US Traffic Accidents: Trend Analysis and Severity Classification

Lohitha Vanteru, Mahe Jabeen Abdul, Pranavi Sandrugu, Sadakhya Narnur

Department of Applied Data Science, San Jose State University

DATA-240

Dr. Taehee Jeong

May 4, 2023

Abstract

Roads are shared by various means of transportation such as cars, trucks, buses, motorcycles, pedestrians, and animals, and they play a significant role in the economic and social growth of many nations. However, every year, a large number of vehicles are involved in collisions that result in numerous fatalities and injuries. There has been a rise in the number of road accidents, posing a challenge to governments, individuals, and communities as these accidents can be deadly and hazardous to society. This paper aims to address this issue by examining the primary factors that contribute to the increase in the rate of car accidents and develop a predictive model that accurately identifies accident-prone areas and helps reduce the frequency and severity of accidents in the USA. The data utilized in this study was obtained from traffic accidents recorded by the United States Department of Transportation, law enforcement agencies, and traffic cameras between 2016 and 2021 through multiple data providers that include various APIs which provide streaming traffic event data and US Census Demographic data that contains information regarding the population of each state and county in the United States. The models utilized in this research are the Apriori algorithm to provide recommendations based on the association rules, the Decision tree classifier with SMOTE for addressing the class imbalance, and the BERT model to predict the consequences of car accidents on road traffic, with a particular emphasis on identifying the main factors that contribute to road accidents. The decision tree classifier despite balanced classes done with SMOTE gave a low accuracy of 71%. The BERT model for solving this problem showed a greater accuracy for very fewer data with 85% accuracy in classifying the severity of accidents. The findings from this study can potentially inform the development of strategies and interventions aimed at reducing the frequency and severity of car accidents in urban areas. These could include initiatives such as encouraging the adoption of autonomous

vehicles, improving public transportation infrastructure, and implementing measures to spread out peak traffic times.

1. Introduction

Road traffic accidents (RTAs) can have serious consequences, resulting in injuries, fatalities, and significant property damage. Unfortunately, the number of road traffic accidents has been increasing due to various factors such as population growth, urbanization, and increased mobility. This trend is not only alarming but also highlights the need for effective and efficient methods to prevent and manage road traffic accidents. The National Highway Traffic Safety Administration has reported that there are over 37,000 fatalities and 2.35 million injuries or disabilities caused by road accidents in the United States annually. The economic cost of these accidents is around $230.6 billion per year, with an average cost of $820 per person. On a global scale, accidents are ranked as the 9th leading cause of death. To gain a comprehensive understanding of the patterns, reasons, and resolutions of this problem, a more detailed analysis is required. The objective of this research is to analyze historical accident data to identify patterns and trends in accident occurrence, contributing factors, and potential solutions.

Data mining is an effective method for drawing essential conclusions and information from vast and complicated databases. Data mining can be used to analyze crash data and identify the causes of accidents in the context of driving safety. We can anticipate the risk of an accident and take proactive steps to prevent them by employing data mining tools. Data mining techniques include various methods such as classification, clustering, and association rule mining. Classification techniques are used to classify accidents into different categories based on their severity or the type of accident—clustering techniques group similar accidents based on specific characteristics such as location, time, or weather conditions. Association rule mining identifies relationships between different factors that contribute to accidents.

2. Motivation

Traffic accidents have a significant economic and societal impact on the United States, costing billions of dollars annually. The majority of these losses stem from a small number of severe accidents. Therefore, it is essential to reduce traffic accidents, particularly the serious ones. One approach to addressing traffic safety problems is the proactive approach, which aims to prevent unsafe road conditions from occurring. Accident prediction and severity prediction are crucial for the successful implementation of this approach. By identifying patterns and key factors associated with serious accidents, informed actions can be taken, and resources can be allocated more effectively. Analyzing accident data can help identify the root causes and factors that contribute to accidents, allowing policymakers and transportation agencies to develop more effective strategies to prevent accidents from occurring. By leveraging advanced analytics techniques and machine learning algorithms, accident data can be transformed into meaningful insights that can inform decision-making, shape policies, and, ultimately, help reduce the number of accidents and fatalities on our roads. Therefore, conducting an in-depth analysis of road accidents is a crucial step toward making our roads safer and reducing the devastating impact of accidents on individuals, families, and society as a whole.

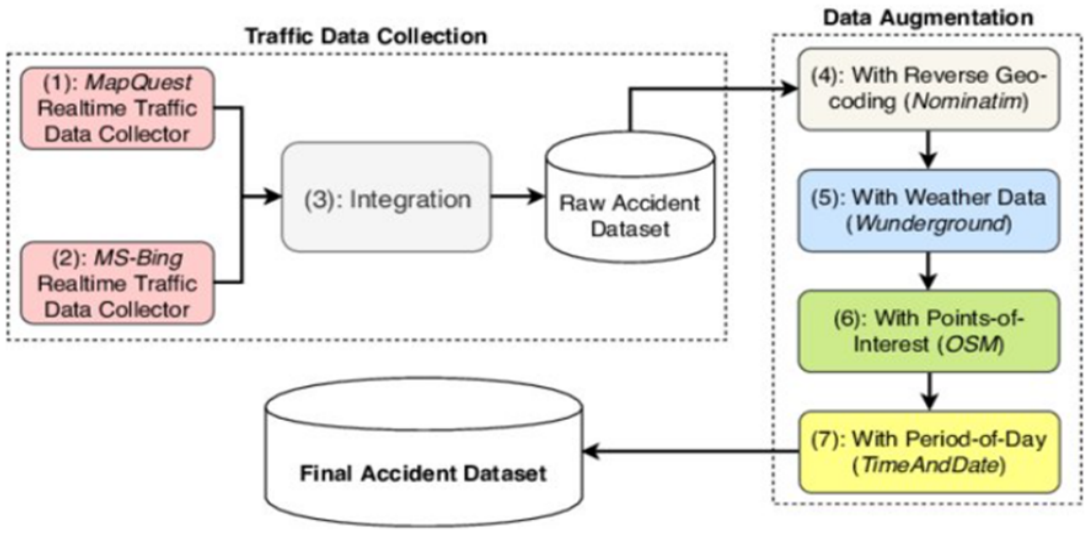
3. Dataset Description

In recent decades, accident analysis has been the subject of considerable research due to the prevalence of road accidents. Decreasing the number of traffic accidents is a crucial issue for public safety. Unfortunately, most studies that focus on analyzing and predicting traffic accidents have relied on small datasets that don't cover a wide range of situations, which limits their usefulness and applicability. Additionally, the few large datasets available are either not accessible to the public, outdated, or lack crucial contextual data such as weather conditions and nearby points of interest. To assist the research community in

overcoming these limitations, (Moosavi et al., 2019) have developed a publicly accessible database of accident information called US-Accidents. This large-scale database was created by gathering, integrating, and supplementing data through a comprehensive process. It includes information on 2.8 million traffic accidents that occurred in the contiguous United States. Each accident record contains intrinsic and contextual attributes, such as location, time, weather, natural language description, points of interest, and time of day. This dataset on traffic accidents covers 49 states of the United States and is continuously updated since February 2016. It is collected through multiple data providers that include various APIs which provide streaming traffic event data. The APIs capture traffic events from different sources such as traffic cameras, traffic sensors in the road network, and law enforcement agencies, among others. Moosavi, Samavatian, Parthasarathy, Teodorescu, et al. (2019) have provided a summary of this process which is shown in Figure 1. These sources include both US and state departments of transportation. The information in this dataset is available in a CSV file. The table 1 below lists the attributes of the data:

Figure 1

Summary of the traffic data process



It is the collection of car accident data from various sources such as MapQuest and Bing. The data set was collected from February 2016 to December 2019 and includes information on traffic events recorded by different entities. The data set contains 47 features, and details on each feature are given below. Some features include TMC, which is a Traffic Message Channel code, Severity, which is a number ranging from 1 to 4 indicating the extent of the impact on traffic and the level of damage or fatalities; and Description, which provides a natural language description of the accident, and Weather Condition, which describes the weather at the time of the accident using natural language keywords.

Table 1

No.	Attribute	Description
1	ID	This is a unique identifier of the accident record.
2	Severity	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).
3	Start_Time	Shows the start time of the accident in the local time zone.
4	End_Time	Shows the end time of the accident in the local time zone. End time here refers to when the impact of the accident on traffic flow was dismissed.
5	Start_Lat	Shows latitude in the GPS coordinate of the start point.
6	Start_Lng	Shows longitude in the GPS coordinate of the start point.
7	End_Lat	Shows latitude in the GPS coordinate of the endpoint.

8	End_Lng	Shows longitude in the GPS coordinate of the endpoint.
9	Distance(mi)	The length of the road extent affected by the accident.
10	Description	Shows natural language description of the accident.
11	Number	Shows the street number in the address field.
12	Street	Shows the street name in the address field.
13	Side	Shows the relative side of the street (Right/Left) in the address field.
14	City	Shows the city in the address field.
15	County	Shows the country in the address field.
16	State	Shows the state in the address field.
17	Zipcode	Shows the zip code in the address field.
18	Country	Shows the country in the address field.
19	Timezone	Shows the timezone based on the location of the accident (eastern, central, etc.).
20	Airport_Code	Denotes an airport-based weather station which is the closest one to the location of the accident.
21	Weather_Timestamp	Shows the time-stamp of the weather observation record (in local time).

22	Temperature(F)	Shows the temperature (in Fahrenheit).
23	Wind_Chill(F)	Shows the wind chill (in Fahrenheit).
24	Humidity(%)	Shows the humidity (in percentage).
25	Pressure(in)	Shows the air pressure (in inches).
26	Visibility(mi)	Shows visibility (in miles).
27	Wind_Direction	Shows wind direction.
28	Wind_Speed(mph)	Shows wind speed (in miles per hour).
29	Precipitation(in)	Shows precipitation amount in inches if there is any.
30	Weather_Condition	Shows the weather condition (rain, snow, thunderstorm, fog, etc.)
31	Amenity	A POI annotation which indicates the presence of amenity in a nearby location.
32	Bump	A POI annotation that indicates the presence of a speed bump or hump in a nearby location.
33	Crossing	A POI annotation which indicates the presence of crossing in a nearby location.
34	Give_Way	A POI annotation that indicates the presence of give_way in a nearby location.
35	Junction	A POI annotation indicates the presence of a junction in a nearby location.

36	No_Exit	A POI annotation that indicates the presence of no_exit in a nearby location.
37	Railway	A POI annotation that indicates the presence of a railway in a nearby location.
38	Roundabout	A POI annotation indicates the presence of a roundabout in a nearby location.
39	Station	A POI annotation indicates the presence of a station in a nearby location.
40	Stop	A POI annotation that indicates the presence of a stop in a nearby location.
41	Traffic_Calming	A POI annotation that indicates the presence of traffic_calming in a nearby location.
42	Traffic_Signal	A POI annotation indicates the presence of traffic_signal in a nearby location.
43	Turning_Loop	A POI annotation indicates the presence of turning_loop in a nearby location.
44	Sunrise_Sunset	Shows the period of day (i.e. day or night) based on sunrise/sunset.
45	Civil_Twilight	Shows the period of day (i.e. day or night) based on civil_twilight.
46	Nautical_Twilight	Shows the period of day (i.e. day or night) based on nautical_twilight.
47	Astronomical_Twilight	Shows the period of day (i.e. day or night) based on astronomical_twilight.

Figure 2:

Sample data from the CSV file

	ID	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	Description	...	Roundabout
230117	A-230118	2	2021-12-22 23:22:00.000000000	2021-12-23 01:46:46.000000000	35.644053	-120.703917	35.643957	-120.703560	0.021	Accident on Slate Ranch Rd from Mustang Spring...	...	False
2042881	A-2042882	2	2020-10-02 19:03:00	2020-10-02 20:36:02	28.529638	-81.385849	28.534218	-81.383669	0.343	Incident on I-4 EB near EXIT 79 Expect long de...	...	False
723274	A-723275	2	2021-10-11 14:57:00	2021-10-11 16:58:30	28.472620	-81.393638	28.472574	-81.397147	0.213	Stationary traffic from FL-527/Hansel Ave (E O...	...	False
752773	A-752774	2	2021-08-23 05:47:00	2021-08-23 13:30:28	39.525951	-77.927979	39.533530	-77.920276	0.665	Vehicle Crash on I-81 NB at Mile Marker 19.0.	...	False
1884971	A-1884972	2	2020-12-29 12:00:00	2020-12-29 15:36:03	34.145604	-117.306185	34.145514	-117.296925	0.530	I210 E H ST. 75-OIC ADV ACCESS F/ STATE IF GOI...	...	False

5 rows × 47 columns

Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop	Sunrise_Sunset	Civil_Twilight	Nautical_Twilight	Astronomical_Twilight
False	False	False	False	False	False	Night	Night	Night	Night
False	False	False	False	False	False	Day	Day	Day	Day
False	False	False	False	False	False	Day	Day	Day	Day
False	False	False	False	False	False	Night	Night	Day	Day
False	False	False	False	False	False	Day	Day	Day	Day

```
# check the no. of columns & rows
print('The Dataset Contains, Rows: {:,d} & Columns: {}'.format(acc_data.shape[0], acc_data.shape[1]))
```

```
The Dataset Contains, Rows: 2,845,342 & Columns: 47
```

We have also used the US Census Demographic dataset collected from the American Community Survey (ACS) for the year 2017. It covers all 52 states of America as well as DC and Puerto Rico. The dataset has two tables for each year and four tables in total. The first two tables are called census_tract_data which contain data for all census tracts within the US.

The second two tables are called county_data which contain data for all counties or county equivalents in the US. Although the accidents we are looking at happened over the span of five years (2016-2021), we do think that the 2017 population data is a decent representation for all of our analysis since we think that the population would not have changed by a significant amount. Each table of the US Census Demographic dataset consists of 37 columns which are identical across the four tables except for the ID column being Census Tract ID for the two census_tract_data tables and County Census ID for the two county_data tables. Many of the columns are considered out of the scope of this project so we only focus on the following attributes given in Table 2:

Table 2:

No.	Attribute	Description
1	State	Name of one of the 52 states of America, or DC or Puerto Rico
2	County	Name of the county or county equivalent
3	TotalPopulation	Total population of the county
4	Drive	Percentage of the county's population commuting alone in a car, van, or truck
5	Transit	Percentage of the county's population commuting on public transport
6	MeanCommute	Mean commute time in minutes
7	Poverty	Percentage of the county's population under the level of poverty

Figure 3:

Sample data from US Census dataset

	State	County	TotalPop	Drive	Transit	MeanCommute	Poverty
1094	Kentucky	Rockcastle County	16815	85.6	0.5	24.7	21.0
1995	North Dakota	Bowman County	3226	77.4	0.0	14.3	8.5
82	Alaska	Lake and Peninsula Borough	1301	22.0	0.2	6.7	16.5
684	Illinois	Tazewell County	134695	87.5	0.5	21.2	8.0
2440	Tennessee	Claiborne County	31566	85.1	0.1	24.5	22.4

```
# check the no. of columns & rows
print('The Dataset Contains, Rows: {:,d} & Columns: {}'.format(pop_data.shape[0], pop_data.shape[1]))
```

```
The Dataset Contains, Rows: 3,220 & Columns: 7
```

4. Related Work

Krishnaveni & Hemalatha (2011), proposes a comprehensive analysis of traffic accidents using data mining techniques. Their study shows the use of data mining techniques to construct predictive models to forecast accidents and identify the factors that cause them. The authors applied classification techniques such as the Naive Bayes Bayesian classifier, AdaBoostM1 Meta classifier, PART Rule classifier, J48 Decision Tree classifier, and Random Forest Tree classifier to predict the severity of the injury that occurred in the road accidents. They used a dataset of 34,575 cases from the Transport Department of Hong Kong. For the data preparation part, they used a WEKA Workbench that includes visualization tools, algorithms, and graphical user interfaces, allowing quick access to these capabilities in data analysis and predictive modeling. The model Random Forest has performed better than the other algorithms. Rather than choosing all of the variables for the modeling, they have used a generic algorithm for feature selection to lessen the dimensionality of the dataset. The test result's classification accuracy is shown for the following three cases, including accident, vehicle, and casualty, finding the severity and cause of the accident.

According to (Li et al., 2017), it is essential to carefully analyze road traffic statistics to identify factors directly associated with fatal incidents to provide safe driving recommendations. Their research tries to solve this problem by applying statistical analysis and data mining algorithms to the FARS Fatal Accident dataset. The study was conducted into the relationship between the fatality rate and other factors such as the type of collision, the environment, the surface, the time of day, and drunk drivers. Using the Apriori method, classification models were created using the Naive Bayes classifier, and clusters were created using the straightforward K-means clustering algorithm. Based on the statistics, association rules, categorization model, and clusters found, some driving safety recommendations were produced. According to the clustering results, some states and regions have a greater fatality rate than others. Therefore, when driving in those risky states or areas might need attention. The authors concluded that there always needs to be more data to support a solid decision. If more data were available, such as non-fatal accident data, weather data, and mileage data, more tests could be run, and more conclusions could be drawn.

According to the research (Shweta et al., 2021), analyzing data on road accidents is challenging due to its heterogeneous nature, and segmentation is a critical task in this regard. The researchers conducted an analysis of road accidents in Bangladesh using machine learning algorithms in their research paper. The data used in this study was obtained from the City of Toronto Police Open Data Portal and pertained to Killed or Seriously Injured (KSI) traffic accidents. It includes details about all reported traffic accidents that occurred from 2007 to 2017. The research work proposes the use of the K-means clustering method to address this issue. The model's second task is to extract data, images, and hidden patterns through a supervised machine-learning algorithm. This information can help in formulating policies to prevent road accidents. This paper has proposed by combining the segmentation and machine learning algorithms, meaningful insights can be obtained on the factors that

cause road accidents. In this research paper, the authors utilized data mining techniques, including pre-processing data to identify locations with different frequencies of road accidents and analyzing those data to determine the factors influencing those locations. They employed the K-means clustering algorithm to classify accident locations into three categories based on frequency count. They also used association rule analysis to identify connections between different attributes and showed that different locations have different accident frequencies. To classify the severity of accidents, the authors employed four supervised machine learning techniques, including Decision Tree, K-Nearest Neighbor, Naive Bayes, and AdaBoost, with AdaBoost achieving the best performance. The authors concluded that the proposed approach achieved approximately 85% accuracy in detecting special situations. They also analyzed the occurrence of road accidents using various machine learning algorithms, including CART, Naive Bayes, and ROC value, and found that applying the CART algorithm resulted in 81.5% accuracy.

Nour et al. (2020) proposed machine learning approaches to classify and compare road accident severity. This paper utilizes sophisticated data analytics methods to forecast injury severity levels and evaluate their efficacy. The research applies predictive modeling techniques to recognize risk and important factors contributing to the severity of accidents. The data utilized in the study is publicly accessible and was obtained from the UK Department of Transport, covering the period from 2005 to 2019. The main objective of this paper is to classify accident severity using various classification methods, specifically five techniques: logistic regression models, deep neural networks, support vector machines, decision trees, and extreme gradient boosting. These methods will be compared to determine which one performs the best. The study applied hyperparameter tuning to the classification methods and split the dataset into two parts: 70% for training and 30% for test data. The evaluation metrics used were balanced accuracy, which is typically used with imbalanced

datasets, and ROC curves. The study found that XGBoost and Random Forest outperformed logistic regression, support vector machines, and neural networks. The paper also analyzed 63 attributes from three data sources to examine their relation to accident severity and highlighted issues related to data quality and imbalanced data, with techniques applied to address these issues.

5. Data Preprocessing

Data preprocessing is a critical step in data analysis that involves cleaning, transforming, and organizing data before the analysis. In the case of US road accident data, preprocessing is particularly important as the data can be noisy, contain missing or incorrect values, and may require some form of feature engineering to make it worthwhile for analysis. The various processes involved in data preparation for US road accident data are discussed below:

5.1 Datatype Conversions

In this step, we have converted the variables "Start_Time" and "End_Time" in the dataset into date-time features using the "pd.to_datetime()" function. It allows for better data analysis, particularly in identifying patterns and trends related to the timing of accidents. Furthermore, it offers crucial insights into the variables influencing traffic accidents by transforming them from strings to date-time data types.

Figure 4

Code snippet of Datatype Conversions

```
# convert the Start_Time & End_Time Variable into Datetime Feature
acc_data.Start_Time = pd.to_datetime(acc_data.Start_Time)
acc_data.End_Time = pd.to_datetime(acc_data.End_Time)
```

5.2 Adding new columns

We have added a new column to the dataset to improve the quality of the analysis. For

example, adding a column for the day of the week or the time of day can help pattern in the data. First, the "Duration(min)" column is created by calculating the time duration between the "Start_Time" and "End_Time" columns in minutes. Next, the new columns are made using the "Start_Time" column. These include the "Start_Hour" column, which extracts the hour from the "Start_Time" column, and the "Day" column, which identifies the day of the week (0 for Monday, 1 for Tuesday, and so on). And the "Month" and "Year" columns, extract the month and year from the "Start_Time" column, respectively.

Finally, a "Weekend" column is created, which maps the "Day" column to a binary indicator of whether the day is a weekend (1) or not (0). This step enables the incorporation of time-based features into the analysis, which may improve the accuracy of the model's predictions or generate new insights into the data.

Figure 5

Code snippet of adding new columns

```
acc_data['Duration(min)'] = (acc_data['End_Time'] - acc_data['Start_Time']).astype('timedelta64[m]')
acc_data['Start_Hour'] = (acc_data['Start_Time']).dt.hour
acc_data['Day'] = (acc_data['Start_Time']).dt.weekday
acc_data['Month'] = acc_data['Start_Time'].dt.month
acc_data['Year'] = acc_data['Start_Time'].dt.year
# Weekend column where 1 = weekend, 0 = weekday
#acc_data['Weekend'] = acc_data['Day'] <= 5
acc_data['Weekend'] = acc_data['Day'].map({0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 1, 6: 1})
```

5.3 Data Quality Analysis

This step involves identifying and correcting inconsistencies or missing data in the dataset. We have removed the attributes that contain the most null instances: Number, Wind_Chill(F), Wind_Speed(mph), and Precipitation(in). The missing data impact the quality and accuracy of data analysis results. The next step is to decide how to handle these missing values by filling them with appropriate values or removing them entirely.

Figure 6

Code snippet of checking missing values

```
# Check for missing values
acc_data.isna().sum()
```

```
ID                          0
Severity                    0
Start_Time                  0
End_Time                    0
Start_Lat                   0
Start_Lng                   0
End_Lat                     0
End_Lng                     0
Distance(mi)                0
Description                 0
Number                1743911
Street                      2
Side                        0
City                      137
County                      0
State                       0
Zipcode                  1319
Country                     0
Timezone                 3659
Airport_Code             9549
Weather_Timestamp       50736
Temperature(F)          69274
Wind_Chill(F)          469643
Humidity(%)             73092
Pressure(in)            59200
Visibility(mi)          70546
Wind_Direction          73775
Wind_Speed(mph)        157944
Precipitation(in)      549458
Weather_Condition       70636
Amenity                     0
Bump                        0
Crossing                    0
```

5.4 Removing redundant Columns

We have also removed the redundant columns that do not add any value to the analysis. For example, the attribute "Wind_Chill(F)" is highly correlated with "Temperature(F)," indicating that including both variables would not provide significant additional information. Therefore, "Wind_Chill(F)" is removed from the dataset. Next, the attributes that contain information that has already been extracted into more relevant features, such as "Weather_Timestamp," "Start_Time," and "End_Time." These attributes are dropped from the dataset, and the information they contain is captured in more relevant features like "Duration," "Day," and "Start_Hour."

Additionally, the geographical attributes such as "Country," "Street," "Side," "Zipcode," "Timezone," and "Airport_Code" are not relevant to the analysis and can be removed. Other columns like "Description," "End_Lat," "End_Lng," and "Wind_Direction" are also identified as irrelevant and are dropped from the dataset.

Figure 7

Code snippet of dropping redundant columns

```
data = acc_data.drop(columns = ['Start_Time','End_Time','End_Lat','End_Lng','Description','Number','Street','Side','Zipcode','Timezone','Country','Airp
data.head()
```

	ID	Severity	Start_Lat	Start_Lng	Distance(mi)	City	County	State	Temperature(F)	Humidity(%)	...	Traffic_Calming	Traffic_Signal	Turning_
0	A-1	3	40.108910	-83.092860	3.230	Dublin	Franklin	OH	42.1	58.0	...	False	False	
1	A-2	2	39.865420	-84.062800	0.747	Dayton	Montgomery	OH	36.9	91.0	...	False	False	
2	A-3	2	39.102660	-84.524680	0.055	Cincinnati	Hamilton	OH	36.0	97.0	...	False	False	
3	A-4	2	41.062130	-81.537840	0.123	Akron	Summit	OH	39.0	55.0	...	False	False	
4	A-5	3	39.172393	-84.492792	0.500	Cincinnati	Hamilton	OH	37.0	93.0	...	False	False	

5 rows × 35 columns

5.5 Filling Null Values

After removing redundant columns and converting data types, we filled in the missing values in the dataset. We have used techniques such as mean, median, mode, are imputation techniques to fill in missing values. For instance, 2.4% of the "Temperature" attribute contains missing values filled with the mean temperature. Similarly, we have filled the missing values with the mean for Humidity and Pressure, which have 2.5% and 2% null values, respectively. For Precipitation, which has a higher percentage of missing values (19.3%), we can assume that the missing values indicate no rain and fill in the missing values with 0. Finally, for Wind Speed, which has 16% null values, we have filled the missing values with the mean wind speed. However, we have dropped the rows containing the missing values for Weather Condition, which has only 2% null values. Filling in missing values allows us to use more available data for analysis, while dropping unnecessary columns can help streamline the research and improve its efficiency.

Figure 8

Code snippet of checking for Null values

```
# percentage of Null values
(data.isna().sum()/data.shape[0]) *100
```

```
ID                     0.000000
Severity               0.000000
Start_Lat              0.000000
Start_Lng              0.000000
Distance(mi)           0.000000
City                   0.004815
County                 0.000000
State                  0.000000
Temperature(F)         2.434646
Humidity(%)            2.568830
Pressure(in)           2.080593
Visibility(mi)         2.479350
Wind_Speed(mph)        5.550967
Precipitation(in)     19.310789
Weather_Condition      2.482514
Amenity                0.000000
Bump                   0.000000
Crossing               0.000000
Give_Way               0.000000
Junction               0.000000
No_Exit                0.000000
Railway                0.000000
Roundabout             0.000000
Station                0.000000
Stop                   0.000000
Traffic_Calming        0.000000
Traffic_Signal         0.000000
Turning_Loop           0.000000
Sunrise_Sunset         0.100761
Duration(min)          0.000000
Start_Hour             0.000000
Day                    0.000000
Month                  0.000000
Year                   0.000000
```

5.6 Converting boolean columns into 0 and 1

We have converted the Boolean columns into 0 and 1 for better analysis. For example, the features, Sunrise_Sunset column, which contains ‘Day’ or ‘Night’ values, are converted into 0 and 1. Converting Boolean columns into numeric format is helpful because many machine learning algorithms only work with numeric data. It also simplifies data analysis and

makes the data easier to understand.

Figure 9

Code snippet of converting Boolean columns into 0 and 1

```
boolean_cols = ['Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway','Roundabout',
data[boolean_cols] = (data[boolean_cols] == True).astype(int)
#Sunrise_Sunset have Day or Night values. Convert them into 0 and 1's.
data['Sunrise_Sunset'] = data['Sunrise_Sunset'].map(dict(Day=0, Night=1))
```

6. Exploratory Data Analysis:

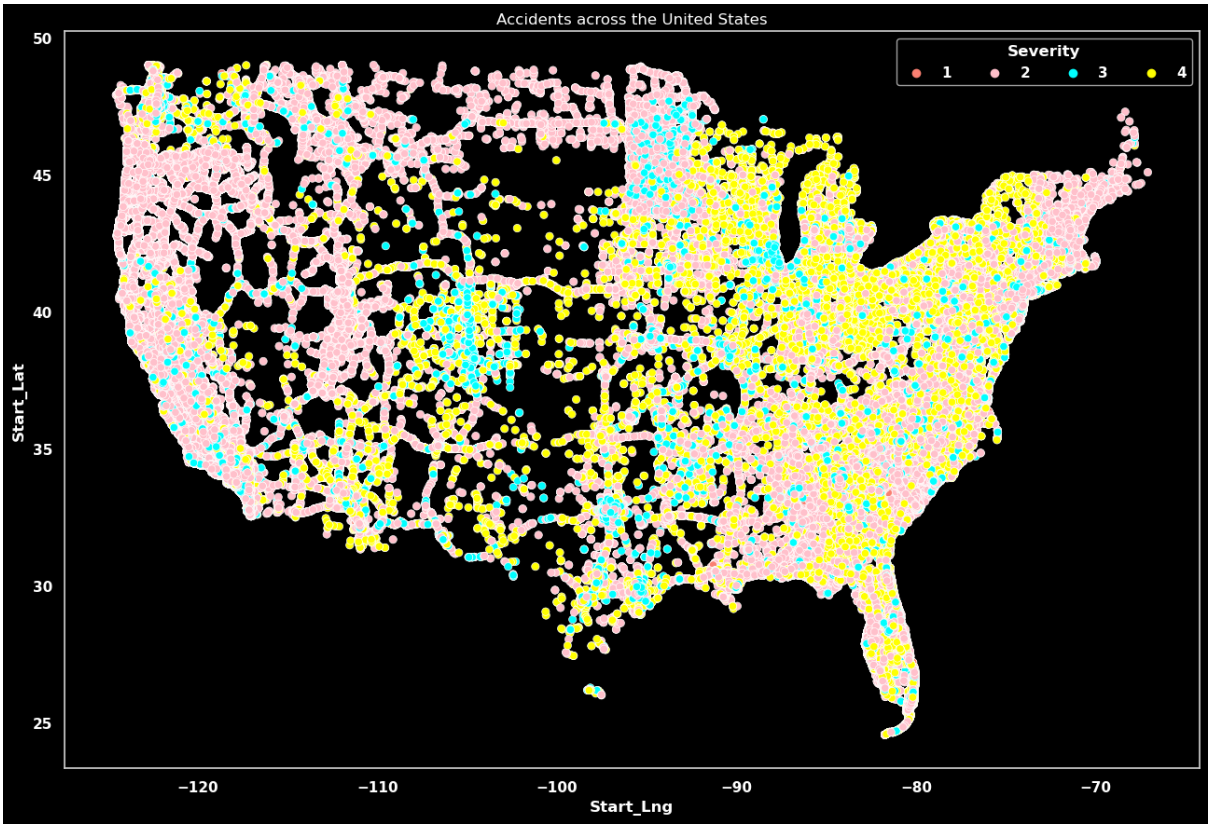
Exploratory Data Analysis (EDA) involves applying a range of statistical methods to a dataset to uncover its main characteristics. EDA often involves using data visualization techniques to create visual representations of the data, making it easier to identify patterns and trends. The goal of EDA is to gain insights into the data and identify any potential issues or anomalies that may need to be addressed before further analysis. By exploring the data in a systematic and comprehensive way, EDA can help analysts to make informed decisions and draw accurate conclusions from the data.

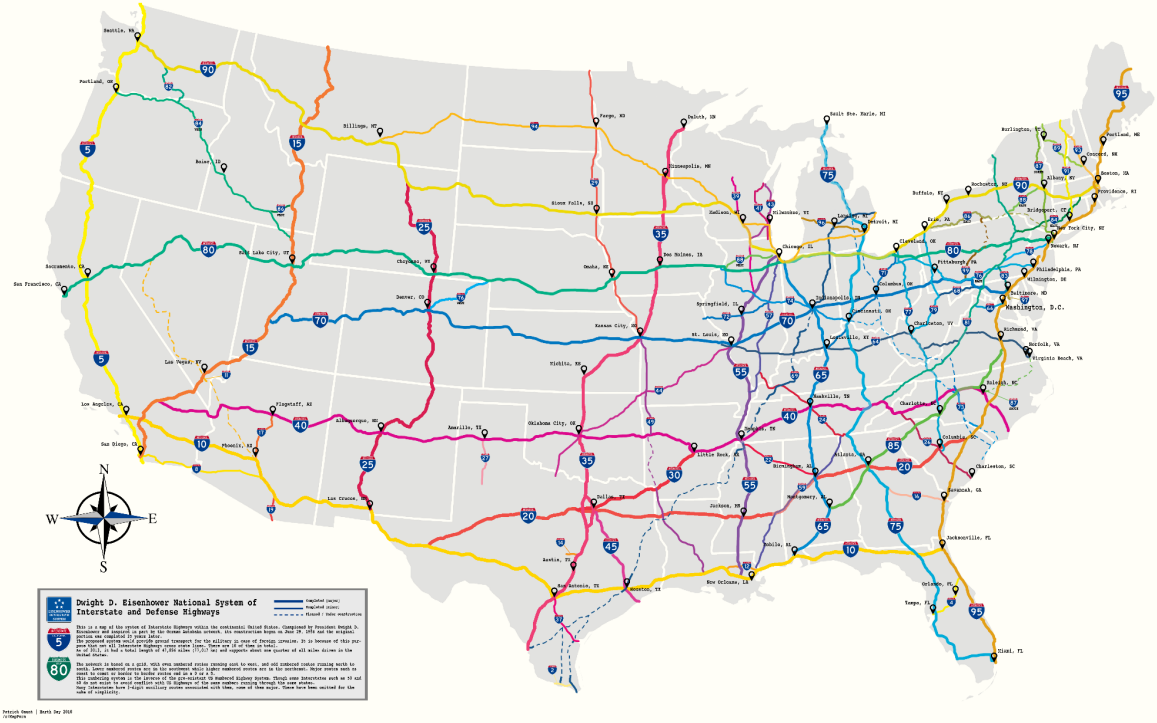
Although accidents cannot be predicted with certainty, analyzing the circumstances under which they occur can be very valuable. Understanding the controllable factors that contribute to accidents can help reduce the number of incidents. By studying past accidents and identifying common contributing factors, such as road conditions, driver behavior, or weather conditions, measures can be taken to mitigate these risks and prevent accidents in the future. Therefore, gaining an overview of these conditions can be helpful in developing strategies and interventions to improve road safety and reduce accidents.

After the data has been cleaned and prepared for analysis, we selected certain columns and applied statistical methods to reveal underlying patterns and relationships in the data. By performing statistical analysis, we can gain a deeper understanding of the data and use this information to make informed decisions or identify areas for further investigation.

Figure 10

A region-based analysis of accident hotspots





Based on the graphs provided, it appears that accidents in general, and Level 4 (Fatal) accidents in particular, tend to occur more frequently along major highways and interstates in the United States. This suggests that there may be certain factors, such as high traffic volumes or hazardous road conditions, that contribute to the occurrence of accidents on these routes. Identifying these factors and taking steps to mitigate them could potentially reduce the number of accidents and fatalities on these highways and interstates.

6.1 We will narrow our analysis to look at accident prevalency per state and city

Our analysis will focus on examining the frequency of accidents in each state and city. By narrowing our analysis to a specific geographical area, we can identify patterns and trends that may not be evident when looking at the data as a whole.

Figure 11

Analysis of accident prevalence per state and city

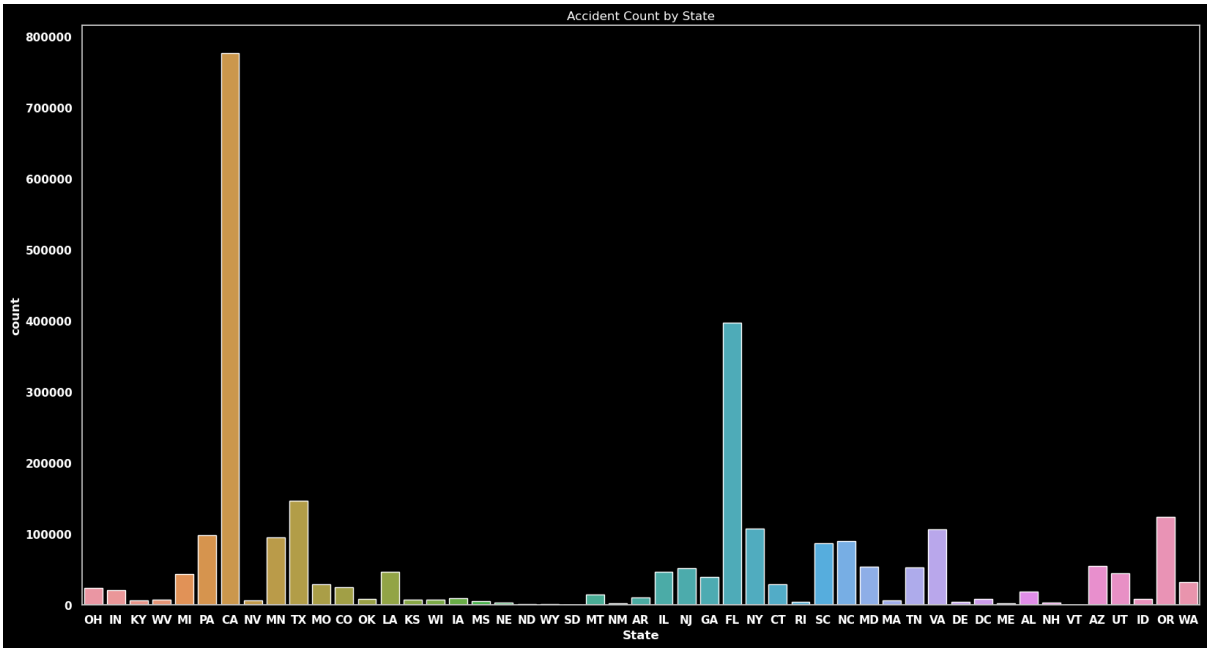


Figure 12

Top 10 states with the highest numbers of accidents

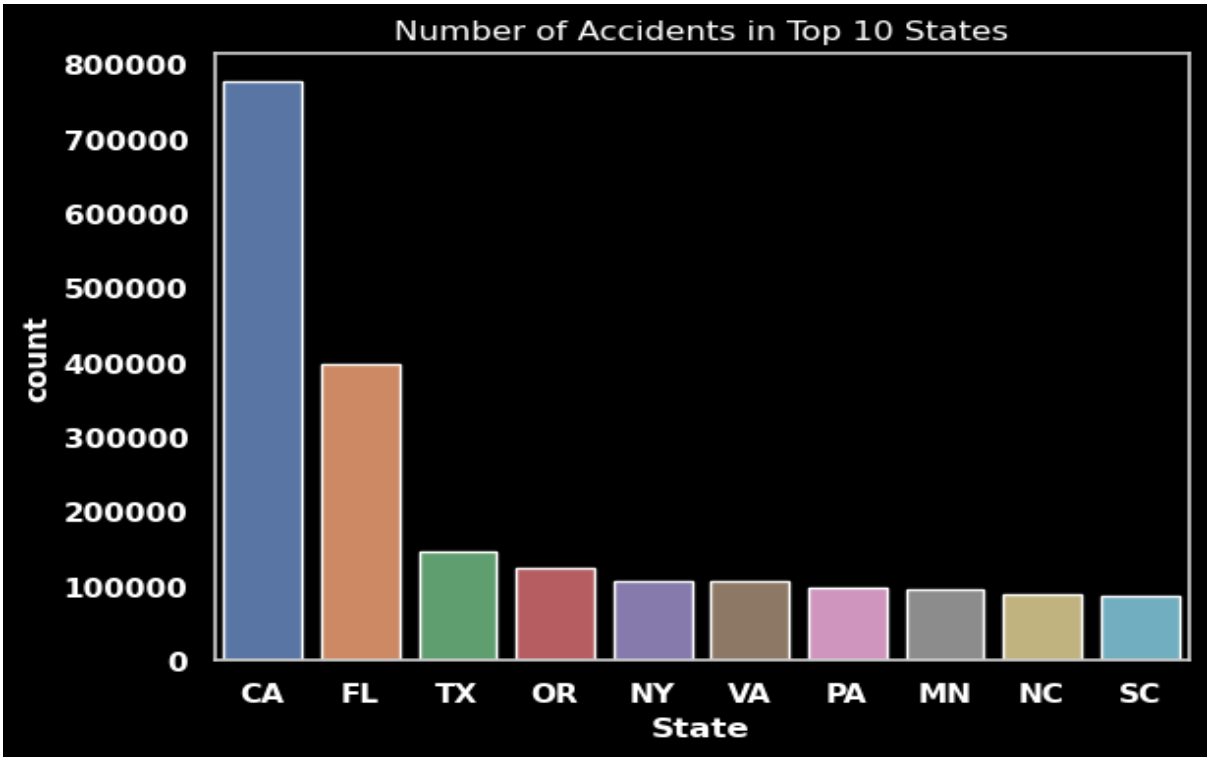
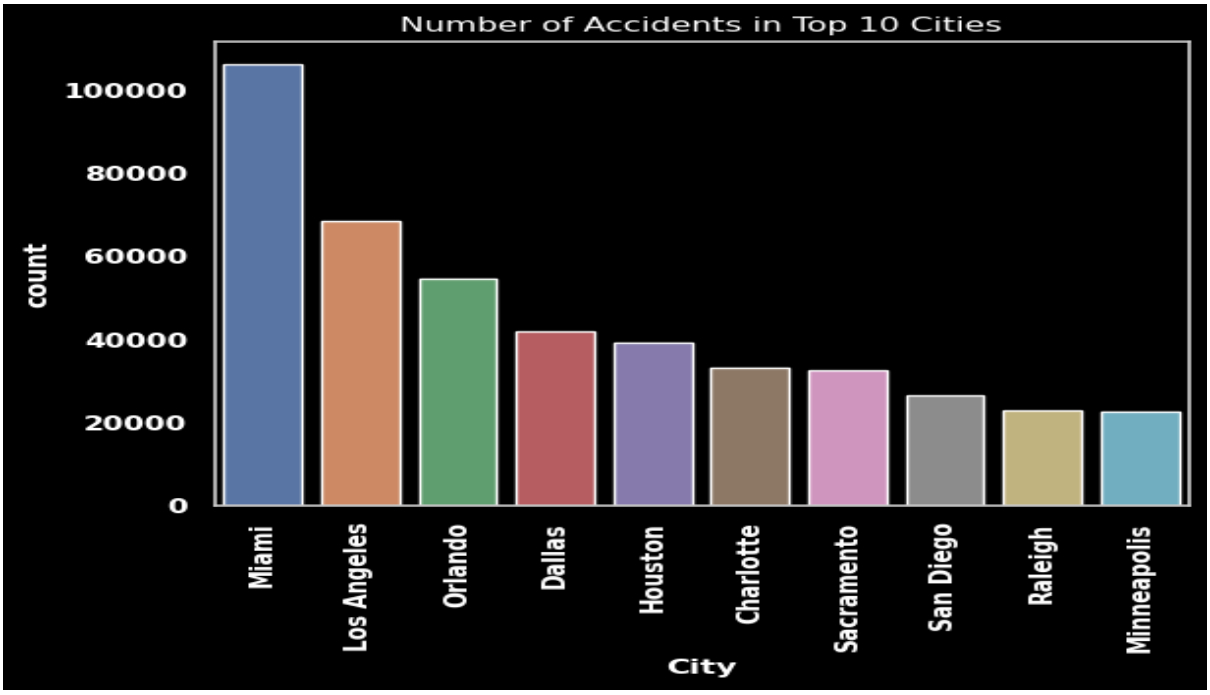


Figure 13

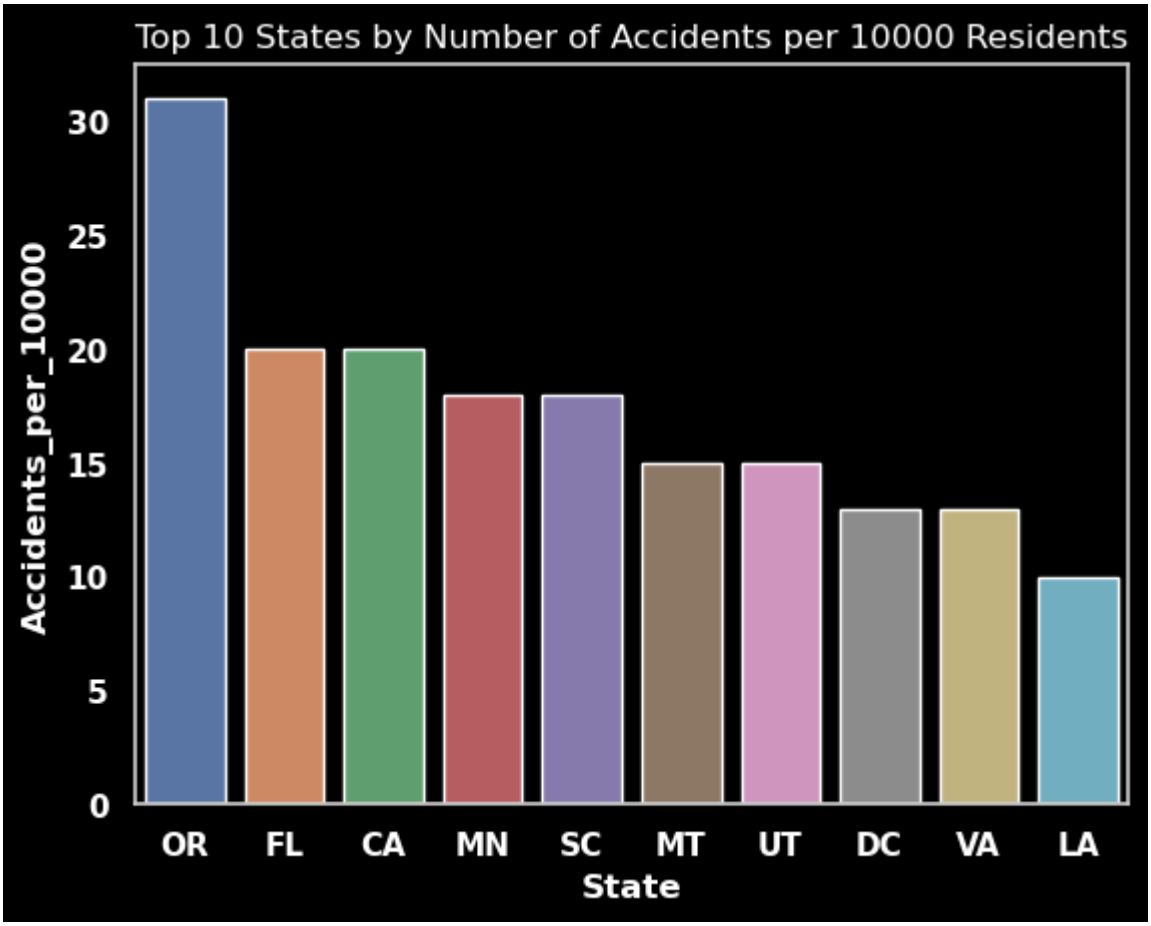
Top 10 cities with highest number of accidents



Initial analysis of the data suggests that California has recorded over 800,000 accidents, a significantly higher number than the other states. Texas, with over 300,000 accidents, has the second-highest number of accidents. But following closer examination, we discover that California, Texas, and Florida are the states with the highest number of accidents. Since these states have greater populations than states with smaller populations, it is expected that we will witness more accidents in these states. We must standardize our data depending on population size to take into consideration the various population sizes so that we can evaluate whether or not this trend in accidents is caused by state policy rather than population.

Figure 14

Top 10 States by Number of Accidents per 10000 Residents



Upon examining the data more closely, we have noticed a shift in the distribution of accidents. One notable change is that California is no longer the state with the highest number of accidents. This suggests that there may have been changes in the factors that contribute to accidents in different states over time.

Insights from the above analysis:

1. Prior to standardization, the data reveals that California had the highest number of road accidents, followed by Florida and Texas.
2. The data also indicates that Miami is the city with the highest number of road accidents across all states in the US.
3. The data reveals that three out of the top 10 cities with the highest number of accidents are located in California.
4. After standardizing the data by state population, the analysis shows that Oregon has the highest number of road accidents per 10,000 residents.

6.2 A Severity Spectrum of Accidents

The severity is a value between 0 and 4, where 0 indicates the least impact on traffic (i.e., short delay as a result of the event) and 4 indicates a significant impact on traffic (i.e., long delay).

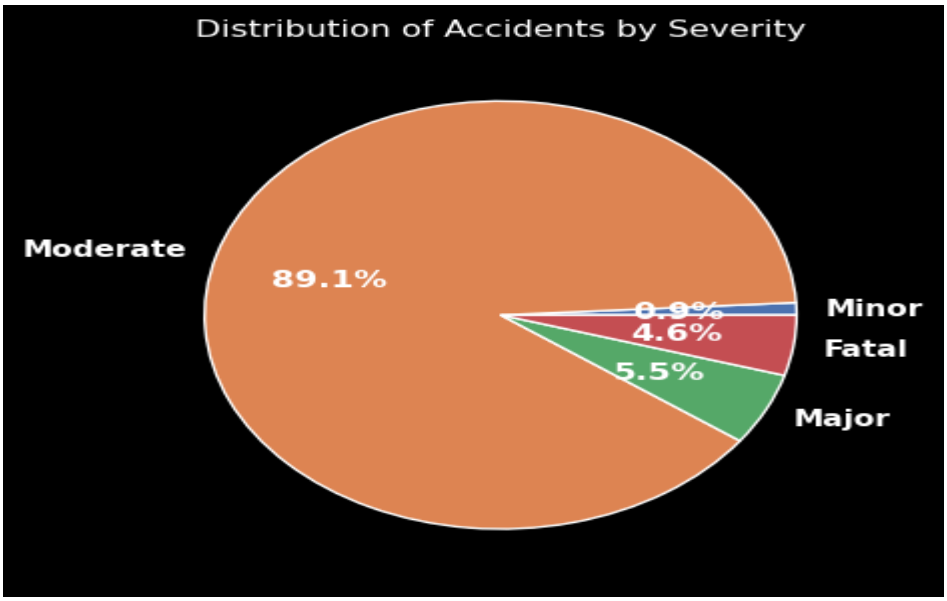
Insights from the analysis:

1. The data analysis indicates that in a large majority of road accident cases, accounting for 89% of all incidents, the impact on traffic was considered moderate (Severity-2). This suggests that, while accidents may still cause significant disruption to traffic flow and safety, most incidents do not result in major, long-lasting impacts.

2. However, the data also reveals that a small but significant percentage of accidents, approximately 4.6%, had a highly severe impact on traffic (Severity-4). These incidents are likely to have caused significant disruption and potentially resulted in serious injuries or fatalities.

Figure 15

Distribution of Accidents by Severity



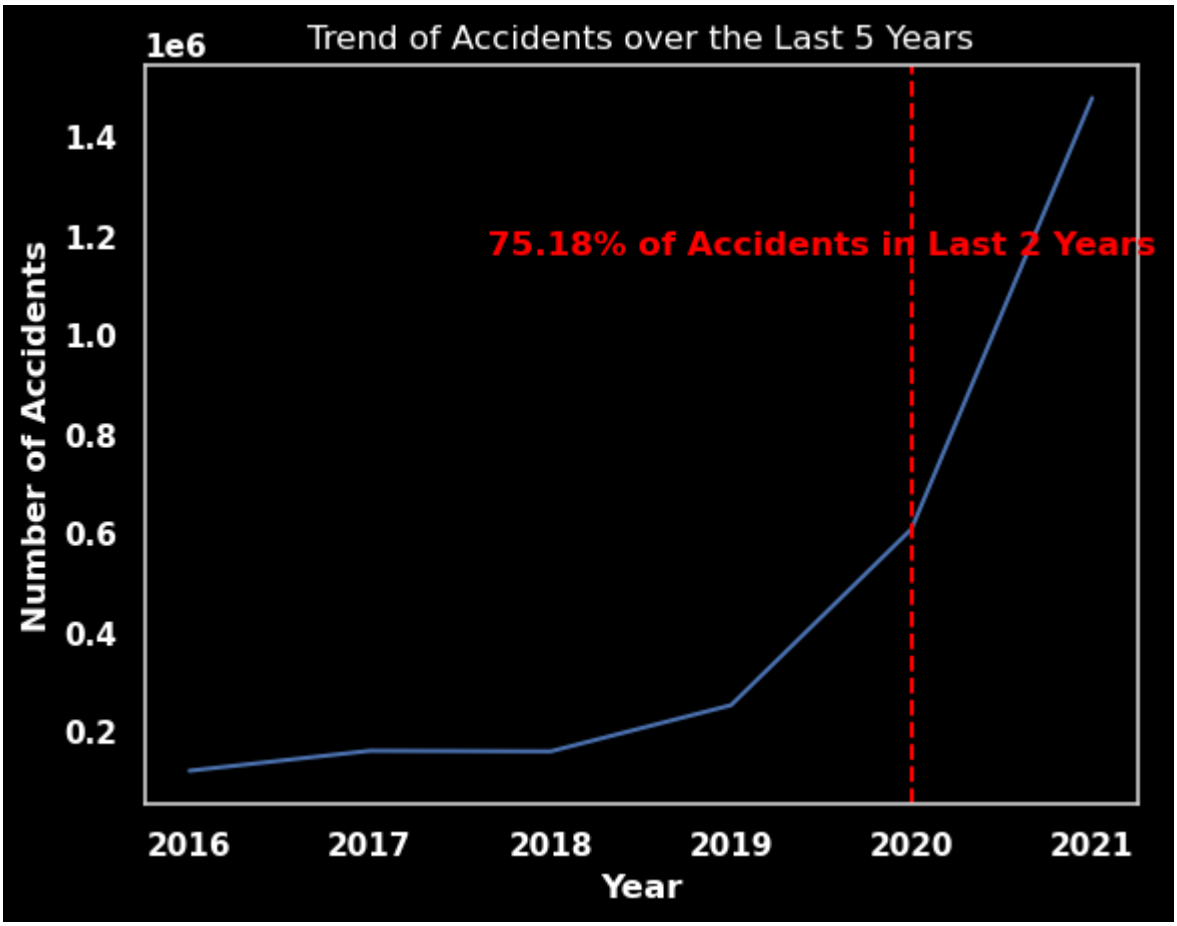
6.3 Uncovering the Timing of Traffic Troubles

Insights from the above figure:

1. The data presented in the figure indicates a clear and significant increase in the percentage of road accidents over the last six years in the United States, from 2016 to 2021.
2. In addition, the data highlights a particularly concerning trend in the concentration of road accidents in the most recent two years, 2020 and 2021. Specifically, 75% of the total road accidents recorded over the last six years occurred during these two years alone.

Figure 16

Trend of Accidents over the last 5 years



6.4 Percentage of accidents by Month

Insights from the above figure:

1. The analysis indicates that a significant percentage, around 16.7%, of road accidents in the US occurred during the month of December. This finding suggests the need for heightened vigilance and attention to road safety during this time period, as well as targeted efforts to address factors contributing to the increased prevalence of accidents during this month.

2. On the other hand, the months of July and March are associated with the least number of road accidents, with only 5.6% of total accidents occurring during these months. This information could be useful in identifying patterns and trends in accident prevalence over time, as well as identifying factors that may contribute to the increased likelihood of accidents during certain months.
3. Additionally, the analysis indicates that a significant proportion, approximately 40%, of all road accidents occurred within the three-month period from October to December, which represents the transition period from autumn to winter. This finding suggests the need for targeted efforts to address the factors that may contribute to the increased prevalence of accidents during this time period, such as inclement weather, decreased visibility, and changes in driving conditions.

Figure 17

Percentage of Accidents by Month

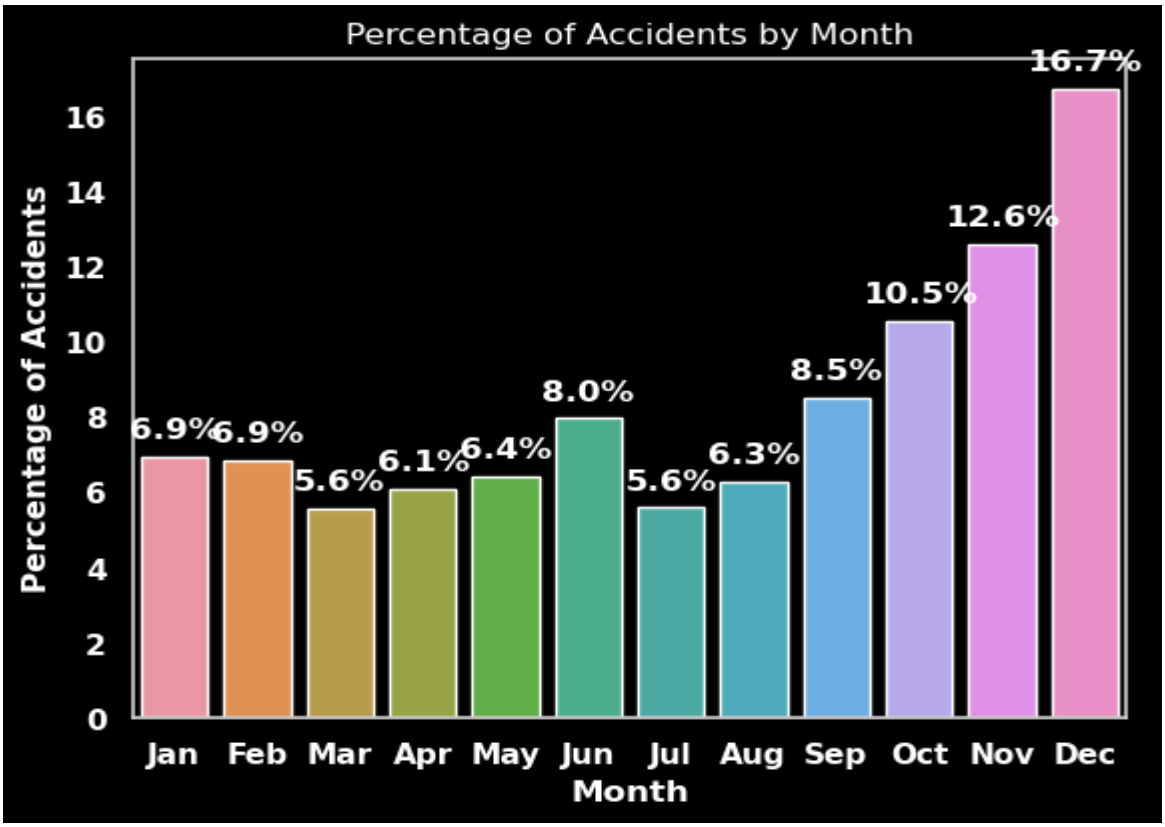


Figure 18

Percentage of Accidents by Weekend vs Weekday

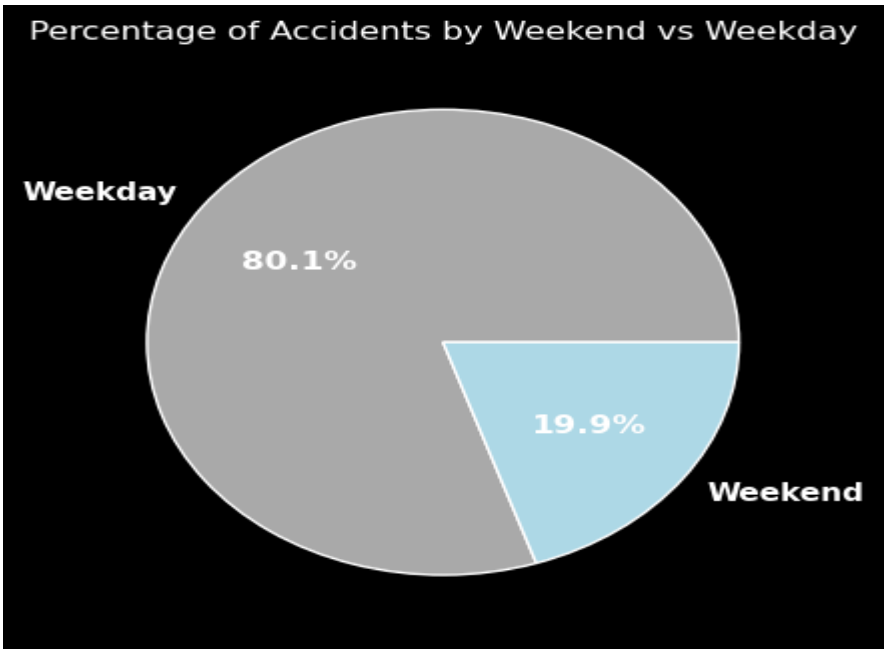
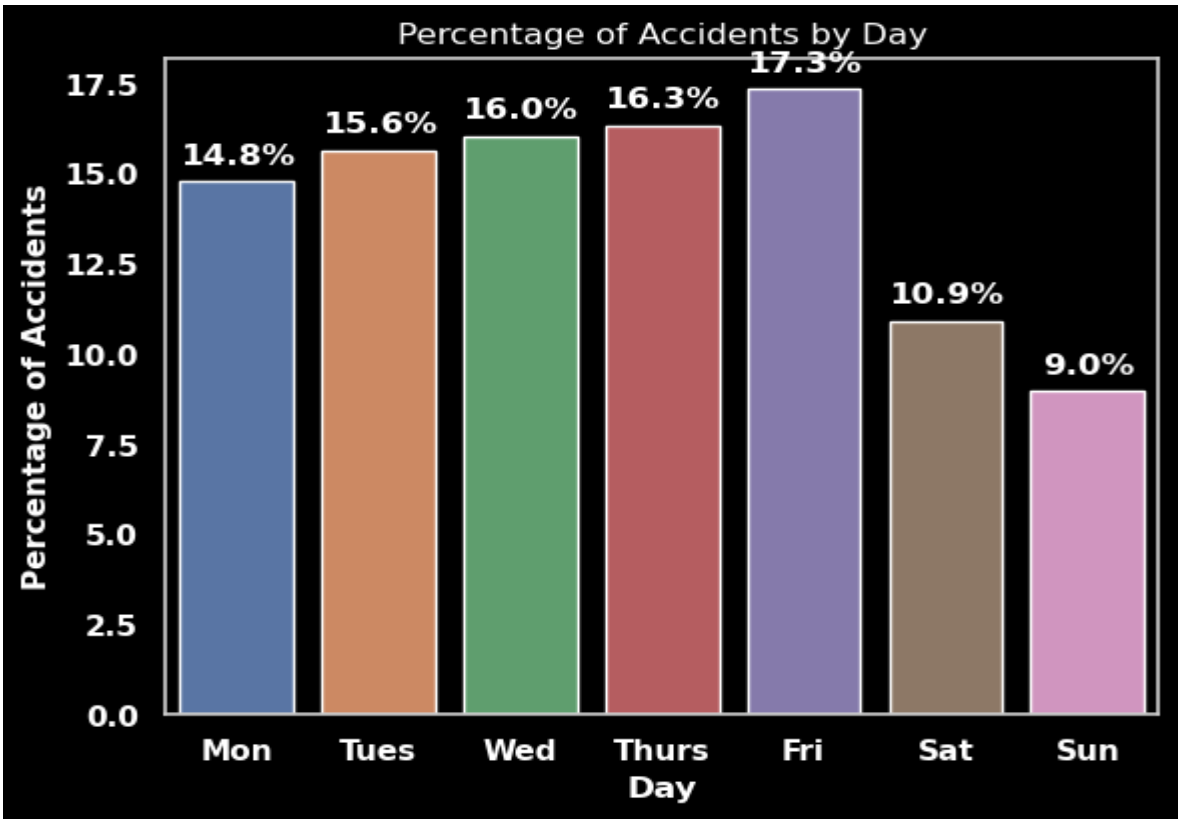


Figure 19

Percentage of Accidents by Day

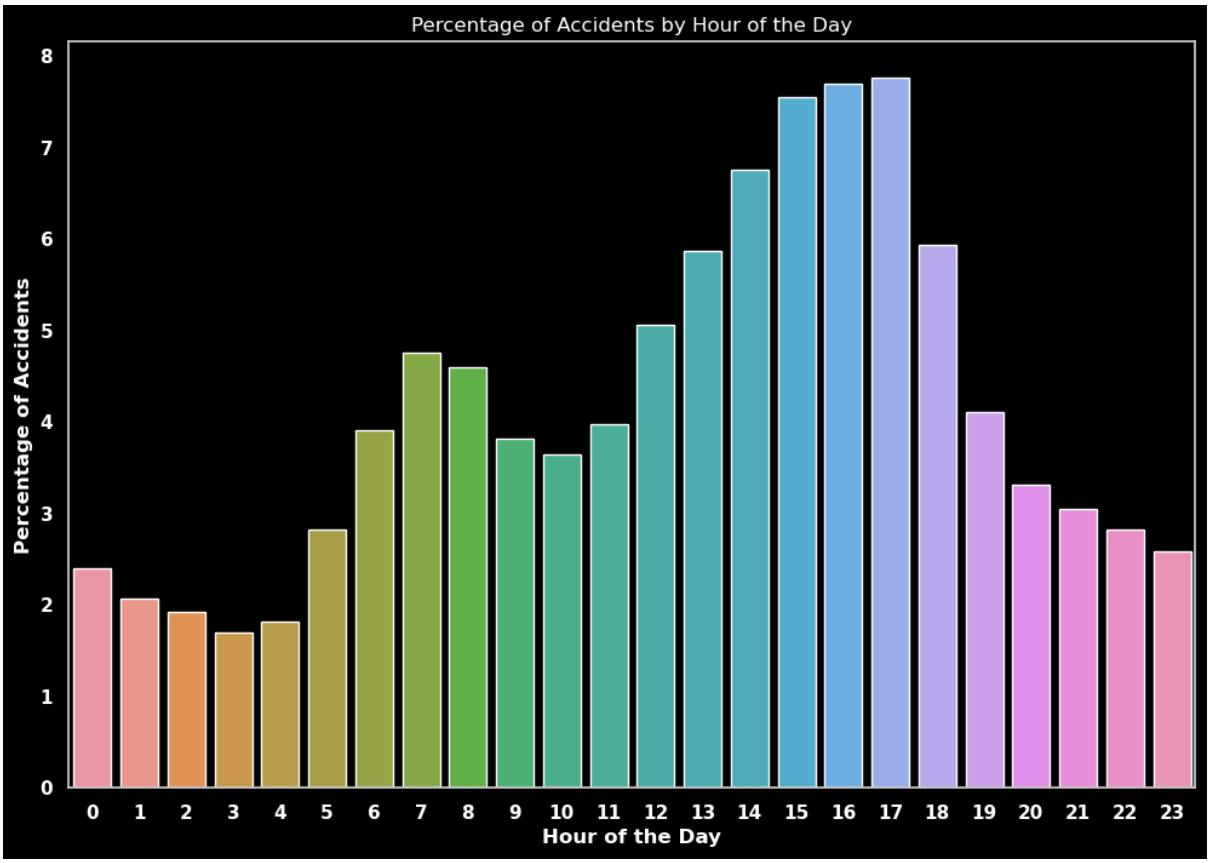


Insights from the above analysis:

1. Weekdays have approximately twice the number of road accidents compared to weekends.
2. Only about 20% of road accidents occurred on weekends.
3. Fridays have the highest percentage of road accidents among weekdays.
4. The percentage of road accidents is lowest on Sundays in the US.

Figure 20

Percentage of Accidents by Hour of the Day



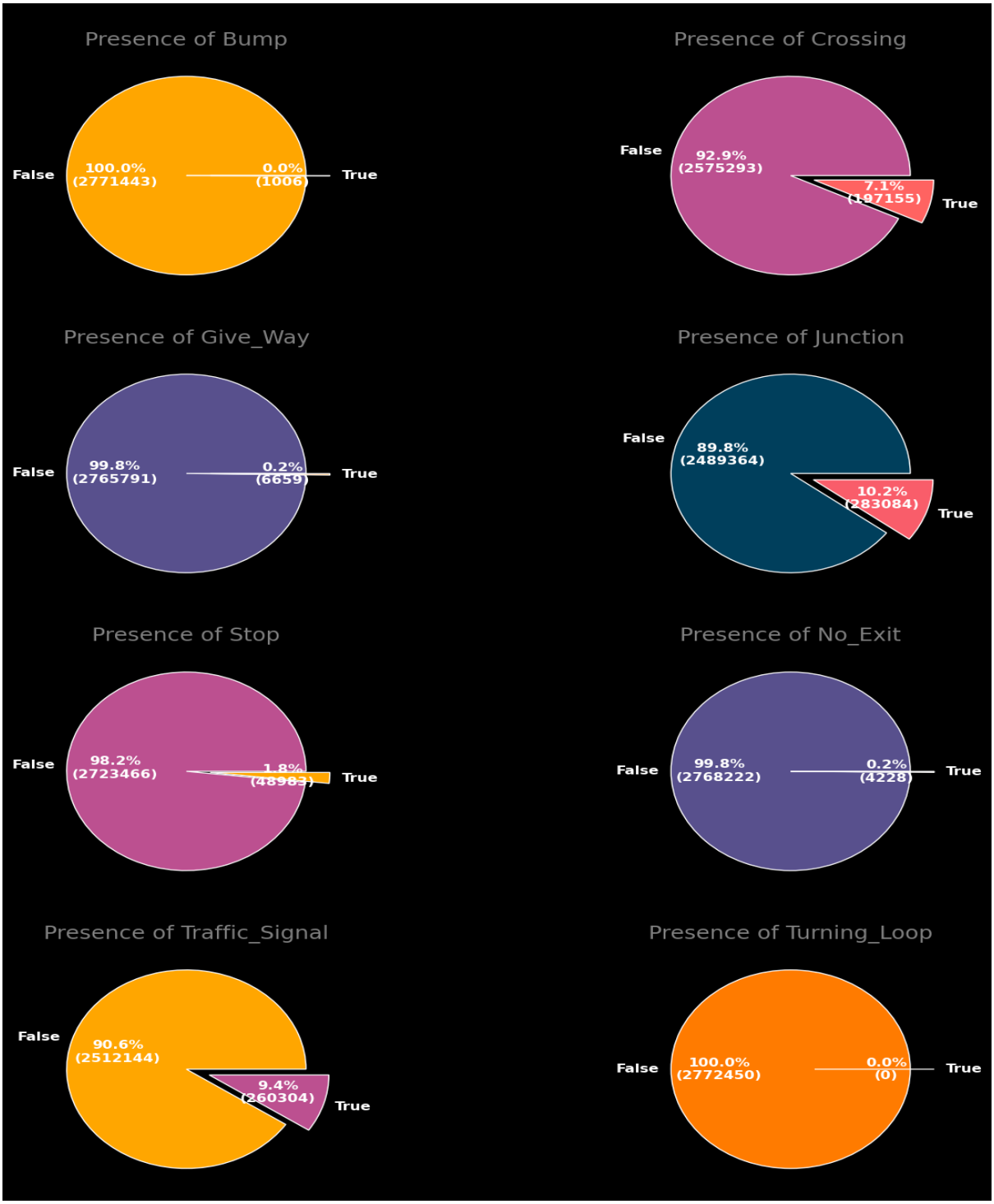
Insights from the above figure:

1. Around 30% of road accidents occur in the evening, specifically between 3:00 PM and 6:00 PM.

2. The deadliest hour for accidents is 5:00 PM, which coincides with the time when many people are returning from work in the evening.

Figure 21

How Road Conditions Can Make or Break Your Commute



Insights from the above figure:

1. Bumper, Yield, and Turning Loop were not present at the accident site in almost every case.
2. 7% of road accidents occurred near crossings, 10% near junctions, and 9.4% near traffic signals.
3. Stop signs were not present near the accident area in 98% of cases.

Figure 22

A look at how weather conditions impact road safety

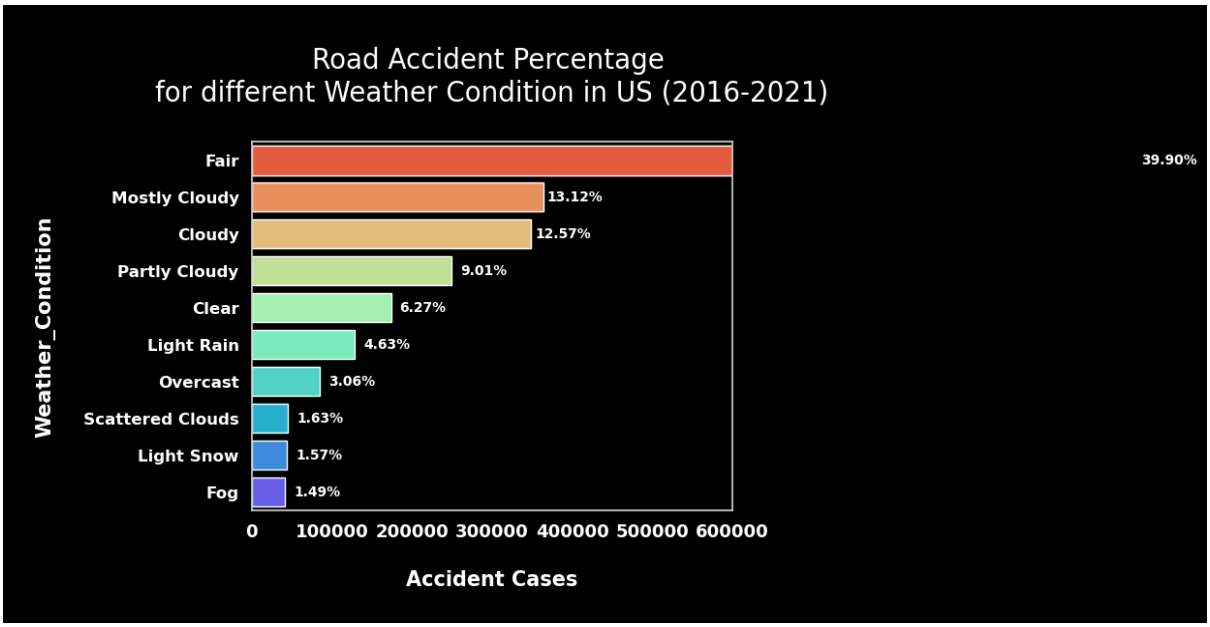
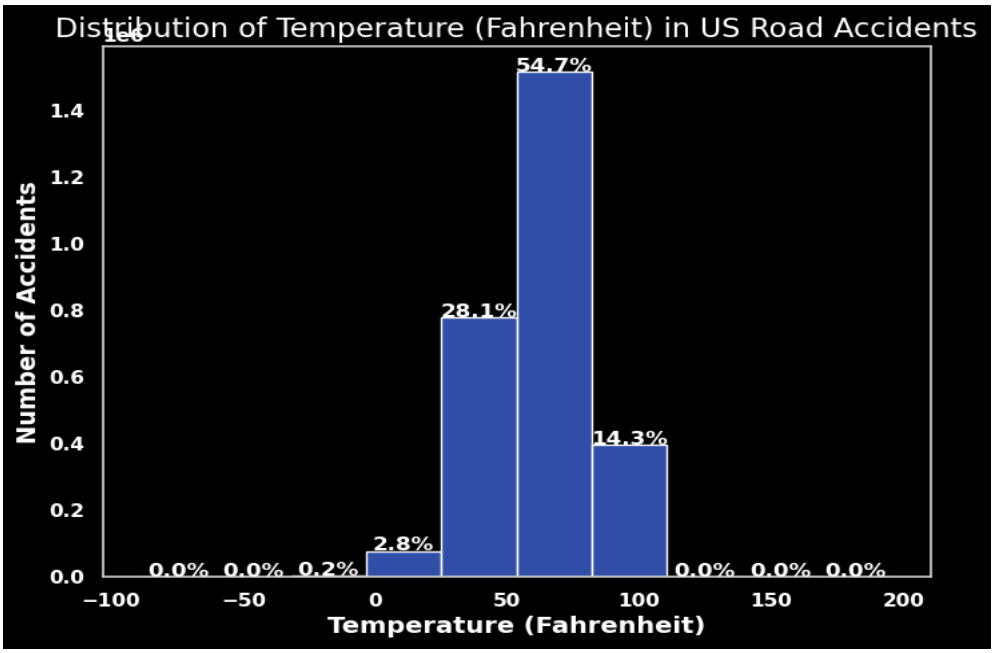
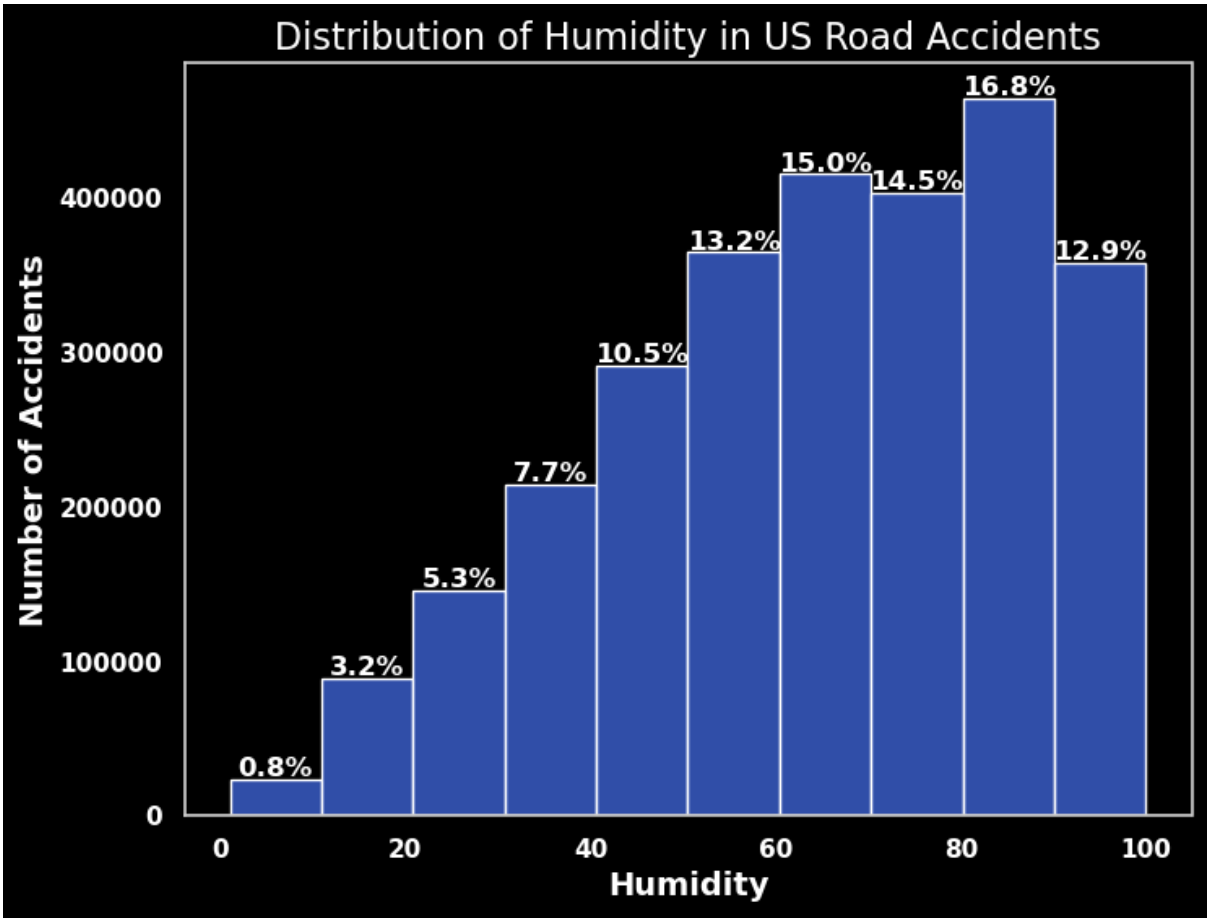
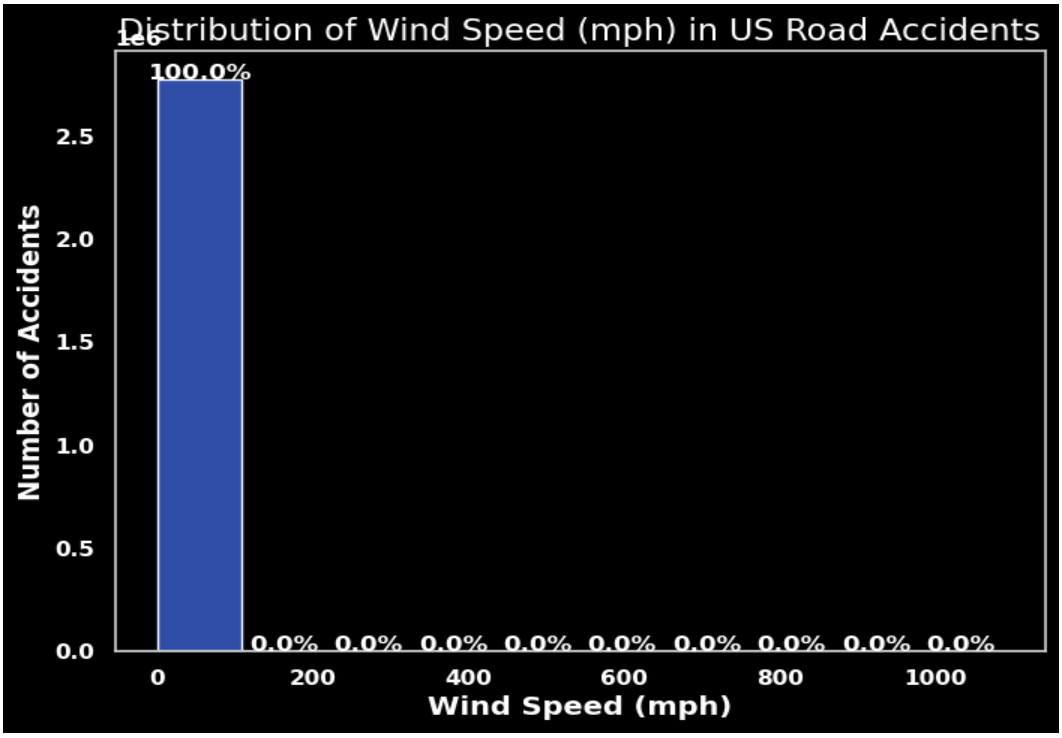


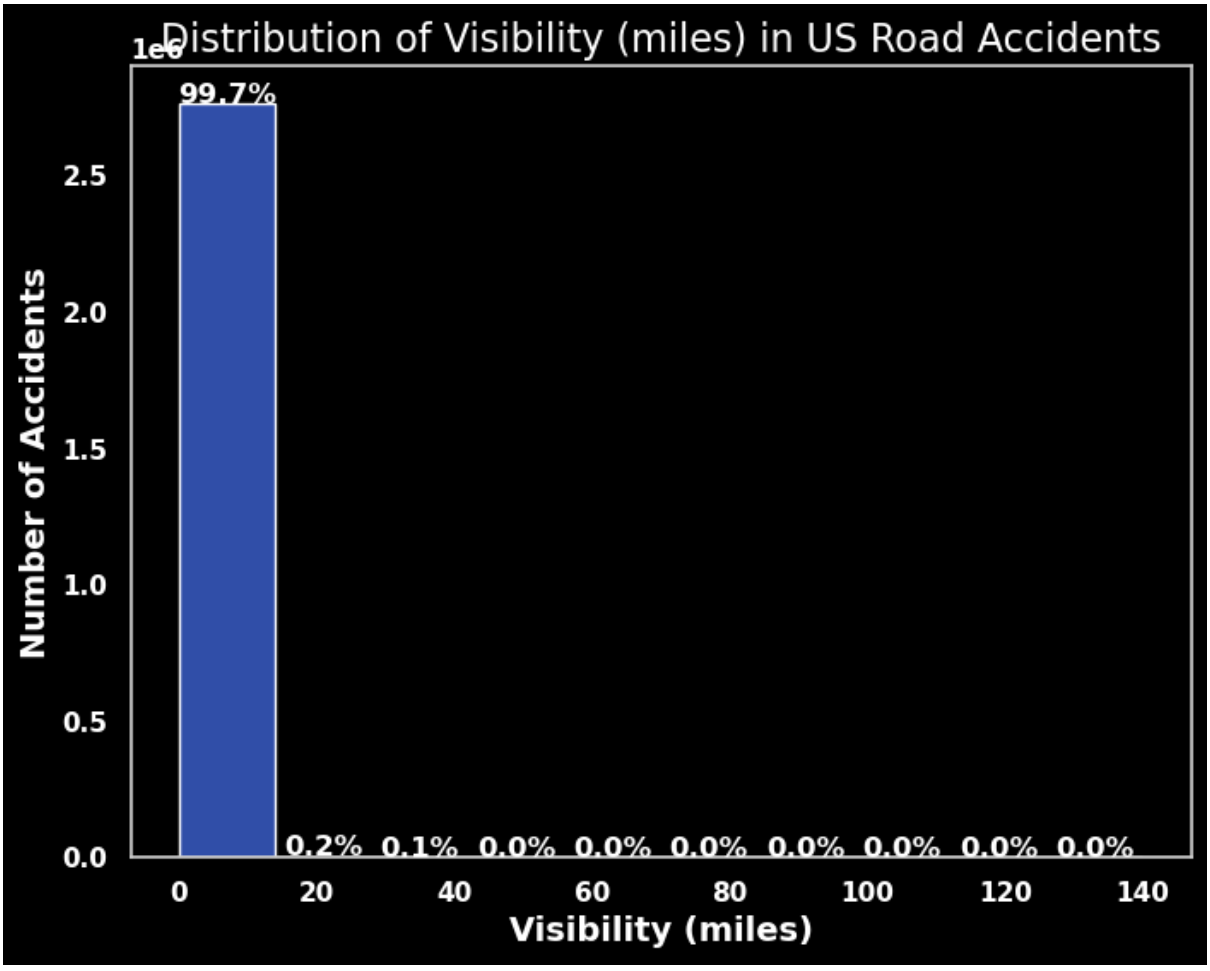
Figure 23

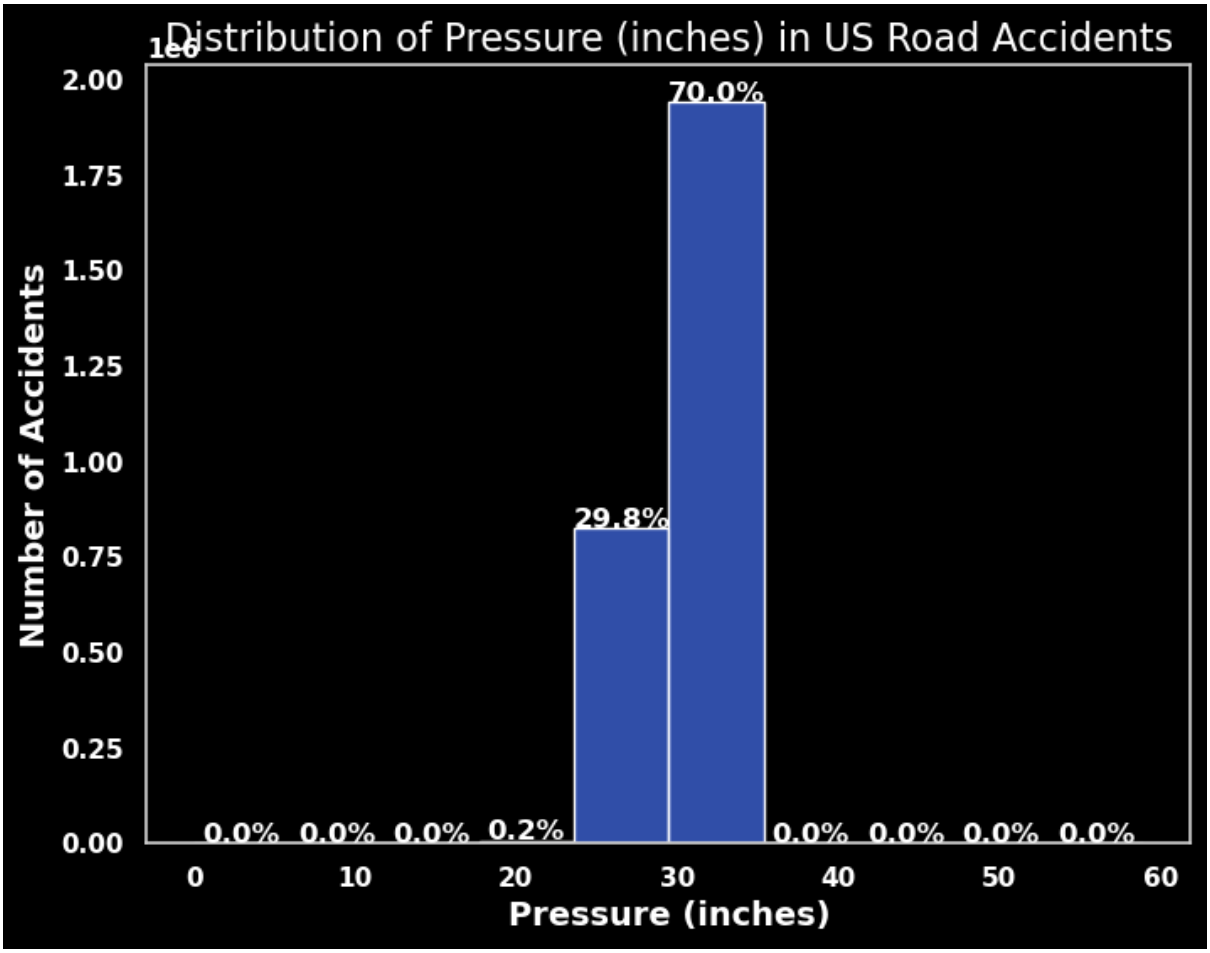
Distribution charts






Distribution of Wind Speed (mph) in US Road Accidents
1e6
100.0%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
2.5
2.0
1.5
1.0
0.5
0.0
Number of Accidents
0
200
400
600
800
1000
Wind Speed (mph)


Distribution of Visibility (miles) in US Road Accidents
1e6
99.7%
0.2%
0.1%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
0.0%
2.5
2.0
1.5
1.0
0.5
0.0
Number of Accidents
0
20
40
60
80
100
120
140
Visibility (miles)



Insights from the above analysis:

1. The most frequent weather condition during road accidents was fair weather, which was present in 39.9% of cases. Around 41% of cases were associated with overcast weather with a certain amount of clouds.
2. The majority of road accidents, 55%, occurred within the temperature range of 61(F) to 91(F).
3. The humidity range between 81% and 91% accounted for the maximum number of road accidents, 16.8%.
4. Almost all road accidents, experienced wind speeds of less than 100 mph.
5. Visibility range between 0(mi) to 15(mi) accounted for the maximum number of road accidents, 99.97%.

6. Air pressure range between 30(in) to 35(in) accounted for 70% of road accidents.

7. Goals of the project

- To analyze historical accident data to identify patterns and trends in accident occurrence, contributing factors, and potential solutions.
- To develop a predictive model that accurately identifies accident-prone areas and helps reduce the frequency and severity of accidents in the USA.
- To identify the most significant predictors of accidents in the USA, including road conditions, vehicle types, and weather patterns.
- To explore the use of advanced technologies, such as machine learning and artificial intelligence, to improve the accuracy and effectiveness of accident prediction models.
- To develop recommendations and guidelines for policymakers, transportation agencies, and other stakeholders to help prevent accidents and reduce their impact on public safety.

8. Community Contribution

Nowadays, car transportation has become an integral part of our daily lives. Considering the persistently high rates of severe accidents and fatalities, it is inevitable to improve automobiles. Regrettably, traffic accidents have always been a part of the driving experience. This highlights the importance of data-driven approaches to road safety and aims to inspire transportation authorities and the public to take action in promoting such approaches.

8.1 By sharing our findings with the community, we hope to raise awareness to prevent accidents.

- Improved road signage and traffic signals.
- Increased law enforcement presence in high-risk areas.

- Public education campaigns to promote safe driving practices.

8.2 By open sourcing our model, we aim to empower communities to improve road safety.

- Real-time accident alerts and risk assessment to educate drivers in accident-prone areas.
- Optimized route planning, avoiding high-risk areas and reducing the likelihood of accidents.

9 Models/Methodology:

9.1 Apriori algorithm to provide recommendations based on the association rules:

Apriori Algorithm:

The Apriori algorithm is a classic algorithm used in data mining to find frequent item sets in a large dataset. It is based on the principle that if an item set is frequent, then all its subsets must also be frequent.

The algorithm works in two phases. In the first phase, called the "candidate generation" phase, the algorithm generates a set of candidate itemsets of length k, where k is the current length of the frequent itemsets. These candidate itemsets are generated by combining frequent itemsets of length k-1.

In the second phase, called the "candidate pruning" phase, the algorithm scans the dataset to count the frequency of each candidate itemset generated in the first phase. If an itemset does not meet the minimum support threshold, it is discarded as a non-frequent itemset. The frequent itemsets generated in this phase are used to generate candidate itemsets of length k+1, and the process continues until no more frequent itemsets can be found.

Apriori Algorithm has three parts:

1. Support - Fraction of transactions that contain an itemset.

For example, the support of item I is defined as the number of transactions containing I divided by the total number of transactions.

Support(I)=

(Number of transactions containing item I) / (Total number of transactions)

2. Confidence - Measures how often items in Y appear in transactions that contain X

Confidence is the likelihood that item Y is also bought if item X is bought. It's calculated as the number of transactions containing X and Y divided by the number of transactions containing X.

Confidence(I1 -> I2) =

(Number of transactions containing I1 and I2) / (Number of transactions containing I1)

3. Lift - Measure of association between two items in a frequent itemset. It measures how much the occurrence of one item in a frequent itemset increases the probability of the other item in the same frequent itemset.

Lift(I1 -> I2) = (Confidence(I1 -> I2) / (Support(I2))

Methodology:

The preprocessed data is further processed for making transactions. We start by replacing the true/false values with string labels that can be self-explanatory. From the previous modeling, we observed that the absence of bumps, signals, stops, stations, etc. did not add any importance to the classification hence we remove them from the rules. Further we use a Transaction encoder which is a class in Python's machine learning library scikit-learn that is used to convert a list of transactions into a one-hot encoded format suitable for use in frequent itemset mining algorithms such as Apriori.

The TransactionEncoder class takes as input a list of transactions, where each transaction is a list of items, and creates a sparse matrix where each row represents a transaction and each

column represents an item. If an item appears in a transaction, the corresponding element in the matrix is set to 1, otherwise, it is set to 0.

The output of the TransactionEncoder can be fed into an Apriori algorithm to generate frequent itemsets. Based on the frequent itemsets hence generated we produce the association rules with ‘lift’ as a metric and sort based on confidence.

9.2 Decision tree classifier for severity classification with SMOTE:

Decision tree Classifier:

A form of machine learning algorithm called a decision tree classifier is used for supervised learning tasks like classification. Each internal node represents a decision based on a particular trait, and each leaf node represents a classification label. Together, these nodes form a tree-like model of decisions and potential outcomes.

Recursively partitioning the input space according to the values of the input characteristics is how the method constructs the decision tree during training. The objective is to develop decision rules that correctly forecast the incoming data's class label. Based on a criterion like information gain or Gini index, the algorithm chooses the optimum feature to partition the data.

Following the path through the decision tree based on the values of the input characteristics allows the decision tree to be used to generate predictions on fresh data after it has been constructed. The algorithm checks the value of each internal node's relevant feature before moving to the left or right child node depending on whether the value meets a predetermined requirement. The algorithm outputs the corresponding class label once it reaches a leaf node. There are many benefits to using decision tree classifiers, including its usability, interpretability, and capacity for both category and numerical data.

SMOTE :

SMOTE stands for Synthetic Minority Over-sampling Technique, which is a data augmentation method used in machine learning to address the class imbalance. Class imbalance occurs when the number of instances in one class is much smaller than the number of instances in another class, which can lead to poor performance of the model on the minority class.

SMOTE works by generating synthetic samples for the minority class by interpolating between existing minority class instances. The basic idea is to randomly select a minority class instance and then select one or more of its nearest neighbors. Synthetic instances are then generated by creating linear combinations of the features of the selected instance and its neighbors, with some random perturbation added to each feature. The result is a set of new instances that are similar to the existing minority class instances, but not identical.

By increasing the number of minority class instances in this way, SMOTE can help to balance the distribution of classes in the training data and improve the performance of the model on the minority class. However, it is important to note that SMOTE should only be used on training data, and not on the validation or test data, as this can lead to overfitting and poor generalization performance.

Methodology:

As observed there is a class imbalance with respect to the Severity column with a very wide disparity between Severity level 2 and rest 1, 3, and 4 as seen in Figure 24.

Figure 24

```
data['Severity'].value_counts()
```

```
1    2469255
2     151145
3     126348
0      25702
Name: Severity, dtype: int64
```

We initially undersample the severity 2 records to 150000 records and then we upsample the remaining classes in a range of 15000 to 151145 such that there is a close balance in the training data. This upsampling is done using SMOTE by specifying a strategy of desired ratio and fit resampling the data. On this balanced data we split into train and test splits in 70:30 ratio with a random state of 42.

We use DecisionTreeClassifier from scikit learn library to fit the data. We have used both entropy and gini index for building the tree with max_depth 8 and random state 1. On evaluating with the test split an accuracy of 71% is observed for both the criterions.

Figure 25

```
[Decision Tree -- entropy] accuracy_score: 0.716.
[Decision Tree -- gini] accuracy_score: 0.718.
```

Despite handling class balance and using the most widely used classifier it was observed that the model was unable to learn from the nuances of the features like the presence of bump, signal, and other categorical features which it considered of less importance. Hence to propose an approach we experiment with the BERT model next.

9.3 RoBERTa for classifying the severity of the accidents into four categories: Robustly Optimized Bidirectional Encoder Representations from Transformers

Approach:

Facebook AI Research created RoBERTa, a pre-trained transformer-based neural network model for tasks involving natural language processing. The popular BERT (Bidirectional Encoder Representations from Transformers) model's architecture serves as the foundation for RoBERTa, which is trained on a sizable corpus of text data using a modified training methodology that incorporates dynamic masking, longer sequences, and other methods to enhance the model's performance.

The Google-developed pre-trained language model BERT can be adjusted for a range of natural language processing applications, including text categorization. Similar to BERT,

RoBERTa is adaptable for a range of natural language processing applications, such as text classification, question resolution, and named entity recognition. RoBERTa is trained on a particular task during fine-tuning, using a smaller labeled dataset as a starting point rather than the pre-trained weights.

We must first hone BERT on a particular classification task, such as topic classification, before we can utilize it for classification. The pre-trained BERT model is then combined with a task-specific layer, and the entire model is subsequently trained on a labeled dataset.

BERT receives a sequence of tokens and a label corresponding to the classification problem as input during training. A probability distribution across the potential labels is the model's output. The difference between the predicted label and the actual label is measured by a loss function, and the model is trained to minimize this difference.
Once the model has been trained, it can be applied to new text inputs to make predictions. Tokenizing and converting the input text into the training data's format comes first. The output of the task-specific layer is then utilized to create the final prediction once the BERT model has been applied to the input text.

BERT can capture the context and links between words in a sentence, which can improve performance on tasks where the meaning of the text is crucial. This is one benefit of utilizing BERT for text categorization. BERT is widely utilized in both industry and academics for a variety of text categorization tasks and has attained state-of-the-art results on several NLP benchmarks.

Methodology:

We start by installing transformers, sentencepiece, contractions and keras preprocessing. We then import all the required libraries. BERT works by analyzing a long string of text with all the required details based on which it can be classified. Hence we

preprocess and make a string out of the accidents data with all the features that seem relevant for classification. The advantage here is it requires no particular data type and accepts the records in the form of a conversation. Due to high GPU and RAM overhead, we considered a small fragment of data with balanced classes of 1000 records for each class and an overall train data of 4000 records. Further some processing specific to BERT training requirements are done like changing True/False (1,0) to self-explanatory strings say “Bump” and “No Bump”.

Further the severity is split from its text and organized to form a sentence like below.

Figure 26

```
['At Kemp Mill Rd - Accident. in Silver Spring,Montgomery,MD when temperature is 84.0 when wind chill is
84.0 when humidity is 43.0 when pressure is 29.83 when visibility is 10.0 when wind speed is 9.0 when
precipitation is 0.0 with weather condition Fair and No Bump and No Crossing and No Junction and No
Roundabout and No Stop and Traffic_Signal in June a Week day',
 'At 211th St/Exit 12 - Earlier accident. in Miami,Miami-Dade,FL when temperature is 72.0 when wind
chill is 72.0 when humidity is 88.0 when pressure is 30.0 when visibility is 10.0 when wind speed is 6.0
when precipitation is 0.0 with weather condition Mostly Cloudy and No Bump and No Crossing and No
Junction and No Roundabout and No Stop and No Traffic_Signal in May a Weekend',
```

Now we install the Roberta tokenizer that is pretrained and tokenize our text. We then take tokenized and encoded sentences and attention masks. We now split this data into train and validation sets. Converting all of our data into torch tensors, the required data type for our model and creating an iterator of our data with torch DataLoader. This helps save on memory during training because, unlike a for loop, an iterator the entire dataset does not need to be loaded into memory. Next we load the model RobertaForSequenceClassification, the pretrained model will include a single linear classification layer on top for classification. We next set the custom optimizer Adam and other parameters. Now we train the model and keep track of the loss. Validation phase comes next where we validate the model and predict the labels.

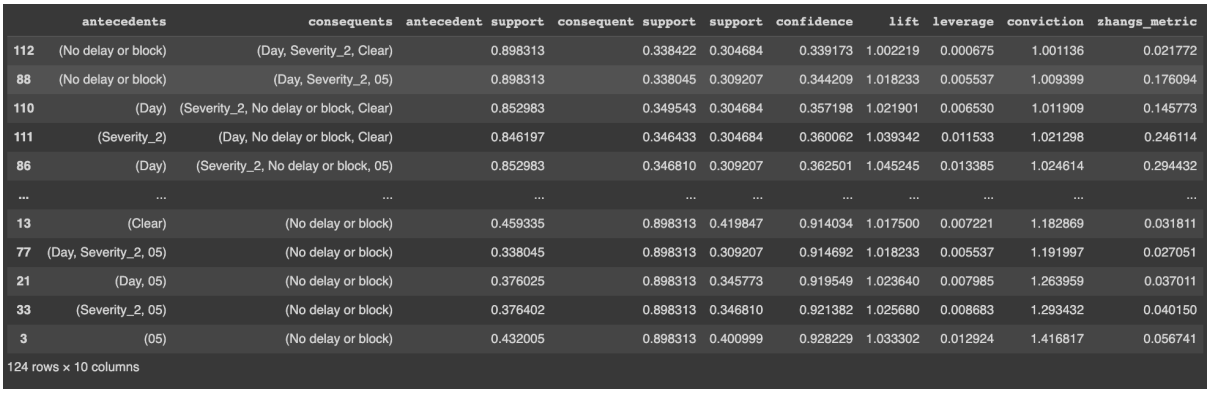
Now we prepare the test data and sampled 10 for each class. We preprocess this test data with the same steps followed for train data preparation. We then tokenize the test text

and predict the classes. Now we find the accuracy and show the classification report. It gave a better accuracy of 85% for just a small fragment of data compared to the decision tree.

Results:

- The association rules show the relation in terms of support and confidence between the features.

  Figure 27



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
112	(No delay or block)	(Day, Severity_2, Clear)	0.898313	0.338422	0.304684	0.339173	1.002219	0.000675	1.001136	0.021772
88	(No delay or block)	(Day, Severity_2, 05)	0.898313	0.338045	0.309207	0.344209	1.018233	0.005537	1.009399	0.176094
110	(Day)	(Severity_2, No delay or block, Clear)	0.852983	0.349543	0.304684	0.357198	1.021901	0.006530	1.011909	0.145773
111	(Severity_2)	(Day, No delay or block, Clear)	0.846197	0.346433	0.304684	0.360062	1.039342	0.011533	1.021298	0.246114
86	(Day)	(Severity_2, No delay or block, 05)	0.852983	0.346810	0.309207	0.362501	1.045245	0.013385	1.024614	0.294432
...	...	...	...	...	...	...	...	...	...	...
13	(Clear)	(No delay or block)	0.459335	0.898313	0.419847	0.914034	1.017500	0.007221	1.182869	0.031811
77	(Day, Severity_2, 05)	(No delay or block)	0.338045	0.898313	0.309207	0.914692	1.018233	0.005537	1.191997	0.027051
21	(Day, 05)	(No delay or block)	0.376025	0.898313	0.345773	0.919549	1.023640	0.007985	1.263959	0.037011
33	(Severity_2, 05)	(No delay or block)	0.376402	0.898313	0.346810	0.921382	1.025680	0.008683	1.293432	0.040150
3	(05)	(No delay or block)	0.432005	0.898313	0.400999	0.928229	1.033302	0.012924	1.416817	0.056741

124 rows × 10 columns

  On sorting the rules based on confidence it can be seen that Severity 2 has very little impact on blockages and delays. Similarly, it was observed that the day and month had a greater confidence with Severity 4 which helps in finding the fatality of any accident based on the day and month.

- Decision tree classifier despite balanced classes done with SMOTE gave a low accuracy of 71% showing that the model couldn't learn how to classify based on the patterns in the data like bumps, crossing, signals, etc. which are the main features in road traffic data.

Figure 28

	precision	recall	f1-score	support
0	0.82	0.94	0.88	44864
1	0.79	0.77	0.78	44799
2	0.62	0.62	0.62	45575
3	0.62	0.55	0.58	45106
accuracy			0.72	180344
macro avg	0.71	0.72	0.71	180344
weighted avg	0.71	0.72	0.71	180344

- Our proposed BERT model for solving this problem showed a greater accuracy for very fewer data. With the availability of better computing power and GPU resources, the model can be fine-tuned and better trained with larger data and achieve greater accuracy. It currently gives 85% accuracy in classifying the severity of accidents.

Figure 29

	precision	recall	f1-score	support
0	0.73	0.80	0.76	10
1	1.00	0.70	0.82	10
2	0.75	0.90	0.82	10
3	1.00	1.00	1.00	10
accuracy			0.85	40
macro avg	0.87	0.85	0.85	40
weighted avg	0.87	0.85	0.85	40

10 Future work:

- Future research could involve conducting similar analyses on different countries to see if the results are consistent. Additionally, we could delve deeper into the characteristics of the drivers and vehicles involved in accidents, such as age, gender, profession, car type, and ownership, as this information could provide insights into the psychological factors influencing driving behavior.

- It may be useful to integrate the findings of this study into a real-time accident risk prediction model or develop a new model to predict severe accident risk in specific grid cells. The BERT classification can be added as an accident help bot where the data could be given in a contextual, conversational way and the bot can help understand the possible severity and give an insight into the preparation needed for handling the situation.

11 Conclusion:

Our proposed approach BERT for solving the severity classification of accidents has shown great results of 85% accuracy, especially a perfect F1 score in identifying the fatal accidents that are considered severity 4. This proves that our intuition of trying the transfer learning of RoBERTa, a generally classification model for Natural Language, can also work for a specific domain like in this case the accident data.

Presentation slides:

https://prezi.com/p/edit/qduev5ychtii/?lid=x1kxl8roj2s0&utm_source=braze&utm_medium=email&utm_content=Tokenization+Variant&utm_campaign=Next_Share_A_Prezi_v2&UID=326231818

References

Krishnaveni, S., & Hemalatha, M. (2011). A perspective analysis of traffic accident using data mining techniques. *International Journal of Computers & Applications, 23(7), 40–48.*

Li, L., Shrestha, S., & Hu, G. (2017). Analysis of road traffic fatal accidents using data mining techniques. *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA),* 363–370.

Moosavi, S., Samavatian, M. H., Parthasarathy, S., & Ramnath, R. (2019). A Countrywide Traffic Accident Dataset. *ArXiv:1906.05409 [Cs]*. https://arxiv.org/abs/1906.05409

Moosavi, S., Samavatian, M. H., Parthasarathy, S., Teodorescu, R., & Ramnath, R. (2019). Accident Risk Prediction based on Heterogeneous Sparse Data. *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '19*. https://doi.org/10.1145/3347146.3359078

Nour, M., Naseer, A., Alkazemi, B., & Jamil, M. (2020). Road Traffic Accidents Injury Data Analytics. *IJACSA) International Journal of Advanced Computer Science and Applications*, *11*(12).https://thesai.org/Downloads/Volume11No12/Paper_87-Road_Traffic_Accidents_Injury_Data_Analytics.pdf

Shweta, Yadav, J., Batra, K., & Goel, A. K. (2021). A Framework for Analyzing Road Accidents Using Machine Learning Paradigms. *Journal of Physics: Conference Series*, *1950*(1), 012072. https://doi.org/10.1088/1742-6596/1950/1/012072