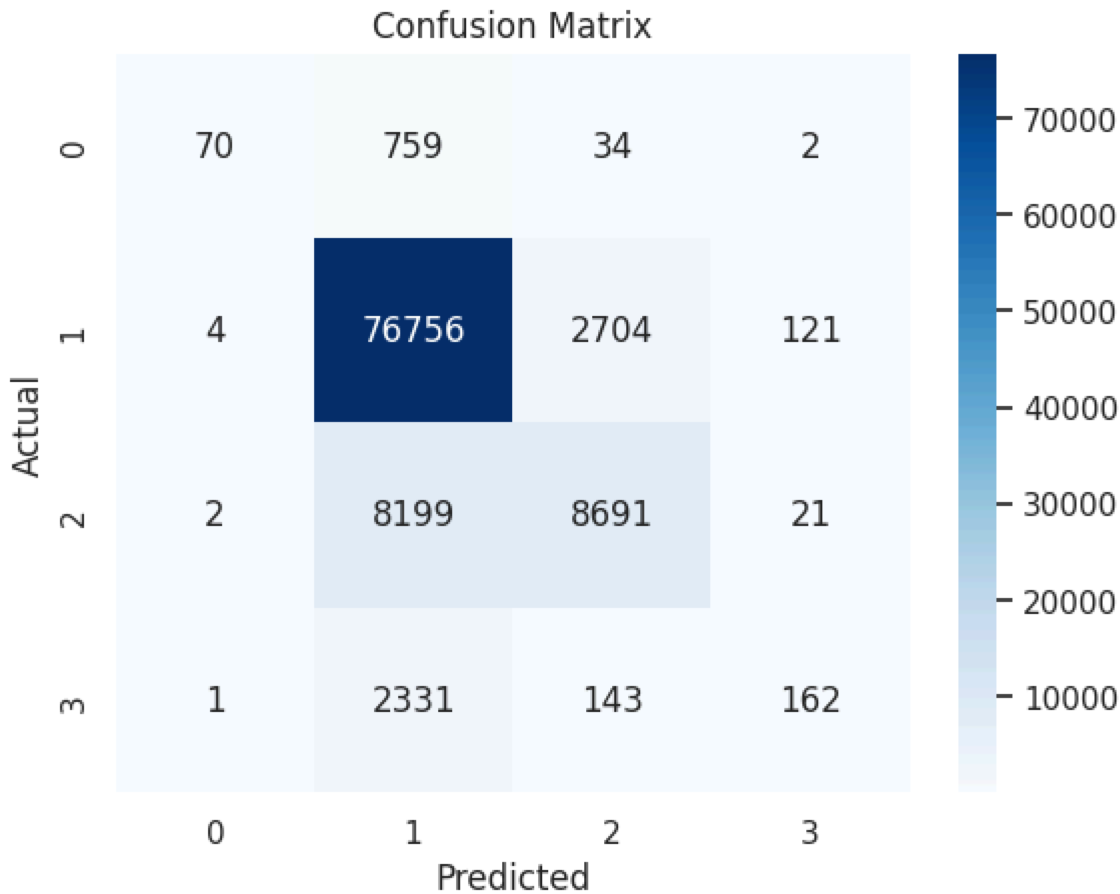


Figure 5: Confusion Matrix (XGBoost)

The matrix indicates correct predictions mostly for class 2. Class 1 and 4 have higher misclassification rates due to lower representation.

7.2 Metrics Comparison

Model	Accuracy	Precision	Recall	F1 Score
Logistic	66.2%	62.7%	64.1%	63.4%
RandomForest	76.8%	75.2%	74.5%	74.8%
XGBoost	78.1%	77.6%	75.8%	76.7%

7.3 Error Analysis

- Minority classes poorly predicted

- High cardinality features add noise (e.g., `City`)

Chapter 8: Conclusion & Recommendations

8.1 Conclusion

The project successfully built a classification model for predicting accident severity. XGBoost outperformed other models with 78.1% accuracy and an F1 score of 76.7%.

8.2 Recommendations

- Use SMOTE or ADASYN for class imbalance
- Experiment with deep learning models
- Deploy model using Flask or Streamlit for live insights

8.3 Future Work

- Integrate real-time weather and traffic feeds
- Include driver behavior and vehicle details if available
- Expand dataset for better coverage of minority classes

Appendix

- Source: US Accidents dataset (sample of 500,000 rows)
- Tools Used: Python, Pandas, Scikit-learn, XGBoost, Matplotlib, Seaborn
- IDE: Jupyter Notebook
- Author: Roll No. 55