Cairo university
Faculty of engineering
Computer engineering department
Big Data [**CMP4011**]
Project Report

US Accidents Analysis

Team 14

Name	section	BN
احمد أسعد درويش محمد درويش	1	1
عمر فريد عبد العاطى لملوم	2	4
محمد نبيل عبد الفتاح فهمى	2	19
ممدوح احمد محمد محمد عطيه	2	26

Presented to:

Eng. Omar Samir

Project Proposal

Problem Description

Our project will use a big dataset of US accidents from Kaggle to understand why accidents happen. We'll use different data analysis methods to find patterns in the data, like what makes accidents more likely in certain areas or weather conditions, and predict how severe can it become relative to many factors featured in the dataset.
The motivation behind this idea is all about making roads safer by giving valuable information to people who can make a difference, like *government officials*, *transportation departments*, and even *everyday drivers*.

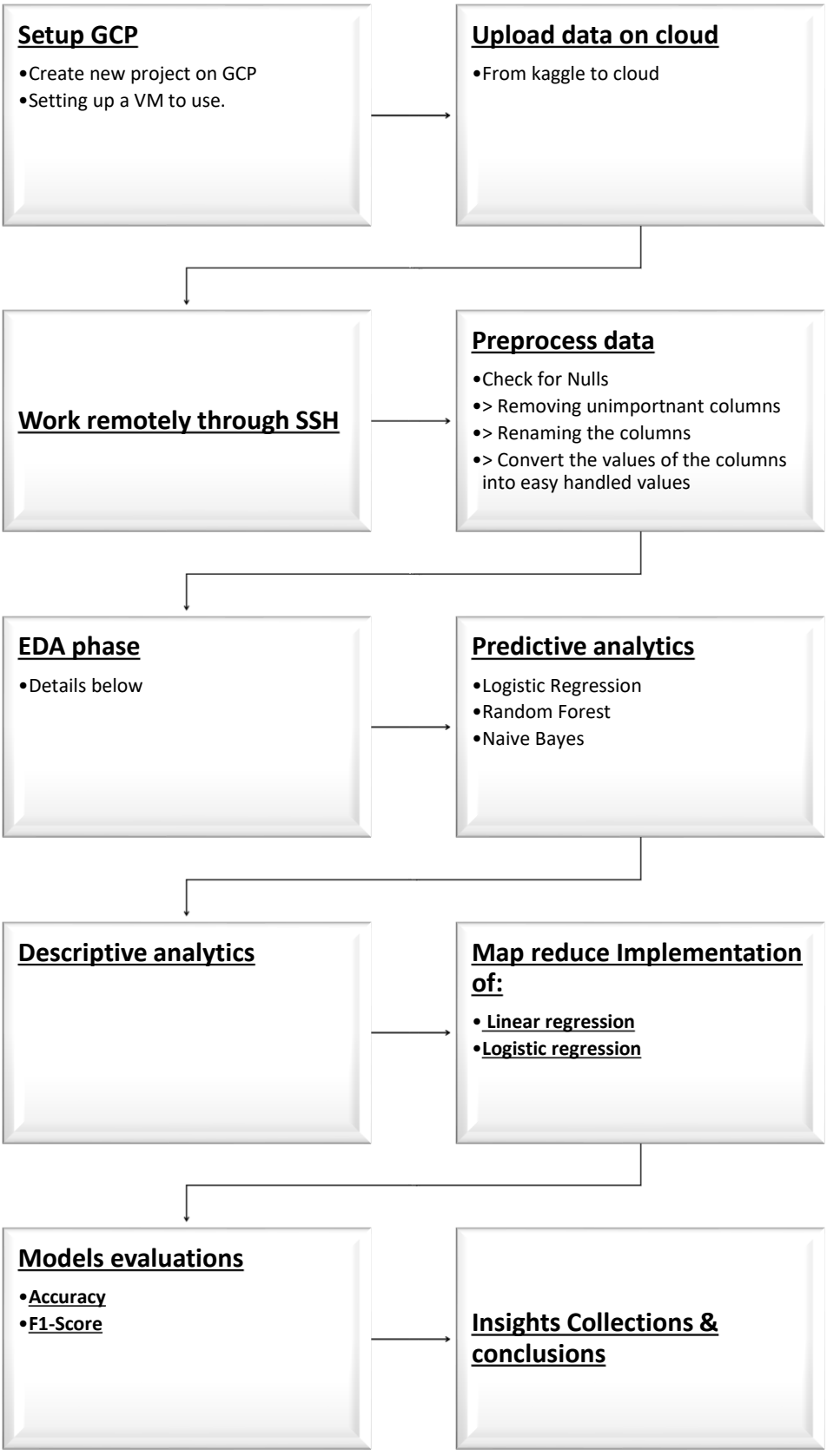
Dataset Link : US Accidents

(49 columns – 7.7 Million records – 3 GB) ➔ Dataset link

Planned approach

- We will use **PySpark** as a framework.
- EDA phase steps: Understanding variables - Data cleansing (e.g.: check nulls – Default values) – dropping unwanted columns – visualization) and more as we explore the data.
- *Gain insights* : for ex: Analyzing the relationship between weather conditions and accident severity – rate of accidents in cities – Map plot of severity of accidents in the US , and so on…
- Predictive analytics include:
    1. [classification - Logistic regression]
       ➔ Predicting the **severity** of an accident based on the factors involved (e.g. : weather conditions – surrounding POIs – place – time …)
    2. [Regression - Linear Regression]
       ➔ Predicting **accident Duration** as indicator of impact on *traffic flow.*
- Descriptive analytics include:
    1. [Association Rules] ➔ find obvious correlation between different circumstances (e.g.: time & place of accidents)
    2. [Clustering - K-Means clustering]
       ➔ Accident location clusters based on latitude information.
- Algorithms to be implemented using Map reduce: Linear regression.
- Cloud to be used ➔ GCP.

Project Pipeline



Problem analysis

We performed an EDA over the data , with the following goals and objectives:

- ***Understanding Accident Trends***: EDA can reveal trends in accident frequency over time, seasonal variations, and any notable spikes or decreases.
- ***Identifying High-Risk Areas***: Analysis of accident locations can help identify regions with a high frequency of accidents, assisting authorities in implementing targeted safety measures.
- ***Examining Contributing Factors***: By exploring variables such as weather conditions, road conditions, and time of day, insights can be gained into factors contributing to accidents.

1. Data preprocessing steps:

a. Get null counts for each row
   i. Count null values for each column
   ii. Calculate total row count
   iii. Subtract non-null counts from total row count to get null counts for each column
   iv. Display null value counts for each column
   v. # Drop nulls

b. Calculate missing value percentages for each column
   i. Filter out columns with non-zero null value percentages

2. Data visualization:

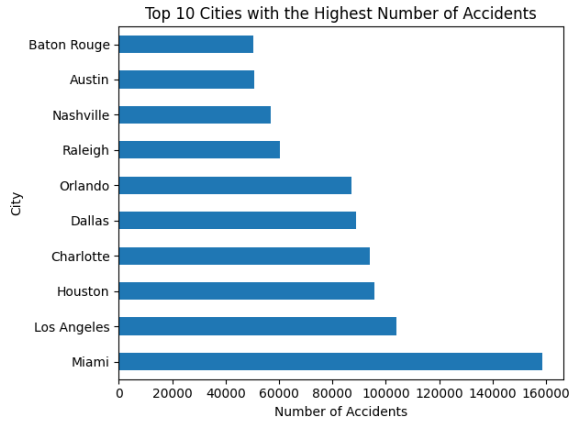
a. Explore city column:
   i. Show the first few rows of the 'City' column
   ii. Count the number of unique cities in the 'City' column

```
1. Number of unique cities: 11729
```

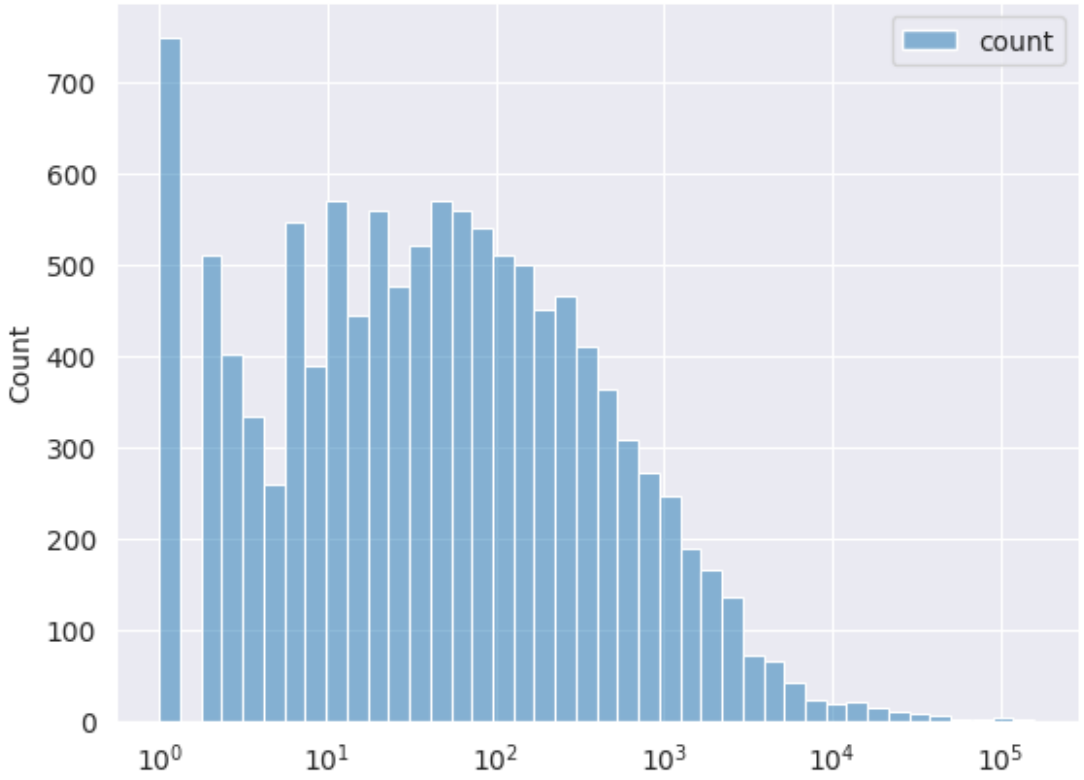
   iii. Group the data by 'City' and count the number of occurrences
   iv. Sort the result in descending order based on the accident counts
   v. Get the city with the highest number of accidents

```
City with the highest number of accidents: Miami
Number of accidents in Miami: 158736
```

   vi. Plot the first 10 cities with the highest number of accidents



vii. Plot a histogram of number of accidents :



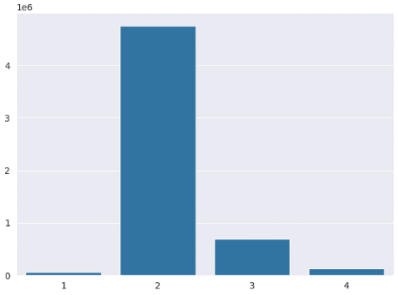
From the above graph we can analyze that distribution of accident is more between ten to hundred. When it goes further its decreasing exponentially.

b. Severity column analysis & preprocessing:
   i. Convert the Severity to integer
   ii. Group data by severity
   iii. Sort the result in descending order based on the severity counts
   iv. Get the Severity with the highest number of accidents

Severity with the highest accidents count: 2

Where the Number of accidents of severity degree: 4756722

v. Plot the severity class frequency:



From the graph we conclude that the class of severity = 2 is the most frequent class.

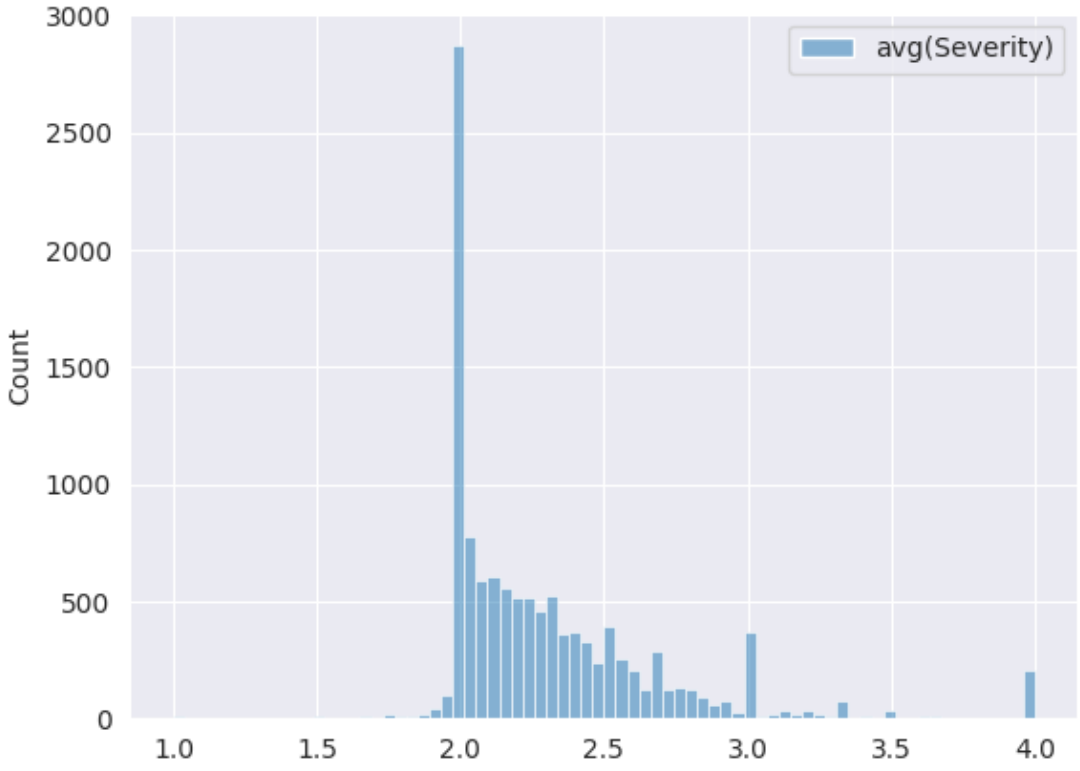
vi. Calculate average severity per city :

1. Result:

City	avg(Severity)
Marinette	4.0
Ackerman	4.0
Cold Brook	4.0
Black River	4.0
Eagle Bay	4.0
Mc Dermott	4.0
S Coffeyville	4.0
Richboro	4.0
Adena	4.0
Iron Mountain	4.0
Mount Tremper	4.0
Mulkeytown	4.0
Valentine	4.0
Bisbee	4.0
Lumberville	4.0
South Whitley	4.0
Juda	4.0
Rose City	4.0
Houghton Lake Hei ...	4.0
Two Rivers	4.0

only showing top 20 rows

2. Histogram plot:



From the above graph, most of the cities have average severity = 2

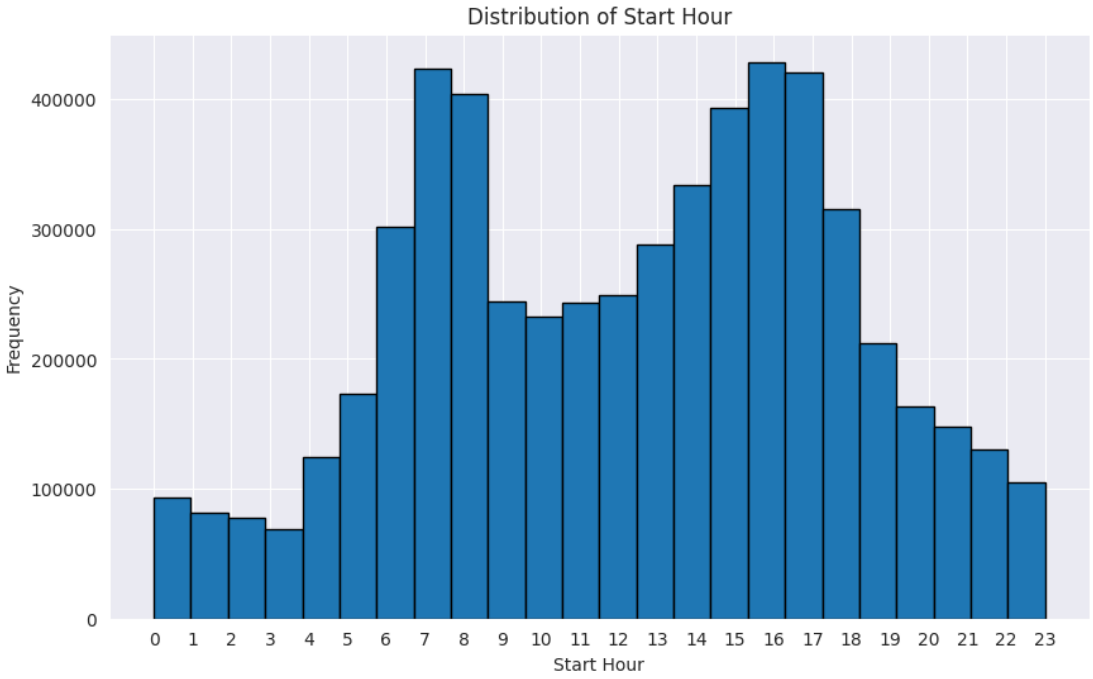
c. Start time analysis

i. Convert "Start_Time" and "End_Time" columns to datetime format

ii. Hour:

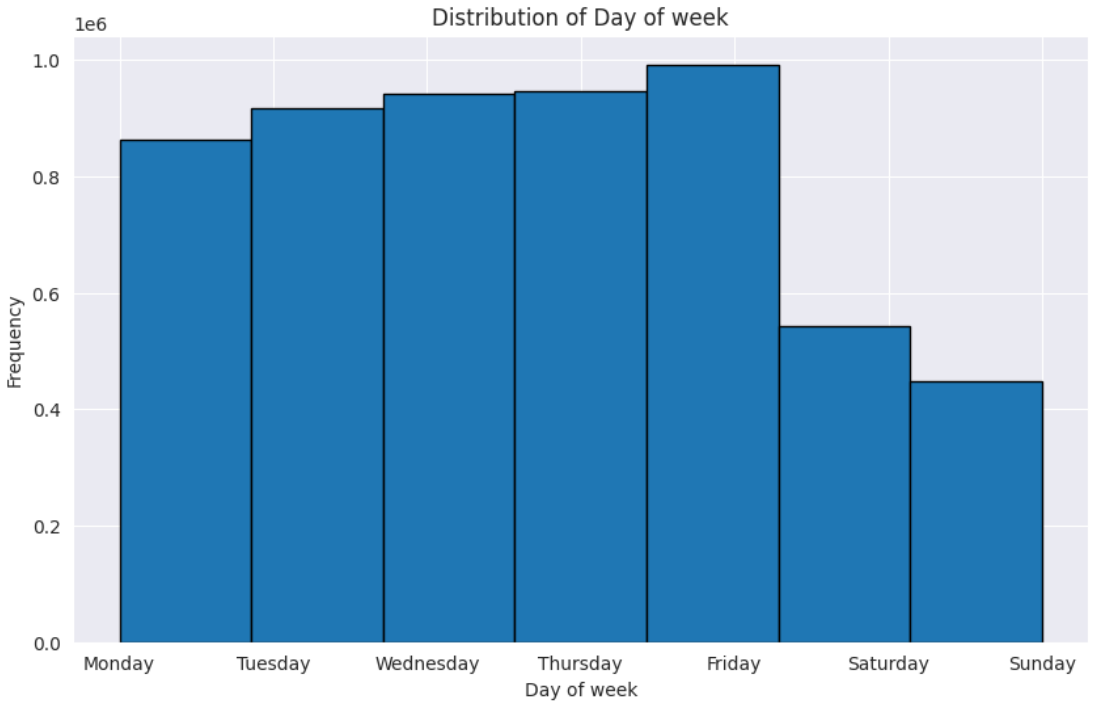
1. Extract the hour from the "Start_Time" column

2. Show the distribution of number of accidents per hours



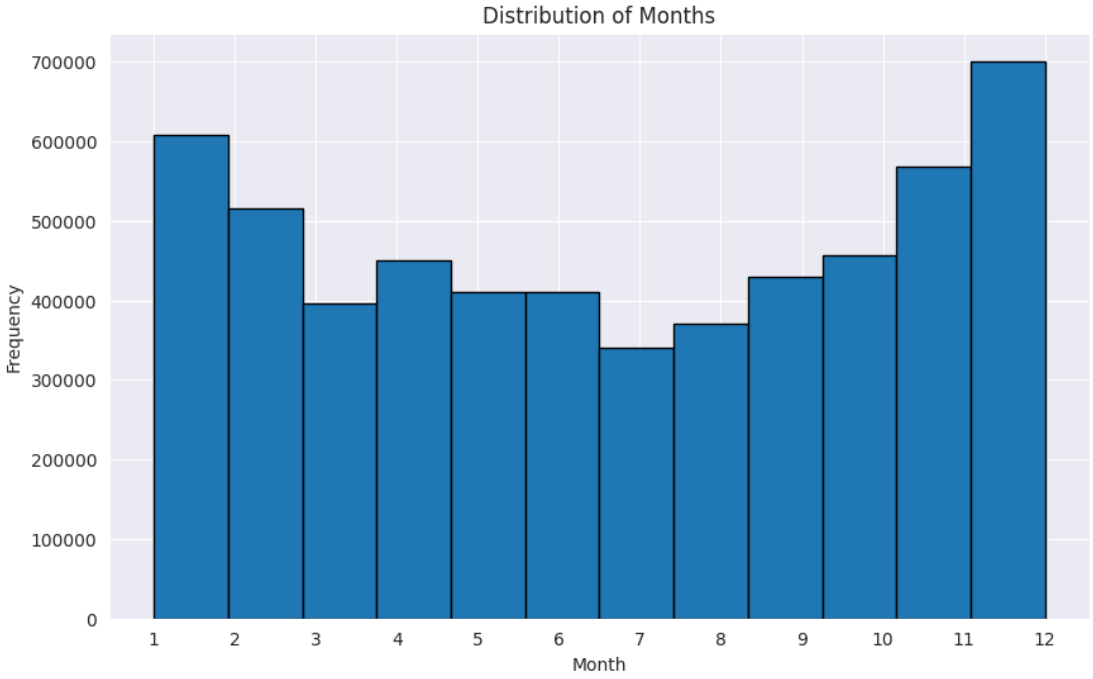
iii. Days:

1. Extract the day from the "Start_Time" column
2. Show the distribution of number of accidents per days of week



iv. Months:

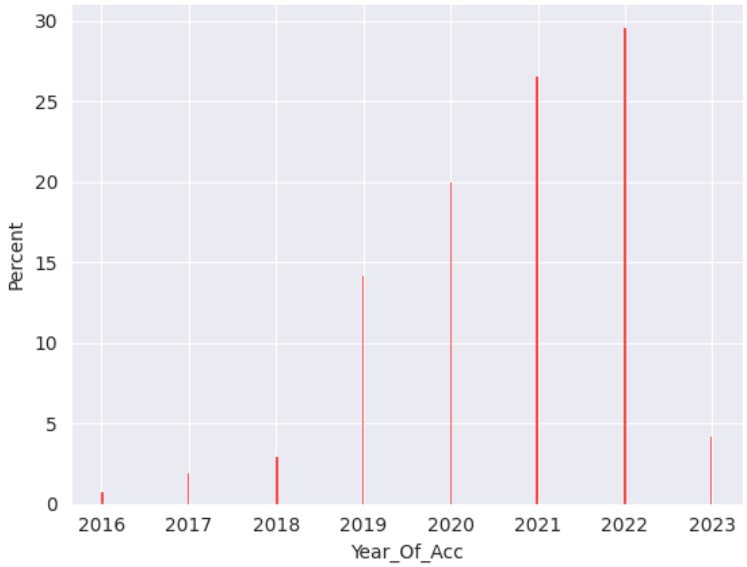
1. Extract the month from the "Start_Time" column
2. Plot histogram of the 'Day_Of_Week' column



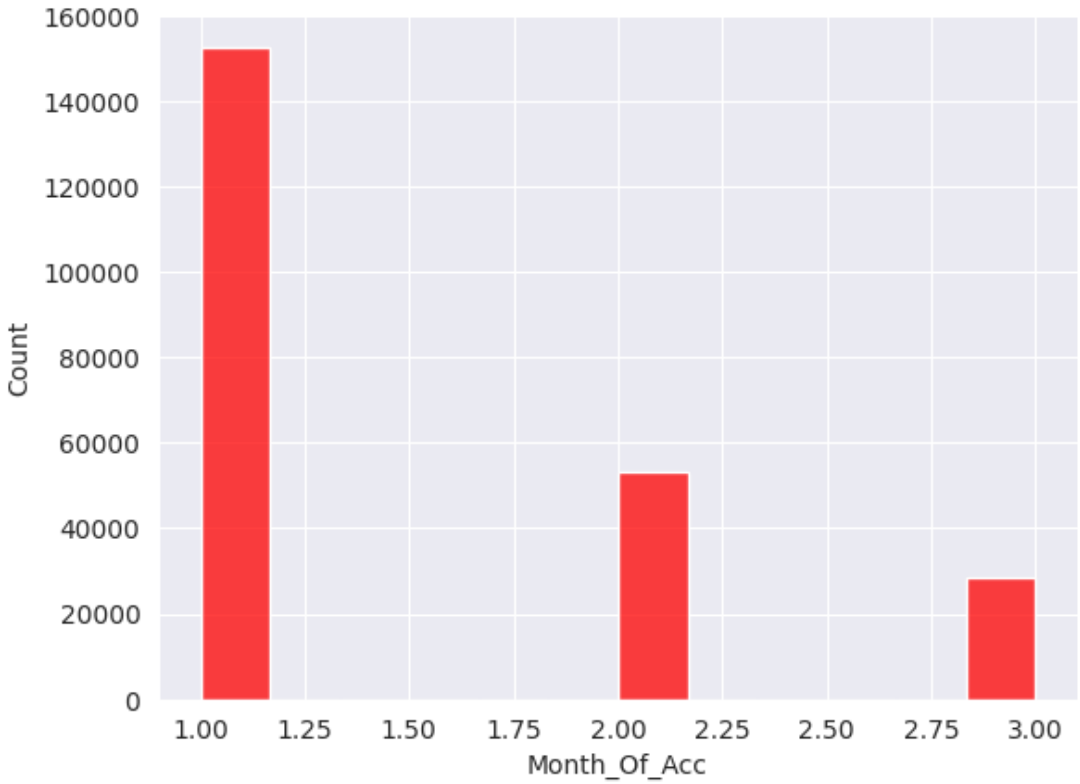
Initially, there is a high number of accidents, which gradually decreases as the months progress. However, towards the end of the year, there is a slight increase in the number of accidents again.

v. Years:

1. Repeat as above we get:

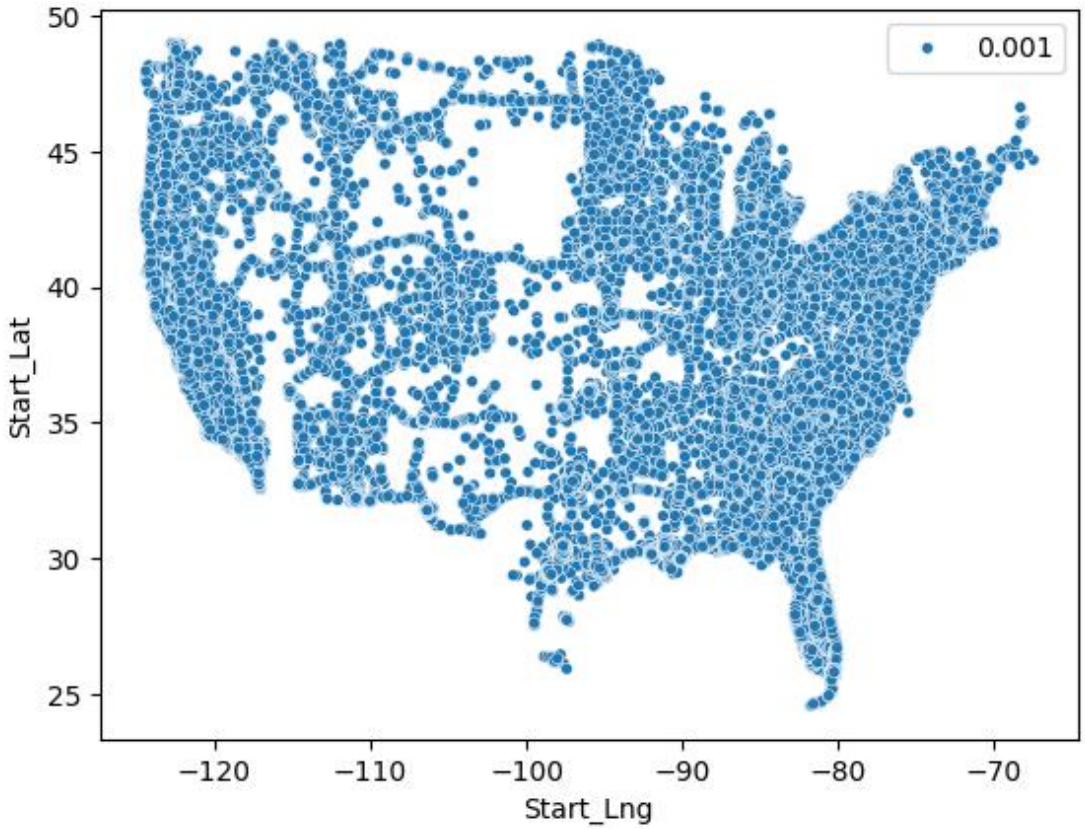


From the graph we can analyze that accidents are increasing every year but we can see 2023 have the lowest accident recorded. Thats really unusual. We need to see the data of 2023.



The reason for the seemingly lower number of accidents in 2023 is primarily due to data availability. The dataset used for analysis only includes data up to March 2023, which means it doesn't cover the entire year. As a result, the observed decrease in accidents for 2023 may not accurately reflect the true accident rate for the entire year, as data for the later months is missing. Therefore, any conclusions drawn about accident trends in 2023 should be approached with caution, keeping in mind the incomplete data for that year.

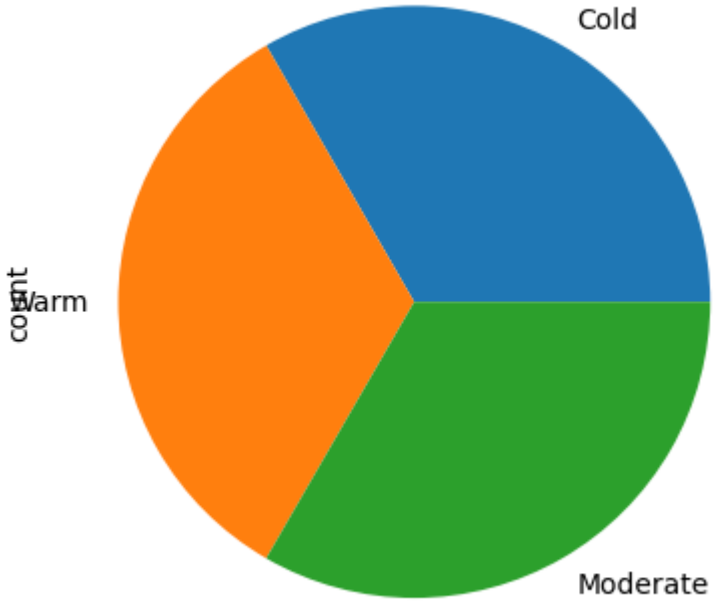
d. Start Latitude and Start Langitude
    i. We plot a scatter plot to distribute the locations of accidents geographically:



e. Analyzing **weather** columns:
    i. ### 1.Are there more accidents in warmer or colder areas?
        1. Define temperature bins and labels
        2. Use 'when' function to assign temperature categories based on the bins
        3. Add the temperature category column to the DataFrame
        4. Group by temperature category and count the number of accidents
        5. Results

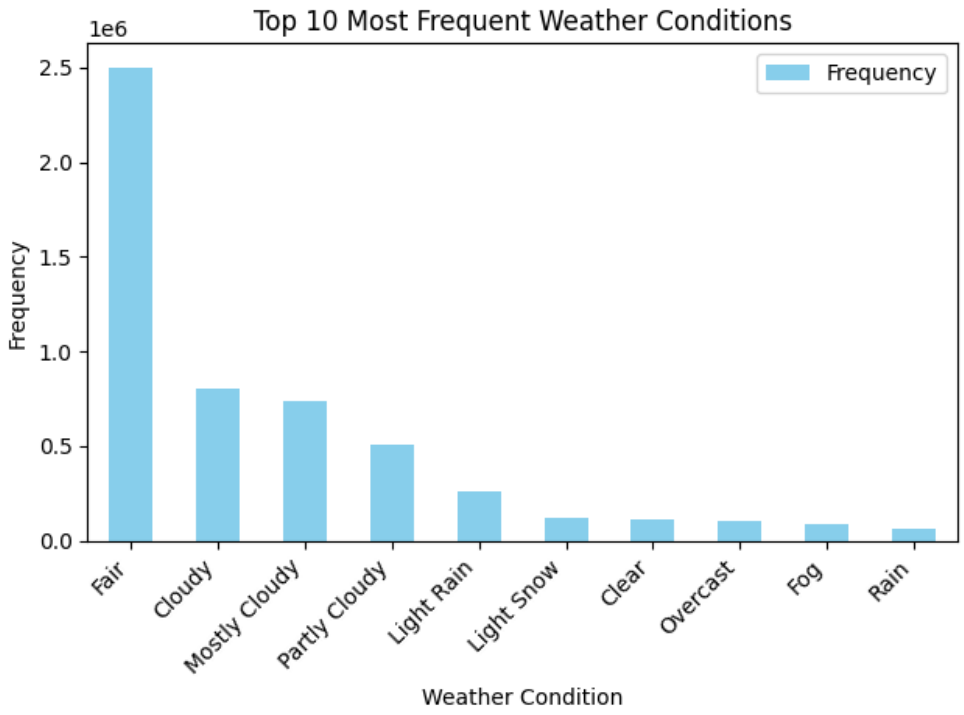
```
+--------------------+---------+
|Temperature_Category|Accidents|
+--------------------+---------+
|                Cold|  1800430|
|                Warm|  1393745|
|            Moderate|  2462664|
+--------------------+---------+
```

6. Convert the result to Pandas DataFrame for plotting

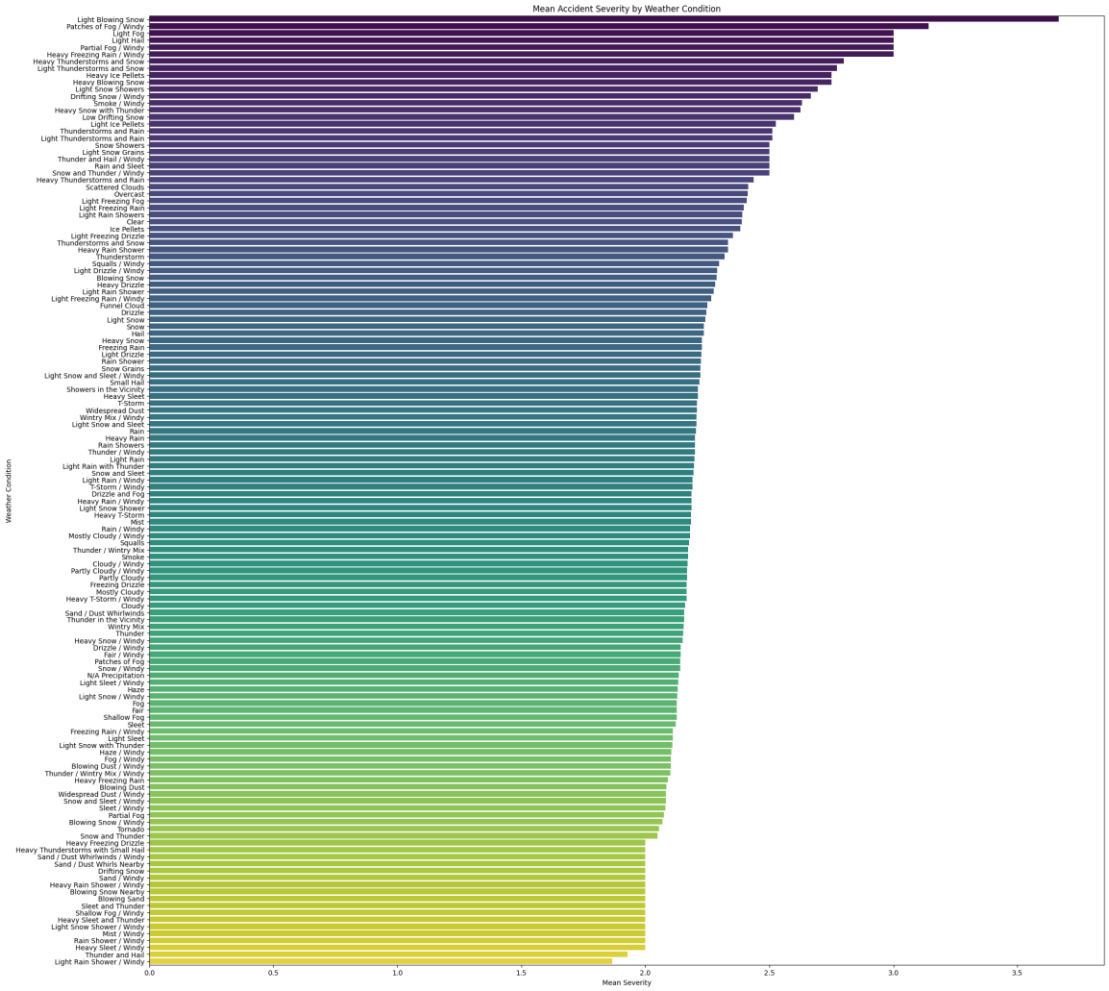


ii. Frequency of Weather Conditions

1. Group by weather condition and count the frequency of each condition
2. Get top 10 most frequent weather conditions:



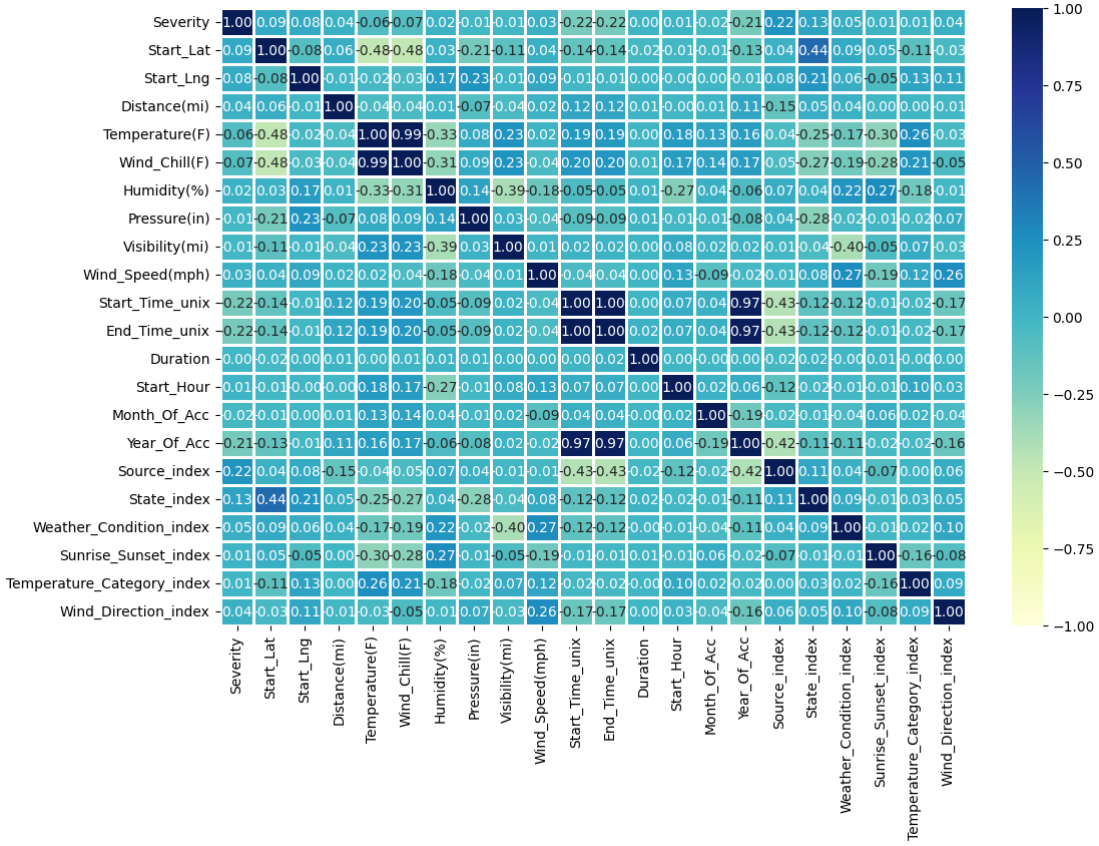
3. Impact of Weather on Accident Severity
    a. Group data by weather condition and calculate mean severity
    b. Plot the values:



Light blowing snow is a weather condition characterized by light snowfall accompanied by strong winds that cause the snow to be lifted from the ground and blown around. This weather condition can significantly reduce visibility and create hazardous driving conditions. Overall, it poses significant hazards to drivers and can contribute to the severity of accidents by reducing visibility, creating slippery road conditions, and increasing the likelihood of collisions.

4. Word cloud for the Description column

5. Plotting the correlation matrix for the features



3. Extracting insights from data

📄🌟 Here is a summary & conclusion of the EDA phase insights we got:

1. The dataset only includes data for 49 states.
2. Miami has the highest number of reported accidents.
3. The frequency of accidents per city follows an exponential decrease pattern.
4. Only 13.5% of cities have more accidents than the average.
5. Accidents are most common between 8 am to 10 am and 3 pm to 6 pm, suggesting a higher likelihood during peak commuting hours.
6. Weekdays show a higher number of accidents compared to weekends.
7. The year 2023 has the lowest number of reported accidents, likely due to data availability only until March 2023.
8. Coastal areas experience higher accident rates compared to inland regions.
9. The top 10 most frequent weather conditions for accidents are: Fair, Mostly Cloudy, Cloudy, Clear, Partly Cloudy, Overcast, Light Rain, Scattered Clouds, Light Snow, and Fog.
10. Analyzing the relationship between weather conditions and accident severity revealed the following insights:
    - Light Blowing Snow has the highest average severity at 3.67.
    - Patches of Fog / Windy, Heavy Freezing Rain / Windy, and Light Fog have average severities of 3.14, 3.00, and 3.00, respectively.
    - Partial Fog / Windy, Heavy Thunderstorms and Snow, Light Thunderstorms and Snow, Heavy Ice Pellets, Heavy Blowing Snow, and Drifting Snow / Windy also show significant average severity levels.

4. Model/classifier training

0. Data preprocessing :
    a. Removing unimportnant columns
    b. Renaming the columns
    c. Convert the values of the columns into easy handled value
        i. EX: Convert "Severity" column from string to integer
    d. We have dropped about 2 millions records because of nulls :)
    e. Create a new column called Duration = End_Time - Start_Time
    f. Use 'when' function to assign temperature categories based on the bins, add to the dataframe
    g. Convert "Distance(mi)" column from miles to meters
    h. The columns were:

```
['ID',
 'Source',
 'Severity',
 'Start_Time',
 'End_Time',
 'Start_Lat',
 'Start_Lng',
 'Description',
 'Street',
 'City',
 'County',
 'State',
 'Temperature(F)',
 'Wind_Chill(F)',
 'Humidity(%)',
 'Pressure(in)',
 'Visibility(mi)',
 'Wind_Direction',
 'Wind_Speed(mph)',
 'Weather_Condition',
 'Crossing',
 'Railway',
 'Station',
 'Traffic_Calming',
 'Traffic_Signal',
 'Sunrise_Sunset',
 'Distance(m)',
 'Start_Time_unix',
 'End_Time_unix',
 'Duration']
```

    i. Convert string columns to numerical representations
    j. Assemble features into 1 feature vector
    k. Split the data into training and testing sets

1.[Regression]➔ Predicting accident Duration as indicator of impact on traffic flow.

Experiment failed as we noticed after the results that most of the features are uncorrelated with the target variable 'Duration' we added earlier, resulting in an MSE of abot 9 millions 😬 This is one of out failed and oh-woaah experiments :)

2. [classification]➔ Predicting the severity of an accident based on the factors involved.

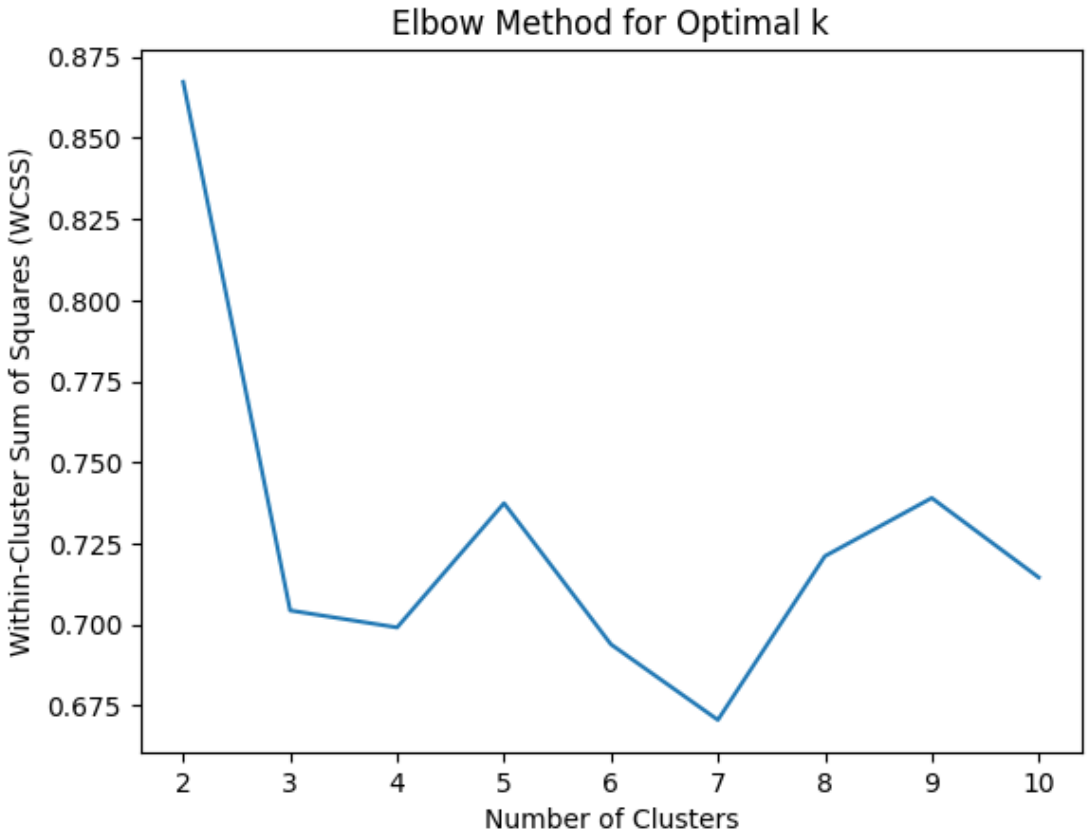
```
### Based on the EDA phase we will use the following features in our model:



### Note:

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).
```

Applied the following models:

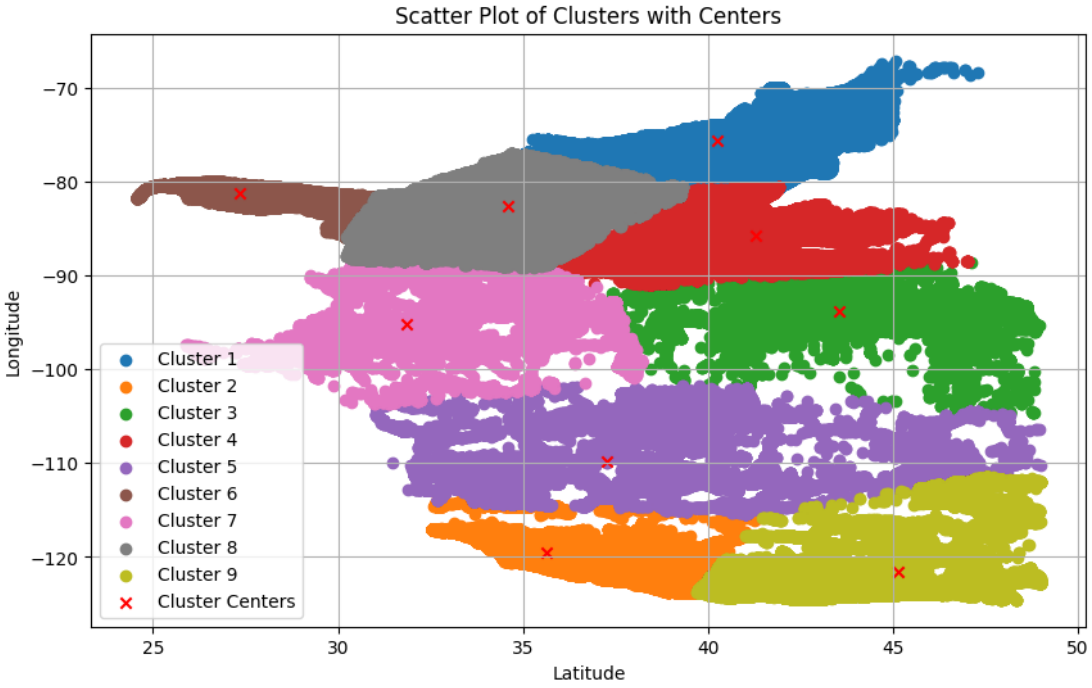
a. Logistic regression [ready & implemented]
b. Random forest
c. Naïve bayes

1. Predictive analytics
    a. ➔ Predicting the severity of an accident:
        i. Logistic regression [Ready & implemented]
    b. ➔ Predicting the duration of an accident:
        i. Linear regression [Ready & implemented]
2. Descriptive analytics:
    a. Clustering
        i. intro
            1. In the context of clustering accident locations based on latitude information, clusters represent groups of accident locations that have similar latitude coordinates.
            2. When applying clustering algorithm like K-means to accident data, the algorithm partitions the accident locations into a predefined number of clusters, with each cluster representing a group of locations that are close to each other in terms of latitude and longitude.
            3. The meaning of these clusters can vary depending on factors such as geographical characteristics, traffic patterns, road conditions, or other contextual information. However, generally, clusters can help identify areas where accidents are more concentrated

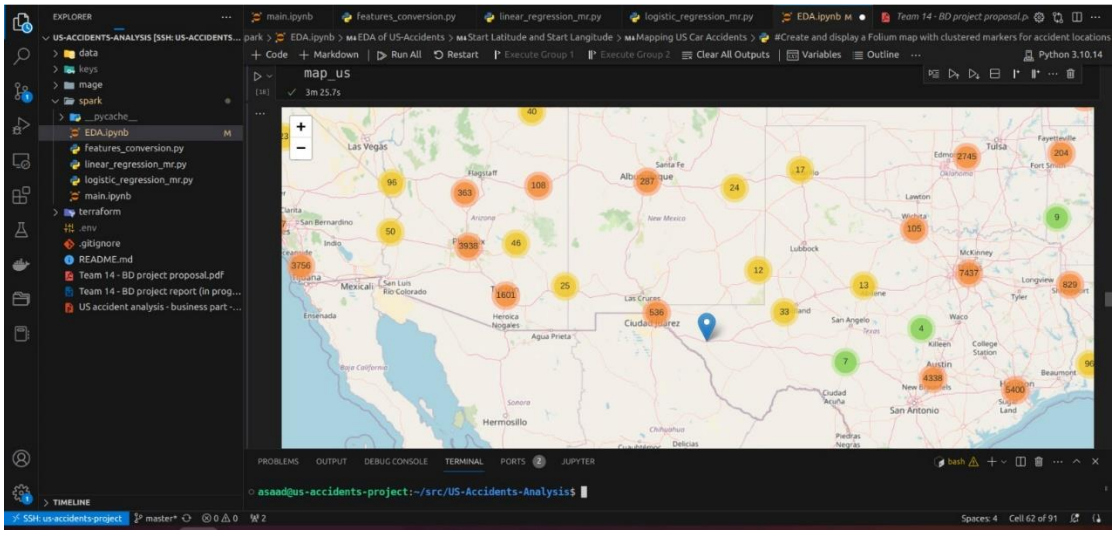
ii. Choosing optimal K [number of clusters] based one
iii. Look for a point where the silhouette score stops increasing significantly with increasing k. This point typically resembles an elbow shape in the plot, hence the name "elbow method."



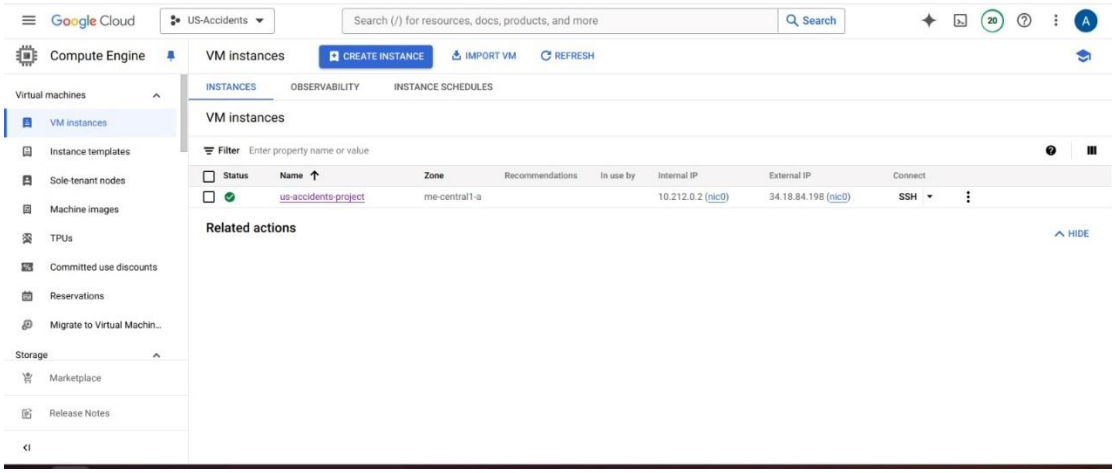
iv. Apply K mean on the chosen K

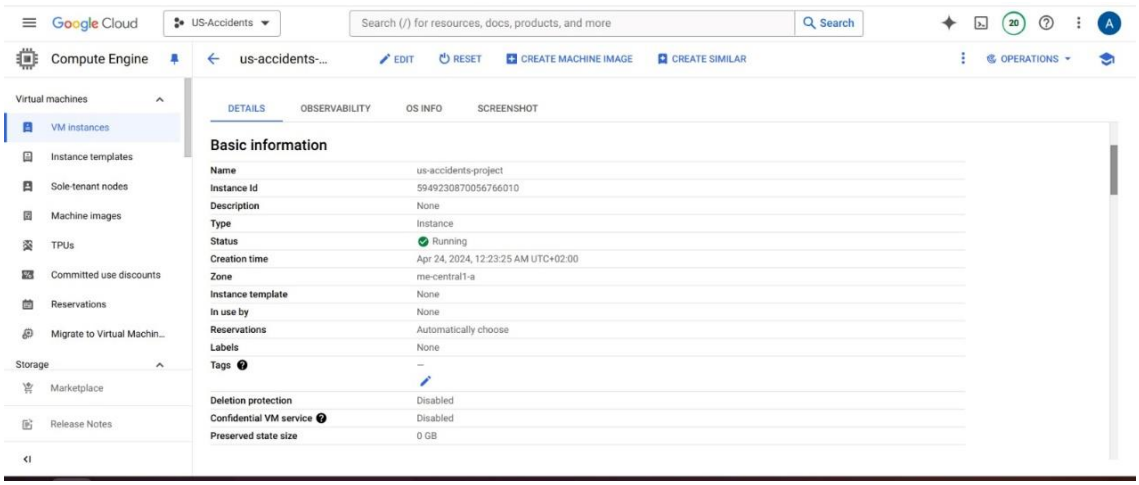


b. Association Rules
    i. We tried to look for a column to apply the association rules on , but after some hours we didn't find any suitable column
c. Map visualizations (note , my mate's electricity went off before manging to upload it , but this is a screenshot of it on his device

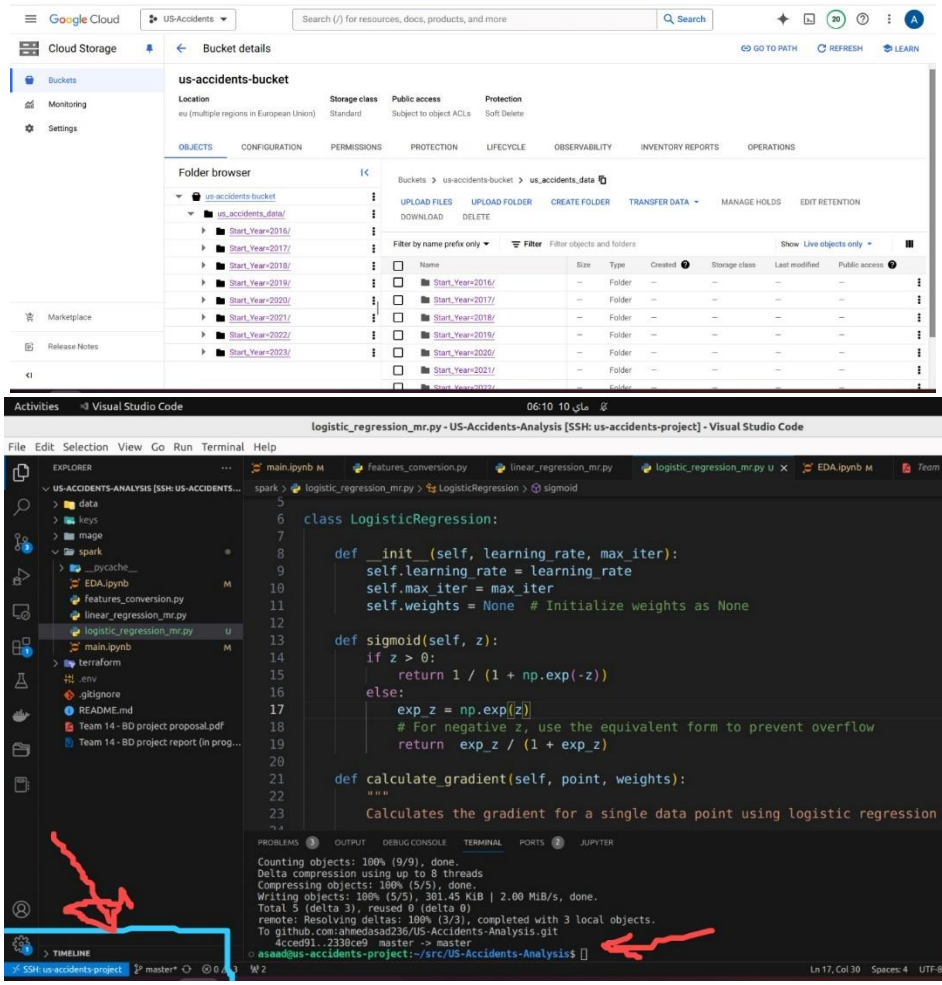


Bonus : The cloud we used : GCP

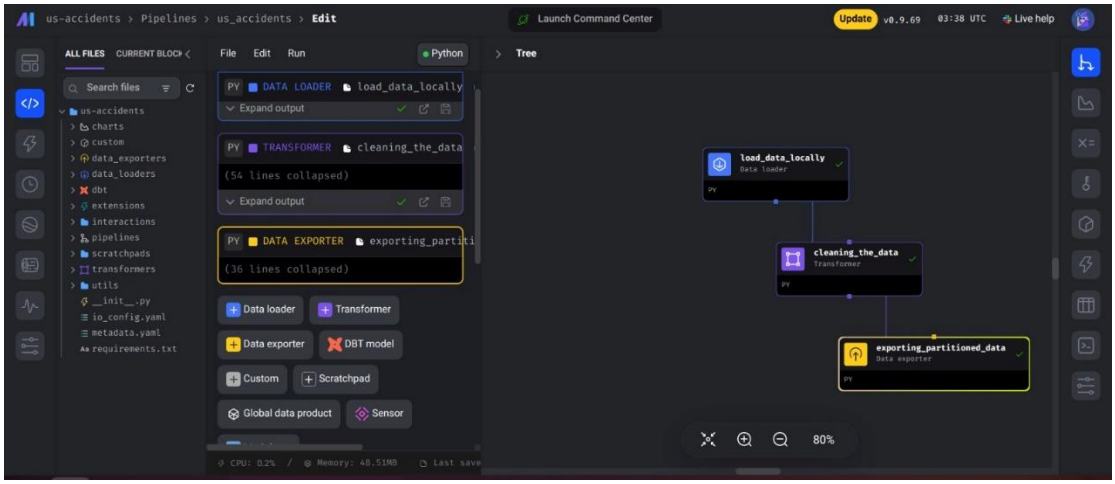




Used **terraform** as *IAC* (infrastructure as Code)➔ to create our resources on GCS



We used a remote VM and accessed it through VScode SSH connection



Use **Mage** for data pipeline Orchestration:

- Load the data from local after downloading from Kaggle
- Apply some cleaning and column removing
- Partition the data into 7 parquet files and export them to GCS (Google Cloud Storage)

Results & Evaluation

Model	Accuracy	F1 – Score
Logistic regression [Ready made]	0.8408217891358007	0.7685677937765466
Logistic regression [Student made]	Errors occurred we didn't manage to solve on time	
Random Forest	0.8406899831395097	0.7680208605030131
Naïve Bayes	0.0114178970079841	0.0002577932869698

Model	RMSE
Linear regression [ready]	906596 [Too large]
Linear regression [student made]	Incompatible with dataframes [needed RDD] , also regression was bad in our case ,so we switched for the logistic and implemented it.

Trials [including unsuccessful ones]

a. At first , VM resources weren't sufficient .
    a. # of CPUs
    b. Memory
b. Spark setup on the VM.
c. Spark connection to GCP.
d. EDA
e. Map-reduce implementation we searched for resources but were scarce.
f. In predictive analysis:
    a. We first started by using pyspark's logistic regression to predict the severity of an accident according to some columns based on our EDA phase. We got 100% accuracy ! why? Because the majority of the values are of class 2 as we showed in the EDA phase.
    b. When we tweaked some modifications in our code , the model ran a bit logical as shown in the table above
    c. Then Used other models , some failed and some worked
        i. Worked:
            1. Random forest
            2. Naïve bayes [ran successfully But failed in accuracy 1.1%]
        ii. Didn't work – error [compilation – logical]
            1. SVM
            2. XGBoost
            3. … don't remember
    d. **Regression**:
        i. At first we applied the linear regression that was ready from library , but the RMS was extremely high 906596 , so we revised the correlation between the column we created and the features to find out the were unexpectedly uncorrelated as seen below:

So we decided it was a bad idea to keep going into this path and stopped pursuing the regression problem after many hours of trials ^_^

g. In Descriptive analysis:
    a. We applied clustering as we wrote in the proposal on the locations of the accident to see where they occurred on the map as clusters using the longitude-latitude info , as it was the nearest thing in our head that we can apply clustering on
    b. Association Rules
        i. We tried to look for a column to apply the association rules on , but after some hours we didn't find any suitable column, so we cancelled looking for it and moved to clustering.
    c. Also , we described the map based on the severity of the accidents and put this information on an interactive map
h. Map reduce student implementations:
    a. At first , we implemented the linear regression in map reduce.
        i. As we mentioned above we stopped pursuing the reg it was incompatible with our notebook flow when integrated with my mate
        ii. After that , we decided to implement logistic regression with map reduce and pyspark vectorized implementation, and it worked , but there was a logical bug as we implemented it mistakenly as binary classification , but we needed multiclass , didn't manage to fix it on time .

Enhancements & fu

الحمد لله الذي بنعمته تتم الصالحات 💜