A Project Report On

US Accidents Analysis and Predictions

Submitted By

Piyush Dewangan
(220940325047)

Mamta Basawaraj Sajjanshetty
(220940325040)

Prafull Dilip Shinde
(220940325049)

Sahadev Shrirang Musale
(220940325059)

Kushal Prakash Wade
(220940325037)

In partial fulfillment of
the requirements for the award of the degree of

Post Graduate Diploma
In
Big Data
Analytics
(PG-DBDA)

Year 2022 - 2023

ABSTRACT

The US road accident analysis and prediction study aims to analyze and predict the trends and patterns of road accidents in the US between 2016-2022. The study will collect and analyze data on the number of accidents, fatalities, injuries, and the factors that contribute to accidents.
The analysis will be conducted using statistical models and machine learning algorithms to identify the underlying patterns and predict the future trends. The study will also examine the effectiveness of existing safety measures and propose new strategies to reduce the number of accidents and fatalities.
The results of the study will be valuable for policymakers, traffic safety professionals, and the general public in understanding the dynamics of road accidents in the US and developing effective strategies to prevent them.

Contents

1.Introduction:

In this project we downloaded the 'US Accident Dataset(2016 June-2021 dec)' of all states in the USA. We used Python libraries and SparkSQL. By using python libraries and ML libraries we have done EDA and built ML models and then used PySpark for data analysis. By analyzing the data we gathered some insights about it and we have described the same in this project report. For visualizing the same analyzed data we used Microsoft Power BI Desktop and created interactive dashboards. We also did prediction using the same dataset. For that we used various machine learning algorithms available in python libraries. The predicted data and information are described in the report.

2.Introduction to Dataset:

This is a countrywide car accident dataset, which covers 49 states of the USA. The accident data are collected from February 2016 to Dec 2021, using multiple APIs that provide streaming traffic incident (or event) data. These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Currently, there are about 2.8 million accident records in this dataset.

1.3 Problem Statement:

To develop a predictive model that can accurately forecast the likelihood of road accidents occurring in the United States, using a dataset of historical road accident records.

This model should take into account various factors that may contribute to accidents, such as weather conditions, Severity, City, and Side.

Additionally, the analysis of the dataset should reveal insights and patterns that can reduce the number of accidents on US roads.

Our goal is to use exploratory analysis and machine learning models to predict and analyze cause of accidents so that will help in reducing no. of accidents.

1.4 Description

This is a countrywide car accident dataset, which covers **49 states of the USA**. The accident data are collected from **February 2016 to Dec 2021**, using multiple APIs that provide streaming traffic incident (or event) data. These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law

enforcement agencies, traffic cameras, and traffic sensors within the road networks. Currently, there are about **2.8 million** accident records in this dataset.

1.5 Objectives

- Analyzing the data according to state base accident in USA
- Analyzing the accident based on country, district and town
- Visualizing the data for better understanding
- Creating interactive dashboards
- Predicting the future data

Chapter II : BIG DATA ANALYTICS

2.1 Data Collection

2.1.1 Dataset Source:

Kaggle website

2.2 Introduction to Spark SQL:

Spark is an analytics engine that is used by data scientists all over the world for Big Data Processing. It is built on top of Hadoop and can process batch as well as streaming data.

There are several operations that can be performed on the Spark DataFrame using DataFrame APIs. It allows us to perform various transformations using various rows and columns from the Spark DataFrame. We can also perform aggregation and windowing operations.

2.3Analysis using Spark SQL

Creating Spark Session

```
from pyspark.sql import SparkSession
from pyspark.sql.types import *
```

```
spark= SparkSession.builder.appName("test_pyspark").getOrCreate()
```

```
from pyspark.sql.types import StringType, StringType, IntegerType, DoubleType, DataType, LongType
```

Creating schema:

```
schema = StructType([
    StructField('ID', StringType(), True),
    StructField('Severity',IntegerType(), True),
    StructField('Side',StringType(), True),
    StructField('City',StringType(), True),
    StructField('County',StringType(), True),
    StructField('State',StringType(), True),
    StructField('Description',StringType(), True),
    StructField('Start_Time',DateType(), True),
    StructField('Weather_Timestamp',DateType(), True),
    StructField('Temperature(F)',DoubleType(), True),
    StructField('End_Time',DateType(), True),
    StructField('Weather_Condition',StringType(), True)
])
```

Loading data to data frame:

```
df = pd.read_csv(path)
```

```
df_US=df[['ID','Severity','Side','City','County','State','Description','Start_Time','Weather_Timestamp','Temperature(F)','End_Time','Weather_Condition']].copy()
```

```
df_US.to_csv("sorted_data.csv",index=False)
```

```
df_US1=spark.read.format("csv").option("header","True").schema(schema).load('/content/sorted_data.csv')
```

```
df = spark.read.format("csv").option("header", "true").option("inferSchema", "true").load("/content/sorted_data.csv") \
    .select("ID", "Severity", "Side", "City","County","State","Description","Start_Time","Weather_Timestamp","Temperature(F)","End_Time","Weather_Condition")
```

```
df_US1= df_US1.withColumn("Date", to_date("Start_Time"))
```

```
df_US1 = df_US1.withColumn("DayOfWeek", date_format("Date", "E"))
```

```
df_US1 = df_US1.withColumn("Month", month("Date"))
```

```
df_US1 = df_US1.withColumn("Year", year("Date"))
```

```
df_US1= df_US1.withColumn("Timestamp", to_timestamp("Start_Time", "MM/dd/yyyy hh:mm:ss a"))
```

```
df_US1 = df_US1.withColumn("HourOfDay", hour("Timestamp"))
```

Registering table:

For performing SQL queries on a schema we have to create a temporary table. This temporary table is automatically dropped after the session ends. To register a 'temp table' –

```
df_US1.registerTempTable("US")
```

After registering the temp table we can perform the SQL queries on the table. We had to answer some business questions listed below. We have answered the first 10 of them using Spark SQL.

```
spark.sql("select * from US").show()
```

```
spark.sql("select * from US").show()
```

```
+---+--------+----+-----------+---------+-----+--------------------+---------+-----------------+-------------+---------+-----------------+---------+---------+-----+----+--------
| ID|Severity|Side|       City|   County|State|         Description|Start_Time|Weather_Timestamp|Temperature(F)|  End_Time|Weather_Condition|     Date|DayOfWeek|Month|Year|
+---+--------+----+-----------+---------+-----+--------------------+---------+-----------------+-------------+---------+-----------------+---------+---------+-----+----+--------
|A-1|       3|   R|     Dublin| Franklin|   OH|Between Sawmill R...|2016-02-08|       2016-02-08|         42.1|2016-02-08|       Light Rain|2016-02-08|      Mon|    2|2016|2016-02-
|A-2|       2|   R|    Dayton|Montgomery|   OH|At OH-4/OH-235/Ex...|2016-02-08|       2016-02-08|         36.9|2016-02-08|       Light Rain|2016-02-08|      Mon|    2|2016|2016-02-
|A-3|       2|   R| Cincinnati| Hamilton|   OH|At I-71/US-50/Exi...|2016-02-08|       2016-02-08|         36.0|2016-02-08|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-4|       2|   R|      Akron|   Summit|   OH|At Dart Ave/Exit ...|2016-02-08|       2016-02-08|         39.0|2016-02-08|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-5|       3|   R| Cincinnati| Hamilton|   OH|At Mitchell Ave/E...|2016-02-08|       2016-02-08|         37.0|2016-02-08|       Light Rain|2016-02-08|      Mon|    2|2016|2016-02-
|A-6|       2|   R|Williamsburg| Clermont|   OH|At Dela Palma Rd ...|2016-02-08|       2016-02-08|         35.6|2016-02-08|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-7|       2|   R|    Dayton|Montgomery|   OH|At OH-4/Exit 54 -...|2016-02-08|       2016-02-08|         33.8|2016-02-08|    Mostly Cloudy|2016-02-08|      Mon|    2|2016|2016-02-
|A-8|       2|   R|  Cleveland|  Cuyahoga|   OH|At Bagley Rd/Exit...|2016-02-08|       2016-02-08|         33.1|2016-02-08|            Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-9|       2|   R|       Lima|    Allen|   OH|At OH-65/Exit 122...|2016-02-08|       2016-02-08|         39.0|2016-02-08|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-10|      2|   R| Westerville| Franklin|   OH|At I-71/Exit 26 -...|2016-02-08|       2016-02-08|         32.0|2016-02-08|            Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-11|      2|   R| Cincinnati| Hamilton|   OH|At OH-4/Paddock R...|2016-02-08|       2016-02-08|         33.8|2016-02-08|       Light Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-12|      2|   R| Cincinnati| Hamilton|   OH|At US-52/Hopple S...|2016-02-08|       2016-02-08|         35.1|2016-02-08|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-13|      2|   R|  Cleveland|  Cuyahoga|   OH|At US-42/Exit 170...|2016-02-08|       2016-02-08|         33.1|2016-02-08|       Light Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-14|      2|   R|  Jamestown|   Greene|   OH|Between OH-72/Exi...|2016-02-08|       2016-02-08|         33.8|2016-02-08|       Light Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-15|      3|   R|   Freeport| Guernsey|   OH|At Shipley Rd - A...|2016-02-08|       2016-02-08|         33.1|2016-02-09|    Mostly Cloudy|2016-02-08|      Mon|    2|2016|2016-02-
|A-16|      3|   L|   Freeport| Harrison|   OH|At Titus Rd - Acc...|2016-02-08|       2016-02-08|         33.1|2016-02-09|    Mostly Cloudy|2016-02-08|      Mon|    2|2016|2016-02-
|A-17|      3|   R|   Columbus| Franklin|   OH|At OH-16/Broad St...|2016-02-08|       2016-02-08|         34.0|2016-02-09|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-18|      2|   R|   Columbus| Franklin|   OH|At I-270 - Accident.|2016-02-08|       2016-02-08|         34.0|2016-02-09|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
|A-19|      3|   R|   Columbus| Franklin|   OH|Between Weber Rd/...|2016-02-08|       2016-02-08|         33.8|2016-02-09|       Light Snow|2016-02-08|      Mon|    2|2016|2016-02-
|A-20|      4|   R|     Toledo|    Lucas|   OH|Closed between I-...|2016-02-08|       2016-02-08|         33.4|2016-02-09|        Overcast|2016-02-08|      Mon|    2|2016|2016-02-
+---+--------+----+-----------+---------+-----+--------------------+---------+-----------------+-------------+---------+-----------------+---------+---------+-----+----+--------
```

Q1)How many unique cities in accident dataset

```
num_cities = df_US1.select('City').distinct().count()
print("Number of cities in the accident dataset:", num_cities)
```

```
Number of cities in the accident dataset: 11682
```

There are 11682 unique cities in the accident dataset.

Q2)How many unique states in accident dataset

```
num_states = df_US1.select('State').distinct().count()
print("Number of states in the accident dataset:", num_states)
```

```
Number of states in the accident dataset: 49
```

There are a total of 49 states in the US accident dataset.

Q3)Find the common causes of the accident

```
most_common_causes = spark.sql("""
    SELECT `Description`, COUNT(*) AS `count`
    FROM US
    WHERE `Description` IS NOT NULL
    GROUP BY `Description`
    ORDER BY `count` DESC
""").limit(10).collect()
```

```
most_common_causes
```

```
[Row(Description='A crash has occurred causing no to minimum delays. Use caution.', count=7978),
 Row(Description='A crash has occurred use caution.', count=2531),
 Row(Description='An unconfirmed report of a crash has been received. Use caution.', count=2308),
 Row(Description='Hazardous debris is causing no to minimum delays. Use caution.', count=2095),
 Row(Description='At I-15 - Accident.', count=2070),
 Row(Description='A disabled vehicle is creating a hazard causing no to minimum delays. Use caution.', count=1912),
 Row(Description='At I-5 - Accident.', count=1907),
 Row(Description='At I-405/San Diego Fwy - Accident.', count=1769),
 Row(Description='At I-605 - Accident.', count=1486),
 Row(Description='Incident on I-95 NB near I-95 Drive with caution.', count=1304)]
```

Q4)find the type of accidents that most occurred in terms of severity

```
spark.sql("""
    SELECT `Severity`, `Description`, COUNT(*) AS `count`
    FROM US
    WHERE `Description` IS NOT NULL
    GROUP BY `Severity`, `Description`
    ORDER BY `count` DESC
""").limit(10).collect()
```

```
[Row(Severity=2, Description='A crash has occurred causing no to minimum delays. Use caution.', count=7978),
 Row(Severity=2, Description='A crash has occurred use caution.', count=2531),
 Row(Severity=2, Description='An unconfirmed report of a crash has been received. Use caution.', count=2308),
 Row(Severity=2, Description='Hazardous debris is causing no to minimum delays. Use caution.', count=2095),
 Row(Severity=2, Description='At I-15 - Accident.', count=2022),
 Row(Severity=2, Description='A disabled vehicle is creating a hazard causing no to minimum delays. Use caution.', count=1912),
 Row(Severity=2, Description='At I-5 - Accident.', count=1814),
 Row(Severity=2, Description='At I-405/San Diego Fwy - Accident.', count=1730),
 Row(Severity=2, Description='At I-605 - Accident.', count=1434),
 Row(Severity=2, Description='Incident on I-95 NB near I-95 Drive with caution.', count=1304)]
```

```
severity_counts = df_US1.groupBy('Severity').count()
```

```
most_frequent_severity = severity_counts.orderBy(severity_counts['count'].d
esc())
most_frequent_severity.show()
```

Severity	count
2	2532991
3	155105
4	131193
1	26053

Q5)Which 5 states have the highest number of accidents?

```
result=spark.sql("SELECT State, COUNT(*) as AccidentCount FROM US GROUP BY
State ORDER BY AccidentCount DESC LIMIT 5")
result.show()
```

State	AccidentCount
CA	795868
FL	401388
TX	149037
OR	126341
VA	113535

california, Florida, Texas, Oregon and Virginia are the top 5 states have the highest accident rate.

Q6)Which days of the week have the most accidents?

```
result = spark.sql("SELECT DayOfWeek, COUNT(*) as AccidentCount FROM US GRO
UP BY DayOfWeek ORDER BY AccidentCount DESC")
result.show()
```

DayOfWeek	AccidentCount
Fri	492074
Thu	463477
Wed	455037
Tue	443968
Mon	419821
Sat	311691
Sun	259274

Friday has the highest accident number which is 492074.

Q7)Which months have the most accidents?

```
result = spark.sql("SELECT Month, COUNT(*) as AccidentCount FROM US GROUP B
Y Month ORDER BY AccidentCount DESC")
result.show()
```

```
+-----+-------------+
|Month|AccidentCount|
+-----+-------------+
|   12|       473943|
|   11|       360696|
|   10|       299131|
|    9|       241822|
|    6|       226561|
|    1|       198365|
|    2|       194995|
|    5|       181944|
|    8|       178670|
|    4|       171880|
|    7|       159111|
|    3|       158224|
+-----+-------------+
```

The last 4 months of the year have the highest rate of accidents.

Q8)What is the trend of accidents year over year (decreasing/increasing)?

```
result = spark.sql("SELECT Year, COUNT(*) as AccidentCount FROM US GROUP BY
 Year ORDER BY Year ASC")
result.show()
```

```
+----+-------------+
|Year|AccidentCount|
+----+-------------+
|2016|       122024|
|2017|       163918|
|2018|       163176|
|2019|       258615|
|2020|       625864|
|2021|      1511745|
+----+-------------+
```

It is exponentially increasing year on year.

Q9)At what time of the day are the accidents most frequent?

```
result = df.groupBy("Start_Time").count()

result = result.orderBy("count", ascending=False)
```

```
result.show()
```

```
+-------------------+-----+
|         Start_Time|count|
+-------------------+-----+
|2021-01-26 16:16:13|  252|
|2021-01-26 16:17:33|  170|
|2021-02-16 06:42:43|  157|
|2021-05-03 06:29:42|  111|
|2021-11-21 18:37:51|  102|
|2021-02-16 06:43:35|   97|
|2021-04-26 08:58:47|   95|
|2021-04-14 13:51:30|   92|
|2020-12-16 13:53:25|   82|
|2021-05-03 06:30:28|   75|
|2021-12-07 14:16:30|   72|
|2021-03-22 06:30:42|   71|
|2021-12-14 10:39:00|   71|
|2021-03-15 06:25:17|   70|
|2021-02-10 14:26:11|   69|
|2021-12-16 14:11:00|   63|
|2020-10-12 11:13:30|   62|
|2020-09-30 12:41:30|   61|
|2021-05-03 06:31:30|   60|
|2017-05-15 09:22:55|   59|
+-------------------+-----+
```

At Afternoon near 4 pm there was a high accident rate.

Q10) Which city has the highest accident rate in the US dataset?

```
query = """
        SELECT City, COUNT(*) as TotalAccidents
        FROM US
        GROUP BY City
        ORDER BY TotalAccidents DESC
        LIMIT 1
        """

# execute the query and display the result
```

```
result = spark.sql(query)

result.show()
```

```
+-----+--------------+
| City|TotalAccidents|
+-----+--------------+
|Miami|        106966|
+-----+--------------+
```

Highest accident occur in Miami City

Chapter III : DATA VISUALIZATION

3.1 Introduction to Power BI

Power BI is a Data Visualization and Business Intelligence tool that converts data from different data sources to interactive dashboards and BI reports. Power BI suite provides multiple software, connector, and services - Power BI desktop, Power BI service based on SaaS, and mobile Power BI apps available for different platforms. These sets of services are used by business users to consume data and build BI reports.

Why is visualization important because with the technology revolution data went from expensive, difficult to find and to abundant, cheap data to store, understand and analyze the data. Microsoft Power BI is a data visualization tool that allows you to quickly connect your data, prepare it, and model it as you like. It gives a professional environment allowing you to acquire analytics and reporting capabilities. You can publish these reports, therefore allowing all the users to avail the latest information. It gives you the power to transform all your data into live interactive visuals, create customized real time business view dashboards, thus extracting business intelligence for enhanced decision making
Here we are using the clustered columns chart ,donut chart , card , map and slicer

3.2 Importing Data into Power BI

To create a dashboard in Power BI, you need to add all data sources in Power BI new . To add a data source, go to the Get data option. Then, select the data source you want to connect and click the Connect button.

3.3 Visuals used for Dashboard

We used the following visuals for visualization.

Clustered column chart : Clustered Bar charts display values (i.e. measures) where the length of the bar or column is proportional to the data. Here we have displayed the count of id by weather conditions.

Maps: You can also use a map to view your sales in different places. Choose the globe icon from the visualization panel. Here we have displayed all the states included in the dataset as per their longitude and latitude positions.

Pie Charts: Pie charts are quite well known when it comes to displaying a lot of data in divided percentage format. Here we have displayed the count of id by wind direction.

Stacked column chart : Stacked column charts display values (i.e. measures) of multiple cases stacked in a single column where the length of the bar or column is proportional to the data. Here we have

created a separate column named ‘SEVERE’ and have displayed the cities by the count of id in two separate graphs.

Line Charts: Let’s visualize our data using a line chart. Select the line chart from the visualization pane and add values for time (day/night) and side (right/left/neutral).

Small Multiples : Whenever we put multiple parameters in a single graph, we can use small multiples. Here we have taken traffic conditions into account so that the count of accidents can be bifurcated as per the given bar graphs.

Ribbon Chart : This chart is most famous for understanding the percentage increase/decrease in the count for two independent situations. For locations, we considered roundabout, railway and junction situations to check their impact on the total count of accidents.

Area Graph: This graph shows the net area which gives exact count of the data provided in y axis. We have used four different area graphs to check the relation of total counts with various atmospheric conditions. Visibility, precipitation, wind speed and humidity are the factors taken into the consideration to understand their impact on the total count.

3.4 Dashboard state and accident severity

3.4.1 Insights gathered from the dashboard on states and severity:

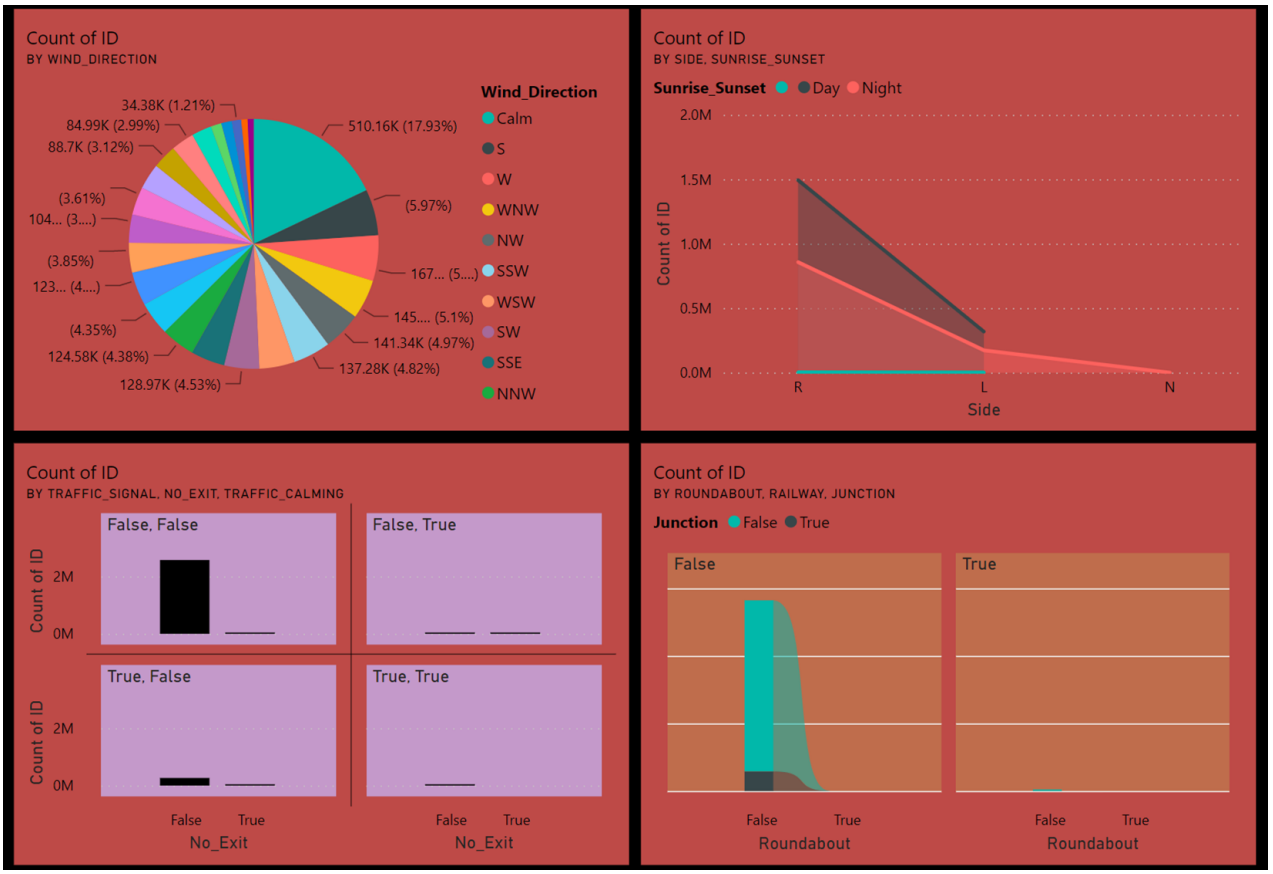
Here, we wanted to know certain details such as the names of states in which the data is divided. Various cities across many states are kept as legend, so that the positions of all states with respect to their longitude and latitude position can be shown on the map.

Clustered column chart of all the possible weather conditions in which accidents have occurred is represented in the above dashboard. This chart gives the idea that the majority of the accidents have occurred in fair conditions that proves the irresponsibleness of the driver. Also, cloudy weather impacted many accidents.

Further, we created an extra column in power BI to understand the severity of the accidents. with a column named ‘severe’,we created four string values viz low, moderate, high and very high.

Two different stacked charts are implemented in the above dashboard. First one is a stacked column chart for the accident severity of ‘high’ and ‘very high’. Second one is a stacked bar chart for the accident severity of ‘moderate’ and ‘low’. It has been observed that cities like chicago, houston and dallas had many severe accident records as compared to other cities.

3.5 Dashboard for count of ids for various locations and conditions



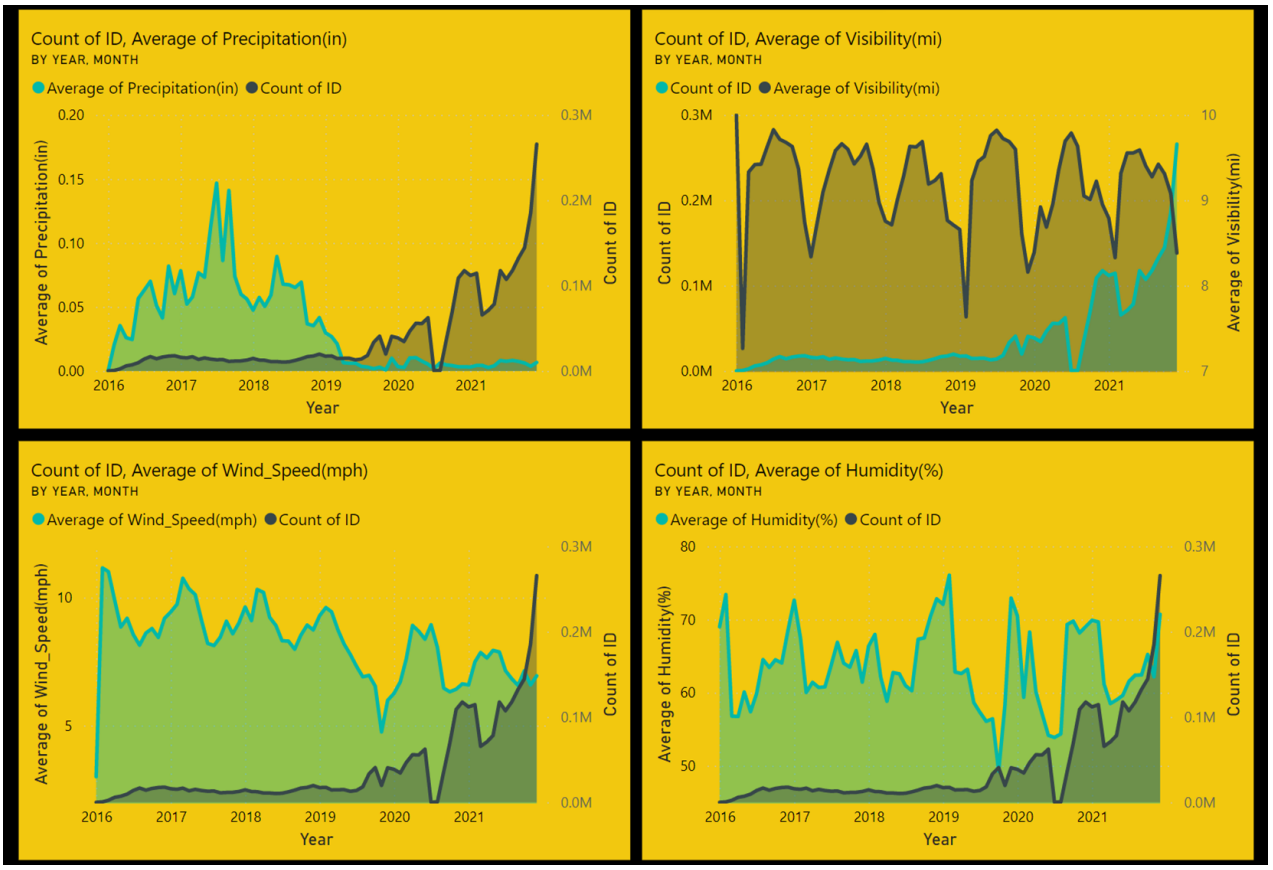
In the pie chart, we have tried to understand how wind flow conditions have impacted accidents in the United States. It has been observed that only 17.93% accidents had occurred in the calm winds. Remaining accidents had occurred in certain wind conditions. Out of all directions, the major impacted zone is west, south, west of north west and north west directions.

Road side of the accident had a major impact on the count of accidents. As shown in the area graph, the majority of the accidents had happened on the right side of the road. Since the USA has Left hand Drive compulsion, these numbers could be justified. Also, during night rides, there is not much difference in the count based on sides. Hardly any accident had happened on the neutral point.

Based on traffic control systems or traffic calming gestures, we have created further sub divided bar graphs. For small multiples, we have used ‘No exit’ boards. It had been observed that Many accidents had occurred when there was no sign of no exit and in the absence of any traffic calming gestures.

Understanding different locations, we have created the ribbon chart to observe how values in the roundabout, railway or junctions vary for total count of accidents. Although, the roundabout and railways did not impact a lot, yet the junction point had registered over 2.8 lakhs accidents in the given duration.

3.6 Dashboard for count of ids for atmospheric conditions



Understanding the various atmospheric conditions and their impacts on the total count, we have created several area charts.

Firstly, we have considered Average precipitation from January 2016 to December 2021, it has been observed that, with a decrease in the average precipitation, the number of accidents went on increasing in the given duration.

Our second graph gives us a brief idea of the visibility factor affecting the number of accidents. As year on year, visibility on the road has decreased, the number of accidents have increased. As the visibility drastically decreased in 2021, the number of accidents have increased by a large amount.

In the next chart, we have tried to relate the number of accidents with the average wind speed for every month. As observed, wind speed has been decreasing for the given duration, yet the number of accidents have been increasing.

Finally our last graph gives us a brief idea of the humidity factor affecting the number of accidents. As year on year, visibility on the road has varied, the number of accidents have also varied. Post year 2019, the graph of humidity looks very much aligned to the number of accidents..

Chapter IV : MACHINE LEARNING

4.1 Introduction to Machine Learning :

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

There Are Different Types of machine Learning algorithm based on dataset we choose the correct model

Models Used in Project:

1) Logistic Regression
2) Random Forest Classifier

4.2 EDA And Data Pre-processing:

4.3.1 Introduction To EDA

What is EDA?

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Need Of EDA Before Machine Learning?

Helps you prepare your dataset for analysis. Allows a machine learning model to predict our dataset better. Gives you more accurate results. It also helps us to choose a better machine learning model.

For EDA we use Python Libraries:-

1. Pandas
2. NumPy
3. Matplotlib
4. Seaborn

Before EDA we need to check following parameter:

1. NULL values in Dataset
2. Outliers in Dataset
3. Duplicate Values
4. Fixing Time stamps
5. Done some feature engineering

4.3.2 - Operation Perform on EDA

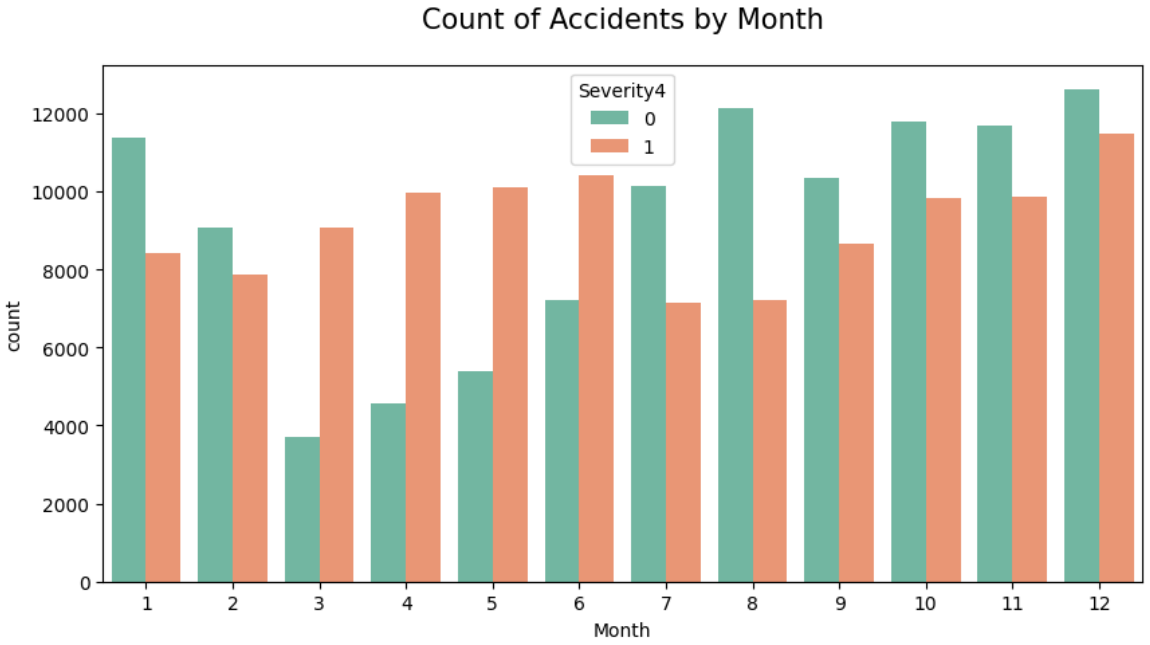
We Drop the Columns which is not Much important as per analysis like–

- 'Distance(mi)', 'End_Time' (we have start time), 'End_Lat', and 'End_Lng'(we have start location) can be collected only after the accident has already happened and hence cannot be predictors for serious accident prediction, so removed from data.

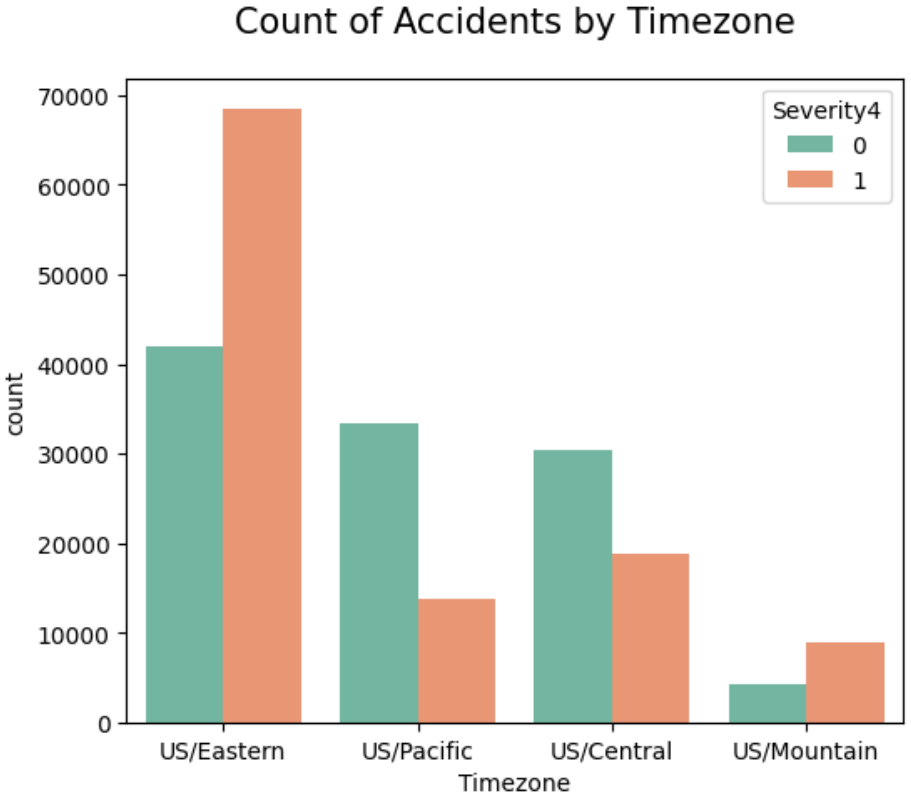
- 'Country' and 'Turning_Loop' for they have only one class. So, we have to drop them otherwise we will get error at the time of splitting the dataset.

EDA operations perform for Analysis the dataset:-

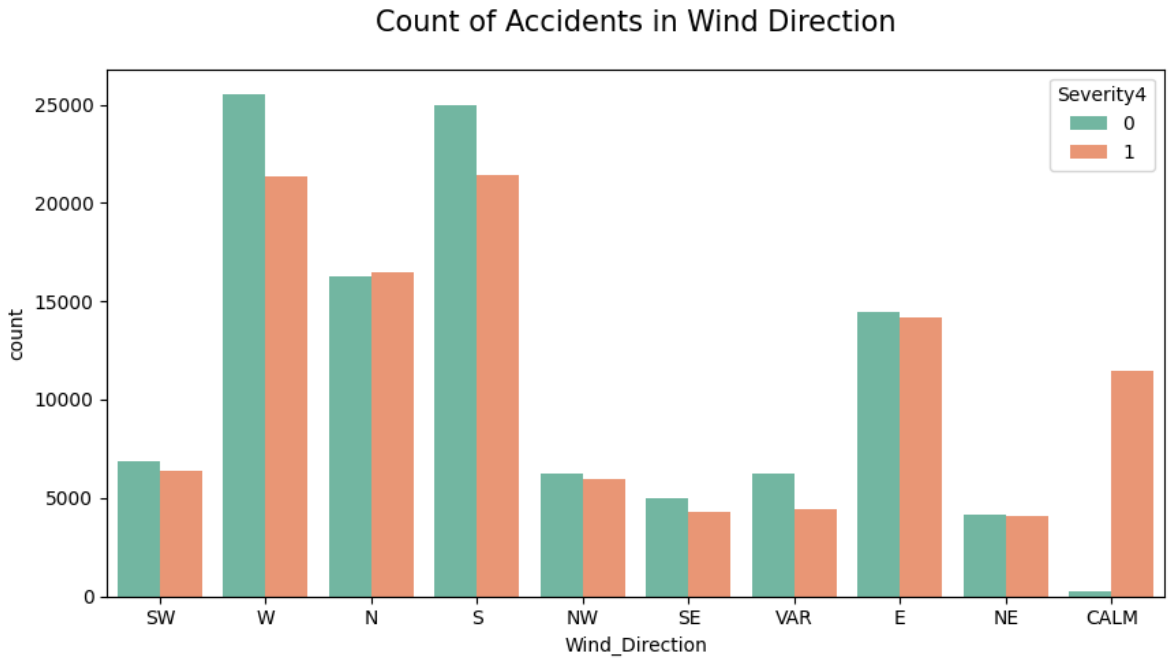
1) Balancing the target as it contains unbalanced labels and recategorizing their classes into two
2) EDA on Time features like month, year, weekday, period-of-day with respect to Severity



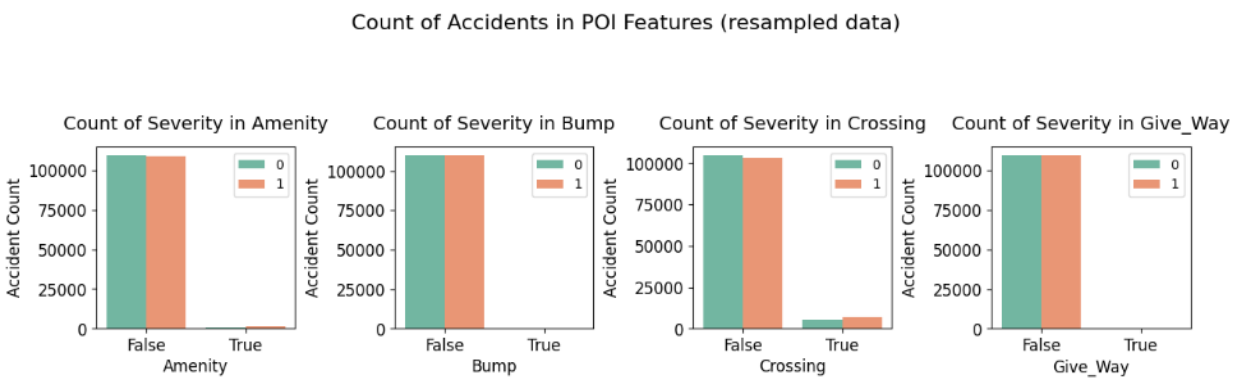
3) EDA on Address features like Timezone, state, side with respect to Severity



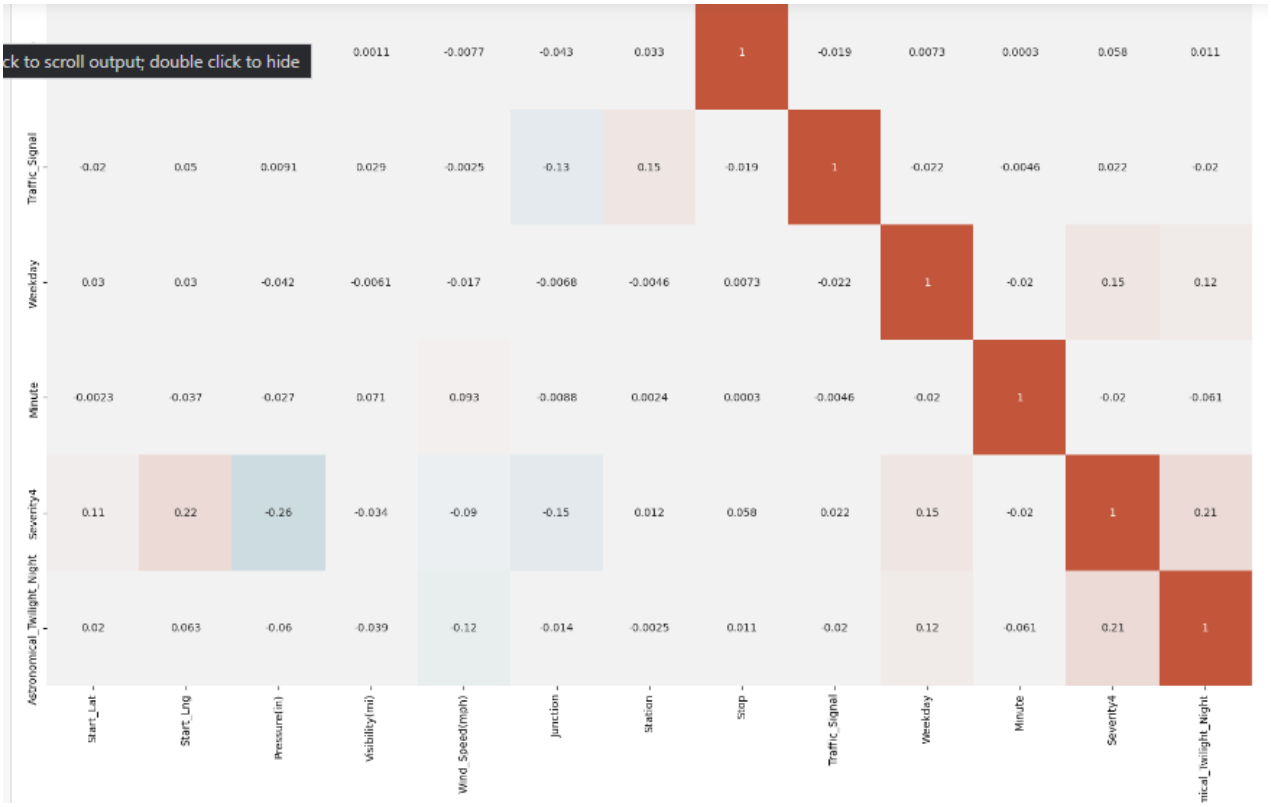
4) EDA on Weather features like different weather conditions & wind directions with respect to Severity



5) EDA on Point of Interest features like Junction, Railway, No_stop etc. with respect to Severity



6) Plotting Heatmap for the co-relation between features



4.3.3 One-Hot-Encoding

One hot encoding is **one method of converting data to prepare it for an algorithm and get a better prediction**. With one-hot, we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector. We convert all the categorical data to numerical format

4.3 Preparing for the Machine Learning Model :

4.4.1 Logistic-Regression Model

Logistic regression is used to solve classification problems, and the most common use case is binary logistic regression, where the outcome is binary (yes or no). In the real world, you can see logistic regression applied across multiple areas and fields.

- In health care, logistic regression can be used to predict if a tumor is likely to be benign or malignant.
- In the financial industry, logistic regression can be used to predict if a transaction is fraudulent or not.
- In marketing, logistic regression can be used to predict if a targeted audience will respond or not.

Mathematics behind logistic regression

Probability always ranges between 0 (does not happen) and 1 (happens). Using our Covid-19

example, in the case of binary classification, the probability of testing positive and not testing positive will sum up to 1. We use logistic function or sigmoid function to calculate probability in logistic regression. The logistic function is a simple S-shaped curve used to convert data into a value between 0 and 1.

$$h\Theta(x) = 1/1 + e - (\beta o + \beta 1X)$$

*'**hΘ(x)**' is output of logistic function , where 0 ≤ hΘ(x) ≥ 1*

*'**β1**' is the slope*
*'**βo**' is the y-intercept*
*'**X**' is the independent variable*

*(βo + β1*x) - derived from equation of a line Y(predicted) = (βo + β1*x) + Error value*

So in our scenario Our **Y** is Target variable means our **Severity Column**, X is the remaining all the features which we have to be predicted.

4.4.1.1 Splitting The data into test train using Sklearn

Python-Library to For Machine learning is sklearn. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

Split the data set into two pieces — a training set and a testing set. This consists of random sampling without replacement about 80 percent of the rows (you can vary this) and putting them into your training set. The remaining 20 percent is put into your test set. ("X_train," "X_test," "y_train," "y_test") for a particular train test split.

4.4.1.2 Train and Fit the model and 5-fold cross-validation

K-fold cross-validation is a superior technique to validate the performance of our model. It evaluates the model using different chunks of the data set as the validation set.

We divide our data set into K-folds. K represents the number of folds into which you want to split your data. If we use 5-folds, the data set divides into five sections. In different iterations, one part becomes the validation set.

Train the model on the training set. This is "X_train" and "y_train"

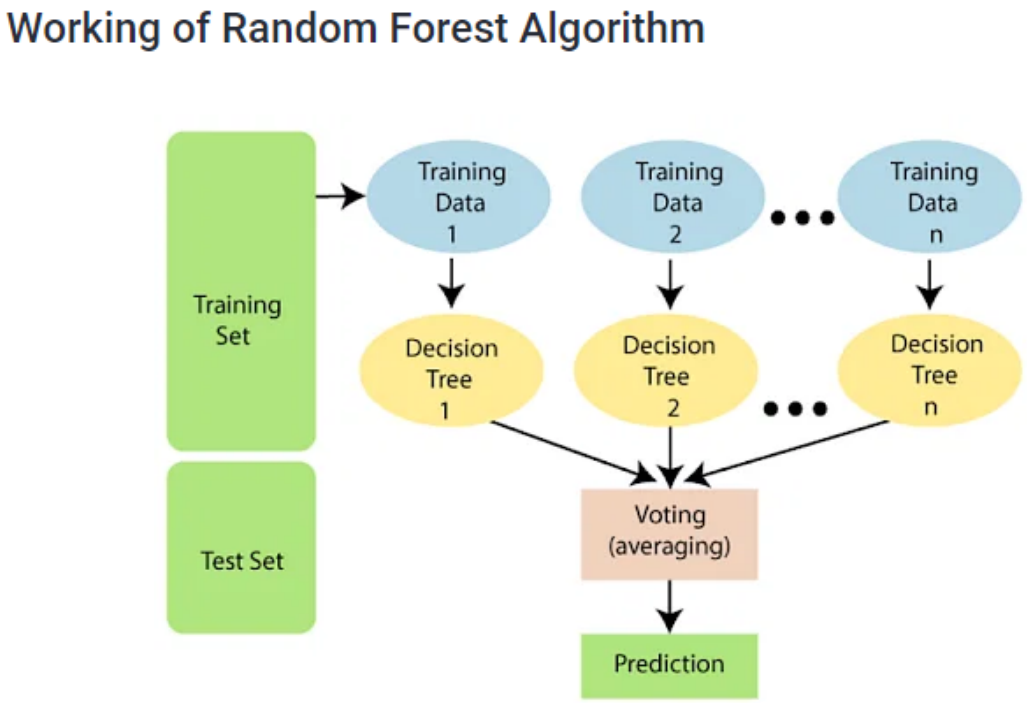
4.1.1.3 Testing and Prediction of model

Test the model on the testing set ("X_test" and "y_test" in the image) and evaluate the performance.

The Accuracy of the model is Near By 70% Which is too low. That's why by this conclusion we have to change the model of the dataset by another algorithm.

4.4.2 Applying Random Forest Classifier Model

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.



The Steps for Random Forest as per Logistic Regression we only import RandomForestClassifier()

- Fitting a Random-Forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)

4.5 Conclusion from machine learning:

Accuracy And Prediction:

After applying Random Forest Algorithm we got the nearby 86% accuracy which is quite good but not the best. Because applying a model on 2.8 million is a big task and also the target column is unbalanced so we trained the model on only 200k rows of data.

Chapter V : CONCLUSION

From this project we could manage to apply a prediction model to the given dataset. This could be done using Machine Learning, Big data and Data Visualization. We did hands on with Pyspark. We did data analysis using PySpark and we found some important insights from the questions.

For visualization we used Power BI Desktop and gathered many insights from the data easily with the help of dashboards we created. All the charts and graphs helped us understand the dataset from the visual perspective.

After that we performed EDA using a Jupyter notebook . Then we used 2.2 million rows out of 2.8 million for the prediction purpose. For prediction we used a Logistic regression model and random forest classifier. We achieved a decent accuracy of 86% and we got to learn many things from this project.