

CMP4011 Big Data and Cloud Computing

Project Report

Team 10

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Sec** | **B.N** | **Code** |
| Ahmed Osama Helmy | 1 | 5 | 9213061 |
| Omar Mahmoud | 1 | 29 | 9210758 |
| Abdallah Ahmed | 1 | 25 | 9210652 |
| Aliaa Gheis | 1 | 27 | 9210694 |

Contents

[Problem Statement: 3](#_Toc196736372)

[Dataset 3](#_Toc196736373)

[Project Pipeline: 4](#_Toc196736374)

[Data Ingestion 4](#_Toc196736375)

[Data Cleaning 4](#_Toc196736376)

[Feature Engineering 6](#_Toc196736377)

[Data Visualization: 7](#_Toc196736378)

[Descriptive Analysis (Insights): 8](#_Toc196736379)

[Predictive Analysis (Insights): 15](#_Toc196736380)

[Cloud Computing Used: 18](#_Toc196736381)

[Fully Clustered Mode: 22](#_Toc196736382)

## Problem Statement:

Road safety is a critical concern, and understanding accident patterns can help cities improve traffic management and reduce accident rates. This project aims to analyze accident data to identify high-risk locations, contributing factors, and potential mitigation strategies. By leveraging big data processing, we will extract valuable insights for transportation authorities and urban planners.

## Dataset

**Dataset Name:** US Accidents (2016 - 2023)

**Link:** <https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents>

**Description:**

* Contains 7.7 million accident records with 46 columns
* Main features of accident are (Severity of Accident, Time, Location, Weather, Road Characteristics)

## Project Pipeline:

### Data Ingestion

This stage involves loading the data and specifying the schema of the loaded data which is as follows:

|  |  |  |
| --- | --- | --- |
| Category | Fields | Data Types |
| Incident Info | ID, Source, Severity, Start\_Time, End\_Time, Description | String, Integer, Timestamp |
| Location | Start/End\_Lat/Lng, Distance(mi), Street, City, County, State, Zipcode, Country, Timezone, Airport\_Code | Double, String |
| Weather | Weather\_Timestamp, Temperature(F), Wind\_Chill(F), Humidity(%), Pressure(in), Visibility(mi), Wind\_Direction, Wind\_Speed(mph), Precipitation(in), Weather\_Condition | Timestamp, Double, String |
| Road Features | Amenity, Bump, Crossing, [...] (14 boolean flags) | Boolean |
| Time of Day | Sunrise\_Sunset, Civil/Nautical/Astronomical\_Twilight | String |

### Data Cleaning

#### Handling Missing Values and Nulls

* This was handled by first checking the percentage of missing values in each column and sorting them descending.
* Dropping Columns like End\_Lat & End\_Lng as the percentage of missing values was greater than 40%
* Imputing missing values in numeric columns by inserting the mean value
* Imputing missing values in categorical columns by inserting the mode value

#### Removing Columns with 1 Unique Value

* Features like Country and Turning\_Loop has only one unique value. Thus, they won’t be helpful in analysis

#### Column Casting

* Ensures that the columns are in the correct data types for further processing and model compatibility.

#### Removing Irrelevant Columns

* Eliminates columns that don’t provide meaningful data for modeling or analysis.
* Columns such as ID, Source, Description, Street, City, Zipcode, Airport\_Code, etc., are dropped as they are not useful for analysis or prediction tasks.

#### Handling Outliers

* Eliminating records with temperature higher than 56.7 C as reported in this [article](https://www.visualcapitalist.com/charted-the-hottest-and-coldest-temperatures-in-u-s-history/#:~:text=The%20hottest%20temperature%20ever%20recorded,%2C%20on%20July%2010%2C%201913.) that the maximum US temperature was 134.4°F (56.7°C)
* For Wind Speed values, we identified outliers by considering the maximum observed wind speeds. According to the World Meteorological Organization, the highest recorded wind speed was 254 mph (408 km/h). We decided to remove any records with wind speeds exceeding this threshold to eliminate extreme outliers.

#### Handling Bias in Data

* The data is not equally distributed on 4 values of Severity. Thus, we tried different techniques to try to solve it.
* The Data first was

### Feature Engineering

#### Adding Time-Related Features

* The raw timestamp (Start\_Time) was parsed to extract granular temporal components:
  + **Hour of the Day**: Captures the time of day when accidents occur (e.g., morning rush hours, nighttime).
  + **Day of the Week**: Identifies whether accidents are more frequent on weekdays or weekends.
  + **Month**: Highlights seasonal trends in accident occurrences (e.g., higher rates during winter months due to adverse weather conditions).
  + **Year**: Tracks long-term trends in accident frequency over multiple years.
  + **Duration**: Tracks the duration of the accident in minutes by subtracting the start time from the end time
  + **Season**: Determines the Season when the accident happened (Summer, …)

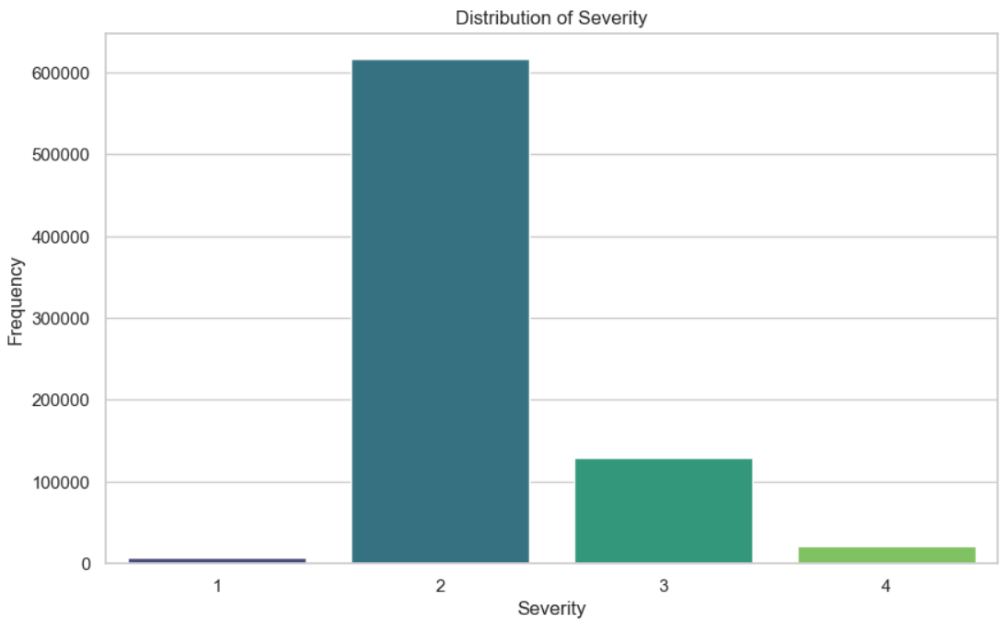
#### Adding Road-Related Features

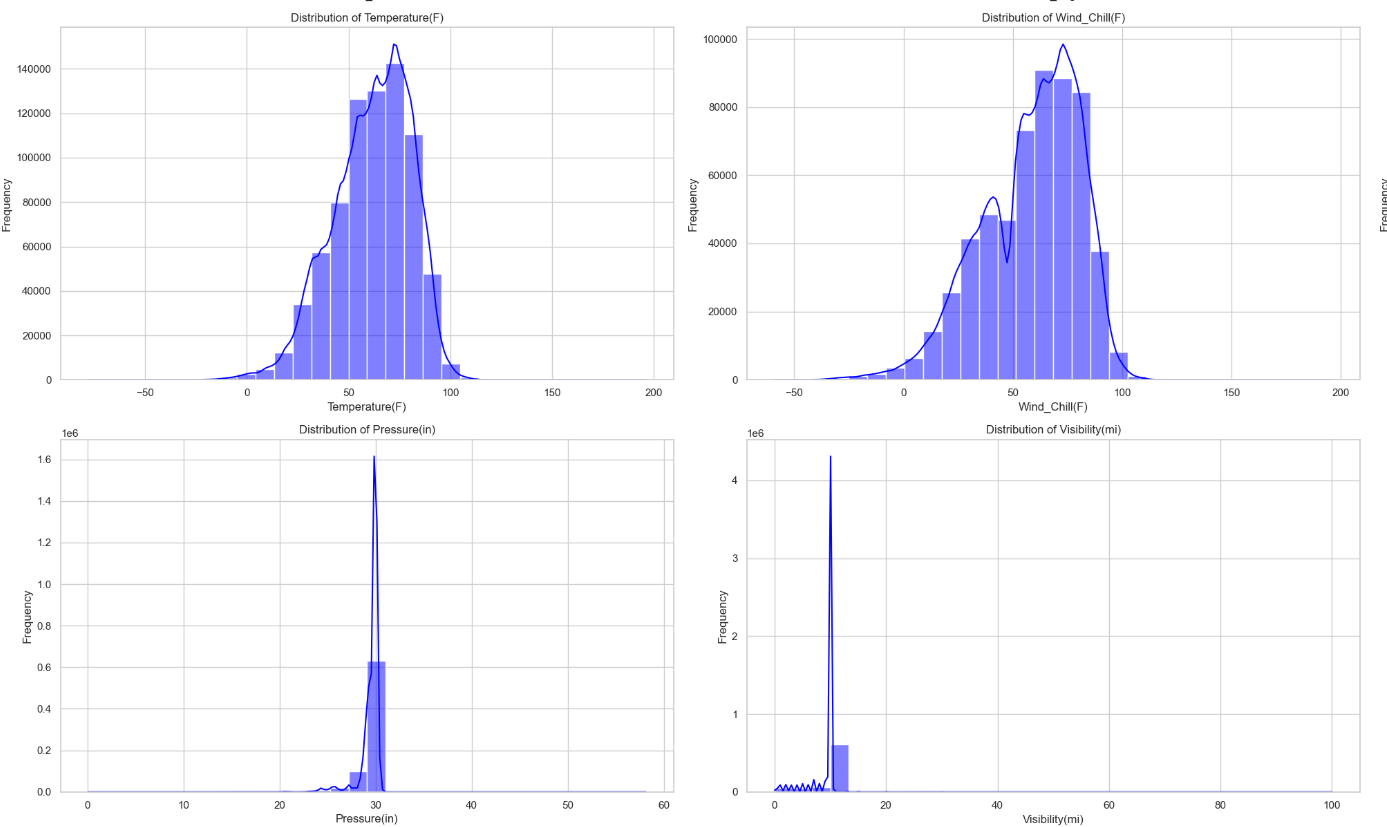
* A Boolean variable Is\_Complex\_Road was added to interpret whether the road is complex by utilizing the other variables like (Junction, Railway, Crossing)
* This will help in giving insights into the effect of complexity of roads.

#### Adding Risk Score for each State

* The dataset was aggregated at the state level to simplify the analysis and provide actionable insights at a regional scale.
* Key metrics such as average accident severity, total accident count, and risk score were computed for each state.
* The **Risk score** was calculated as the product of average severity and accident count, capturing both the frequency and severity of accidents.
* The **Risk score** was then normalized to be from 0 to 1.
* Also, A Boolean variable **Is\_High\_Risk** was added to detect if a state was high risky or not by using the 75th quartile.

## Data Visualization:

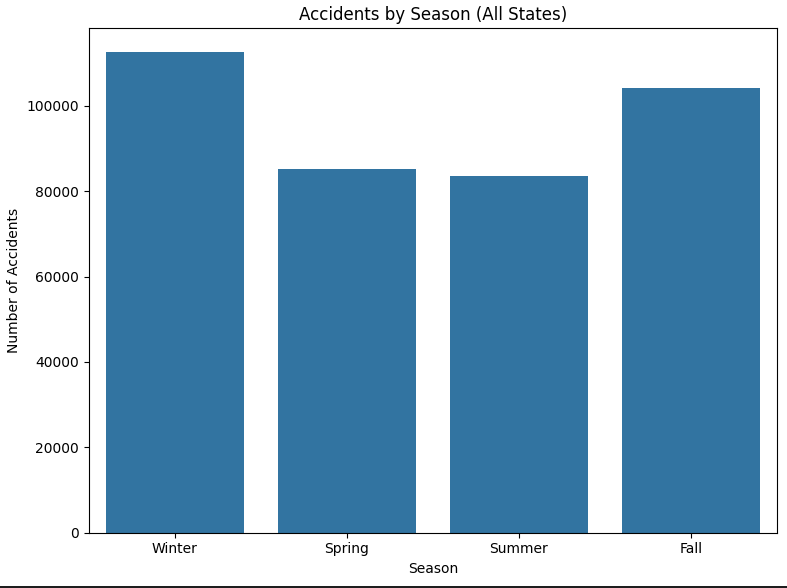
* The distribution of severity shows that the data is unbalanced where the number of accidents with severity 2 is leading. This needs to be handled by techniques like oversampling or under sampling.
* The Distribution below can show that some weather features have outliers that needs to be handled.



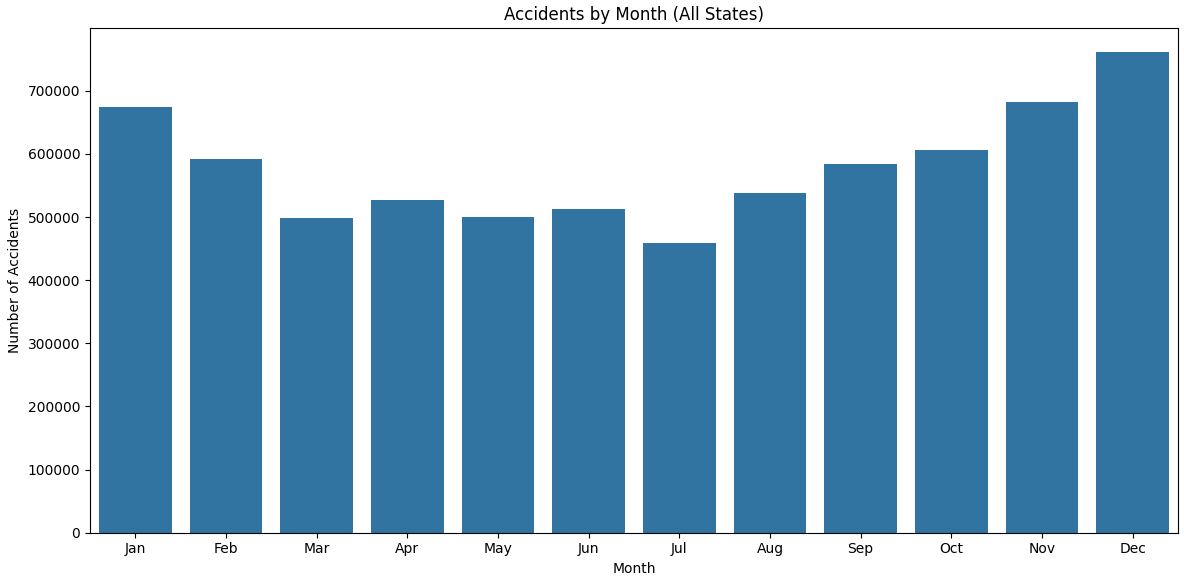
## Descriptive Analysis (Insights):

#### Accident Trends by Season.

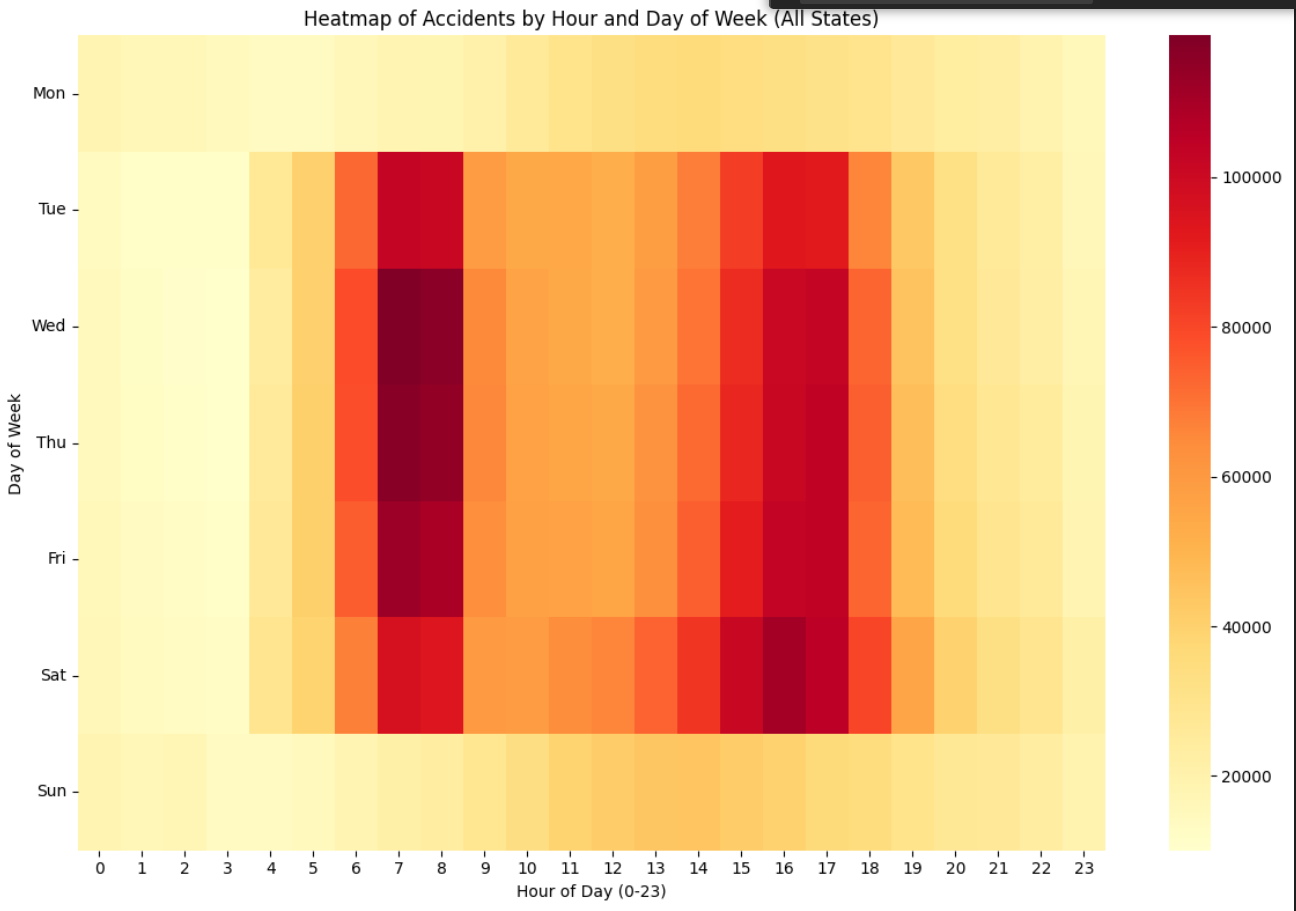
* Winter and Fall experience more accidents due to hazardous weather, such as snow, ice, and reduced daylight.



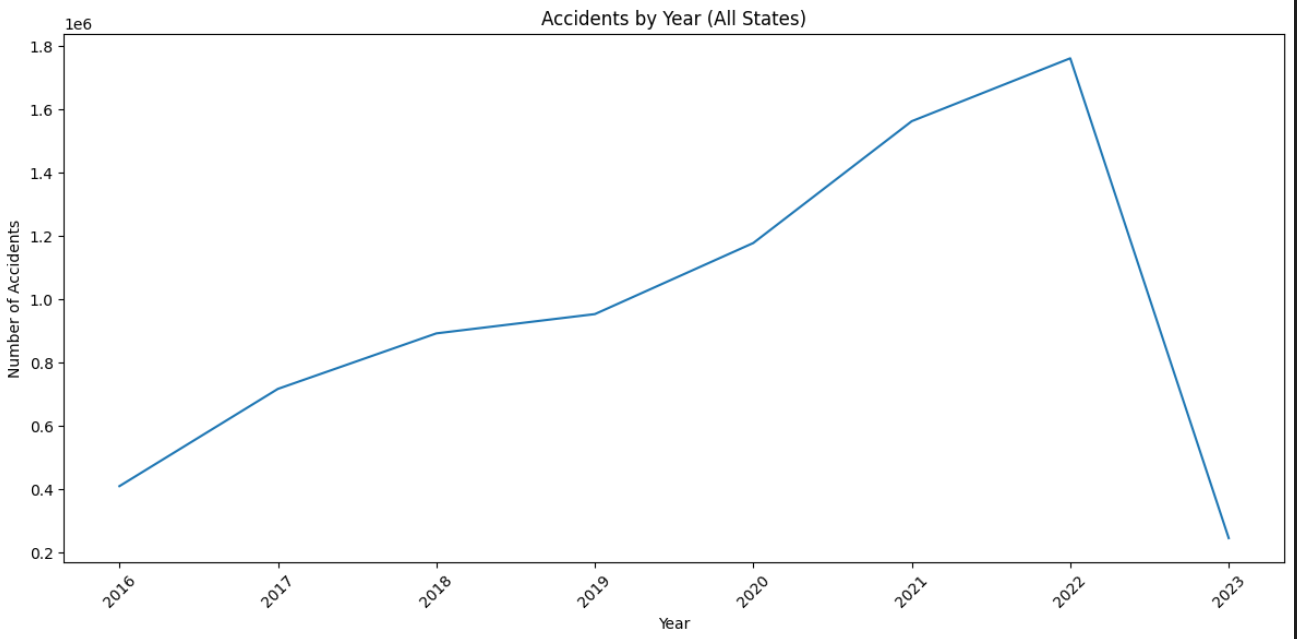
#### Accident Trends by Month.

* July seems the safest month
* Jan & Dec having high records of accidents probably due to holidays

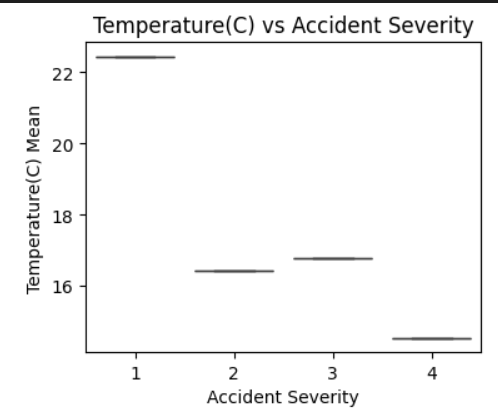
#### Accident Trends by Hour.

* Rush hours (7-9 AM and 3-6 PM) have the highest accident frequency due to increased traffic.

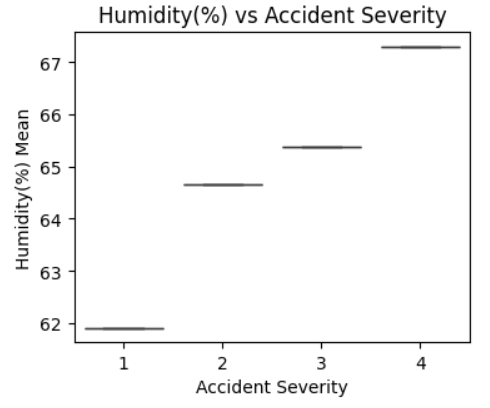
#### Accident Trends by Year.

* Accident counts steadily grew from 2016 to 2022, nearly quadrupling in this period, indicating an increase in accident frequency.
* The sharp drop in 2023 accidents is likely due to incomplete data (up to March), not a true decline.

#### Relation between Temperature and Accident Severity.

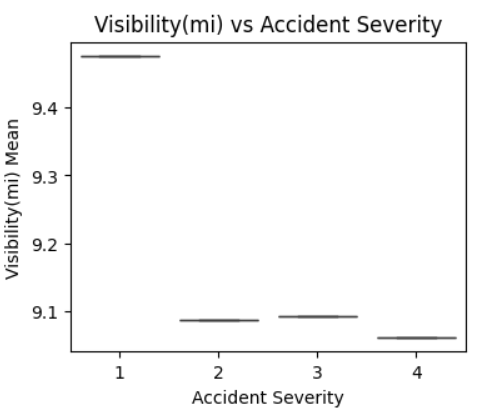
* The data suggests that accidents occurring at lower temperatures tend to be associated with greater severity.
* Probably, Icy conditions or other cold-weather hazards could play a significant role in the seriousness of these incidents

#### Relation between Humidity and Accident Severity.



* Accidents happening with higher humidity levels appear to correlate with increased severity.
* Perhaps rain or other moisture-related factors contribute to more impactful collisions.

#### Relation between Visibility and Accident Severity.

* Lower visibility is strongly linked to more severe accidents.
* This underscores the critical impact of clear sight on road safety and the potential dangers of driving in compromised visual conditions.

#### Top Cities contributing with Accidents.

* A graph of blue rectangular bars

  AI-generated content may be incorrect.Miami, Houston & Los Angeles are the top cities contributing to accidents

#### Top States contributing with Accidents.

* California, Florida & Texas are leading the Accidents contribution

A graph of a number of states with the highest accident counts

AI-generated content may be incorrect.

#### Clustering using K-Means based on Location.

* Accidents are geographically concentrated into 5 major clusters across the US.
* A map of different colors

  AI-generated content may be incorrect.Each cluster corresponds to a distinct geographic zone, showing different accident patterns based on location.
* K used here is 5

#### MapReduce K-means.

* Map Reduce was implemented using Spark RDDs to perform K-means
* Feature Extraction: The dataset was enriched by extracting time-based features (hour\_of\_day, day\_of\_week, month, year) and creating categorical columns like weather\_condition\_cat (for weather conditions) and binary flags like is\_night (indicating night accidents) and severe\_accident (indicating high-severity accidents).
* Missing values in columns like Temperature(F), Wind\_Speed(mph), and Humidity(%) were imputed using the Imputer transformer, while rows with missing critical columns (Severity, Start\_Lat, Start\_Lng) were dropped to ensure a clean dataset for further analysis.
* The DataFrame was transformed into an RDD of tuples, where each tuple contains a point (latitude and longitude) and metadata (such as accident severity, weather condition, and city/state). This conversion is essential for applying the MapReduce process efficiently in the next steps.
* **Map Phase:** 
  + The task is to assign each point to the nearest cluster using Euclidean Distance by iterating over all the centroids and choosing the least distance between the point and the centroids
* **Reduce Phase:** 
  + Map Phase Output is that all points are assigned to a cluster
  + The reduce phase (using reduceByKey) recalculates the new centroids based on the new assignments

## Predictive Analysis (Insights):

#### Predicting Accident Severity using Random Forest.

* Feature engineering was applied: extracting hour of day, day of week, month, and creating new features like is\_night and weather\_condition\_cat.
* **Weather conditions** were **categorized** into groups like clear, rain, snow, fog, etc., using **label encoding** (weather\_condition\_cat) to numerically represent different weather scenarios.
* Features were assembled into a single vector for modeling using VectorAssembler.
* A Random Forest Classifier was trained to predict accident severity.
* The model evaluation showed an 81% accuracy in predicting the severity of accidents.
* Weather conditions (rain, snow, fog, etc.) had the highest impact on accident severity, with an Importance score of 59.60%, emphasizing that businesses must account for weather factors in operational planning.

A graph with blue bars

AI-generated content may be incorrect.

* Another Trial was performed after performing feature engineering which is to use under sampling to make Severity uniformly distributed and the following was shown:
  + **Accuracy**: The accuracy decreased from 0.7967 to 0.6822 after undersampling, indicating that the model's overall correctness reduced.
  + **Precision**: Precision increased from 0.6347 to 0.6848, suggesting that the model became better at predicting positive instances correctly.
  + **Recall**: Recall decreased from 0.7967 to 0.6822, showing that the model's ability to identify all positive instances reduced.
  + **F1 Score:** The F1 Score decreased from 0.7065 to 0.6557, reflecting a balance between precision and recall.

#### Binary Classification for Severity

* Is\_Severe has been identified true if it’s 3 or 4 and 1 or 2 as false
* Logistic Regression, Random Forest, Gradient-Boosted Trees, and Decision Tree models have been tried
* Gradient-Boosted Trees performed the best in terms of AUC (0.8089), F1 Score (0.7660), and Accuracy (0.8192), indicating it has the highest overall performance among the models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | AUC | F1 | Accuracy |
| Logistic Regression | 0.7297 | 0.7492 | 0.8077 |
| Random Forest | 0.7509 | 0.7184 | 0.8053 |
| Gradient-Boosted Trees | 0.8089 | 0.7660 | 0.8192 |
| Decision Tree | 0.6452 | 0.7438 | 0.8112 |

Severity (Multiclass)

Random Forest (Undersampled)

Features:   
Accuracy: 0.6821712330546371

Precision: 0.6848113710718194

Recall: 0.6821712330546371

F1 Score: 0.6557234283316703

categorical\_cols = ["State", "Timezone", "Weather\_Condition", "Sunrise\_Sunset", "Season"]

numerical\_cols = [

    "Start\_Lat", "Start\_Lng", "Distance", "Temperature", "Humidity",

    "Pressure", "Visibility", "Wind\_Speed", "Is\_Complex\_Road",

    "Hour", "Day\_of\_Week", "Month", "Year", "DayOfWeek", "Duration",

    "Is\_Weekend"

]

boolean\_cols = [

    "Amenity", "Bump", "Crossing", "Give\_Way", "Junction",

    "No\_Exit", "Railway", "Roundabout", "Station", "Stop",

    "Traffic\_Calming", "Traffic\_Signal"

A blue squares with numbers and labels

AI-generated content may be incorrect.]

Random Forest (No Undersampling)

Features:   
Accuracy: 0.7966703509506979

Precision: 0.6346836480839083

Recall: 0.7966703509506979

F1 Score: 0.7065109609540438

categorical\_cols = ["State", "Timezone", "Weather\_Condition", "Sunrise\_Sunset", "Season"]

numerical\_cols = [

    "Start\_Lat", "Start\_Lng", "Distance", "Temperature", "Humidity",

    "Pressure", "Visibility", "Wind\_Speed", "Is\_Complex\_Road",

    "Hour", "Day\_of\_Week", "Month", "Year", "DayOfWeek", "Duration",

    "Is\_Weekend"

]

boolean\_cols = [

    "Amenity", "Bump", "Crossing", "Give\_Way", "Junction",

    "No\_Exit", "Railway", "Roundabout", "Station", "Stop",

    "Traffic\_Calming", "Traffic\_Signal"

]

Random Forest (Undersampled)

Features:   
Accuracy: 0.6821712330546371

Precision: 0.6848113710718194

Recall: 0.6821712330546371

F1 Score: 0.6557234283316703

categorical\_cols = ["Weather\_Condition", "Season"]

numerical\_cols = [

    "Start\_Lat", "Start\_Lng", "Distance", "Temperature", "Humidity",

    "Pressure", "Visibility", "Wind\_Speed", "Is\_Complex\_Road",

    "Hour", "Day\_of\_Week", "Month", "Year", "DayOfWeek", "Duration",

    "Is\_Weekend"

]

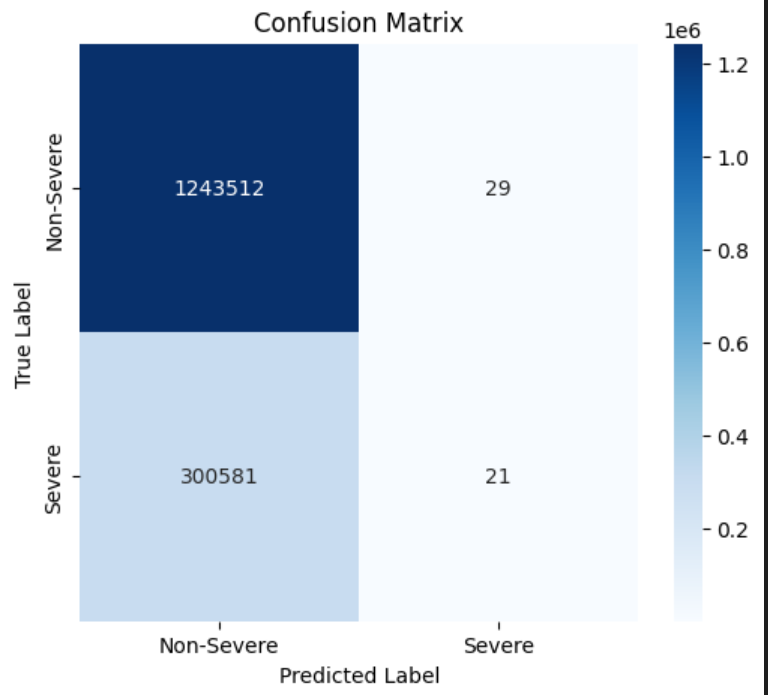
#### Predicting Accident Distance using Linear Regression.

* The Linear Regression model achieved **Root Mean Squared Error (RMSE):** 1.75
* This means, on average, the model's predicted accident distance is about 1.75 miles away from the actual distance.
* Considering the scale of the 'Distance(mi)' which can go up to over 200 miles according to the plot, an average error of 1.75 miles, while seemingly small in isolation, is significant given the model's inability to predict larger distances, further reinforcing the conclusion that the model is not performing well across the full range of possible values.
* It also achieved R2 of 0.01 and MSE of 3.07 which indicates poor performance.

A graph with a red line and a dotted line

AI-generated content may be incorrect.

#### Predicting Accident Severity using Logistic Regression.

* The **Logistic Regression** model achieved an accuracy of 81%, indicating good overall performance in predicting severe accidents.
* The model shows a balance between Precision (73%) and Recall (81%), with a F1-Score of 72%, reflecting its ability to identify severe accidents while minimizing false positives.
* The feature importance analysis of the Logistic Regression model shows that weather conditions (weather\_condition\_cat) have the greatest impact, with a negative coefficient of -0.29, suggesting that adverse weather conditions reduce the likelihood of a severe accident.
* A graph with blue squares

  AI-generated content may be incorrect.Wind speed (Wind\_Speed\_imputed) and nighttime conditions (is\_night) are also significant factors, with positive coefficients indicating that both higher wind speeds and nighttime conditions increase the likelihood of severe accidents.

## Cloud Computing Used:

* A screenshot of a computer

  AI-generated content may be incorrect.Used Azure Databricks to use the clusters with 2 setups, one with single Machine and other cluster with 1 driver and 2 worker nodes
* After creating the Azure-Databricks instance, a cluster was created and configured with 2 workers and one driver. In addition, the cluster was then modified to have only one node working to optimize cost.

A screenshot of a computer

AI-generated content may be incorrect.

* After creating the cluster, a notebook was created and attached to the cluster created
* **Databricks Runtime**: Version 15.4 LTS, which includes Apache Spark 3.5.0, ensuring compatibility with our PySpark codebase.
* **Cluster Composition:** One driver node and two worker nodes, each using the Standard\_E4d\_v4 virtual machine type (4 vCPUs, 32 GB RAM), to enable parallel task execution across multiple nodes.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

* A screenshot of a computer

  AI-generated content may be incorrect.A screenshot of a computer

  AI-generated content may be incorrect.A screenshot of a computer

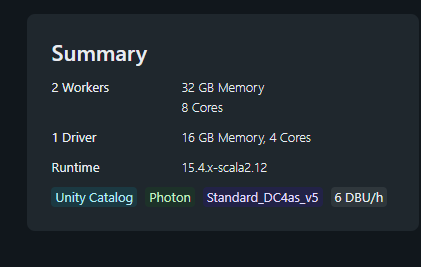
  AI-generated content may be incorrect.During the execution, the Spark jobs were monitored to check their performance and their success
* A screenshot of a computer screen

  AI-generated content may be incorrect.Several Algorithms were trained on the cluster utilizing the resources given. Below, the results of K-means are shown as an example.

A screen shot of a computer screen

AI-generated content may be incorrect.

## Fully Clustered Mode:

* Using 2 worker nodes on Azure Databricks, Fully Distributed mode was implemented dividing jobs among the workers.

