# ANALYSIS ON US ACCIDENTS FROM 2016 -2020 USING PYSPARK

**ECE 590 - Big Data Technologies| Prof. Erton Boci,Ph.D.**

**George Mason University Spring 2021**

***Team 5***

***Bhavana Emmadi, Keerthi Gollamudi, Deepa Kapse***

## ABSTRACT

**Road Accidents is a global problem for all age groups and the leading cause of serious injuries and deaths. Every year almost 1.35 million people are killed due to the accidents over the world and if we consider it as per day than 3,700 people have been killed globally by the crashes involving cars, buses, motorcycles, trucks[1]. The injuries caused by road traffic is an economic loss for individuals, families and to nations. This problem costs most of the countries with 3% of their gross domestic product[2]. The research contributes to the cause of United States accidents which is the busiest country in terms of road traffic. The dataset is taken from Kaggle which contains 3 million records with 49 columns also covers 49 states of the USA along with the accidents data collected from February 2016 to December 2020[3]. This project is used to determine and predict the factors which are influencing accidents that happened in the United states. The dataset contains some of the important factors like severity, city, state, time zone, weather conditions. The end goal of this research is to perform data cleaning, data exploratory and visualizations using PySpark-Python Spark API framework.**

## INTRODUCTION

Apache Spark is an open-source cluster computing framework which can be used to process huge volumes of real-time data. Spark has a distributed computing framework which processes the big data fast, easy and scalable. For processing the huge volumes of data, spark follows a master-slave architecture. The spark architecture[4] mainly consists of 2 components i.e. Driver program and Executor. The user writes the spark code in the driver program using any of the Spark API's namely Scala, Java, Python, R, SQL. Executor is the Java Virtual Machine (JVM) which provides hardware resources for running the tasks which are launched by the driver program. The basic entity of the Spark is Resilient Distributed Dataset(RDD) which is a read-only partitioned collection of data. Spark can distribute a collection of records using an RDD, and process them on different machines parallelly. In RDD, the data is splitted into many chunks based on the key. RDD's are immutable and follow lazy transformations.

In this paper, we are transforming the data to Apache parquet[5]. Apache parquet is an open source which supports columnar storage of data whereas CSV and TSV files support row-based storage of data. Parquet is available to any project in the Hadoop environment regardless of the choice of data model, programming language or data processing framework. Parquet is very helpful to reduce storage requirements of large datasets to a great extent[6]. It also improves scan and deserialization time which will reduce the overall costs as a whole. It also helps for fast querying on large datasets. So we have transformed the input csv file of the US-Accidents dataset which has around 3 million rows of data into the parquet file and implemented many queries to get some useful insights from the data in a small amount of time.

The intent behind this project is to research US accidents dataset from 2016 to 2020 at a country-

level of granularity covering 6 years of data with various analytical approaches to investigate.

- ❏ Trends in accidents throughout United states based on time zone and states.
- ❏ Trends in accidents over the period.
- ❏ Trends in accidents based on day of the week.
- ❏ Number of accidents based on the hour of day.
- ❏ Weather-related events that resulted in most accidents.

This project was put down to answer the above problem statement.

## LITERATURE REVIEW

The authors in [3] talk about their system setup to capture US accident data from various systems. They have set up ETL to fetch data from a variety of sources such as: MapQuest Traffic [7] and Microsoft Bing Map Traffic [8] for traffic data and Weather Underground API [9] for weather data. They have performed data augmentation tools such as and Nominatim tool [10] to geomap the data. I would like to perform my analysis using PySpark and perform exploratory analysis.

## 1. SYSTEM ARCHITECTURE:

In the first step, spark is connected to a cluster then it is hosted on a virtual machine that was created using Microsoft azure. We loaded the US accidents .csv datasets into spark data frame performed data cleaning. To improve the query performance we converted the PySpark data frame into .parquet file. The Parquet supports efficient encoding schemes and compression options. PySpark SQL provides support for both reading and writing Parquet files that automatically capture the schema of the original data, It also reduces data storage by 75% on average. Further the exploratory analysis was achieved using matplotlib pyplot.

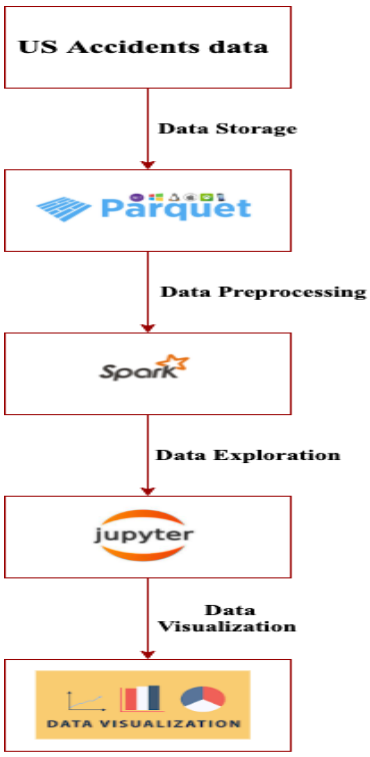



**Figure:1 : System Architecture**

The above **Figure 1** displays the system architecture for the analysis of USA accidents data by using Apache spark.

Hardware specifications of this project are listed below.

- ❏ Operating system: Windows 10, 64 bit
- ❏ CPU: 4 core Intel64 Family 6 Model 85
- ❏ Memory: 16 GB
- ❏ Hard Disk:128GB

Software's used are shown below.

- ❏ Apache Spark: version 2.4.7
- ❏ Apache Parquet is used for data transformation.
- ❏ Jupyter Notebook for operating queries and performing visualizations: 6.1.4

## 2. ABOUT DATASET:

Dataset contains 49 features related to location, weather condition, road condition and contains 3.5 million records. It consisted of geographical details based on time zone, geographic location, weather conditions. It contained several time-based features start, end time. There were several binary categorical features such as 'Traffic Calming', 'Stop', 'Station', 'Roundabout', 'Railway', 'No Exit', 'Give Way', 'Crossing', 'Bump', 'Amenity' which contained categorical values 'Boolean – True or false'. The features related to address categories such as country, city, county, zip code, street, airport code dealt with related sets of information. Features such as id, description, street, weather timestamp, turning loop etc. had large variability within the data around 40-70% of the data values were unique.

| Dataset | | | |
|---|---|---|---|
| **Attribute** | **Data Type** | **Attribute** | **Data Type** |
| ID | Varchar | Zip Code | Integer |
| Severity | Integer | Timezone | Varchar |
| Start_Time | Timestamp | Weather_Condition | Varchar |
| End_Time | Timestamp | Junction | Boolean |
| Street | Varchar | Railway | Boolean |
| Side | Char | Wind_Direction | Varchar |
| City | Varchar | Wind_Speed | Float |
| County | Varchar | Sunrise_Sunset | Boolean |
| State | Varchar | Country | Varchar |

**Table 1: Attributes and their data types**

## 3. HIGHER-LEVEL FRAMEWORK:

**Figure 2** higher level framework the research project relied on PySpark API and SparkSQL for the purpose of data cleaning, data wrangling, data exploration, data analysis and visualizations. PySpark is the primary API used for data cleaning and wrangling. Once data is prepared, then Jupyter, SparkSQL are used for data exploration and visualizations.


**Figure 2 : High Level Architecture**

### 3.1 Data Cleaning and Data Wrangling

We have used PySpark for the data cleaning, data wrangling and data transformation. PySpark is an API which helps to use Python along with the Spark framework.

- There were several features that did not add much meaning towards analysis as most of the values were null and the dataset had few other features such as weather condition, county, airport code which represented the same purpose.

- Features such as id, description, street, weather timestamp, turning loop etc. had large variability within the data around 40-70% of the data values were unique.
- There were several binary categorical features such as 'Traffic Calming', 'Stop', 'Station', 'Roundabout', 'Railway', 'No Exit', 'Give Way', 'Crossing', 'Bump', 'Amenity' where 99% of the data belonged to only one category 'Boolean – false'. Including such features with no variance did not add value in predicting the severity of accidents.

**3.1.1 Importing the Libraries:**

Firstly, we have imported find spark[11] which helps in searching the PySpark installation on the server and add PySpark installation path to the sys.path at the runtime so that we can import the PySpark modules. This in turn helped us to import the PySpark library.

```
import findspark
findspark.init()
import pyspark
```

We imported a Spark Session which is used to create data frame, register data frame as tables, execute SQL on tables, cache tables, read parquet files. So to create a Spark Session[12], we have used the builder pattern. The command below returns an existing Spark Session if there is already one in the environment or creates a new one if necessary.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Python
Spark project").config("spark.some.config.option",
"some-value").getOrCreate()
```

**3.1.2 Importing the Data:**

The dataset is in CSV format. We have read the input US Accidents csv file into a data frame object named accidents_df.

```
accidents_df = spark.read.option("sep",
",").option("header",
"true").csv("../US_Accidents_Dec20.csv")
```

**3.1.3 Data Cleaning:**

**a. Handling the Special Characters:**

Some of the column names in our input file have special characters in them. While transforming and saving the file in parquet format from the CSV, it will not allow special characters and spaces in the column names. So we have replaced everything except the alphabets(both lower and upper case) and digits into the "_" using regular expressions. This can be implemented by importing the 'functions' and 're' as shown below.

```
from pyspark.sql import functions as F
import re
df =accidents_df.select([F.col(col).alias(re.sub("[^0-
9a-zA-Z$]+","_",col)) for col in
accidents_df.columns])
```

**b. Dropping the unnecessary columns and handling the missing values:**

Initially the dataset has 49 columns. We dropped the columns which had more than 70% of data records as null which might not help for analysis. We dropped some of the columns which had high correlations with other features and did not add much value for our analysis. In this process, we have dropped 11 columns after which we are left with 38 columns for our analysis further.

### 3.1.4 Data Transformation:

The input dataset which we considered was in CSV format. CSV works well with the small datasets. As the dataset size grows, load times become impractical and reads cannot be optimized with the CSV. However if we want to deal with millions of records, Parquet and Avro come in. We have transformed our input CSV file to the Parquet file which enables the optimized columnar storage[13]. We have written the data frame which had CSV to the parquet file which was named as USAccidents.parquet. Once the parquet file is written, it is read into a parquet Accidents data frame.

```
df.write.parquet("USAccidents.parquet")
parquetAccidents = spark.read.parquet("USAccidents.parquet")
```

### 3.1.5 Data Querying

The data we are analyzing is 1.577GB, the queries are performed using SparkSQL to generate descriptive statistics. There are 10 queries which are executed in Jupyter Notebook.

**Query 1:** Number of accidents for each state.
**Query 2:** Top 5 accidents based on states.
**Query 3:** Least 5 accidents based on states.
**Query 4:** Time zone of the states which had more/less no of accidents.
**Query 5:** Top 10 accidents based on cities.
**Query 6:** The number of accidents over the years.
**Query 7:** The most and least accidents over the week.
**Query 8:** Hour of the day has the peak and least accidents.
**Query 9:** Top 10 weather conditions that resulted in most accidents.
**Query 10:** Frequency of severity of various accidents. Do accidents have a high impact on traffic?

We have used SparkSQL for querying the database. SparkSQL is a spark module for structured data processing. We have executed the above 10 queries in the Jupyter notebook, and their execution time is as follows:

| Query | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Time(S) | 2 | 2 | 2.8 | 1.3 | 4.7 | 1.1 | 1.6 | 1.1 | 2.7 | 1.2 |

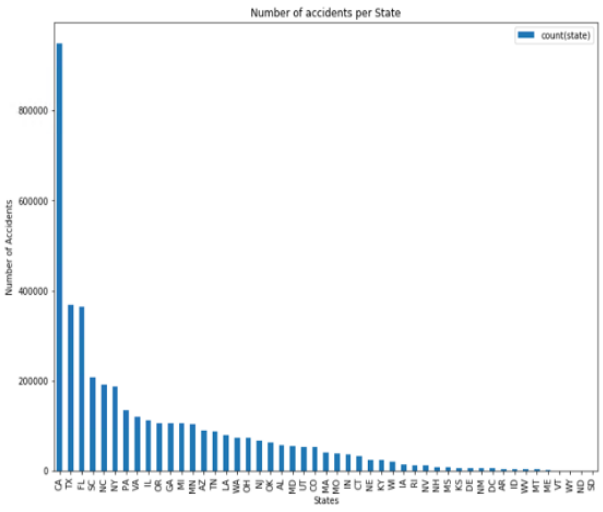
**Table 2: Execution time of each query in seconds**

## 4. US ACCIDENTS EXPLORATORY ANALYSIS

In this project, US accidents were analyzed from various perspectives. First, the number of accidents per state and sorted it by top and least 5 accidents based on state, and time zone of the states which had more/less no of accidents. The severity of the US accidents was dependent on state, weather conditions, time zone and cities variables in the dataset. The variables that had least impact on severity were traffic calming, stop, station and railway where 99% of the data is in the form of Boolean. There are some factors that are analyzed using visualizations like bar charts, line charts, to get insights and trends from the data. This section is divided into subsections 4.1 Analysis based on state, 4.2 City wise analysis, 4.3 Period wise analysis, 4.4 Analysis on weather conditions, 4.5 Severity of accidents.

### 4.1 Analysis based on state:

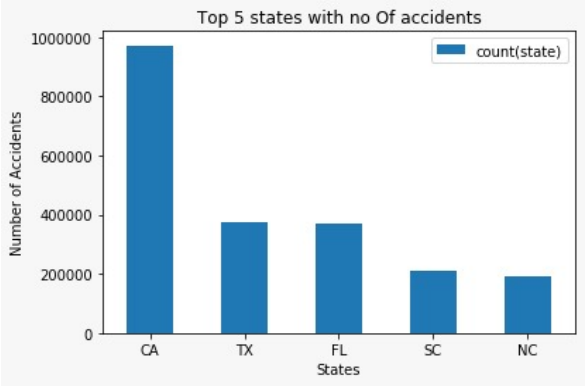
This exploratory analysis is based on the variables states and cities and how the accidents count is varying from different states and cities. From the figures below, the number of accidents were significantly different for various states. Moreover, if we consider **figure 3** the maximum number of accidents have occurred in California and the second most accidents are in Texas and then followed by Florida. These are the top 3 states in the United states with the highest population that was given in census data[14]




| state | count(state) |
|---|---|
| CA | 948331 |
| TX | 367846 |
| FL | 363871 |
| SC | 207958 |
| NC | 191554 |
| NY | 186919 |
| PA | 133563 |
| VA | 119276 |
| IL | 111190 |
| OR | 106213 |
| GA | 105479 |
| MI | 105335 |
| MN | 103348 |
| AZ | 89322 |
| TN | 88119 |
| LA | 79467 |
| WA | 73954 |
| OH | 72673 |
| NJ | 66788 |
| OK | 64078 |

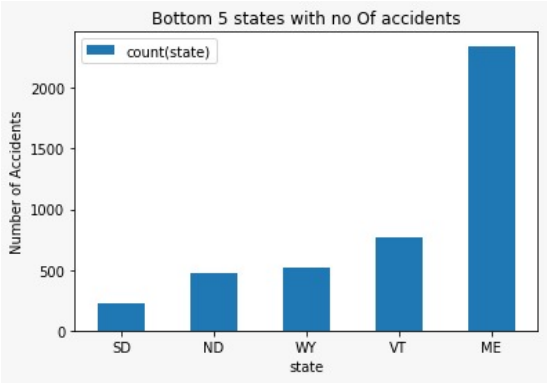
**Figure 3. Number of accidents for each state**

**Figure 4** shows the top five states with the most number of accidents. The frequency distribution from the graph represents CA, TX and FL, which are the top three states with the greatest number of accidents as these are states with the highest number of populations across the United states. Whereas South Carolina and North Carolina have the moderate number of populations with less number of accidents when compared with the top 3 states. Overall, it can be inferred that population of the states might be a factor to predict the number of accidents.




| state | count(state) |
|---|---|
| CA | 948331 |
| TX | 367846 |
| FL | 363871 |
| SC | 207958 |
| NC | 191554 |

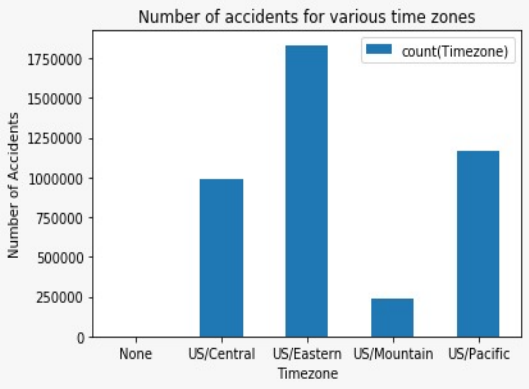
**Figure 4: Top 5 accidents based on states.**




| state | count(state) |
|---|---|
| SD | 220 |
| ND | 430 |
| WY | 508 |
| VT | 751 |
| ME | 2296 |

**Figure 5. Least 5 accidents based on states.**

The Frequency distribution of the bottom 5 states **Figure 5** above with least number of accidents it can be observed that the bottom 3 states SD,ND,WY, VT which has least population across United states Census Ranking[14] are more likely to have a smaller number of accidents. So it can be inferred that the state's population might be a factor to predict the number of accidents.

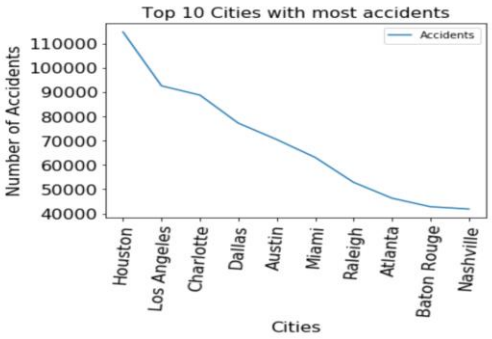



| Timezone | count(Timezone) |
|---|---|
| US/Central | 972236 |
| US/Eastern | 1782992 |
| US/Mountain | 224355 |
| US/Pacific | 1144578 |

**Figure 6. Time zone of the states which had more/less no of accidents.**

From the exploratory analysis for the variables time zone and number of accidents per time zone. We observed the number of accidents were significantly different for various time zones. From above **Figure 6** the maximum US accidents occurred in the US- Eastern time zone. A moderate number of accidents at Pacific and central time zone. The Mountain time zone had the least number of accidents.

## 4.2 City wise analysis

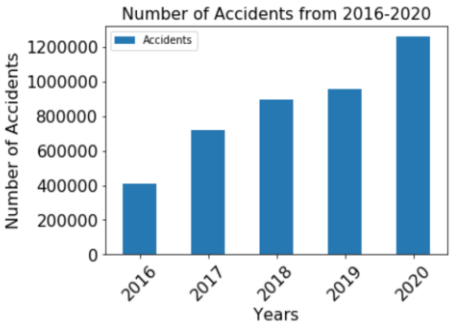



| City | Accidents |
|---|---|
| Houston | 114905 |
| Los Angeles | 92701 |
| Charlotte | 88887 |
| Dallas | 77303 |
| Austin | 70538 |
| Miami | 63162 |
| Raleigh | 52876 |
| Atlanta | 46328 |
| Baton Rouge | 42814 |
| Nashville | 41850 |

**Figure 7. Top 10 accidents based on cities.**

We had plotted the number of accidents vs Cities to know the names of cities where most of the accidents happened **Figure 7**.The above figure displays the top 10 cities with the greatest number of accidents occurring. The X-axis has the Top 10 cities listed whereas the Y-axis has the number of accidents. Houston has the highest number of accidents when compared with the other cities from the data whereas Nashville is the least among the top 10 cities.

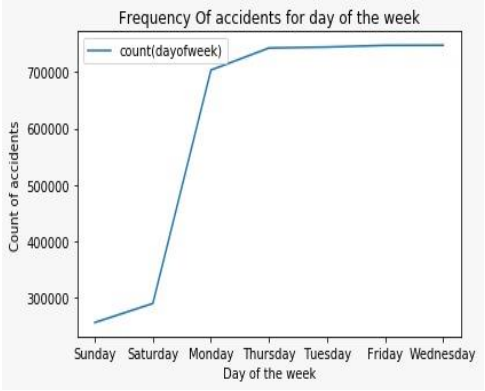
## 4.3 Period wise analysis




| YEAR | Accidents |
|---|---|
| 2020 | 1222895 |
| 2019 | 933668 |
| 2018 | 868982 |
| 2017 | 698187 |
| 2016 | 400429 |

**Figure 8. The number of accidents over the years.**

The bar chart from the above **Figure 8** represents the number of accidents over the years 2016 to 2020. The greatest number of accidents has occurred in the year 2020 and least was in 2016. From the graph, the number of accidents is gradually increasing over the years. The number of Accidents were lower and showed a more significant change in the early years as compared with those in recent years when the number of accidents raised high.

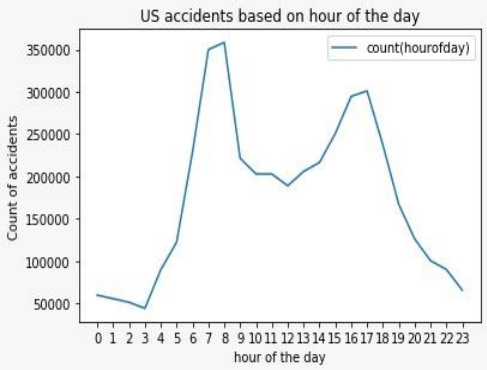



| dayofweek | count(dayofweek) |
|---|---|
| Sunday | 255920 |
| Saturday | 289754 |
| Monday | 703779 |
| Thursday | 742968 |
| Tuesday | 744645 |
| Friday | 747624 |
| Wednesday | 747851 |

**Figure 9: The Frequency of accidents over the week.**

The x-axis on the graph represents days of the week as follows: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and Saturday. The **Figure 9** above shows that the highest number of road traffic incidents occurs from Monday to Friday. This means a lot of road accidents happen during weekdays and this could be due to commute to offices during weekdays, schools and for most of the sector's operations are carried out on weekdays with partial or complete off on weekends. The fewer number of road accidents occurred during the weekend. This led to the conclusion that the higher number of vehicles on the roads during the working days of the week are more likely to yield more Road accidents. The data analysis on field day of week suggests that the factors during the weekday such as large traffic over weekdays, huge commute, many forums being operational contribute to the higher number of road traffic accidents. The number of road accidents are less during weekends compared to weekdays, during the week most

people are travelling to different workplaces. I assume most people travel during the weekday for work, schools and other essential activities and very few travels during weekends. So it can be inferred that the day of the week might be an important factor to predict the number of accidents.

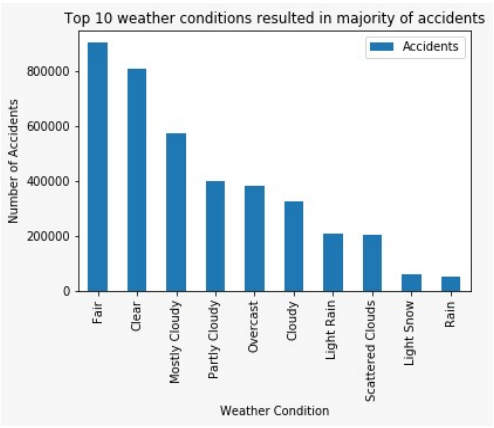



| hourofday | count(hourofday) |
|---|---|
| 0 | 57740 |
| 1 | 53615 |
| 2 | 49493 |
| 3 | 42722 |
| 4 | 87024 |
| 5 | 118758 |
| 6 | 222709 |
| 7 | 341774 |
| 8 | 350077 |
| 9 | 216073 |
| 10 | 197938 |
| 11 | 197995 |
| 12 | 184478 |
| 13 | 200770 |
| 14 | 211112 |
| 15 | 244606 |
| 16 | 287517 |
| 17 | 294216 |
| 18 | 231417 |
| 19 | 162778 |
| 20 | 122819 |
| 21 | 97636 |
| 22 | 87313 |
| 23 | 63581 |

**Figure 10: Hour of the day has the peak and least accidents.**

To understand the trends of US accidents over the hour of the day **Figure 10**. We plotted the total number of accidents for an hour of the day. The line graph had spikes and dips. It can be observed that the number of accidents were at peak at 8 am and decreased gradually from 8 am to 10 am. The number of accidents were stable from 10 am to 3 pm and further started increasing with peak saturation from 3pm to 5pm.The total number of accidents takes a sharp dip after 8pm. The number of accidents is least for the time range of 8pm to 5 am. We assume that most people travel during the day and 8 am, 9 am, 3pm , 4pm and 5 pm are peak timings for office, schools and other essential activities. So it can be inferred that hour of the day might be an important factor to predict the number of accidents. Morning 7 AM to 9 AM and evening 4 PM to 6 PM are the prime hours when most of the accidents happened.

## 4.4 Weather conditions



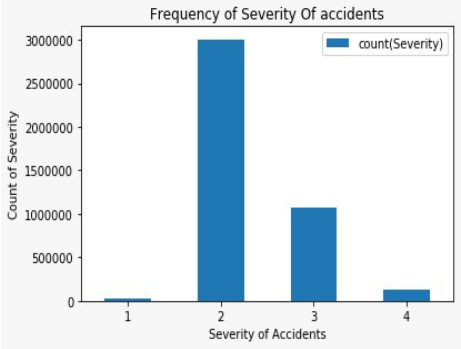

| Weather_Condition | Accidents |
|---|---|
| Fair | 900872 |
| Clear | 808181 |
| Mostly Cloudy | 571743 |
| Partly Cloudy | 397415 |
| Overcast | 382485 |
| Cloudy | 323340 |
| Light Rain | 206389 |
| Scattered Clouds | 204661 |
| Light Snow | 57148 |
| Rain | 48641 |

**Figure 11. Top 10 weather conditions that resulted in the majority of accidents.**

We have considered the Weather conditions column from the dataset to understand whether the weather conditions are affecting the number of accidents. In **Figure 11** shows the top 10 weather conditions that resulted in the majority of the accidents. X-axis has the list of Top 10 weather conditions during which the maximum number of accidents occurred. Y-axis has the number of accidents. Surprisingly, from the above bar plot, we can say that when the weather is 'Fair', most accidents happen. This is followed

by the weather condition 'Clear'. Also the weather condition 'Rain' resulted in a smaller number of accidents among the Top 10 weather conditions, when compared to the weather conditions 'Fair' and 'Clear'. From this we can say that weather conditions are not majorly impacting the number of accidents based on our dataset. From this plot we came to the conclusion that the reason behind the major number of accidents are not the weather conditions as the weather condition Fair resulted in most of the accidents.

### 4.5 Severity of accidents




| Severity | count(Severity) |
|---|---|
| 1 | 29312 |
| 2 | 3006626 |
| 3 | 1072821 |
| 4 | 123782 |

**Figure 12: Frequency of Severity of Accidents**

Plotted the severity of accidents versus number of accidents **Figure 12**. The severity of the accidents had 4 categorical values ranging from 1 to 4 where 1 is the least severity accident with minimal impact on traffic and damage due to accidents and 4 being the severe severity accident with high impact on traffic and personal damage due to accidents. In **Figure 12** it can be inferred that the accidents with moderate severity are more compared to least and high severity accidents i.e. 1 and 4.

## 5. CHALLENGES:

- Data cleaning - The headers in the dataset contained special characters because of which we were unable to convert it into a .parquet file. We handled the special character for flexible .parquet conversion.
- Huge data and time-consuming retrieval- csv file was huge for querying and fetching the data was time consuming. We converted it into .parquet which improved our query performance.
- Retrieving day of the week, hour of the day from timestamp.

## 6. CONCLUSION:

The features such as state, city, time zone, day of the week, hour of the day were the important factors deciding the US accidents. Based on the exploratory analysis it can be found the states which are densely populated were likely to have more accidents. Also, based on the feature hour of the day it can be found that the accidents are highest during the peak hours of the day Morning 7 AM to 9 AM and evening 4 PM to 6 PM compared to other hours of the day. Based on our analysis of time zone features pacific and eastern time zones had more number of accidents compared to central and mountain view time zones. Further, based on the analysis of day of the week the number of road accidents are less during weekends compared to weekdays. For various weather conditions the accidents were more during fair and clear weather conditions followed by Rain from this we can infer weather was not the major factor for the accident.

## 7. DIRECTIONS FOR FUTURE WORK:

We would like to do following additions:

1. We plan to add additional demographic and traffic features to this data set and try out exploratory and predictive analysis.
    a. Exploratory analysis using county level demographics.
    b. Exploratory analysis using county level traffic information.

2. Perform a state wise exploratory and predictive analysis to understand the varying patterns.

## REFERENCES:


[1] Global road safety. (2020, December 14). Retrieved May 05, 2021, from https://www.cdc.gov/injury/features/global-road-safety/index.html

[2] Road traffic injuries. (n.d.). Retrieved May 05, 2021, from https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[3] Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. "A Countrywide Traffic Accident Dataset."

[4] Says:, S. (2020, November 25). Apache spark architecture: Distributed system architecture explained. Retrieved May 05, 2021, from https://www.edureka.co/blog/spark-architecture/

[5] Apache parquet. (n.d.). Retrieved May 05, 2021, from https://parquet.apache.org/

[6] What is apache parquet. (2021, April 13). Retrieved May 05, 2021, from https://databricks.com/glossary/what-is-parquet

[7] MapQuest Traffic API. 2019. https://www.mapquest.com/

[8] Bing Map Traffic API. 2019. https://www.bingmapsportal.com/

[9] Weather Underground. 2014-2019. https://www.wunderground.com/

[10] Nominatim Tool. 2019. https://wiki.openstreetmap.org/wiki/Nominatim.

[11] Nnk. (2021, April 17). How to import pyspark in python script - sparkbyexamples. Retrieved May 05, 2021, from https://sparkbyexamples.com/pyspark/how-to-import-pyspark-in-python-script/

[12] Pyspark.Sql module¶. (n.d.). Retrieved May 05, 2021, from https://spark.apache.org/docs/2.1.0/api/python/pyspark.sql.html?highlight=sparksession

[13] Levkovsky, M. (2019, August 01). CSV vs PARQUET vs avro: Choosing the right tool for the right job. Retrieved May 05, 2021, from https://medium.com/ssense-tech/csv-vs-parquet-vs-avro-choosing-the-right-tool-for-the-right-job-79c9f56914a8

[14]Statista.2021.https://www.statista.com/statistics/183497/population-in-the-federal-states-of-the-us/ (2021)