BEHIND THE WHEEL

CENG790 Big Data Analytics

2023-2024 Spring

Term Project Final Report

Berk GÜLER - 2310092

Mustafa Barış EMEKTAR - 2304533

Middle East Technical University

Department of Computer Engineering

TABLE OF CONTENTS

INTRODUCTION

According to information from the National Highway Traffic Safety Administration (NHTSA), there are 30,000 to 40,000 deaths in traffic accidents in the United States in a year. The analysis of traffic accident data in the United States presents an opportunity to gain valuable insights into the factors contributing to road safety and accident occurrence. In this project, we delve into a comprehensive dataset encompassing various variables related to traffic accidents across the nation.

We now have the tools to identify important patterns and trends from this enormous information thanks to the development of big data analytics. Through the utilization of sophisticated analytical methods, our objective is to detect underlying trends, identify risk variables, and create models for prediction that will help in improving traffic management systems and devising focused safety measures.

We utilized methodologies such as exploratory data analysis and machine learning algorithms. Our aim is to simplify the understanding of road safety, facilitating informed decision-making and ultimately contributing to a safer transportation environment for all.

As we navigate through the dataset, our objective is to highlight the potential of big data analytics in reshaping the future of road safety initiatives. By unraveling hidden insights within the data, we endeavor to pave the way towards a transportation ecosystem that is safer and more resilient.

METHODOLOGY

The problem being addressed in this project is to analyze and understand accident patterns and severity using the US Accidents dataset. Specifically, the exploration of relationships between various features (such as weather conditions, time of day, and location) and their impact on accident outcomes is aimed at. By doing so, we can identify actionable insights for improving road safety and minimizing the consequences of accidents.

DATASET

For this project, we used the US Accidents dataset from Kaggle. It covers accidents that occurred in the United States from February 2016 to March 2023. Here is why this dataset is appropriate to use:

- **Time Scope and Size:** The dataset covers 7 years of accidents that occurred in the 49 states of the USA. It consists of 7.7 million accident records with 3.06GB of data.
- **Rich Features:** The dataset provides a rich source of data for our analysis. It contains detailed information about each accident, including features such as location, severity, weather conditions, time of day, and road characteristics. It contains 46 columns (different features). These (possibly)

relevant features can help us understand accident patterns and severity.

- **Severity Labels:** The dataset includes labels for accident severity (e.g., fatal, minor), which is crucial for our analysis. Without these labels, we would treat all accidents equally, missing critical insights into the impact of accidents. Here are some of the features (columns) of dataset:

Below table shows some of the most important features (columns) of US Accidents Dataset:

Column	Explanation	Column	Explanation
ID	*This is a unique identifier of the accident record.*	text_formatWeather_Condition	*Shows the weather condition (rain, snow, thunderstorm, fog, etc.)*
text_formatSource	*Source of raw accident data*	grid_3x3Temperature(F)	*Shows the temperature (in Fahrenheit).*
grid_3x3Severity	*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).*	grid_3x3Wind_Chill(F)	*Shows the wind chill (in Fahrenheit).*
calendar_todayStart_Time	*Shows start time of the accident in local time zone.*	grid_3x3Humidity(%)	*Shows the humidity (in percentage).*
calendar_todayEnd_Time	*Shows end time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow was dismissed.*	grid_3x3Pressure(in)	*Shows the air pressure (in inches).*
text_formatStreet	*Shows the street name in address field.*	checkTraffic_Signal	*A POI annotation which indicates presence of traffic_signal in a nearby location.*
text_formatCity	*Shows the city in address field.*	checkBump	*A POI annotation which indicates presence of speed bump or hump in a nearby location.*
text_formatCounty	*Shows the county in address field.*	checkCrossing	*A POI annotation which indicates presence of crossing in a nearby location.*
text_formatState	*Shows the state in address field.*	checkGive_Way	*A POI annotation which indicates presence of give_way in a nearby location.*
text_formatZipcode	*Shows the zipcode in address field.*	checkJunction	*A POI annotation which indicates presence of junction in a nearby location.*

Table 1: Some of the features (columns) of US Accidents Dataset

TECHNICAL APPROACH

We implemented our project using **Python** on **Intellij IDEA**. We used **Apache Spark** for the development of the project and this has been a great benefit. Because it would have taken much more time to process and graph a large data of 3 GB without spark. Spark shortened this time considerably for us.

We also used **Pandas** for our project because it is an easy-to-use and efficient data analysis tool. It is successful in managing large datasets. However, the main purpose of using Pandas is to visualize the data more easily.

EXPLORATIONS & INSIGHTS

In this section of the report, we would like to share some of our discoveries of the dataset and give basic insights.

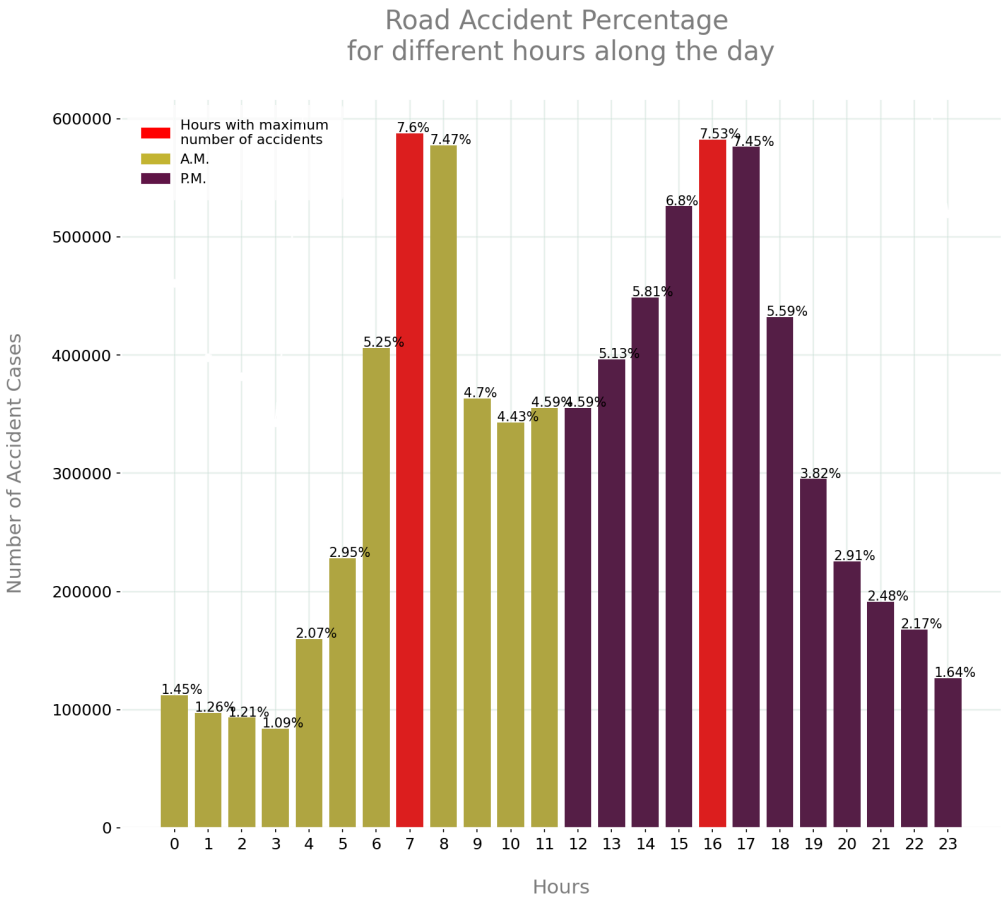

Figure 1: Visualization of Traffic Accidents According to the Hours of the Day

According to Figure 1:

- ➔ 15.07% of accidents happen between 7AM and 9AM.
- ➔ 14.98% of accidents happen between 4PM and 6PM

On average **30% of accidents happen in rush hours** which are the deadliest hours.

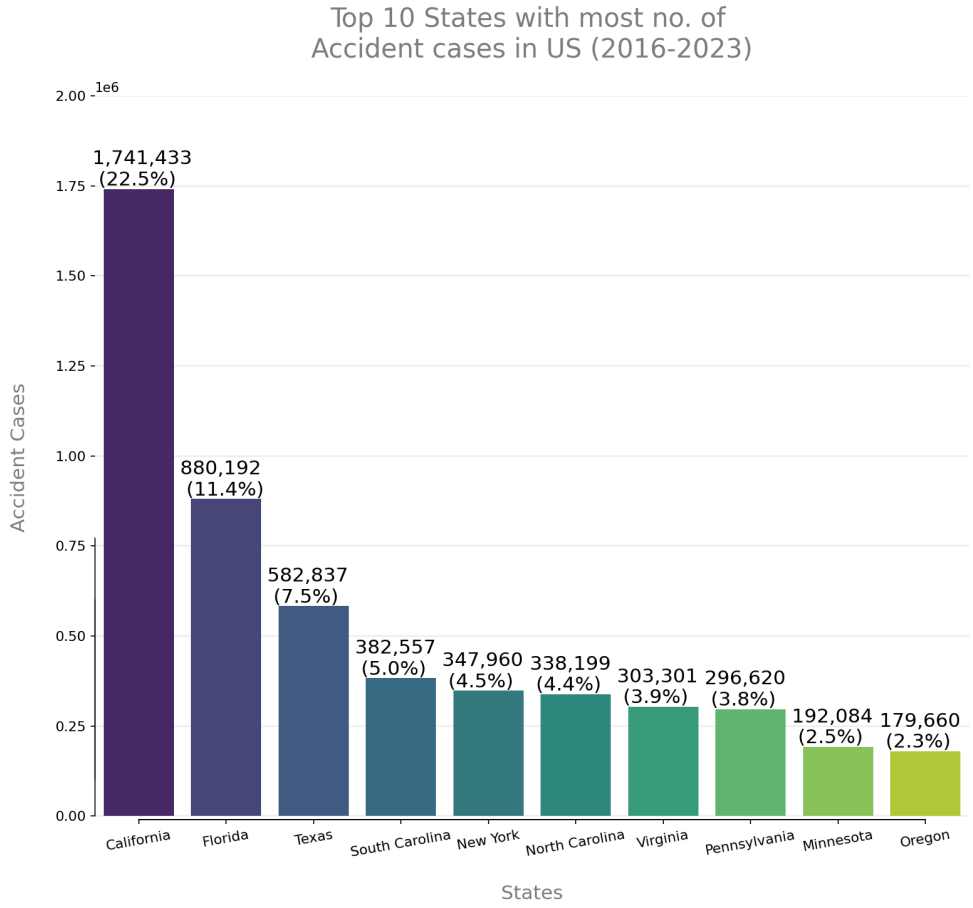

Figure 2: Visualization of the States with the Highest Number of Traffic Accidents

According to Figure 2:

- ➔ **California leads by far in the number of accidents** with 22.5 among the 49 states.
- ➔ Over the last eight years, California has experienced an average of 596 accidents every day, or around 29 accidents every hour.

The number of accidents by states are useful to gain insight about states. However, it would be much more useful if we could normalize it with the state populations or number of cars that exist in states. Our dataset doesn't provide that information. For future studies, we can use this dataset together with another dataset containing more detailed information on states and cities to get deeper insights. For example, with the number of car drivers, we could find out proportionally in which state drivers are more careful.

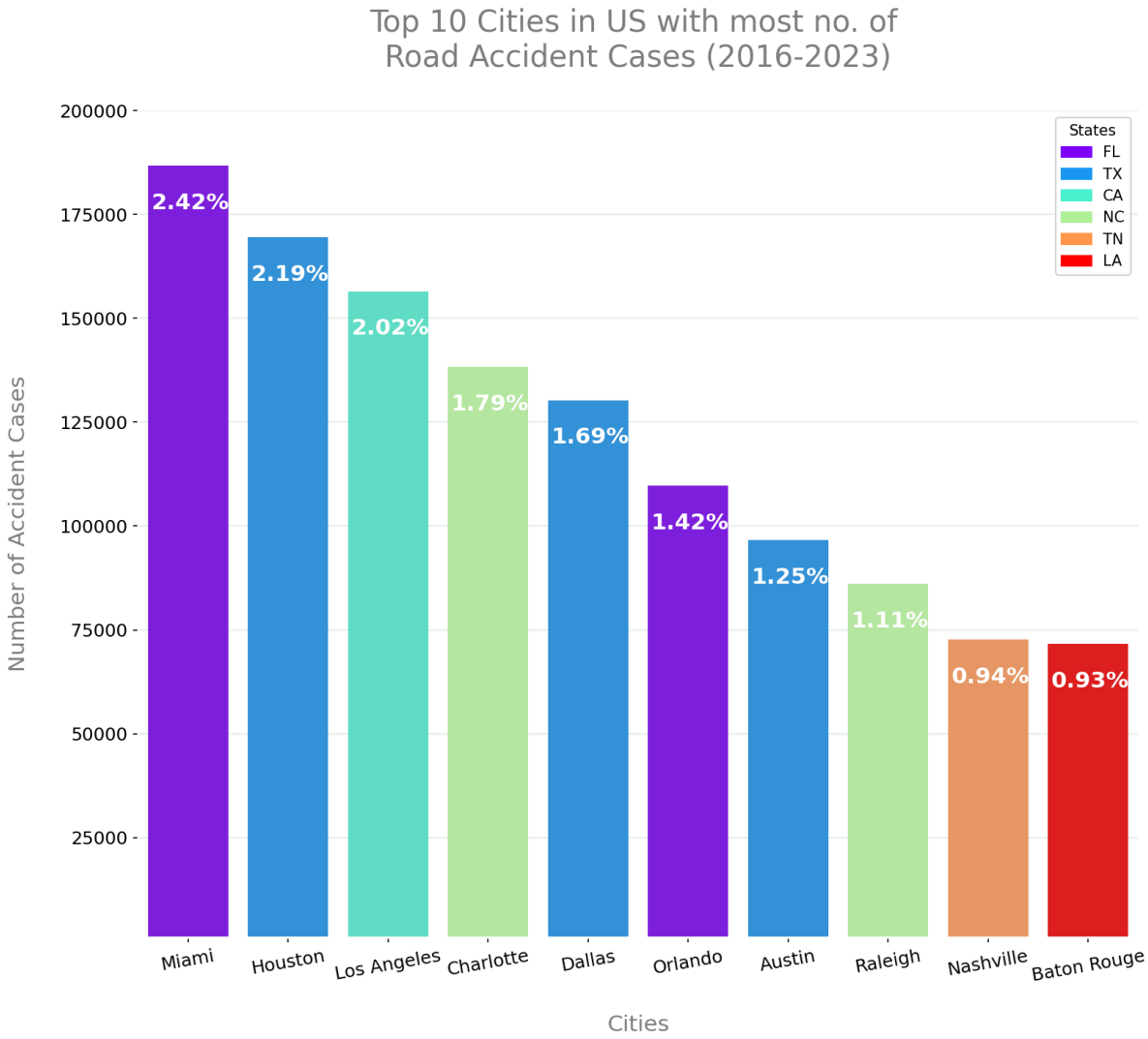

Figure 3: Visualization of the Cities with the Highest Number of Traffic Accidents (according to the colors of the states where they are located)

According to Figure 3:

- ➔ **Miami has the highest (2.42%) number of road accidents** in the United States.
- ➔ In the past 8 years, around 15% of the accident records are only from 10 cities out of over 13,678 cities in the United States.
- ➔ Texas has three cities in the top 10 cities with the most number of road accidents. Florida and New York follow it with two cities.

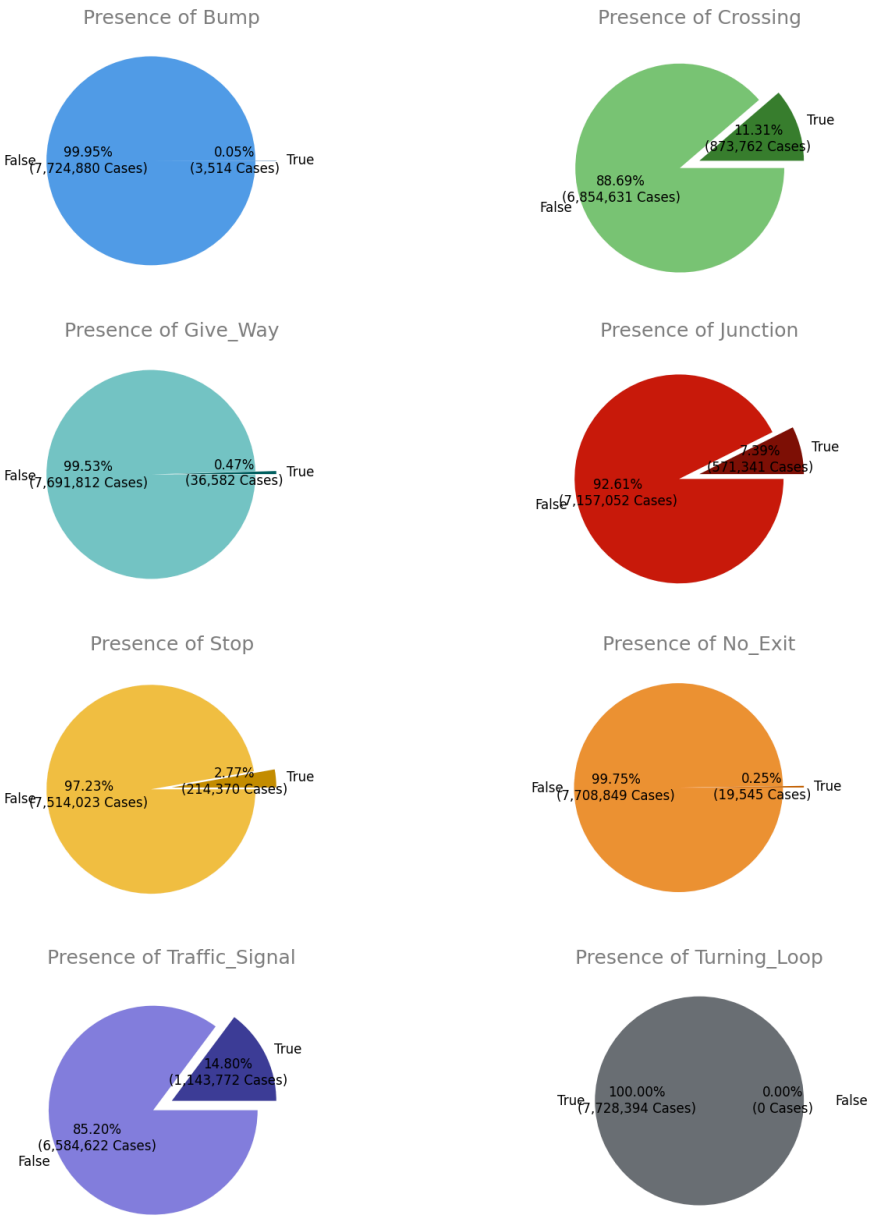

Figure 4: Visualization of Road Conditions in the Area of Traffic Accidents

According to Figure 4:

- ➔ There were no stop signs near the accident area, in 98.83% cases.
- ➔ 11.31% of the road accidents occurred near the junctions;
  - ◆ However only 2.77% of the accidents occurred near stop signs. That means the junctions lack stop signs. **More stop signs should be placed on the roads.**
  - ◆ Similarly, in only 0.47% cases there were give way signs and 0.05% cases there were give way signs. Thus, **more give way signs OR bumps should be placed on the roads.**

The features under road conditions are fun to analyze; however, some of them seem meaningless by themselves. If there were more information about the accident they would be more meaningful.

For example, the cases categorized with traffic signals don't tell us too much because we don't know whether the accident happened in urban or out of town. So we cannot courageously say that more traffic lights should be installed. Also as another example; we can say that the bumps are very useful as there are very few accidents where there are bumps. Or, on the contrary, this may have happened because there are no bumps where they are needed.

Severity	Delay
1	2.5 minutes and 30 seconds
2	3 minutes and 15 seconds
3	8 minutes
4	18 minutes

Table 2: Impact of the Different Severity Levels on Traffic

Before further explorations, knowing the details of the severity levels would be useful. The details of the severity labels are not given explicitly in the dataset. However it is stated that: "It 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)." Thus, we know that it gets worse as the number of severity increases. It is correlated with the impact on the traffic.

In one of the discussions, we found an analysis of the relationship between the delay caused by the accident and the severity. The relationship can be seen from Table-2.

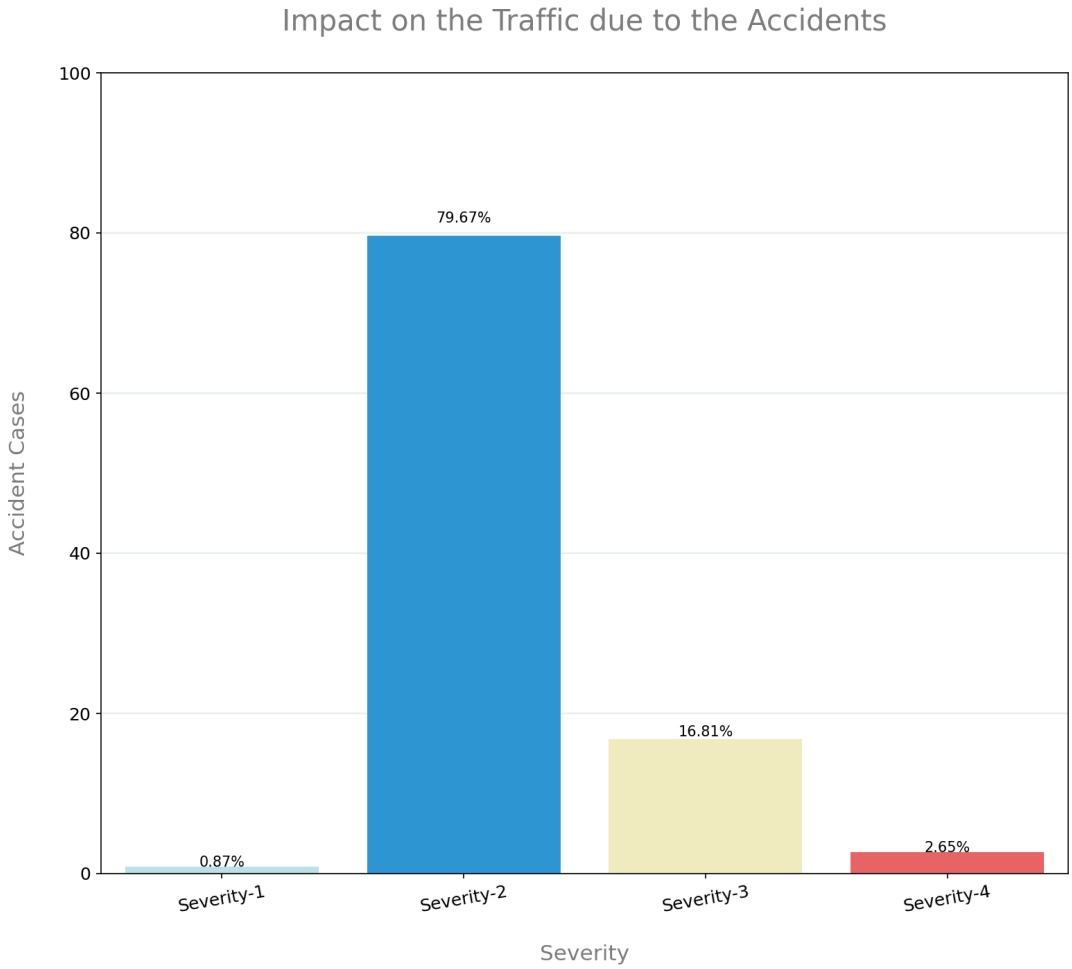

Figure 5: Visualization of Severity Levels by Accident Cases

According to Figure 5:

- ➔ 79.67% of the accidents are categorized as Severity-2.
- ➔ 2.65% of the accidents are categorized as Severity-4 which is deadly..

Since Severity depends on the normalization time of the traffic, it is questionable how reliable it is. Although it is the most important parameter of this dataset, critical information such as the number of deaths or the speed of the vehicles is not included in the dataset. Therefore, the points we were able to explore were limited.

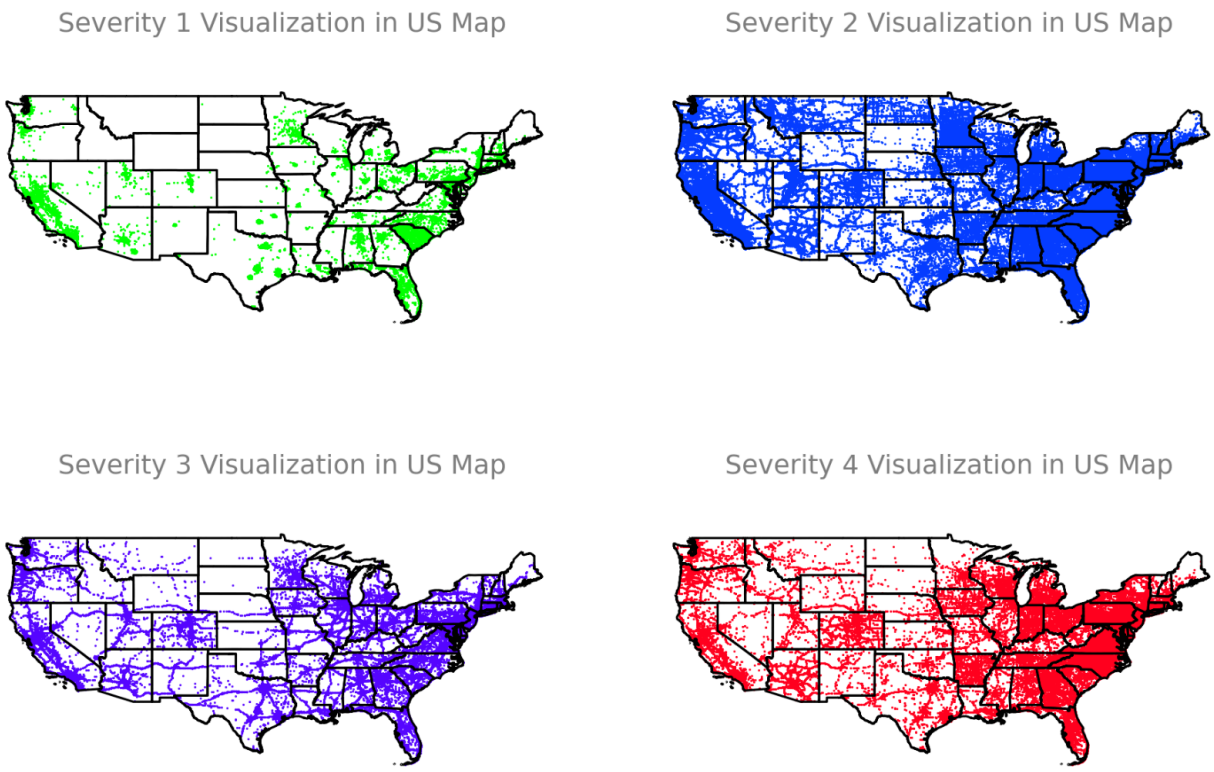

Figure 6: Severity Visualizations on the USA Map With Four Different Level

As seen in Table 1, the dataset includes latitude and longitude values indicating the location of the accident. In this case, we thought it would be nice to observe this data on a USA map. The figure above shows the locations of the accidents divided by severity and plotted on the USA map.

To create this graph, another dataset called **us-map-state** was used. It helped to draw the USA map using the **Geopandas** framework.

As seen in these graphs, the number of accidents is higher in places with high population density and urbanization. Especially in the northern parts of the country, we can observe that Severity-2 accidents are much more common than others.

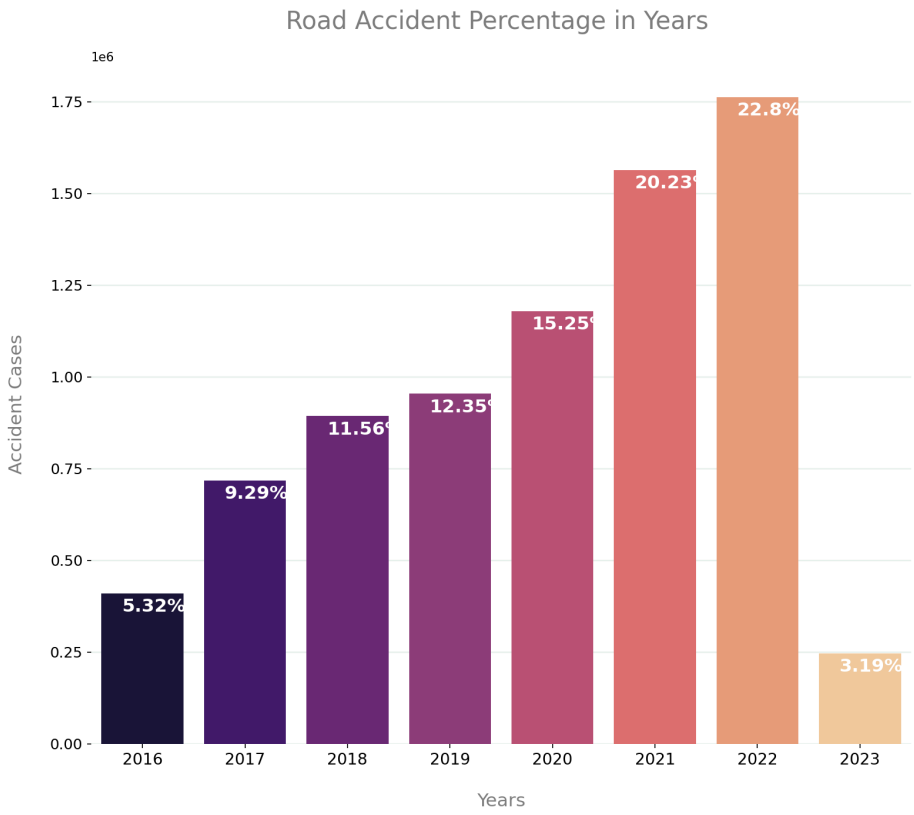

Figure 7: Visualizations of traffic accidents by years

The figure above shows the years in which the traffic accidents in the dataset occurred. As mentioned earlier, the dataset includes accidents between February 2016 and March 2023. So we know that the data shown in 2016 and 2023 are incomplete. Therefore, when we look at the period between 2017 and 2022, we see a steady increase in accidents as the years progress. After observing this, we started to investigate the reason for this, and we came to the conclusion that the biggest reason for this increase in accidents is digitalization. Because while the population or the number of vehicles is not increasing at this rate, we could not find any other logical explanation for the increase in accidents at this rate.

In addition, it can be seen that the rate of increase in 2019 and 2020 slowed down. It can be easily said that this situation is due to the Covid-19 pandemic.

These were the insights we have extracted using this dataset. More detailed results can be obtained by using other datasets. For example, while finding the top 10 states with the highest number of accidents, the results found by normalizing these states by their population or the number of registered vehicles will be healthier. In this way, we would perhaps be able to get an output such as which states have a higher rate of compliance with traffic rules.

SEVERITY PREDICTIONS

The following part of the report provides a detailed explanation of a script that utilizes PySpark and various machine learning models to **predict accident severity** in Alameda County, California. The models employed include **Logistic Regression**, **Decision Tree**, and **Random Forest**. The data is preprocessed and analyzed to evaluate the performance of each model.

We first created a spark session and read the 'US_Accidents_March23.csv' file. Then, convert 'Start_Time' and 'End_Time' columns into timestamp and extract year month day hour weekday columns. Calculated time duration as minutes and deleted the rows with negative or 0 values of time duration.

```
# Create Spark Session
spark = SparkSession.builder \
    .appName("Create DataFrame") \
    .getOrCreate()

df = spark.read Choose schema .csv( path: "US_Accidents_March23.csv", header=True, inferSchema=True)

# Convert 'Start_Time' and 'End_Time' columns into timestamp
df = df.withColumn( colName: 'Start_Time', to_timestamp(col('Start_Time'))) \
    .withColumn( colName: 'End_Time', to_timestamp(col('End_Time')))

# Extract year month day hour weekday columns
df = df.withColumn( colName: 'Year', year(col('Start_Time'))) \
    .withColumn( colName: 'Month', date_format(col('Start_Time'), format: 'MMM')) \
    .withColumn( colName: 'Day', dayofmonth(col('Start_Time'))) \
    .withColumn( colName: 'Hour', hour(col('Start_Time'))) \
    .withColumn( colName: 'Weekday', date_format(col('Start_Time'), format: 'E'))

# Calculate time duration as minutes
df = df.withColumn( colName: 'Time_Duration(min)', round((col('End_Time').cast("long") - col('Start_Time').cast("long")) / 60))

# Delete rows with negative or 0 values of time duration
df = df.filter(col('Time_Duration(min)') > 0)
```

Then, we calculated median and standard deviation to change outliers with NaN, and fill NaN values with median.

```
# Calculate median and standard deviation
td = 'Time_Duration(min)'
median_val = df.approxQuantile(td, probabilities: [0.5], relativeError: 0.01)[0]
stddev_val = df.select(stddev(col(td))).collect()[0][0]

# Change outliers with NaN
df = df.withColumn(td, F.when(F.abs(col(td) - median_val) > stddev_val * 3, value: None).otherwise(col(td)))

# Fill NaN values with median
df = df.na.fill({td: median_val})
```

Finally we chose feature columns for our Machine Learning Algorithms and filtered out our DataFrame accordingly. Since doing these operations on a large data like ours last much time we write dataframe into a csv file in order to use in MLlib algorithms.

```
# Choose feature columns for Machine Learning algorithms
feature_lst = ['Severity', 'Start_Lng', 'Start_Lat', 'Distance(mi)', 'City', 'County', 'State', 'Timezone',
               'Temperature(F)', 'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction', 'Weather_Condition',
               'Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway', 'Roundabout', 'Station',
               'Stop', 'Traffic_Calming', 'Traffic_Signal', 'Turning_Loop', 'Sunrise_Sunset', 'Hour', 'Weekday',
               'Time_Duration(min)']
```

```
# Create new dataframe with selected features.
df_sel = df.select(*feature_lst)

# Delete rows with null values
selected_columns = [col_name for col_name in df_sel.columns if
                    df_sel.select(col_name).na.drop().count() != df_sel.count()]
df_sel = df_sel.dropna(subset=selected_columns)

# Print dataframe row and column size
print((df_sel.count(), len(df_sel.columns)))

# Write dataframe into a csv file
df_sel.write.csv('US_Accidents_March23_clean_sel_dropna.csv', header=True, mode='overwrite')
```

The data is filtered to include only accidents that occurred in California and, more specifically, in Alameda County. After filtering the data, we drop the state and county columns. We set the target as 'Severity' since we are trying to predict it.

```
# Create Spark Session
spark = SparkSession.builder \
    .appName("Predict Severity") \
    .getOrCreate()

df_sel_dropna = spark.read Choose schema .csv( path: "US_Accidents_March23_clean_sel_dropna.csv", header=True, inferSchema=True)

# Choose the state
state = 'CA'
# Choose state as California
df_state = df_sel_dropna.filter(col('State') == state).drop('State')
# Set county
county = 'Alameda'
# Select the county of Alameda
df_county = df_state.filter(col('County') == county).drop('County')
```

All features, except the target variable (Severity), are combined into a single feature vector using VectorAssembler.

```
# Combine features as vector
feature_columns = [col for col in df.columns if col != target]
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
df_assembled = assembler.transform(df)
```

The data is split into training (80%) and testing (20%) sets.

```
# Split data into train and test data
(train_data, test_data) = df_assembled.randomSplit( weights: [0.8, 0.2], seed=21)
```

A MulticlassClassificationEvaluator is created to evaluate the performance of the models based on various metrics.

```
# Create evaluator
evaluator = MulticlassClassificationEvaluator(labelCol=target, predictionCol="prediction")
```

We used 3 different models to predict the severity of the traffic accident, namely logistic regression, decision tree and random forest model.

A Logistic Regression model is created and fitted to the training data. Predictions are made on the test data.

```
# Create and fit Logistic Regression Model
lr = LogisticRegression(featuresCol="features", labelCol=target)
lr_model = lr.fit(train_data)
```

A Decision Tree model is created and fitted to the training data. Predictions are made on the test data.

```
# Create and fit Decision Tree Model
dt = DecisionTreeClassifier(labelCol="Severity", featuresCol="features")
dt_model = dt.fit(train_data)

# Make predictions on test data
dt_predictions = dt_model.transform(test_data)
```

A Random Forest model is created and fitted to the training data. Predictions are made on the test data.

```
# Create and fit Random Forest Model
rf = RandomForestClassifier(labelCol="Severity", featuresCol="features", numTrees=100)
rf_model = rf.fit(train_data)

# Make predictions on test data
rf_predictions = rf_model.transform(test_data)
```

The accuracy, precision, recall, and F1-score for each model are calculated and printed.

```
# Calculate scores
accuracy = evaluator.evaluate(predictions, params: {evaluator.metricName: "accuracy"})
precision = evaluator.evaluate(predictions, params: {evaluator.metricName: "weightedPrecision"})
recall = evaluator.evaluate(predictions, params: {evaluator.metricName: "weightedRecall"})
f1 = evaluator.evaluate(predictions, params: {evaluator.metricName: "f1"})
```

These metrics help assess the models' performance in predicting accident severity. The table below summarizes the performance metrics for Logistic Regression, Decision Tree, and Random Forest models in predicting accident severity:

	Accuracy	Precision	Recall	F1
Logistic Regression	0.794	0.784	0.794	0.786
Decision Tree	0.793	0.786	0.793	0.779
Random Forest	0.775	0.772	0.775	0.746

Table 3: Model Performance Metrics

Logistic Regression shows the highest accuracy and recall, indicating that it is the best model for correctly predicting accident severity among the given models. Its precision is also high, indicating that the predictions are reliable.

The Decision Tree model performs similarly to Logistic Regression in terms of accuracy and precision, with a slight drop in F1 Score. This model is also effective but slightly less balanced in performance compared to Logistic Regression.

The Random Forest model, although robust due to its ensemble nature, has the lowest performance metrics among the three models. This indicates that while it can capture complex relationships, it might be overfitting or not as well-tuned for this specific dataset.

```
# Get feature importances
feature_importances = rf_model.featureImportances.toArray()

# Print feature importances
feature_importance_list = sorted(zip(feature_columns, feature_importances), key=lambda x: x[1], reverse=True)[:10]
print("Feature Importances:")
for feature, importance in feature_importance_list:
    print(f"{feature}: {importance}")
```

Feature importance for Decision Tree and Random Forest models is presented in the following tables:

Decision Tree	
Feature	**Score**
Distance	0.6400
Time_Duration	0.2238
Crossing	0.0928
Start_Lat	0.0188
Stop	0.0175
Wind_Direction_West	0.0037
Start_Lng	0.0015
Amenity	0.0011
Wind_Direction_Calm	0.0003
Pressure	0.0001

Table 4: Decision Tree Feature Importance

Random Forest	
Feature	**Score**
Distance	0.3045
Time_Duration	0.2498
Weather_Condition_Fair	0.1200
Crossing	0.0445
Stop	0.0408
Wind_Direction_Calm	0.0407
Wind_Direction_West	0.0392
Weather_Condition_Overcast	0.0319
Wind_Direction_W	0.0311
Weather_Condition_Cloudy	0.0136

Table 5: Random Forest Feature Importance

Decision Tree

- **Distance:** This feature is overwhelmingly the most important for the Decision Tree model, indicating that the distance traveled is a critical factor in determining accident severity.
- **Time_Duration:** This feature is also significant, suggesting that the duration of the trip correlates with the severity of the accident.
- **Crossing:** The presence of a crossing is another important factor, potentially indicating higher risk areas.
- Other features like 'Start_Lat', 'Stop', and 'Wind_Direction' have lesser influence but are still part of the model's decision process.

Random Forest

- **Distance and Time_Duration:** Similar to the Decision Tree, these are the top features, highlighting their critical roles.
- **Weather Conditions:** This model gives significant importance to various weather conditions ('Fair', 'Overcast', 'Cloudy'), indicating that weather plays a vital role in accident severity.
- **Crossing and Stop:** These features are also important, consistent with the Decision Tree's findings.
- **Wind_Direction:** Different wind types ('Calm', 'West', 'W') appear among the top features, showing their influence on accident outcomes.

RESULTS

The Logistic Regression model showed the best overall performance, with the highest accuracy, recall, and F1 score, indicating its robustness in predicting accident severity. The Decision Tree model also performed well, slightly trailing behind Logistic Regression, while the Random Forest model had the lowest performance metrics.

The feature importance analysis revealed that Distance and Time_Duration are the most influential features across both Decision Tree and Random Forest models. Weather conditions and specific location attributes (e.g., crossings) also significantly impact accident severity predictions.

These insights can help in designing better traffic management and safety measures, emphasizing the importance of monitoring distance, trip duration, and weather conditions to mitigate severe accidents.

CONCLUSION

To conclude, we explored a huge US Accidents dataset with this project. Our project was based on Apache Spark and we used frameworks like Pandas to help us plot graphs. We conducted in-depth research on the severity and trends of accidents. Also, we created visualizations by considering different features and investigated what causes accidents and tried to provide insights that could reduce the number of accidents.

We also used Logistic Regression, Decision Tree, and Random Forest models to predict accident severity. We made performance comparisons of these models and examined which feature is more important for which model. Finally, we shared the results and made comments.

REFERENCES

1. Moosavi, S. (2023) *US accidents (2016 - 2023)*, *Kaggle*. Available at: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents/data (Accessed: 02 June 2024).
2. *Machine Learning Library (mllib) guide* (no date) *MLlib: Main Guide - Spark 3.5.1 Documentation*. Available at: https://spark.apache.org/docs/latest/ml-guide.html (Accessed: 02 June 2024).