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:

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 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 (e.g. Oversampling (SMOTE) , Undersampling) to try to solve it.

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

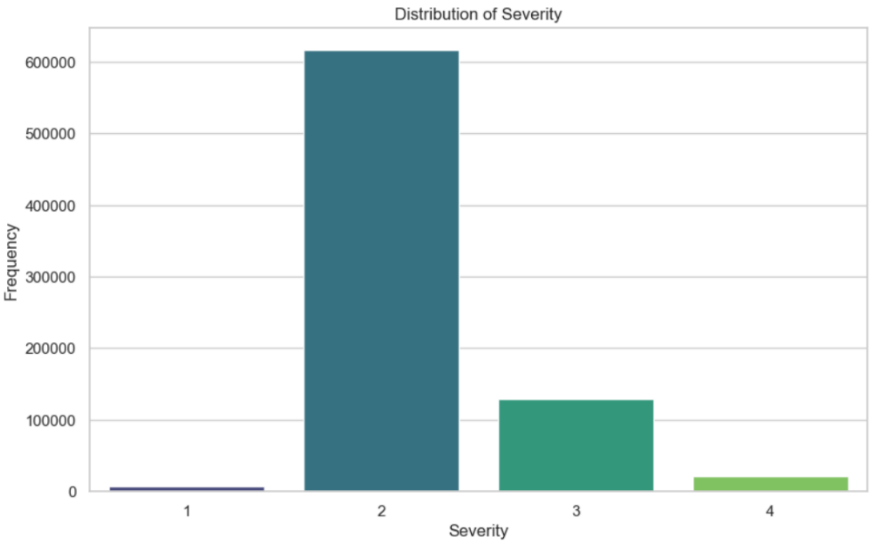
- A Boolean variable Is_Complex_Road was added to interpret whether the road is complex by utilizing the other variables like (Junction, Railway, Crossing)
- This will help in giving insights into the effect of complexity of roads.

Adding Risk Score for each State

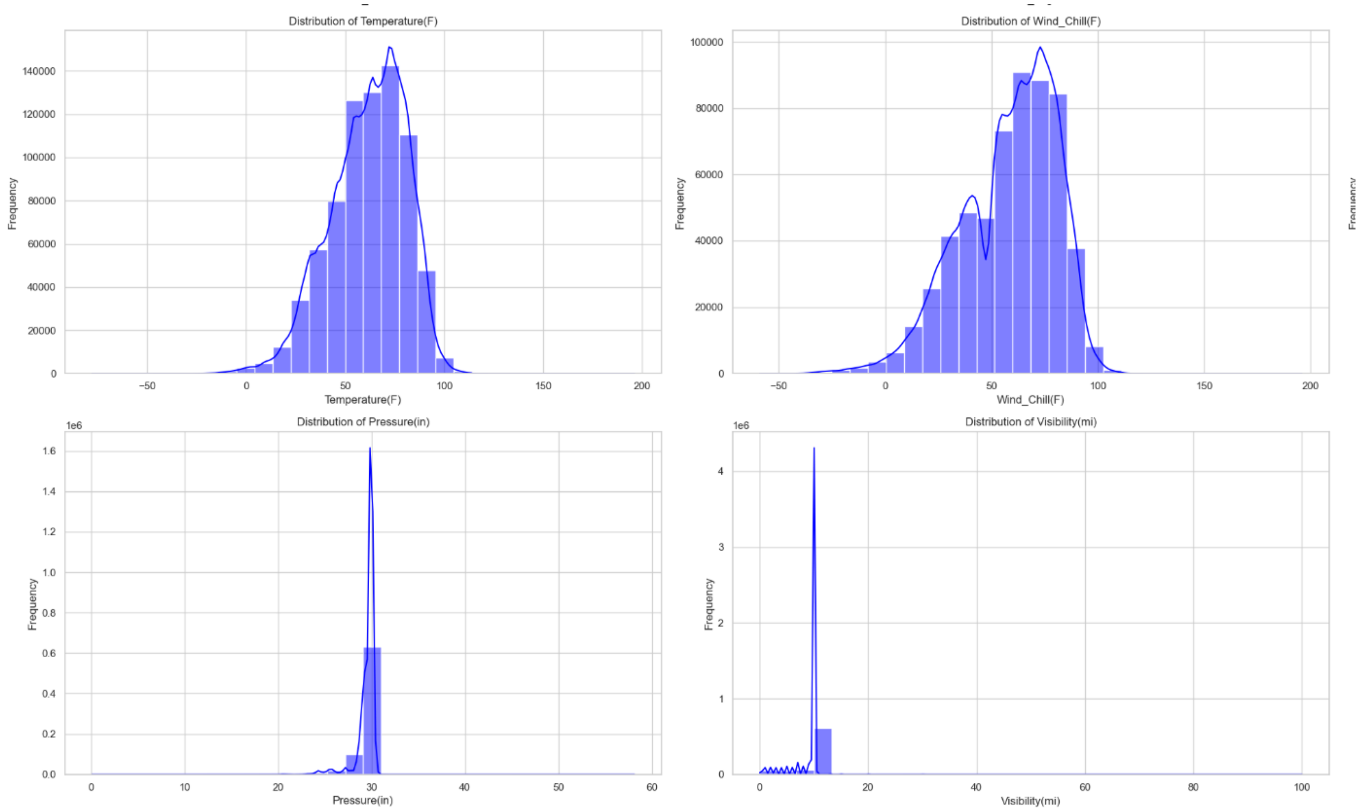
- The dataset was aggregated at the state level to simplify the analysis and provide actionable insights at a regional scale.
- Key metrics such as average accident severity, total accident count, and risk score were computed for each state.
- The **Risk score** was calculated as the product of average 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.



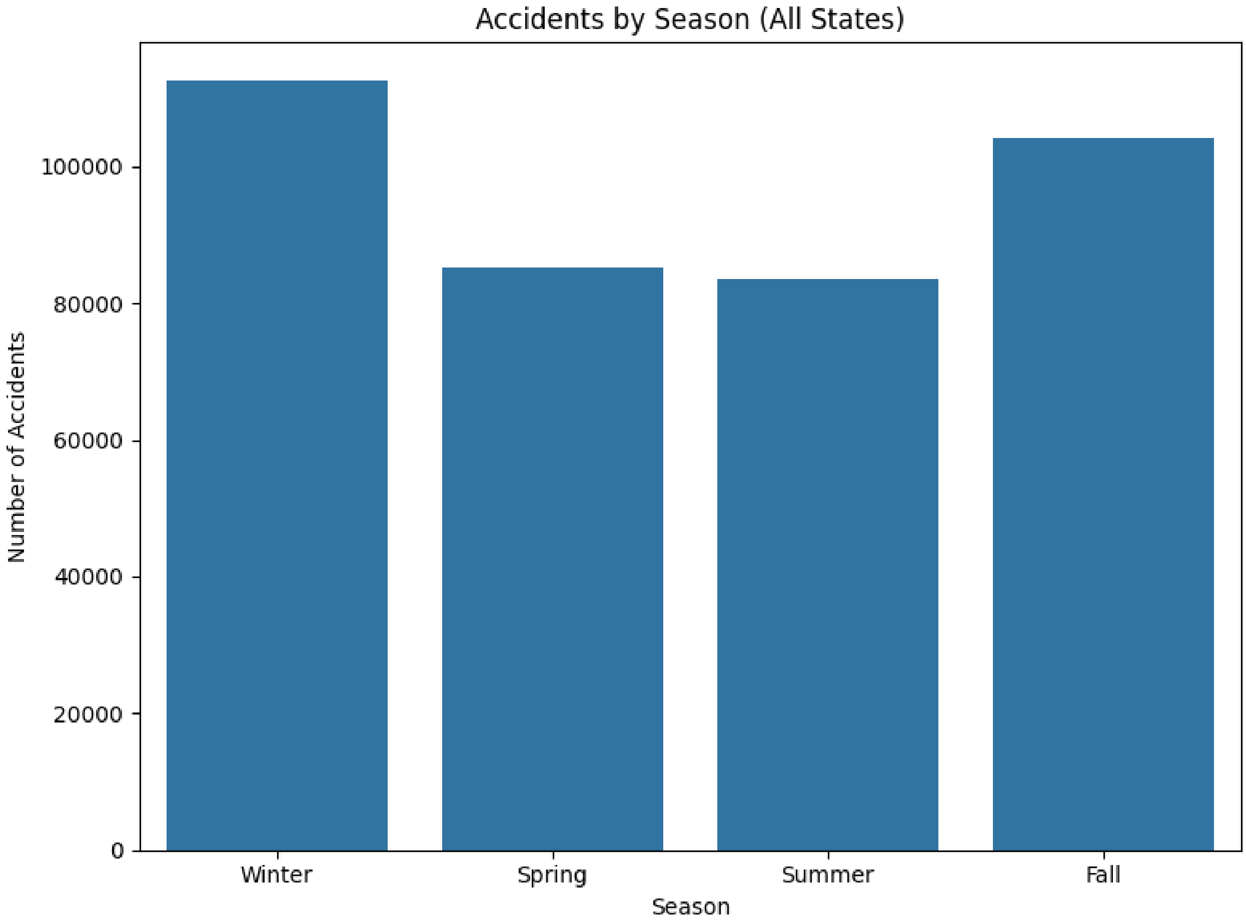
- The Distribution below can show that some weather features have outliers that need to be handled.



Descriptive Analysis (Insights):

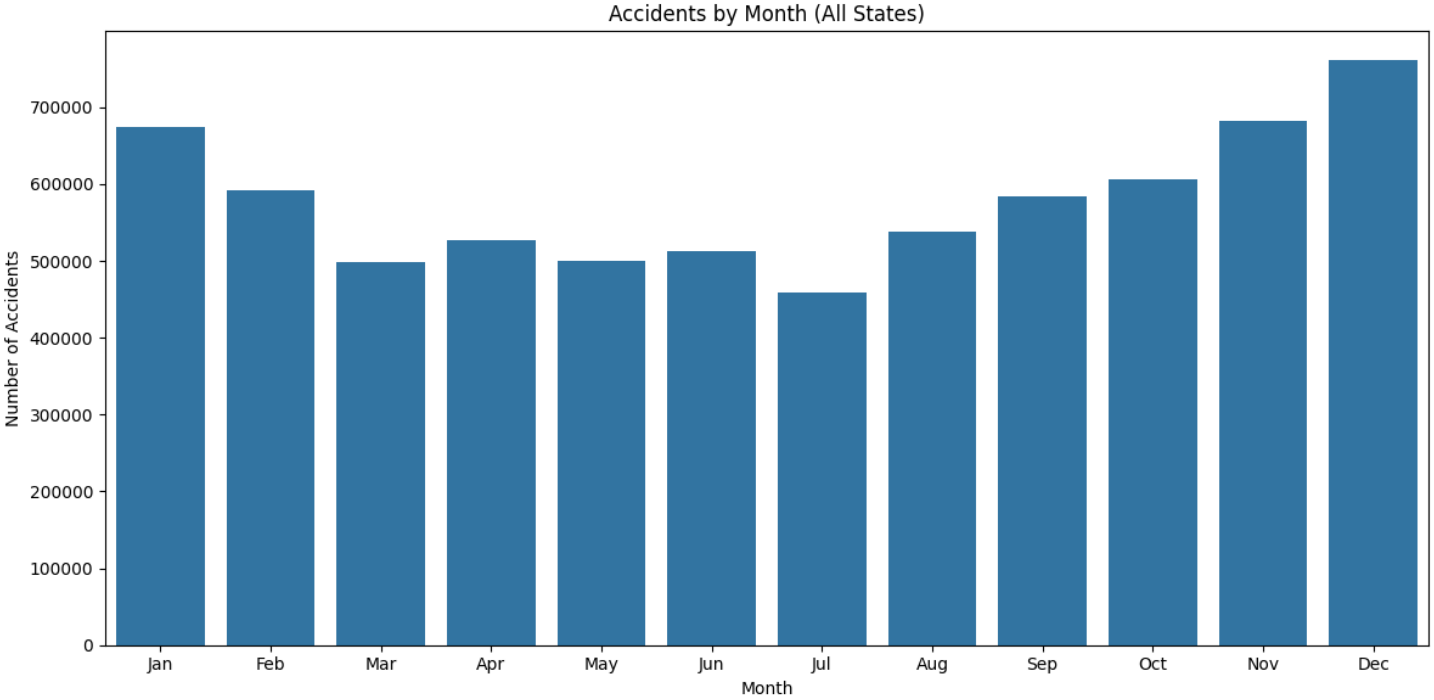
1. Accident Trends by Season.

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



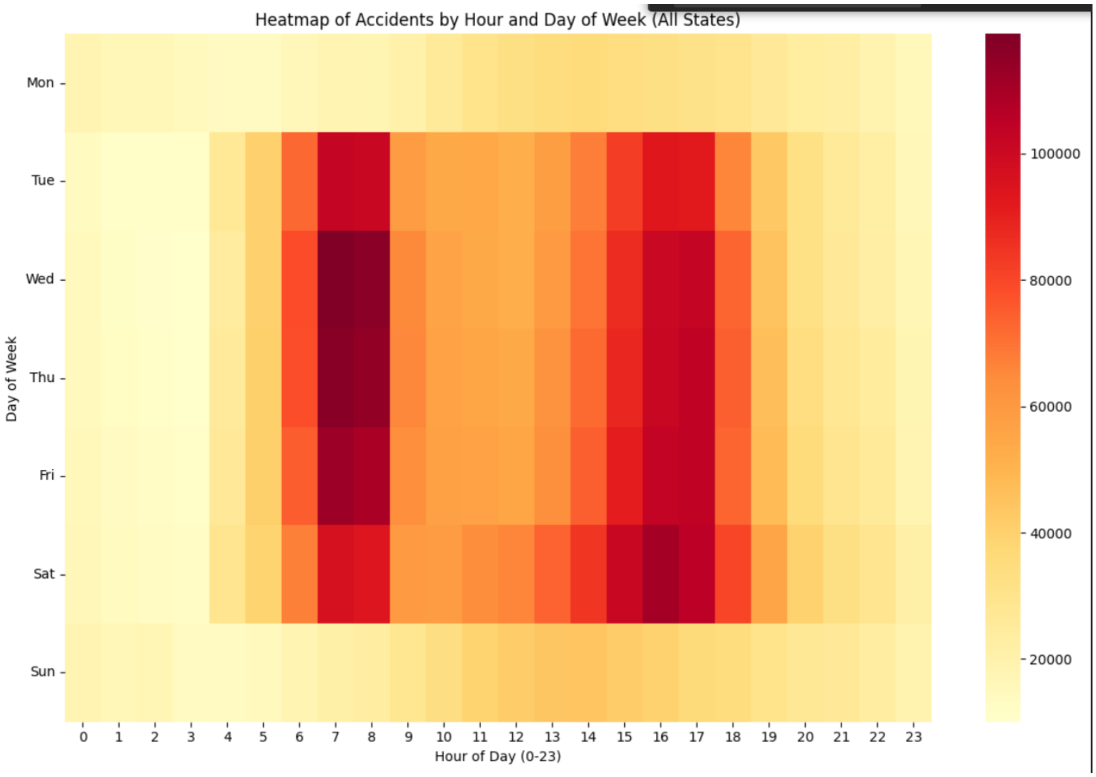
2. *Accident Trends by Month.*

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



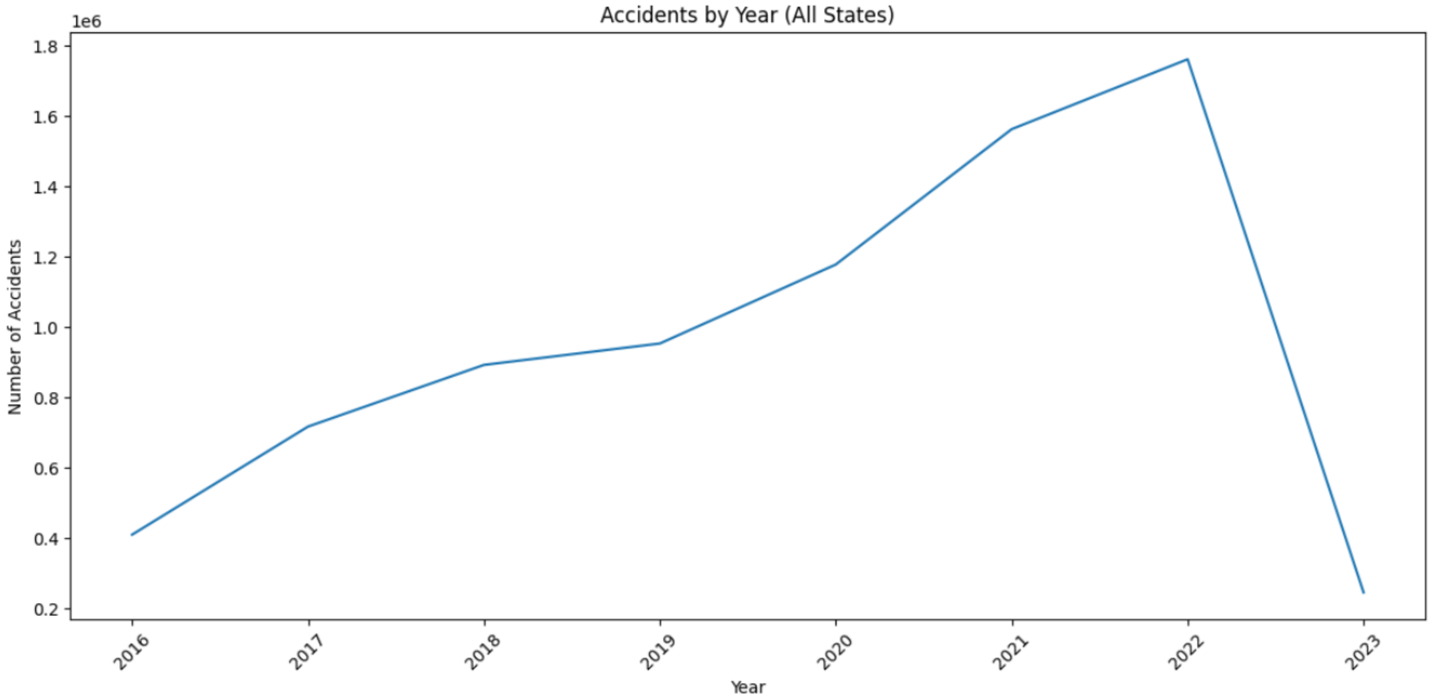
3. *Accident Trends by Hour.*

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



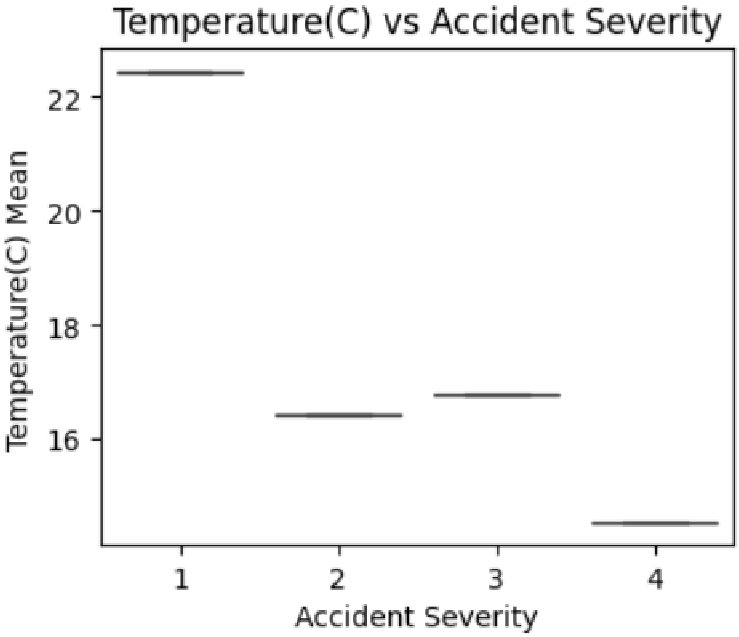
5. *Accident Trends by Year.*

- 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.



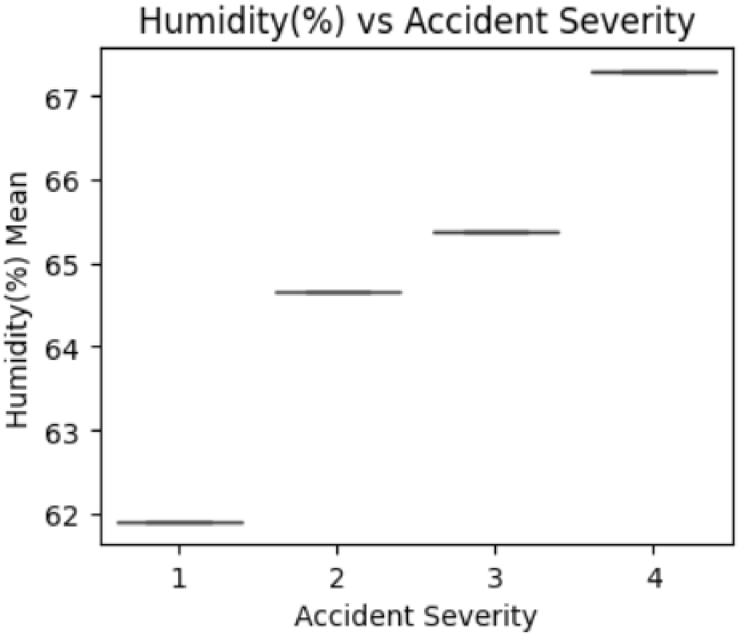
6. *Relation between Temperature and Accident Severity.*

- 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



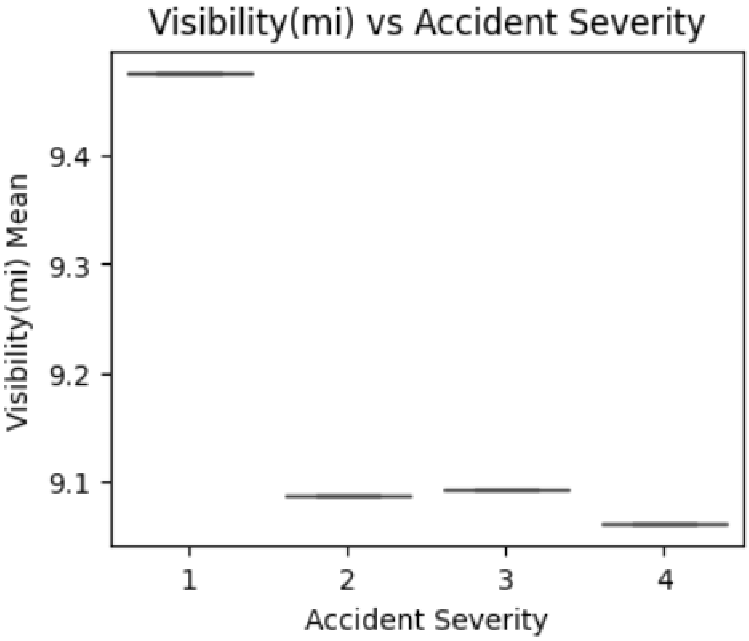
7. *Relation between Humidity and Accident Severity.*

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



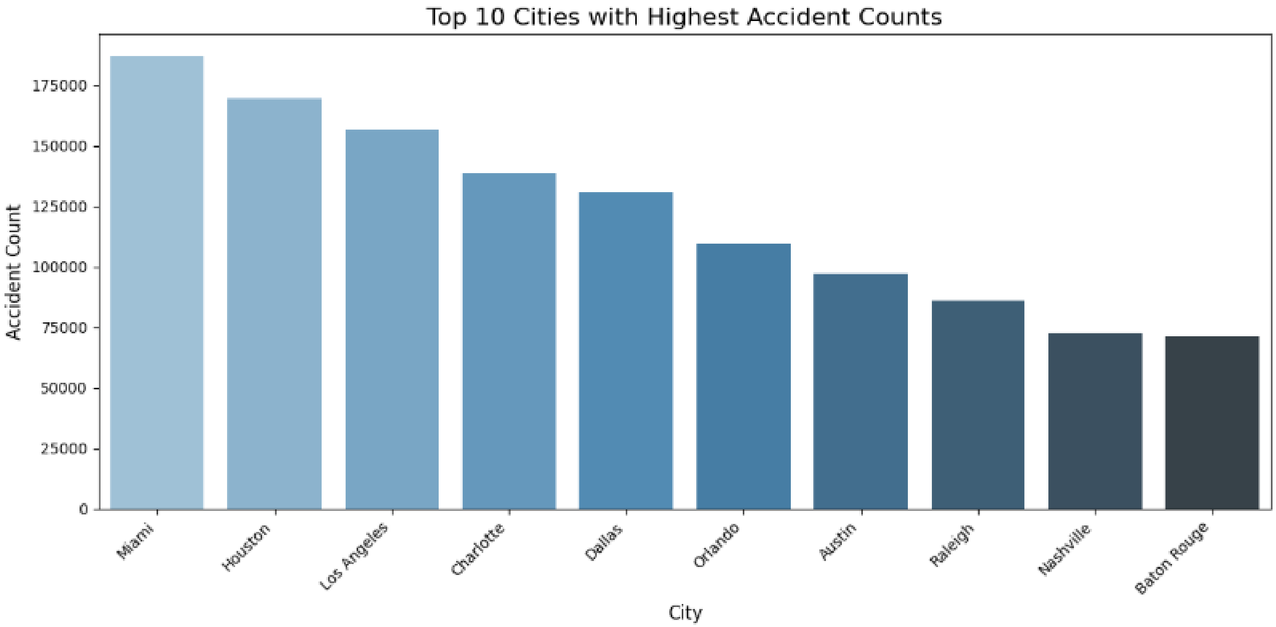
8. *Relation between Visibility and Accident Severity.*

- 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.



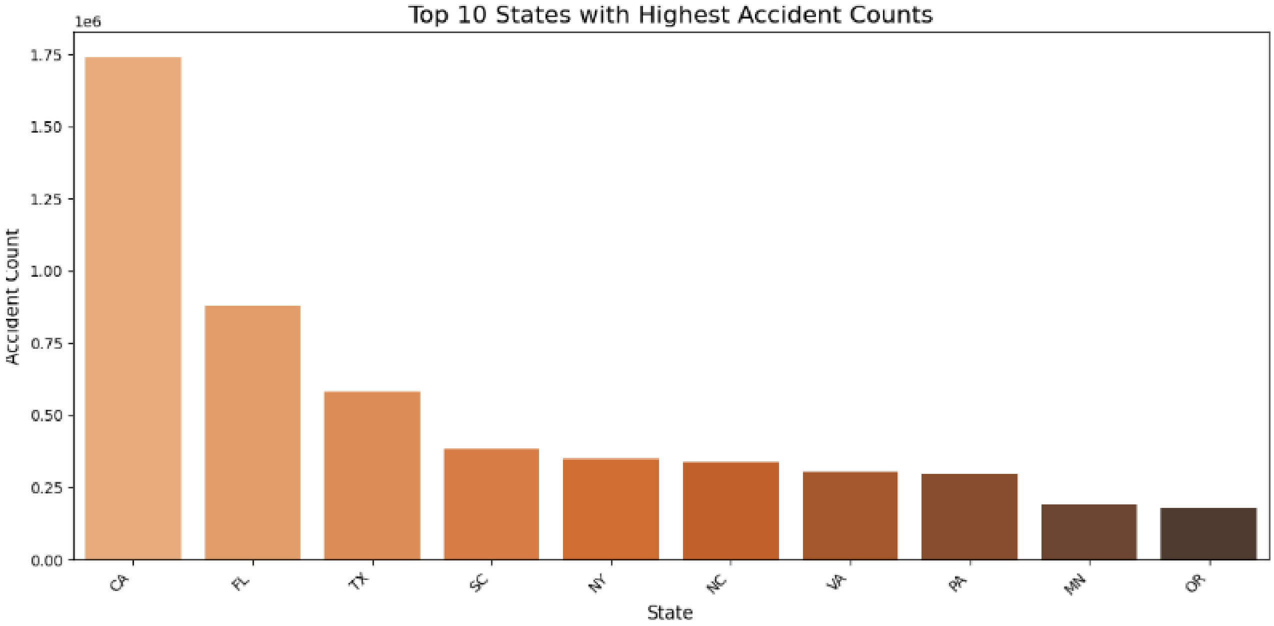
9. *Top Cities contributing with Accidents.*

- Miami, Houston & Los Angeles are the top cities contributing to accidents



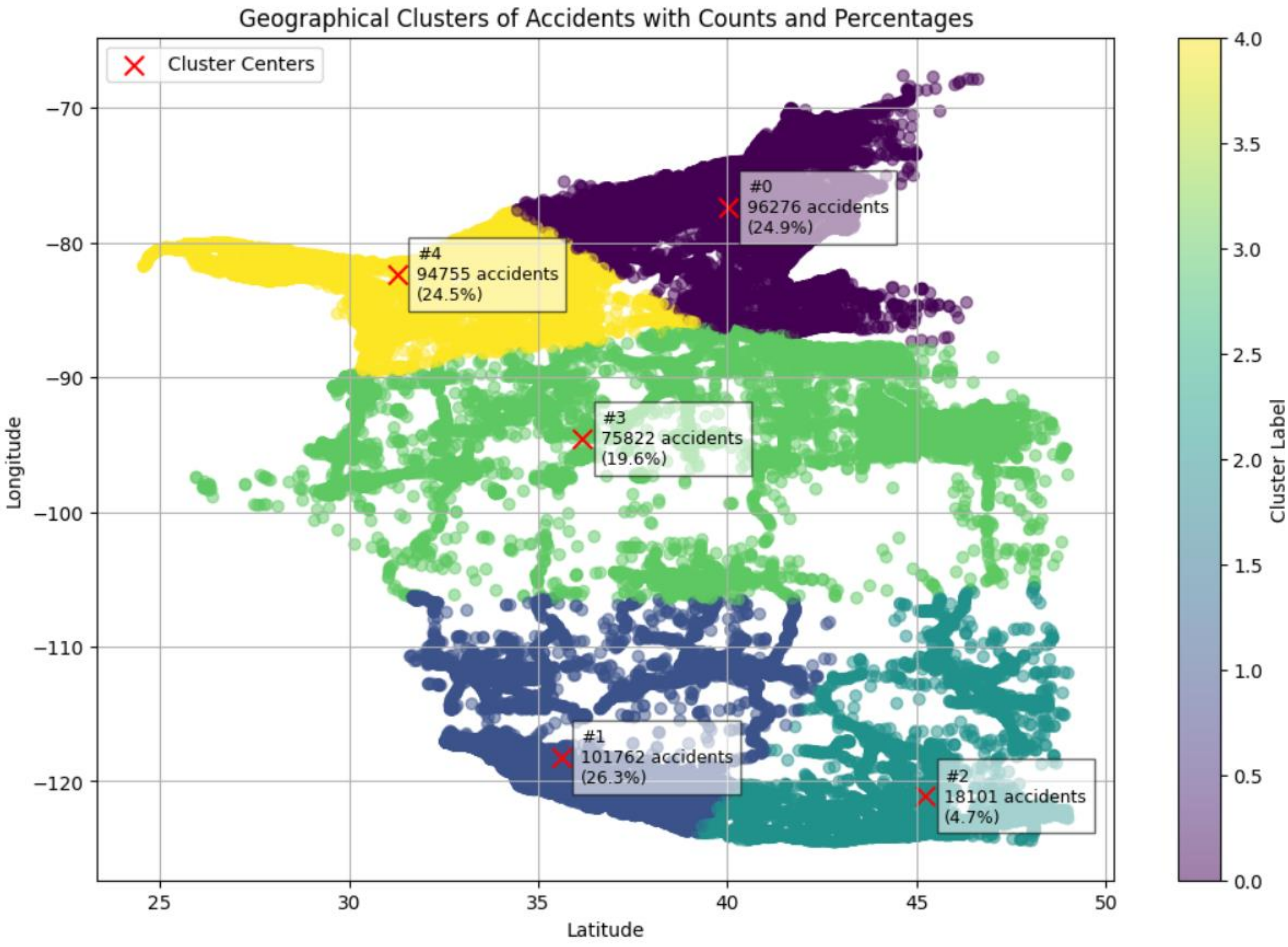
10. *Top States contributing with Accidents.*

- California, Florida & Texas are leading the Accidents contribution



11. *Clustering using K-Means based on Location.*

- Accidents are geographically concentrated into 5 major clusters across the US.
- Each cluster corresponds to a distinct geographic zone, showing different accident patterns based on location.
- K used here is 5

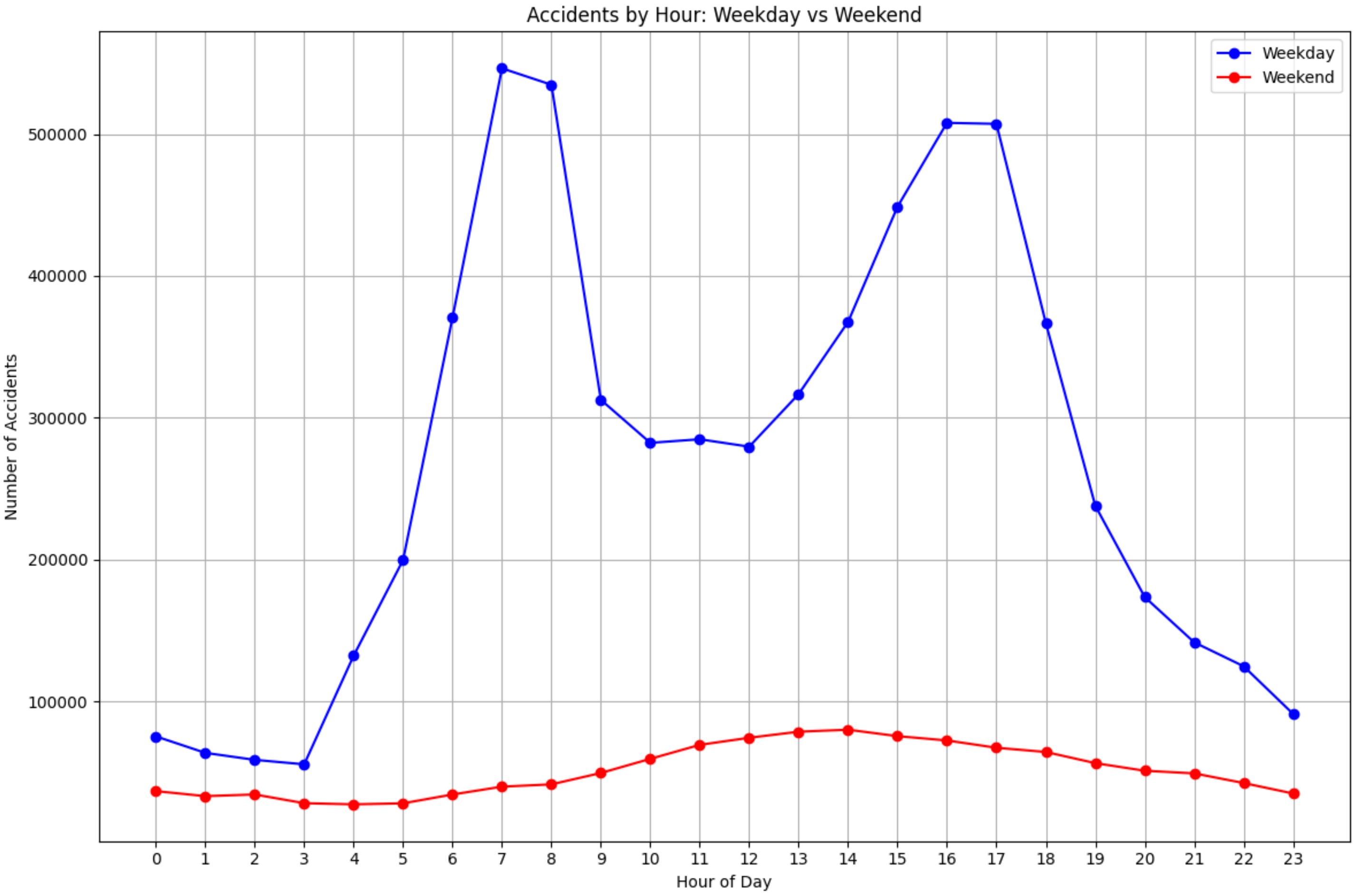


12. MapReduce K-means.

- 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

13. Weekday vs Weekend Accidents

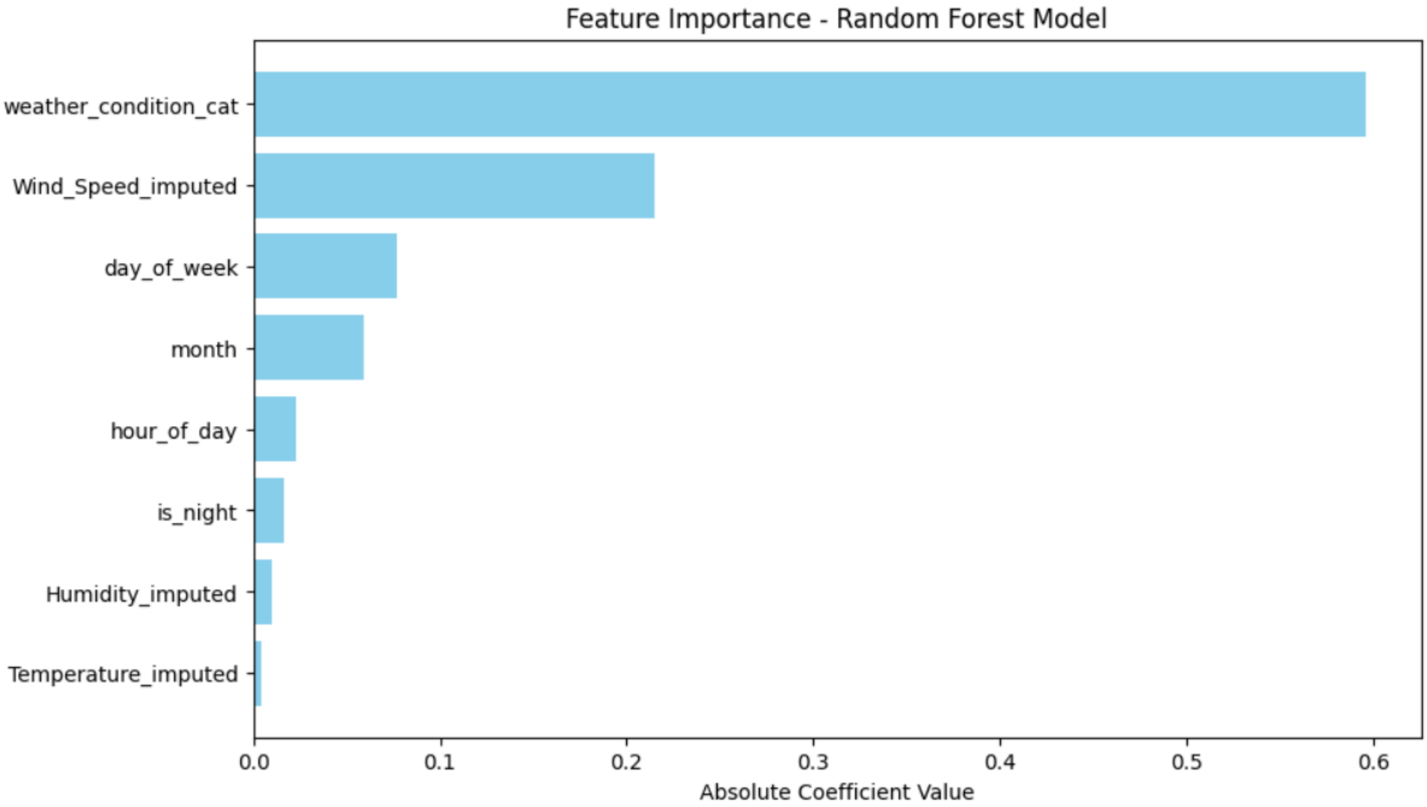
- It can be shown from the following graph that accidents happen on weekends the most from 11 AM to 4 PM, while in weekdays it happens mostly in the rush hours (6-9 AM and 3 to 7 PM.



Predictive Analysis (Insights):

14. 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.
- Precision: 0.80 Recall: 0.81 F1-Score: 0.72
- 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.



- 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.

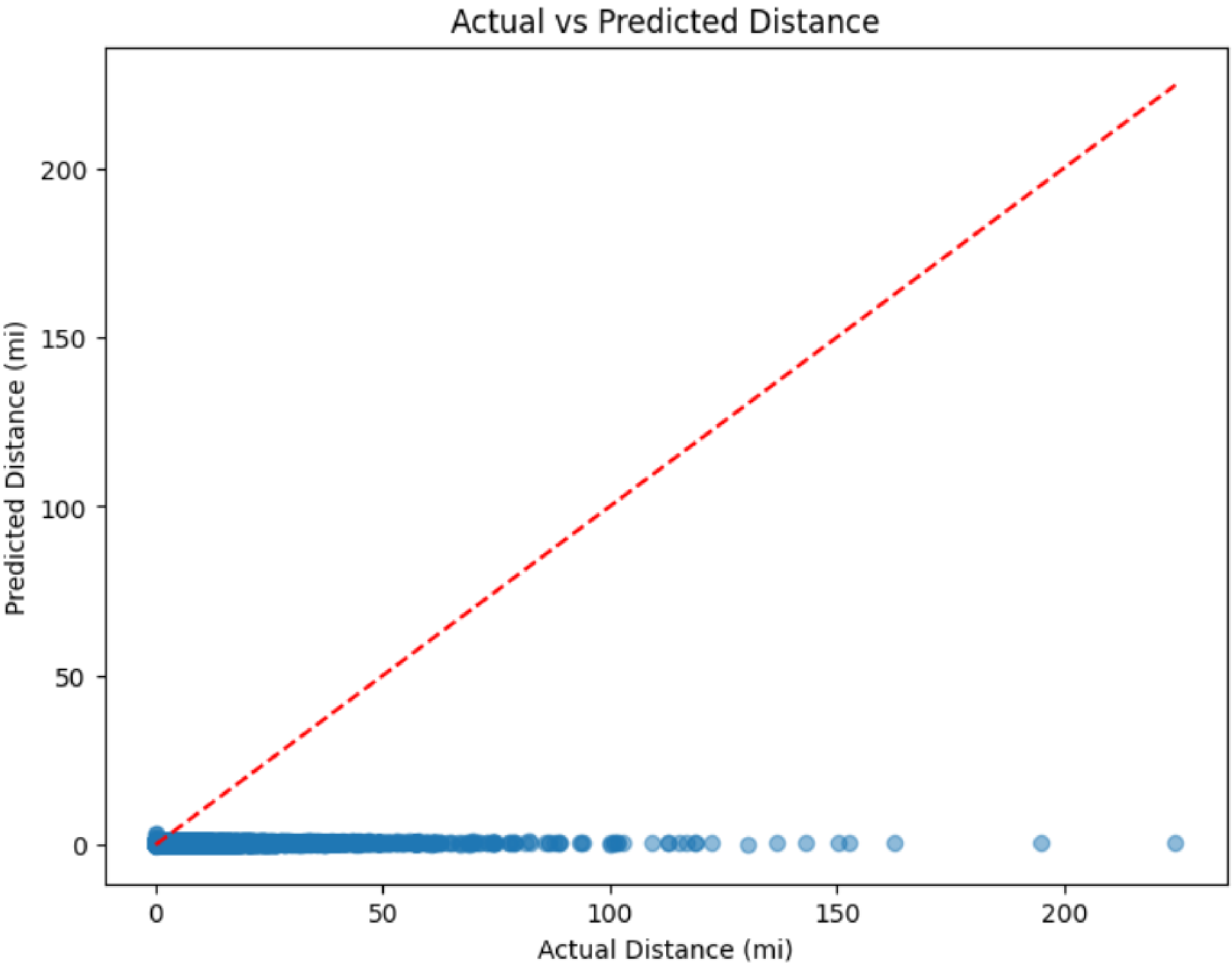
15. Binary Classification for Severity

- 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

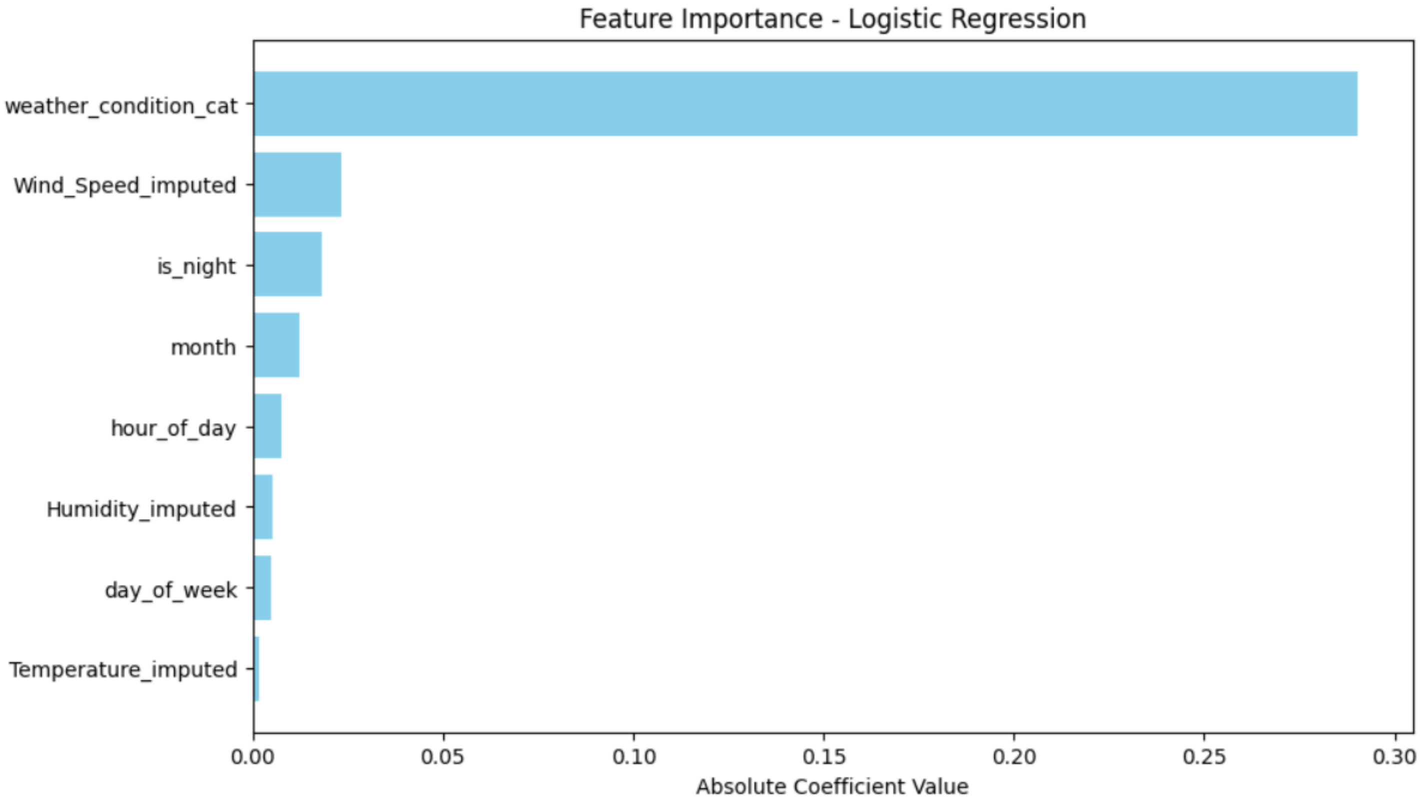
15. Predicting Accident Distance using Linear Regression.

- 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.

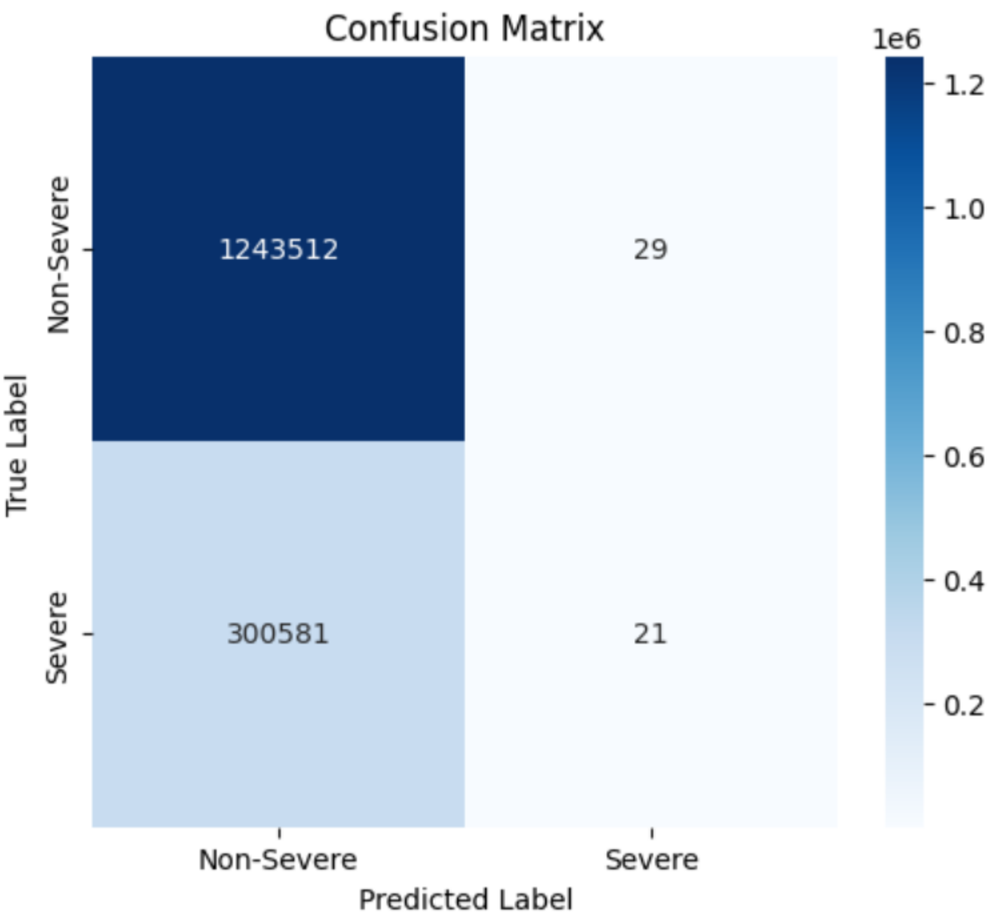


16. Binary Severity Prediction using Logistic Regression.

- 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.
- Undersampling reduced the accuracy into 55%
- 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.
- 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.



- The confusion matrix is shown below, where it can show unbalance in the severity



17. Predicting States' Accident Risk Scores.

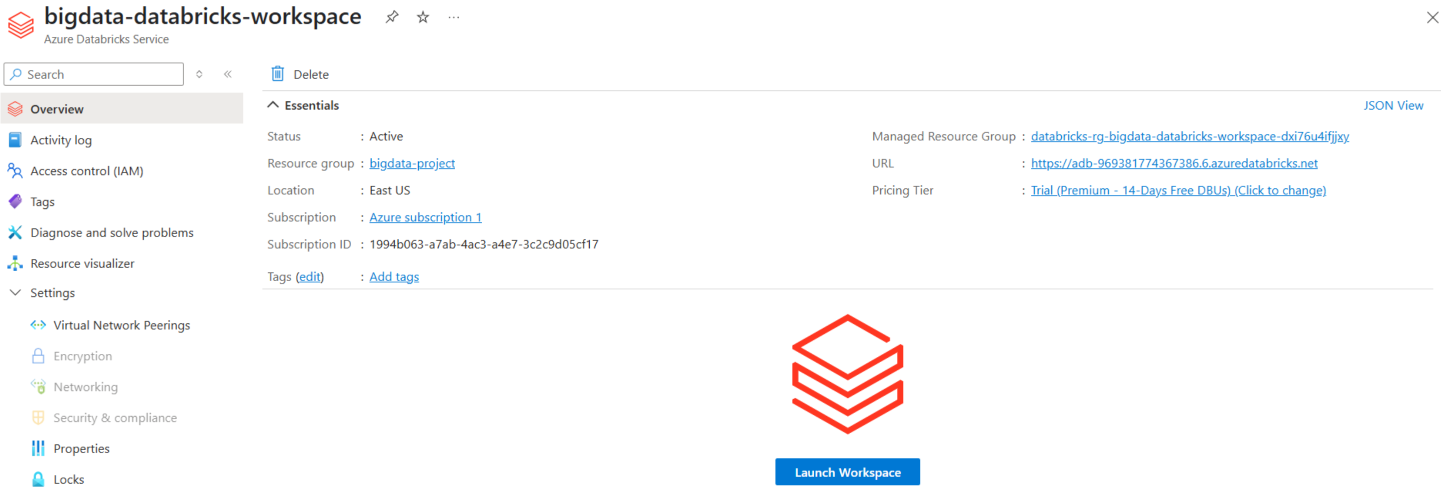
- Linear Regression was used to predict Severity for each state
- **Model Performance:** The regression model demonstrates a moderate level of accuracy with an RMSE of 317,208.92, MAE of 200,413.47, and an R2 score of 0.5838. These metrics suggest room for improvement in predictive accuracy.
- **Key Drivers: Avg_Precipitation** is the dominant factor influencing predictions (importance score: 0.9957), while other features like Avg_Visibility (0.0034) and Avg_Temperature (0.0007) have minimal impact. This insight can guide strategic decisions on data collection and feature engineering.

18. Risk Classification for States

- Linear Regression was used to predict Severity for each state
- Model Performance:
  - AUC-ROC: 0.7778
  - Accuracy: 81.82%
  - Weighted Precision: 81.82%
  - Weighted Recall: 81.82%
  - F1 Score: 81.82%
- Feature Importances:
  - Avg_Precipitation: -505.8624 (most influential, higher precipitation increases risk)
  - Avg_Accident_Distance: -1.0414 (longer distances slightly increase risk)
  - Avg_Visibility: -0.7750 (lower visibility increases risk)
  - Avg_Temperature: 0.2671 (higher temperatures decrease risk)
  - Num_Unique_Cities: 0.0154 (minimal impact)

Cloud Computing Used:

- Used Azure Databricks to use the clusters with 2 setups, one with single Machine and other cluster with 1 driver and 2 worker nodes



bigdata-databricks-workspace

Azure Databricks Service

Essentials

Status : Active

Resource group : bigdata-project

Location : East US

Subscription : Azure subscription 1

Subscription ID : 1994b063-a7ab-4ac3-a4e7-3c2c9d05cf17

Tags (edit) : Add tags

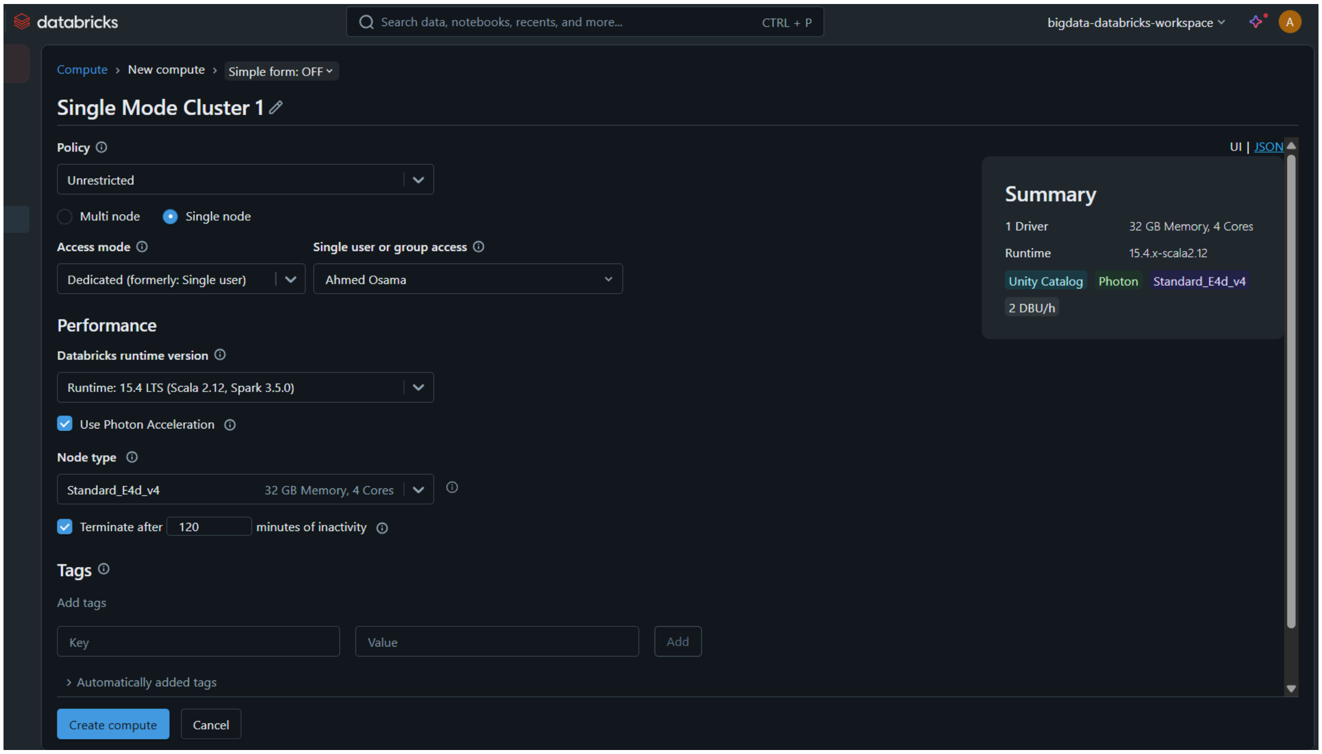
Managed Resource Group : databricks-rg-bigdata-databricks-workspace-dxi76u4ifjjxy

URL : https://adb-969381774367386.6.azuredatabricks.net

Pricing Tier : Trial (Premium - 14-Days Free DBUs) (Click to change)

- 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.

- 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.


databricks
Search data, notebooks, recents, and more... CTRL + P
bigdata-databricks-workspace
Compute › New compute › Simple form: OFF
UI | JSON
Create compute
Cancel

Single Mode Cluster 1

Policy: Unrestricted

○ Multi node ● Single node

Access mode: Dedicated (formerly: Single user)

Single user or group access: Ahmed Osama

Performance

Databricks runtime version: Runtime: 15.4 LTS (Scala 2.12, Spark 3.5.0)

☑ Use Photon Acceleration

Node type: Standard_E4d_v4 32 GB Memory, 4 Cores

☑ Terminate after 120 minutes of inactivity

Tags

Add tags

Key | Value | Add

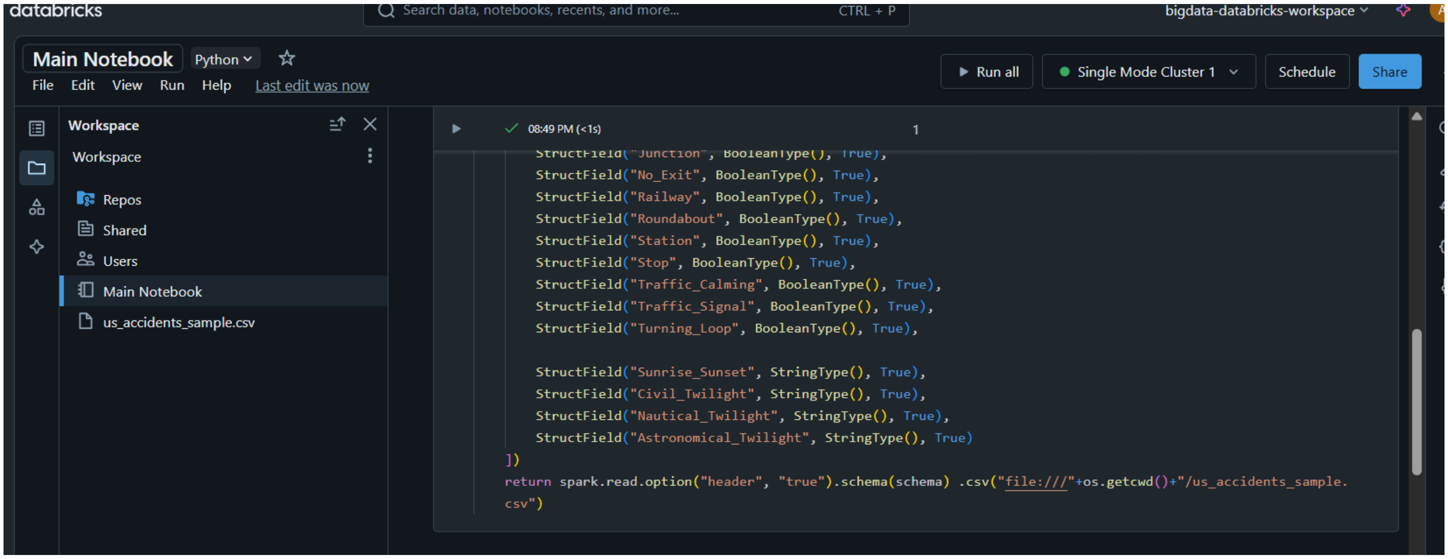
› Automatically added tags

Summary

1 Driver	32 GB Memory, 4 Cores
Runtime	15.4.x-scala2.12

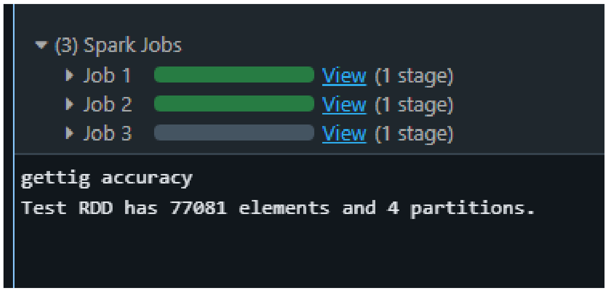
Unity Catalog Photon Standard_E4d_v4

2 DBU/h



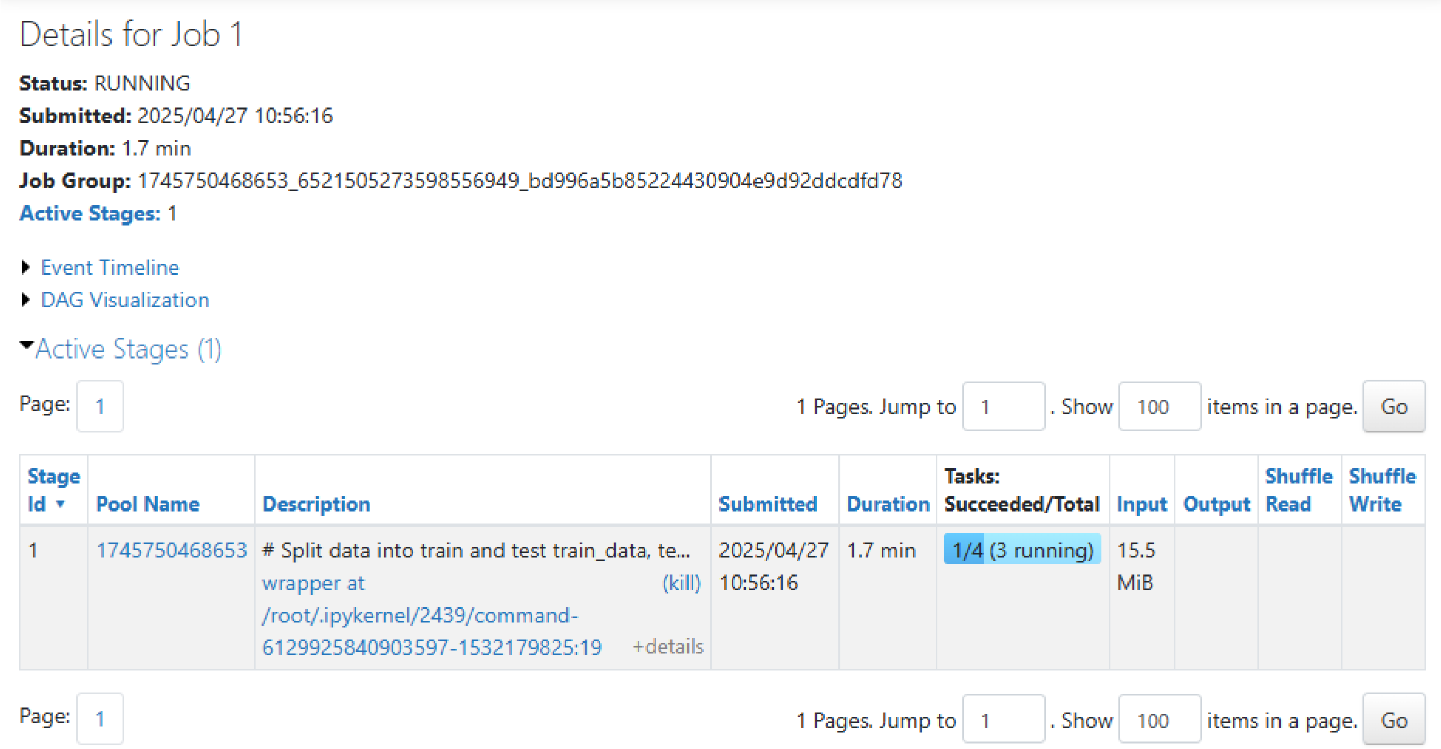
```
        StructField("Junction", BooleanType(), True),
        StructField("No_Exit", BooleanType(), True),
        StructField("Railway", BooleanType(), True),
        StructField("Roundabout", BooleanType(), True),
        StructField("Station", BooleanType(), True),
        StructField("Stop", BooleanType(), True),
        StructField("Traffic_Calming", BooleanType(), True),
        StructField("Traffic_Signal", BooleanType(), True),
        StructField("Turning_Loop", BooleanType(), True),

        StructField("Sunrise_Sunset", StringType(), True),
        StructField("Civil_Twilight", StringType(), True),
        StructField("Nautical_Twilight", StringType(), True),
        StructField("Astronomical_Twilight", StringType(), True)
    ])
    return spark.read.option("header", "true").schema(schema) .csv("file:///"+os.getcwd()+"/us_accidents_sample.
csv")
```

- During the execution, the Spark jobs were monitored to check their performance and their success



▾ (3) Spark Jobs
- ▸ Job 1 View (1 stage)
- ▸ Job 2 View (1 stage)
- ▸ Job 3 View (1 stage)

```
gettig accuracy
Test RDD has 77081 elements and 4 partitions.
```



Details for Job 1

Status: RUNNING
Submitted: 2025/04/27 10:56:16
Duration: 1.7 min
Job Group: 1745750468653_6521505273598556949_bd996a5b85224430904e9d92ddcdfd78
Active Stages: 1

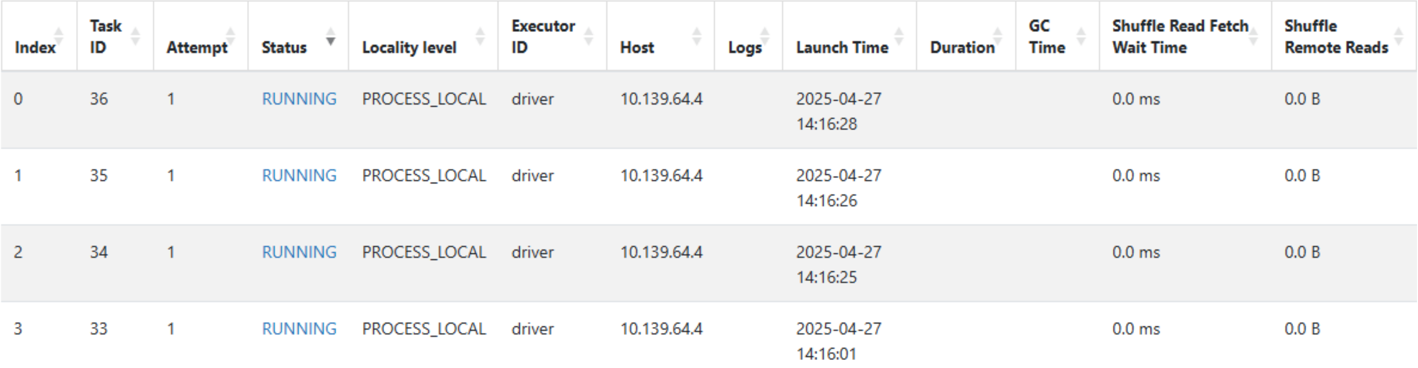
- ▸ Event Timeline
- ▸ DAG Visualization

▾Active Stages (1)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

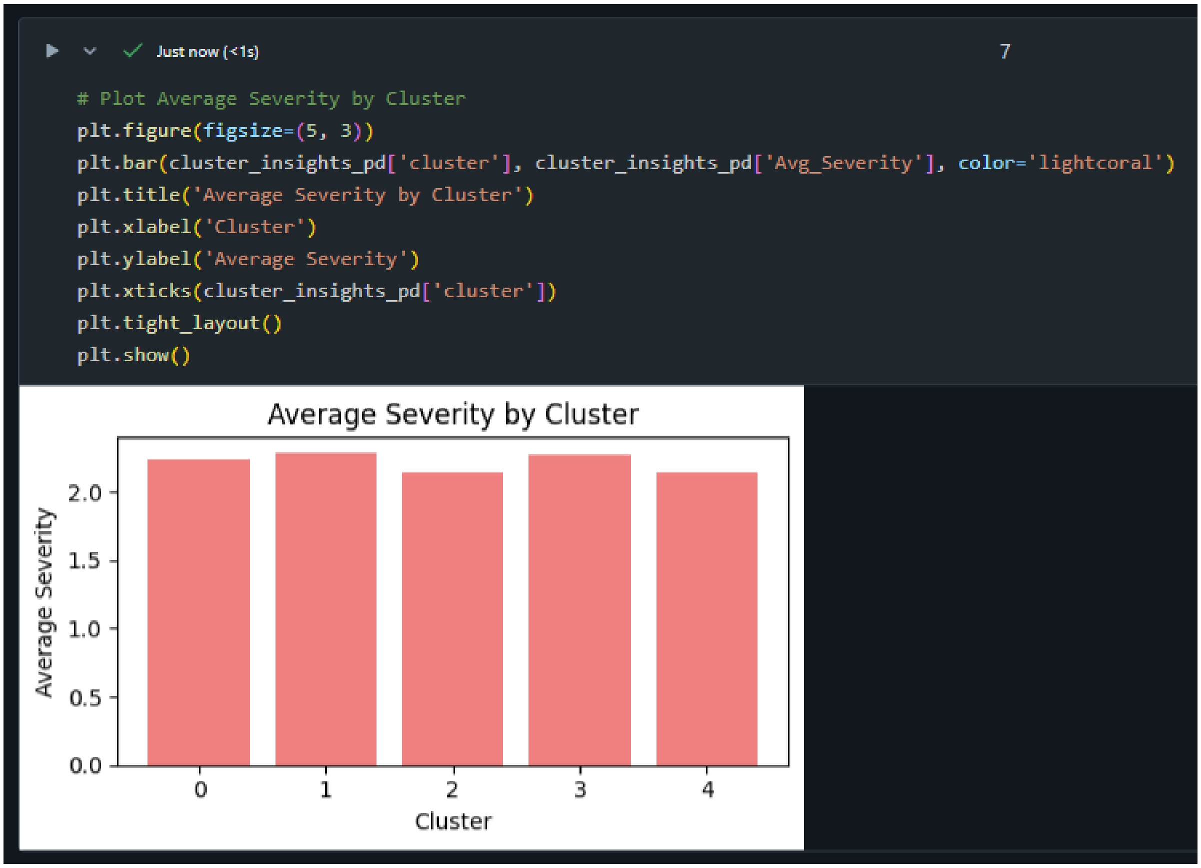
Stage Id ▾	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
1	1745750468653	# Split data into train and test train_data, te... wrapper at /root/.ipykernel/2439/command-6129925840903597-1532179825:19 (kill) +details	2025/04/27 10:56:16	1.7 min	1/4 (3 running)	15.5 MiB			

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

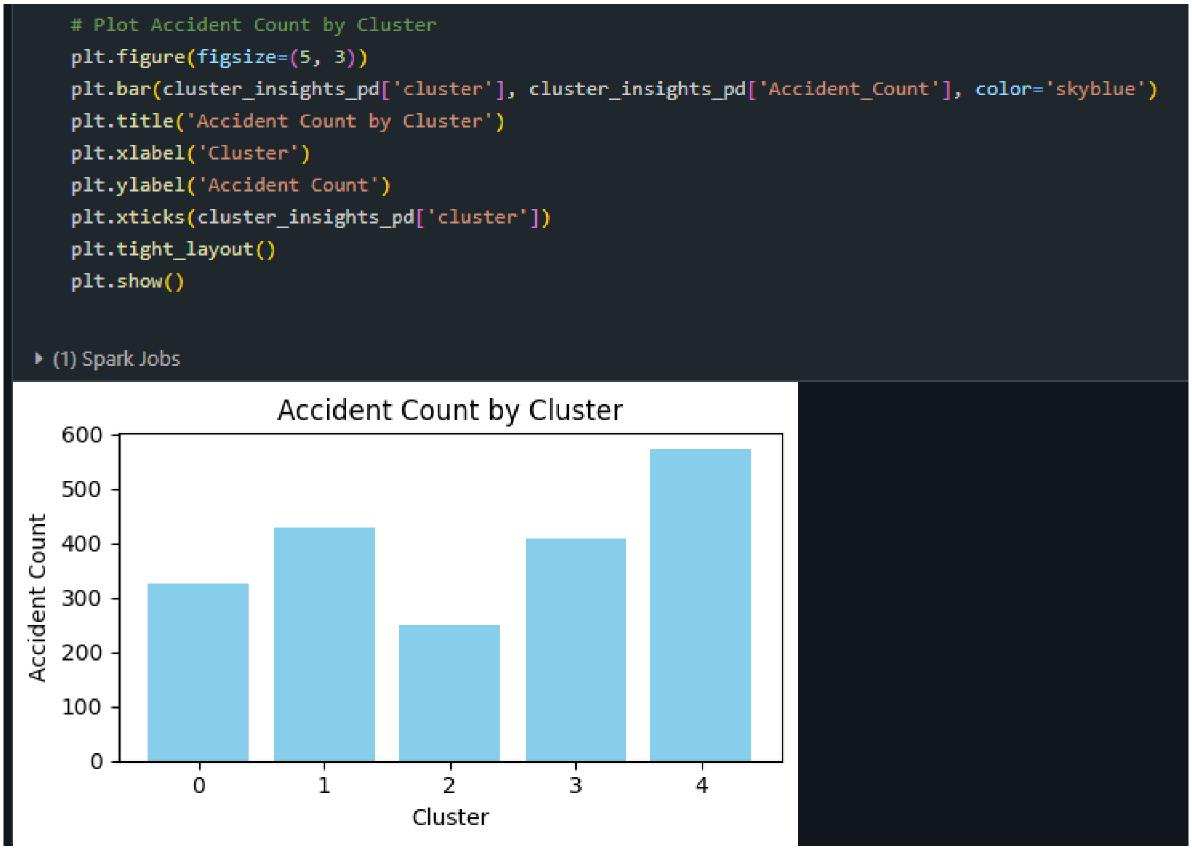


Index	Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Shuffle Read Fetch Wait Time	Shuffle Remote Reads
0	36	1	RUNNING	PROCESS_LOCAL	driver	10.139.64.4		2025-04-27 14:16:28			0.0 ms	0.0 B
1	35	1	RUNNING	PROCESS_LOCAL	driver	10.139.64.4		2025-04-27 14:16:26			0.0 ms	0.0 B
2	34	1	RUNNING	PROCESS_LOCAL	driver	10.139.64.4		2025-04-27 14:16:25			0.0 ms	0.0 B
3	33	1	RUNNING	PROCESS_LOCAL	driver	10.139.64.4		2025-04-27 14:16:01			0.0 ms	0.0 B

- Several Algorithms were trained on the cluster utilizing the resources given. Below, the results of K-means are shown as an example.



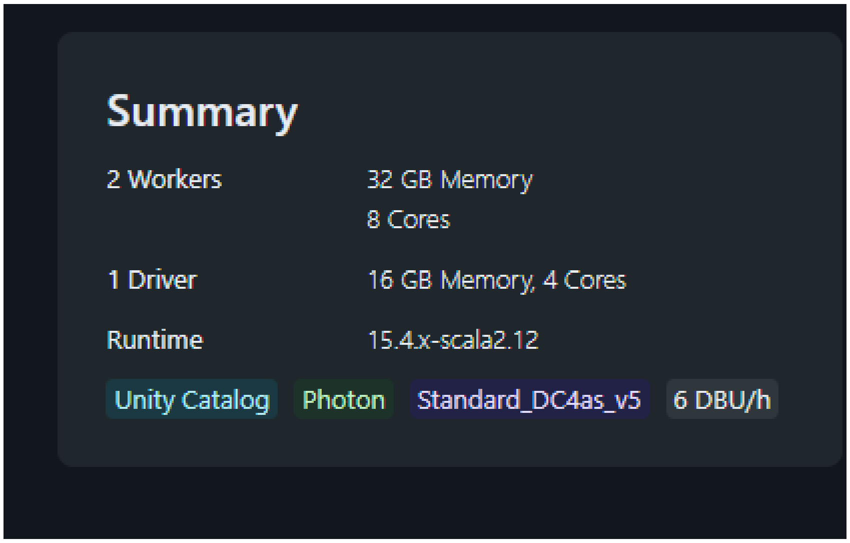
```
# Plot Average Severity by Cluster
plt.figure(figsize=(5, 3))
plt.bar(cluster_insights_pd['cluster'], cluster_insights_pd['Avg_Severity'], color='lightcoral')
plt.title('Average Severity by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Average Severity')
plt.xticks(cluster_insights_pd['cluster'])
plt.tight_layout()
plt.show()
```



```
# Plot Accident Count by Cluster
plt.figure(figsize=(5, 3))
plt.bar(cluster_insights_pd['cluster'], cluster_insights_pd['Accident_Count'], color='skyblue')
plt.title('Accident Count by Cluster')
plt.xlabel('Cluster')
plt.ylabel('Accident Count')
plt.xticks(cluster_insights_pd['cluster'])
plt.tight_layout()
plt.show()
```

Fully Clustered Mode:

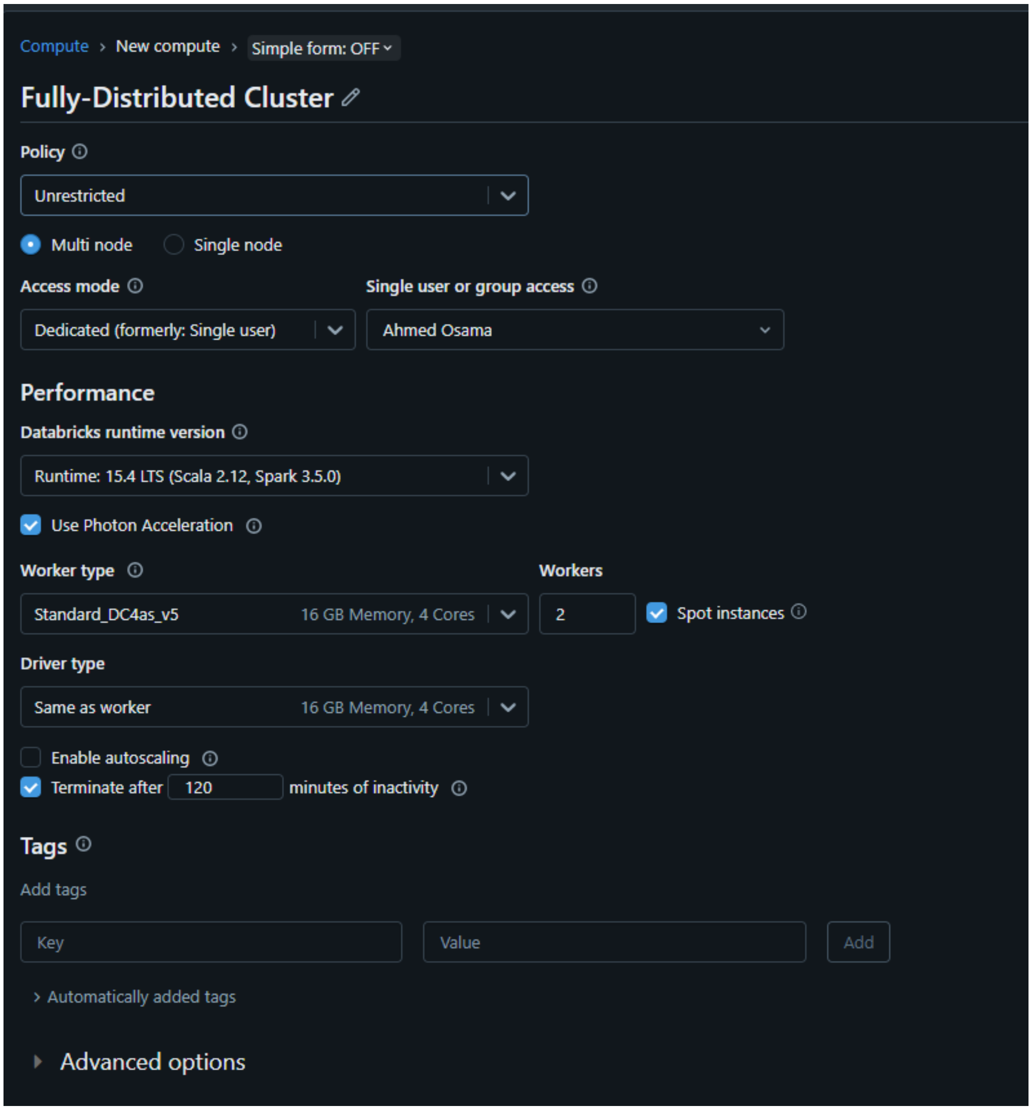
- Using 2 worker nodes on Azure Databricks, Fully Distributed mode was implemented dividing jobs among the workers.



Summary

2 Workers	32 GB Memory 8 Cores
1 Driver	16 GB Memory, 4 Cores
Runtime	15.4.x-scala2.12

Unity Catalog Photon Standard_DC4as_v5 6 DBU/h



Fully-Distributed Cluster

Policy: Unrestricted

● Multi node ○ Single node

Access mode: Dedicated (formerly: Single user)

Single user or group access: Ahmed Osama

Performance

Databricks runtime version: Runtime: 15.4 LTS (Scala 2.12, Spark 3.5.0)

☑ Use Photon Acceleration

Worker type: Standard_DC4as_v5 16 GB Memory, 4 Cores

Workers: 2 ☑ Spot instances

Driver type: Same as worker 16 GB Memory, 4 Cores

□ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Tags

Unsuccessful Trials

- **Apriori:** Apriori was used to generate association rules to predict common patterns among the accidents. However, the rules generated didn't have much business value.
- **Principal Component Analysis (PCA):** Applied PCA to reduce dimensionality and identify key features. The reduced features did not significantly improve model performance or provide clearer insights.
- **Support Vector Machines (SVM):** Tried SVM for classification tasks. The model struggled with the high dimensionality of the data and did not achieve satisfactory accuracy.
- **Train/Test Splitting:** Several splits on the data were tried but did not achieve better results

Future Work & Recommendations

- Include other metrics for each state from other data sources like Population of the state and their characteristics and integrate to have more meaningful insights
- Explore advanced machine learning and deep learning techniques to improve predictive accuracy and uncover complex relationships between features. Techniques like ensemble methods, neural networks, and time-series analysis can be particularly useful.
- Integrate data on driver behavior, such as speeding, distracted driving, and compliance with traffic signals, to understand their influence on accident risk. This can inform educational campaigns and policy changes.