UNIVERSITI PUTRA MALAYSIA

BERILMU BERBAKTI

SSK4409-1: BIG DATA ANALYTICS

PROJECT

PROF. MADYA TS. DR. ISKANDAR BIN ISHAK

Prepared by:

No.	Name	Matric No.
1.	Anas Zulkifli bin Mohd Jeffry	206520
2.	Nik Muhammad Asyraf bin Nik Ismail	206630
3.	Amir Nurhakim bin Mohd Zaid	207092
4.	Muhammad Ikhwan Khuzairi bin Rozdi	206568

Part 1 - Big Data Platform

1. Apache Hadoop Installation

This part provides evidence of the successful installation of Apache Hadoop on our machine. Part 1 covers installation details, environment information, daemon status, web interfaces.

1.1. Installation details

Key configuration files, such as ‘hadoop-env’, ‘core-site.xml’, and ‘hdfs-site.xml’, were modified as per the installation requirements.

- hadoop-env

hadoop-env

File Edit View

```
@rem Unless required by applicable law or agreed to in writing, software
@rem distributed under the License is distributed on an "AS IS" BASIS,
@rem WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
@rem See the License for the specific language governing permissions and
@rem limitations under the License.

@rem Set Hadoop-specific environment variables here.

@rem The only required environment variable is JAVA_HOME.  All others are
@rem optional.  When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use.  Required.
set JAVA_HOME=C:\Progra~1\Java\jdk-21

@rem The jsvc implementation to use. Jsvc is required to run secure datanodes.
@rem set JSVC_HOME=%JSVC_HOME%
```

- core-site.xml

core-site.xml

C: > Hadoop > hadoop-2.9.2 > etc > hadoop > core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>


<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
 </property>
</configuration>
```

- hdfs-site.xml

```
hdfs-site.xml
C: > Hadoop > hadoop-2.9.2 > etc > hadoop > hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>


<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
  <name>dfs.namenode.name.dir</name>
  <value>C:\Hadoop\hadoop-2.9.2\data\namenode</value>
  <final>true</final>
 </property>
 <property>
  <name>dfs.datanode.data.dir</name>
  <value>C:\Hadoop\hadoop-2.9.2\data\datanode</value>
  <final>true</final>
 </property>
</configuration>
```

1.2. Environment Information

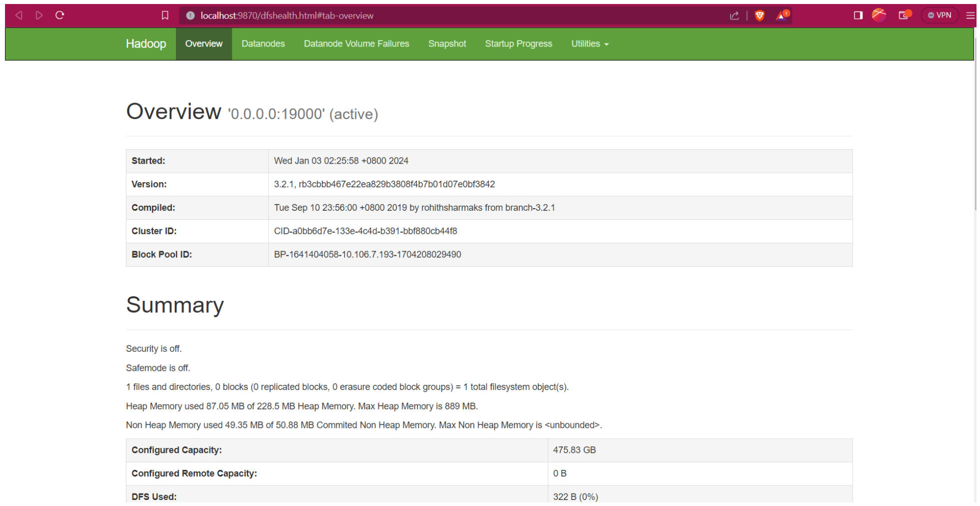
The installed Apache Hadoop version is verified using the following command:

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Hadoop\hadoop-2.9.2\bin>hadoop version
Hadoop 2.9.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 826afbeae31ca687bc2f8471dc841b66ed2c6704
Compiled by ajisaka on 2018-11-13T12:42Z
Compiled with protoc 2.5.0
From source with checksum 3a9939967262218aa556c684d107985
This command was run using /C:/Hadoop/hadoop-2.9.2/share/hadoop/common/hadoop-common-2.9.2.jar
```

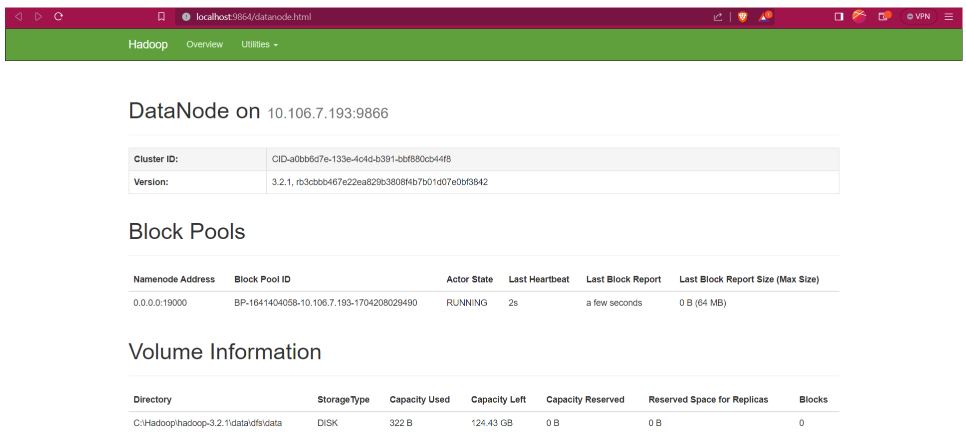
1.3. Web Interfaces

Accessing Hadoop web interfaces to monitor system status

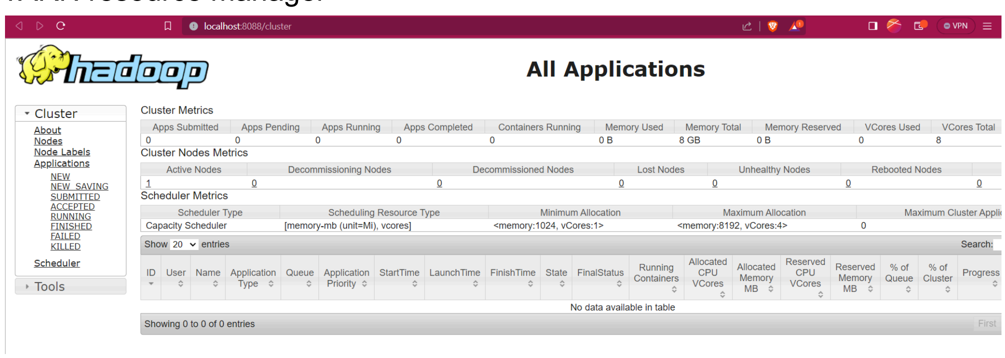
- HDFS NameNode



- HDFS datanode

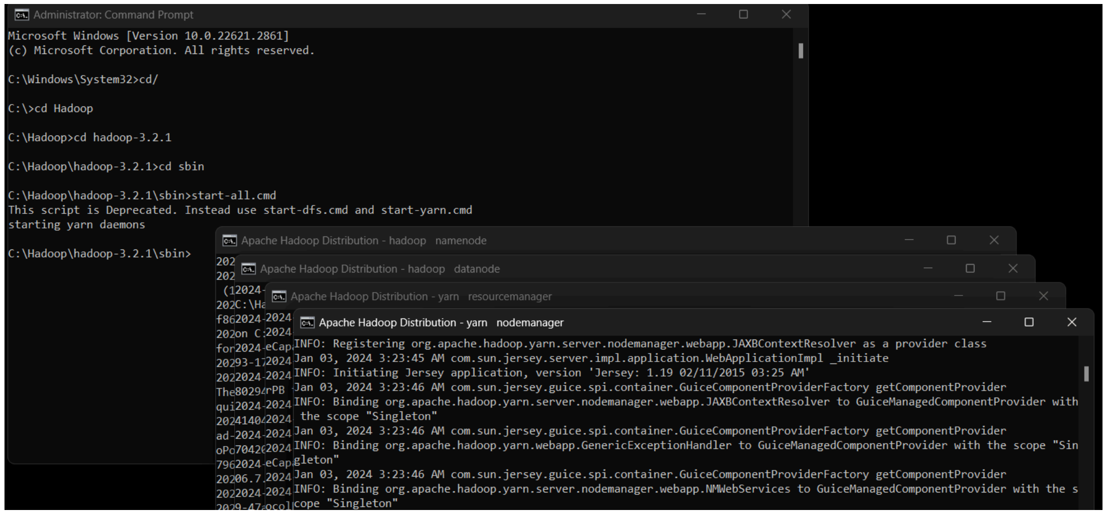


- YARN resource manager



2. MapReduce program on word counting on a text file

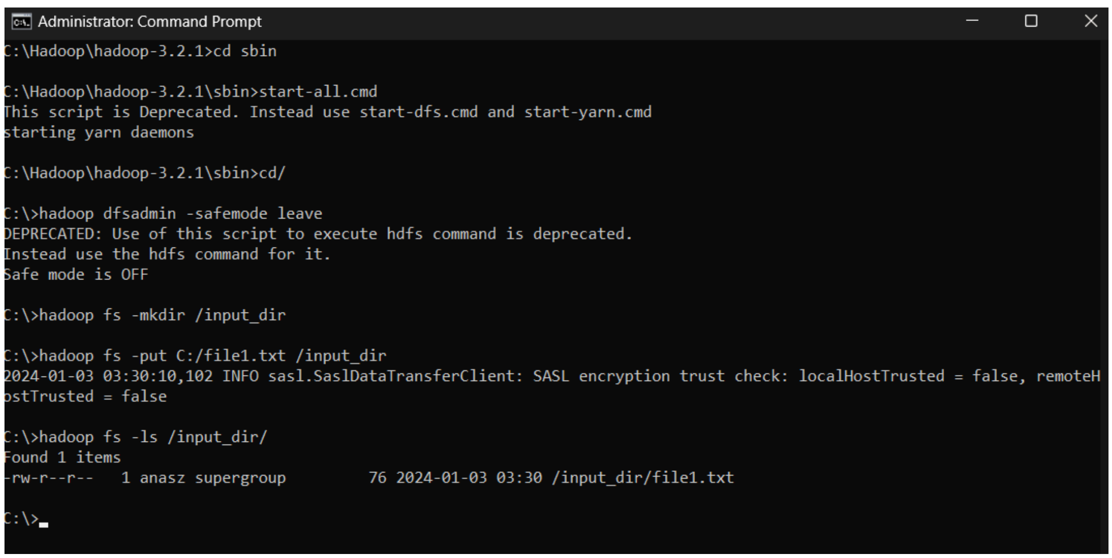
2.1. Open cmd in Administrator mode and move to “C:\Hadoop\hadoop-3.2.1\sbin” and start cluster



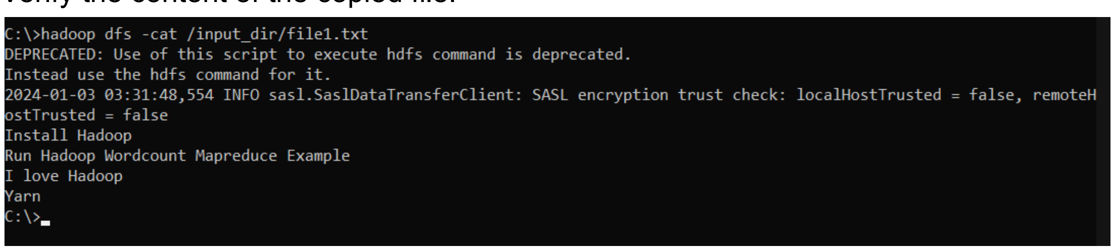
2.2. Create an input directory in HDFS.

2.3. Copy the input text file named file1.txt in the input directory (input_dir)of HDFS.

2.4. Verify file1.txt available in HDFS input directory (input_dir).



2.5. Verify the content of the copied file.

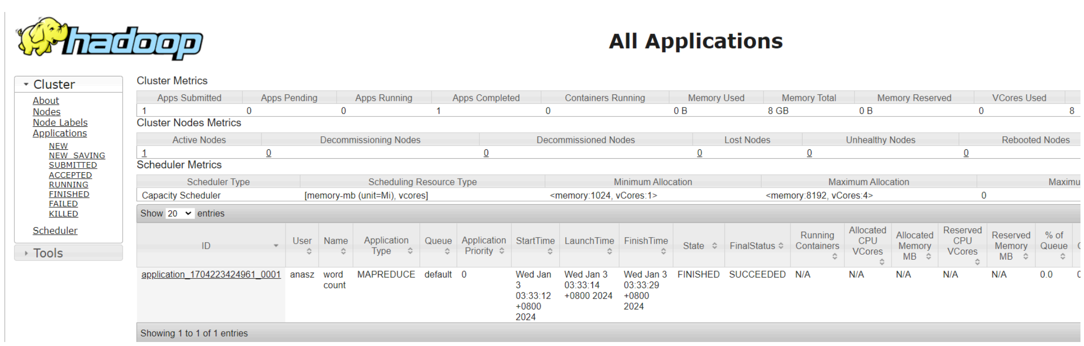


2.6. Run MapReduceClient.jar and also provide input and out directories.

```
Administrator: Command Prompt
                Map output materialized bytes=119
                Input split bytes=105
                Combine input records=11
                Combine output records=9
                Reduce input groups=9
                Reduce shuffle bytes=119
                Reduce input records=9
                Reduce output records=9
                Spilled Records=18
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=62
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=404750336
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=76
        File Output Format Counters
                Bytes Written=77

:\>_
```

2.7. Verify content for the generated output file.

```
C:\>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2024-01-03 03:34:44,023 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteH
ostTrusted = false
Example 1
Hadoop  3
I       1
Install 1
Mapreduce       1
Run     1
Wordcount       1
Yarn    1
love    1

C:\>_
```



Part 2: Data Analytics

2.1 Data Analytics Tasks

- Dataset link: https://www.kaggle.com/datasets/nezukokamaado/road-accident-casualties-dataset/data

2.1.1 Pre-processing

- Importing essential libraries for data processing, visualization and statistics

```
# Packages
# Data Processing
import numpy as np
import pandas as pd
# Visualization
import matplotlib.pyplot as plt
plt.rcParams['figure.dpi'] = 200
import seaborn as sns
# Statistics
import math
from scipy import stats
from scipy.stats import norm
# # Deep Learning
# import tensorflow as tf
# File Path
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

- Defining root path and a random seed for reproducibility

```
# setting
path_root = "C:/Users/Asus/Desktop/UPM/sem 7/BDA/project/"
seed = 394
```

- Adjusting Pandas display settings

```
# pandas display setting
pd.set_option('display.max_columns', 200)
```

- Reading a CSV file from a specified path into a Pandas DataFrame

```
df_accident = pd.read_csv(path_root + "caraccident.csv")
```

- Standardize column names for easier referencing

```
# rename
df_accident.rename(columns = {'Accident Date': 'Accident_Date', 'District Area': 'District_Area'}, inplace = True)
```

- Categorical features processing

```
# categorical features
list_categorical_features = [
    'Accident_Severity', 'Light_Conditions', 'District_Area', 'Road_Surface_Conditions',
    'Road_Type', 'Urban_or_Rural_Area', 'Weather_Conditions', 'Vehicle_Type'
]
df_accident[list_categorical_features] = df_accident[list_categorical_features].astype('category')
df_accident.info()
```

2.1.2 Data Cleaning

- Refining the dataset by selecting relevant columns, converting a date column to DateTime, handling duplicates and missing values, removing specific rows, and verifying the resulting DataFrame's structure.

```
# subsetting
df_accident = df_accident[[
#     'Index',
    'Accident_Severity', 'Accident_Date', 'Latitude',
    'Light_Conditions', 'District_Area', 'Longitude',
    'Number_of_Casualties', 'Number_of_Vehicles', 'Road_Surface_Conditions',
    'Road_Type', 'Urban_or_Rural_Area', 'Weather_Conditions',
    'Vehicle_Type'
]].copy()

# to_datetime
df_accident['Accident_Date'] = pd.to_datetime(df_accident['Accident_Date'], dayfirst = True)
df_accident.head()

# dropping duplicated rows and missing values
df_accident = df_accident.loc[~df_accident.duplicated()].reset_index(drop = True).copy()
df_accident = df_accident.dropna()

# remove "Unallocated" rows(only 3 cases)
df_accident = df_accident.loc[
    (df_accident['Urban_or_Rural_Area'] == "Urban") |
    (df_accident['Urban_or_Rural_Area'] == "Rural")
]

# check
df_accident.info()
```

2.1.3 Modeling

- Imports modules from the scikit-learn library for machine learning tasks.

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

- Prepares the feature set for modeling by removing unnecessary columns, creates a new DataFrame df_X with one-hot encoding for categorical variables to avoid multicollinearity, and sets up a new DataFrame df_y for the target variable "Accident_Severity.

```
df_accident.drop([
    "Accident_Severity", "Accident_Date", "District_Area", "Number_of_Casualties", "Number_of_Vehicles"
], axis = 1)

df_X = df_accident.drop([
    "Accident_Severity", "Accident_Date", "District_Area", "Number_of_Casualties", "Number_of_Vehicles"
], axis = 1)
df_X = pd.get_dummies(df_X, columns = [
    "Light_Conditions", "Road_Surface_Conditions", "Road_Type", "Urban_or_Rural_Area", "Weather_Conditions", "Vehicle_Type"
], drop_first = True)

df_y = df_accident["Accident_Severity"]

df_X.head()
```

- Splits the datasets into training and validation sets based on the target variable "Accident_Severity" to ensure a proportional representation of classes in both sets.

```
X_tr, X_val, y_tr, y_val = train_test_split(df_X, df_y, test_size = 0.3, random_state = seed, stratify = df_y)
```

- Initializes four classification models: Logistic Regression (model_lr), Decision Tree (model_dt), Random Forest (model_rf), and k-Nearest Neighbors (model_knn).

```
list_model = [model_lr, model_dt, model_rf, model_knn]
```

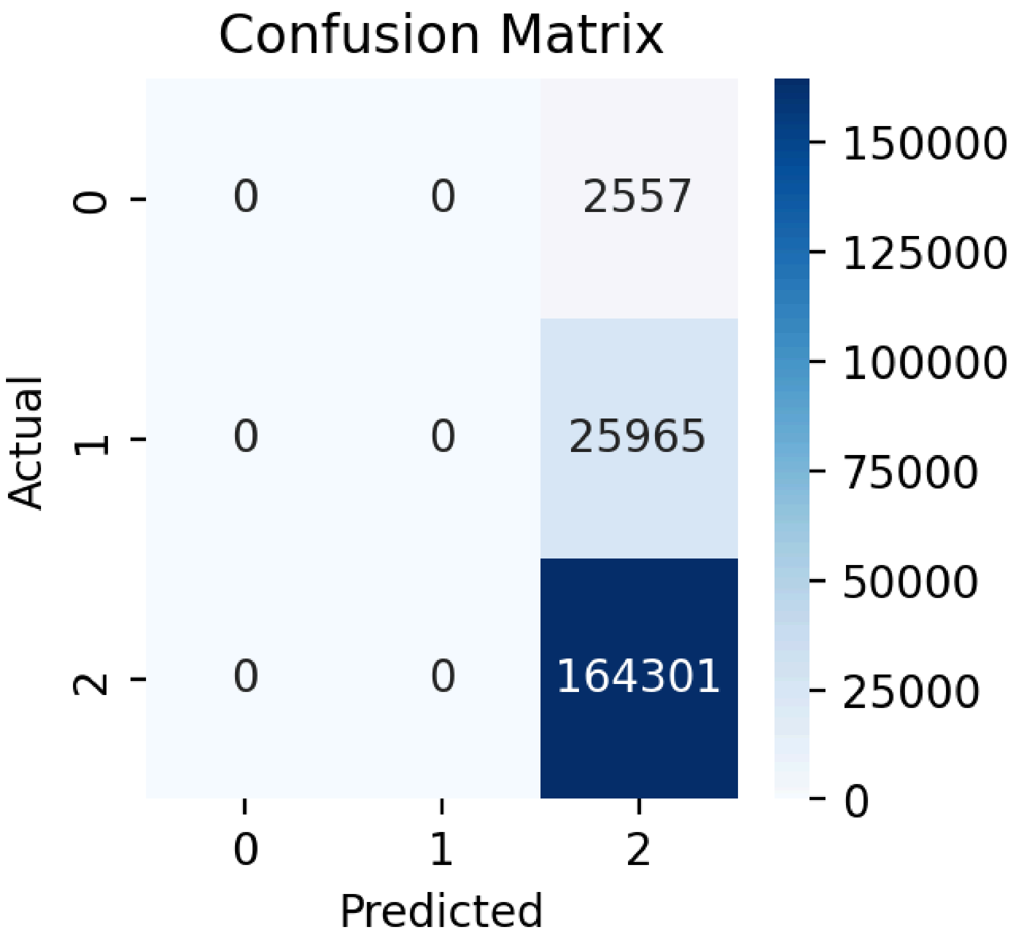
- Iterates through the list of models, fitting each model on the training data (X_tr, y_tr). Predicts the target variable on the validation data (X_val) and lastly prints the classification report, providing precision, recall, F1-score, and accuracy metrics for each model.

```python
for model in list_model:

    print("")
    print(str(model))

    model.fit(X_tr, y_tr)
    y_pred = model.predict(X_val)

    report = classification_report(y_val, y_pred)
    print(report)

    # confusion matrix
    temp_confusion_matrix = confusion_matrix(y_val, y_pred)
    plt.figure(figsize = (3, 3), facecolor = "white")
    sns.heatmap(
        temp_confusion_matrix,
        annot = True, fmt = 'd', cmap = 'Blues'
    )
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()
```

2.1.4 Results

- Confusion Matrix (Logistic Regression)

  This model performed well overall with an accuracy of 85% but struggled when it came to predicting Fatal and Serious accidents as well as did not correctly identify any instances of these severe outcomes, resulting in 0% precision for both. However, the model excelled in correctly classifying less severe accidents, resulting in a 100% recall and a solid F1-score of 92% for the Slight class. While it may have limitations in dealing with more severe incidents, it proves reliable in recognizing less critical accidents.

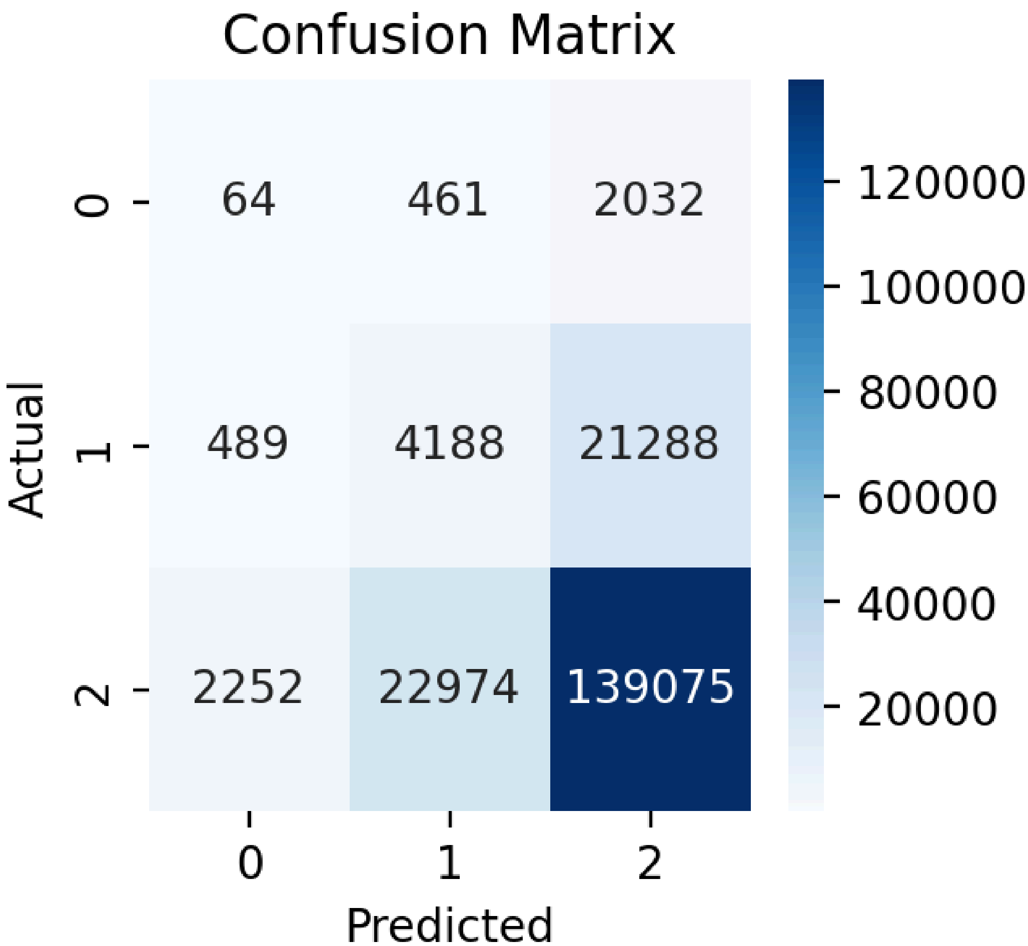


Output:

	precision	recall	f1-score	support
Fatal	0.00	0.00	0.00	2557
Serious	0.00	0.00	0.00	25965
Slight	0.85	1.00	0.92	164301
accuracy			0.85	192823
macro avg	0.28	0.33	0.31	192823
weighted avg	0.73	0.85	0.78	192823

- Confusion Matrix (Decision Tree)

  With a 74% accuracy, the model improved in identifying Fatal and Serious accidents compared to Logistic Regression. However, it still had low precision for these severe cases (2% for Fatal and 15% for Serious), indicating the need for enhancements, especially in handling more severe accidents.

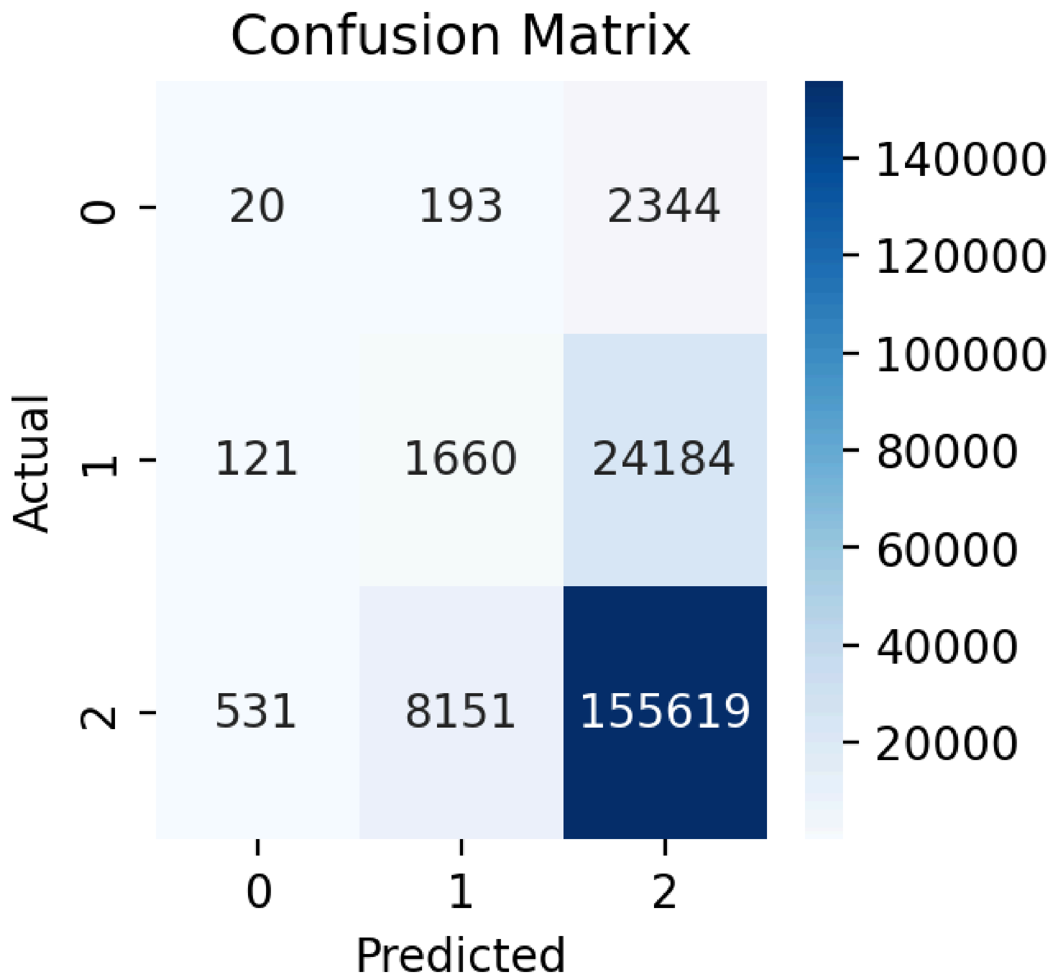


Output:

	precision	recall	f1-score	support
Fatal	0.02	0.03	0.02	2557
Serious	0.15	0.16	0.16	25965
Slight	0.86	0.85	0.85	164301
accuracy			0.74	192823
macro avg	0.34	0.34	0.34	192823
weighted avg	0.75	0.74	0.75	192823

- Confusion Matrix (Random Forest)

  The model achieved an overall accuracy of 82%, showing better precision for Fatal (3%) and Serious (17%) accidents compared to the Decision Tree. However, it still has limitations in precision for these severe cases, and the model struggles to capture and classify instances of severe accidents accurately, as indicated by lower recall scores.

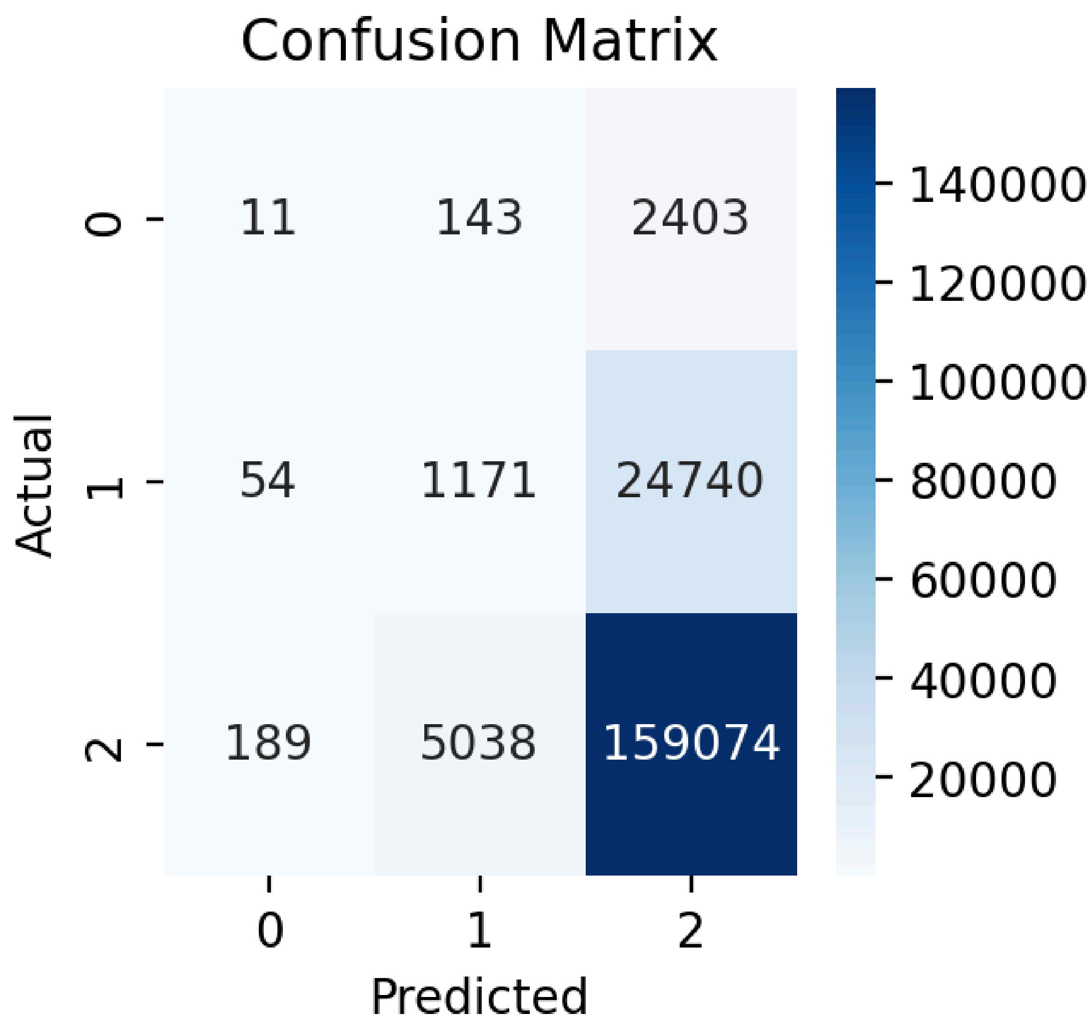


Output:

	precision	recall	f1-score	support
Fatal	0.03	0.01	0.01	2557
Serious	0.17	0.06	0.09	25965
Slight	0.85	0.95	0.90	164301
accuracy			0.82	192823
macro avg	0.35	0.34	0.34	192823
weighted avg	0.75	0.82	0.78	192823

- Confusion Matrix (K-Nearest Neighbor)

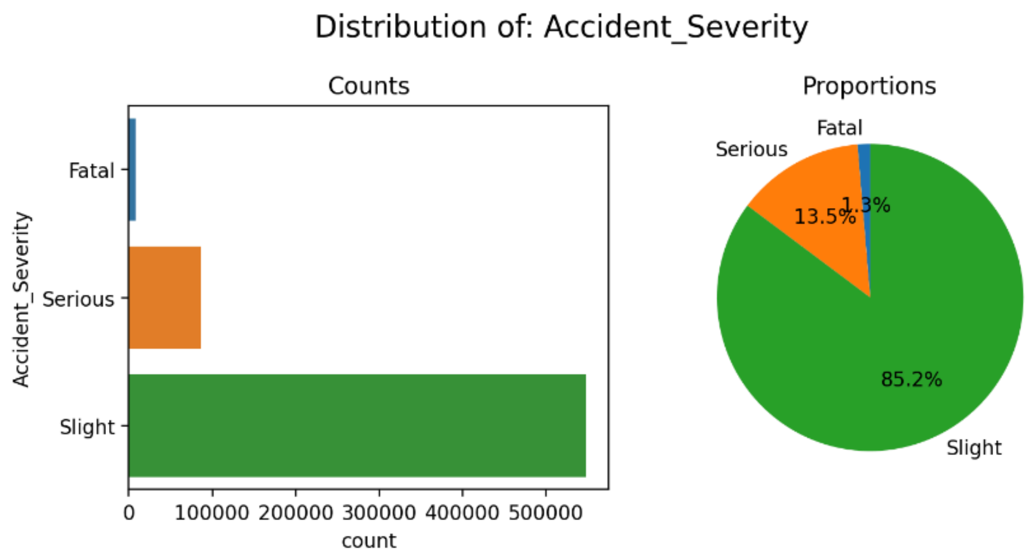
  With an 83% accuracy, KNN showed relevant results in predicting all accident severity levels. It had balanced precision, recall, and F1-score for each class. Thus, KNN outperformed Logistic Regression, Decision Tree, and Random Forest in recall scores for Fatal and Serious accidents.



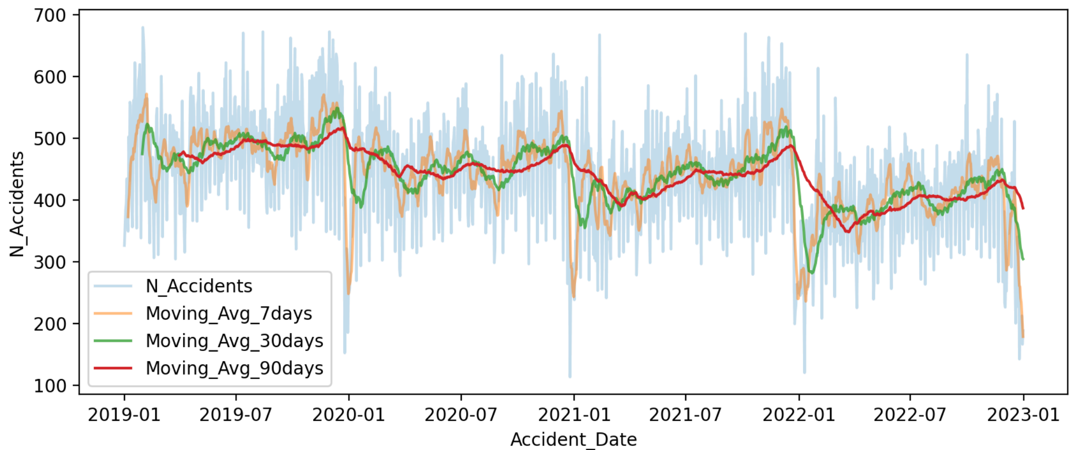
Output:

	precision	recall	f1-score	support
Fatal	0.04	0.00	0.01	2557
Serious	0.18	0.05	0.07	25965
Slight	0.85	0.97	0.91	164301
accuracy			0.83	192823
macro avg	0.36	0.34	0.33	192823
weighted avg	0.75	0.83	0.78	192823

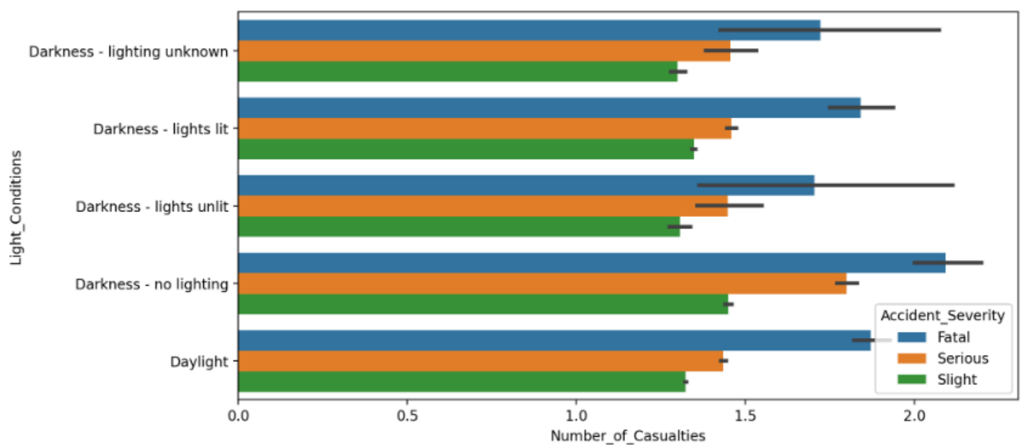
- Result Visualization

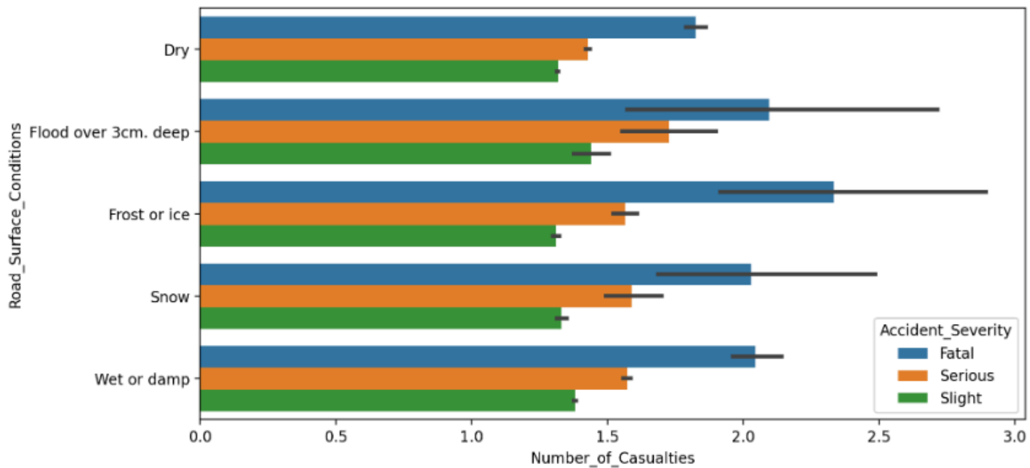


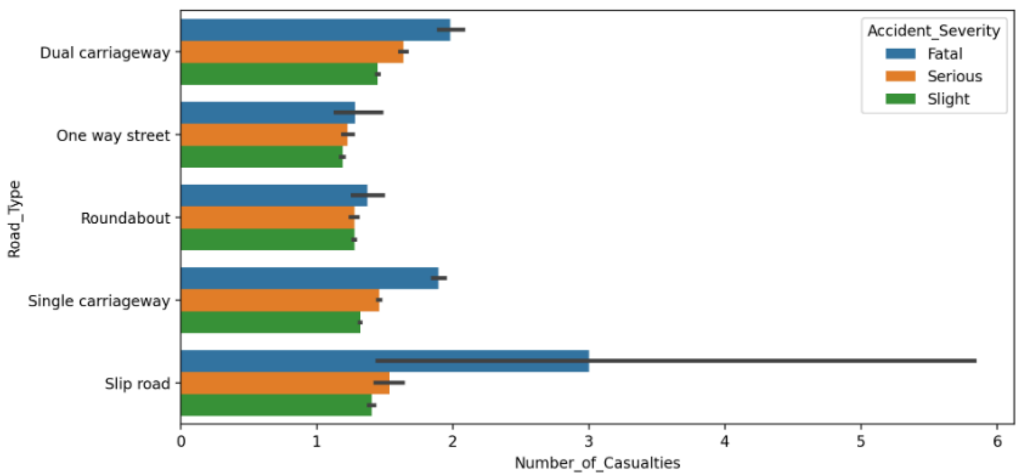
The majority of accidents, comprising 85.2%, are categorized as "Slight," denoting events with minor consequences. A lesser but notable proportion, accounting for 13.5%, falls under the "Serious" category, signifying incidents of greater impact. The least prevalent, at 1.3%, are "Fatal" accidents, emphasizing their infrequency and heightened gravity.

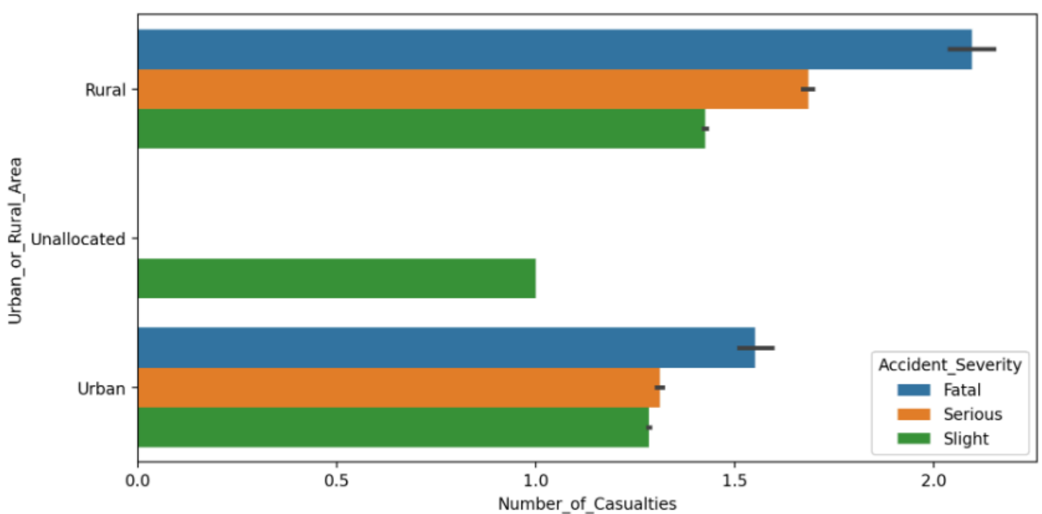


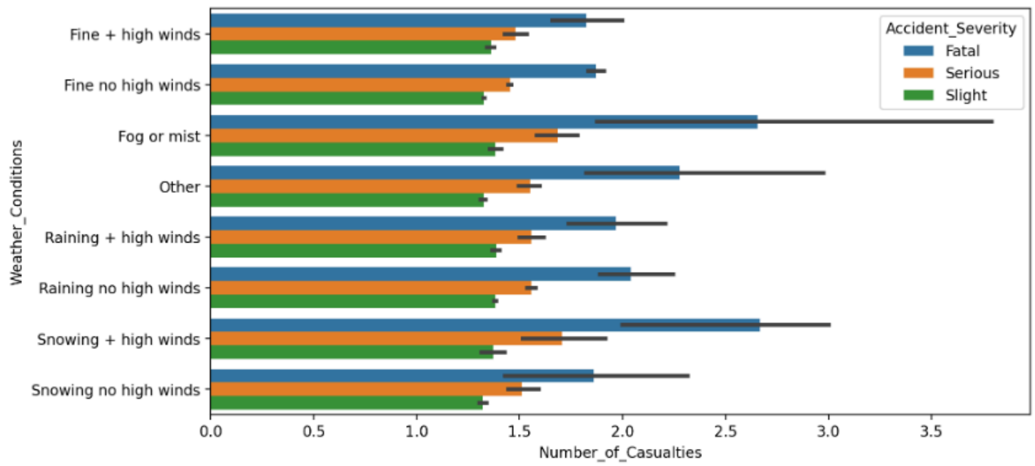
From the visualization of the trend in the number of car accidents occurring, there is a general decrease in the total incidents. It is however still noteworthy that specific periods throughout the year exhibit a distinct reduction in accident occurrences.




Road_Surface_Conditions
Dry
Flood over 3cm. deep
Frost or ice
Snow
Wet or damp
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Number_of_Casualties
Accident_Severity
Fatal
Serious
Slight


Road_Type
Dual carriageway
One way street
Roundabout
Single carriageway
Slip road
0
1
2
3
4
5
6
Number_of_Casualties
Accident_Severity
Fatal
Serious
Slight


Urban_or_Rural_Area
Rural
Unallocated
Urban
0.0
0.5
1.0
1.5
2.0
Number_of_Casualties
Accident_Severity
Fatal
Serious
Slight



The boxplots showcase the impact of diverse variables—namely, light conditions, road surface conditions, road type, weather conditions, and urban or rural areas—on the severity of accidents. Within each distinct condition, these boxplots specify the statistical distribution of casualties across different severity levels, thereby offering a comprehensive visualization of accident occurrences. This results in visualization establishes a foundational basis for future research attempts focused on predicting the risk of car accidents by examining the interplay of these environmental conditions.

2.1.5 Summary

The results regarding critical aspects of road incidents offer valuable recommendations and insights to enhance road safety. Recognizing the severity of accidents is deemed crucial for providing effective road management strategies as well as the significance of understanding where and when accidents occur, guiding interventions in specific regions and timely safety measures. A comprehensive dataset is advocated, serving as a relevant foundation for ongoing research and informed policymaking. Weather and road conditions' impact on accident rates is highlighted, urging the development of weather-responsive safety protocols and awareness initiatives.

By identifying accident hotspots and associated risk factors, targeted preventative measures can be implemented to effectively allocate resources to high-risk areas. These results suggest utilizing data-driven techniques, such as predictive modelling, to proactively tackle road safety issues. Additionally, incorporating traffic collision analysis into urban planning can help create safer urban environments through improved road design and infrastructure. Another key factor is understanding patterns of driver behavior and the importance of applicable educational campaigns and regulations to improve overall road safety. In conclusion, these results offer a comprehensive roadmap for evidence-based interventions, showcasing the value of a refined and targeted approach.

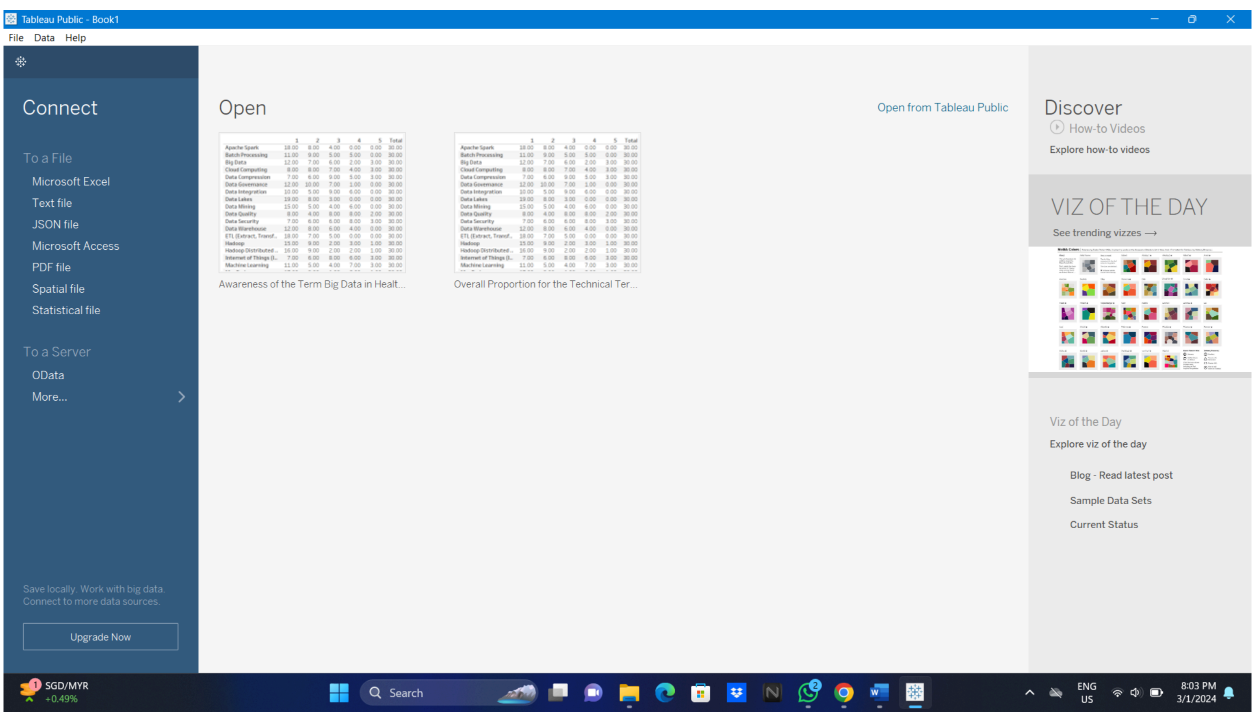
Part 3: Data Visualization

3.1 Public Tableau Installation

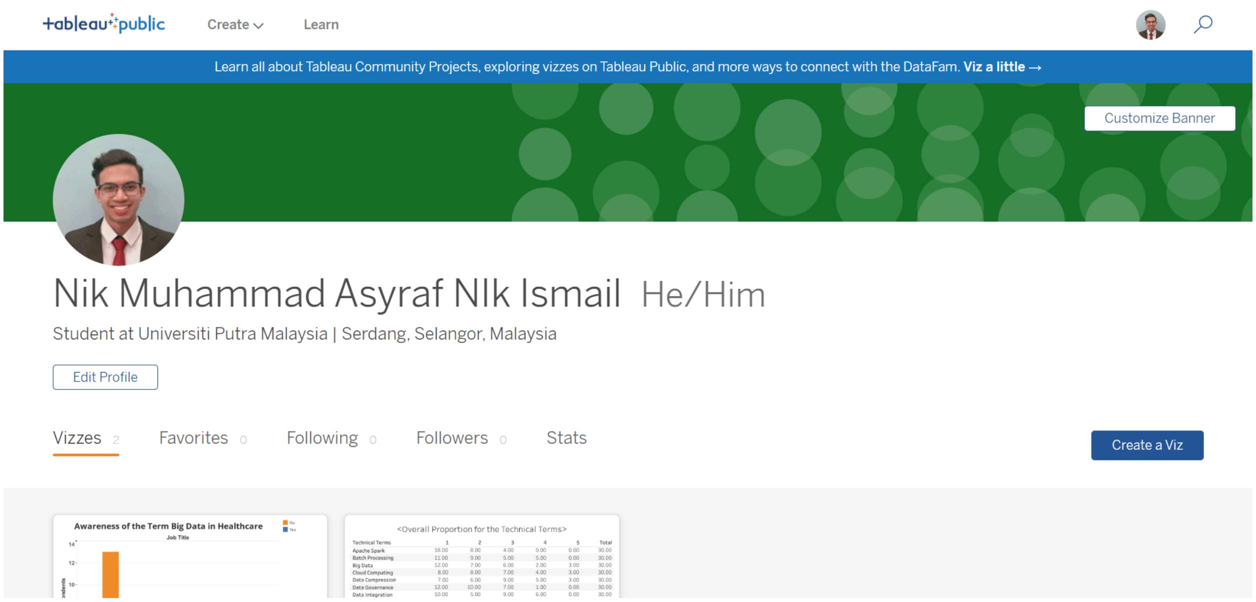
This part provides evidence of the successful installation of Public Tableau on our machine. Part 3 covers installation details, creation of a Tableau account on a Public Tableau Site, dataset analysis, data relationships analysis and a combination of Tableau worksheets within a single story.

3.1.1 Installation Details

- The start Screen of Tableau is shown:

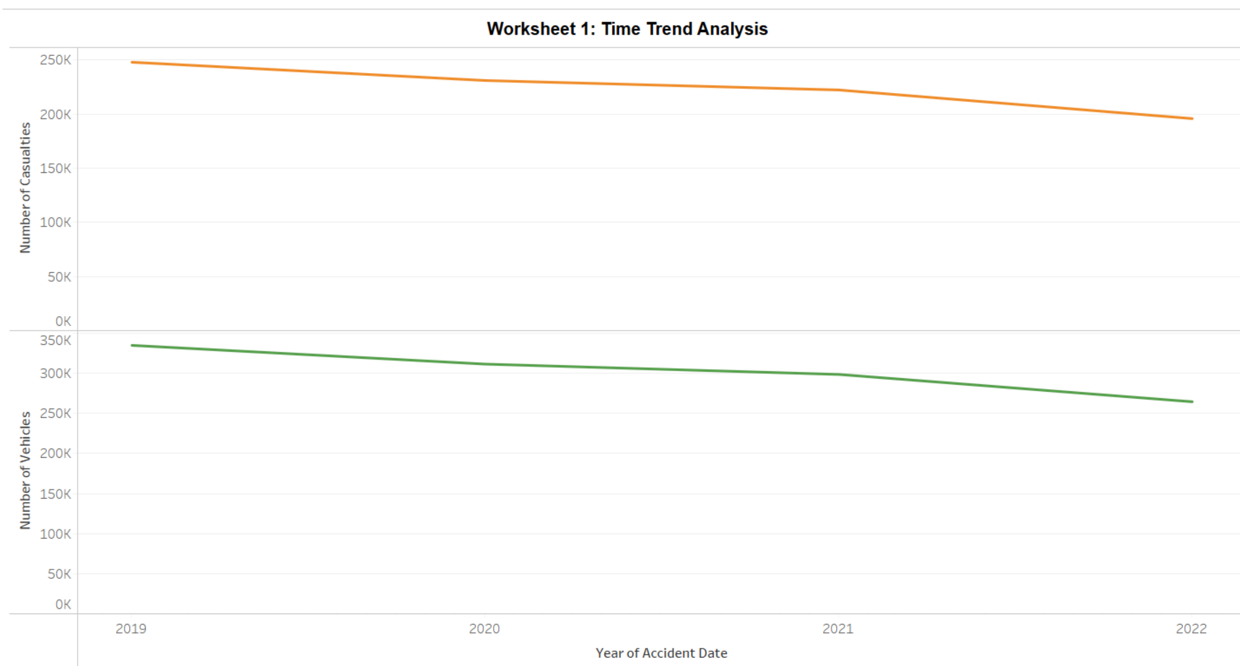


3.1.2 Creation of Public Tableau account in Public Tableau Site



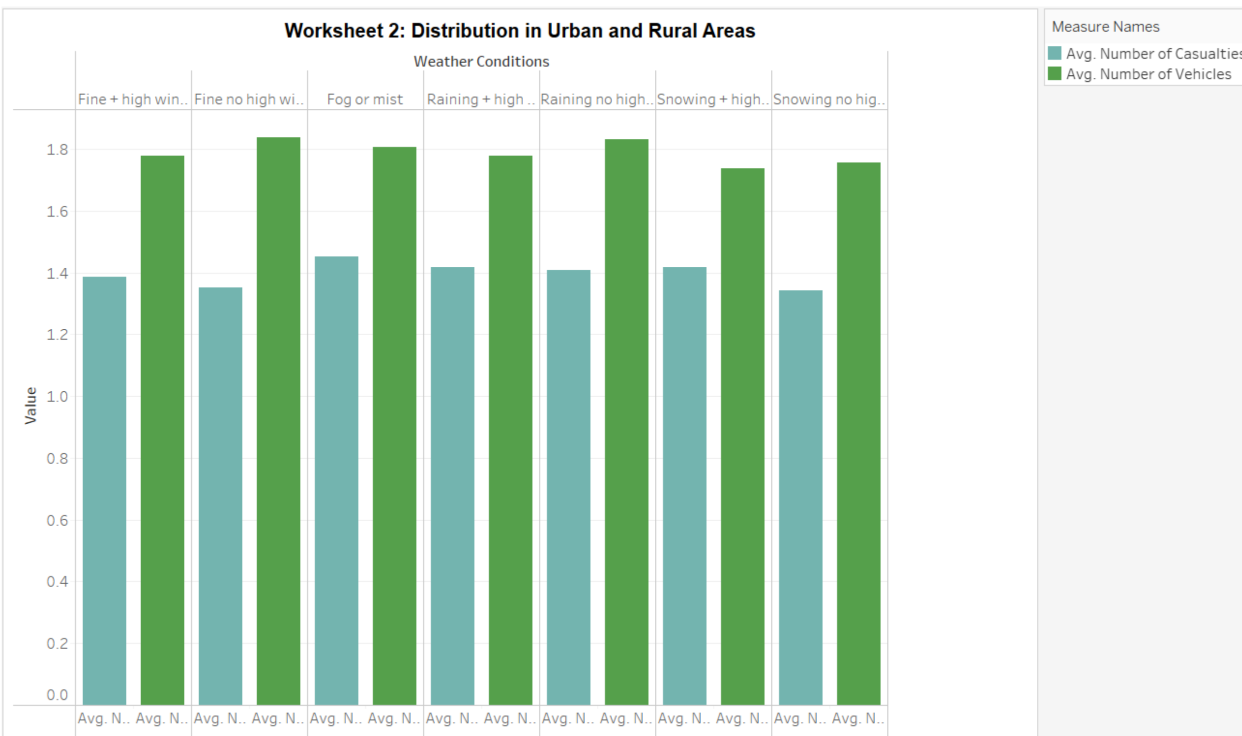
3.2 Relationships in The Chosen Dataset with Tableau Analysis

3.2.1 Worksheet 1: Time Trend Analysis



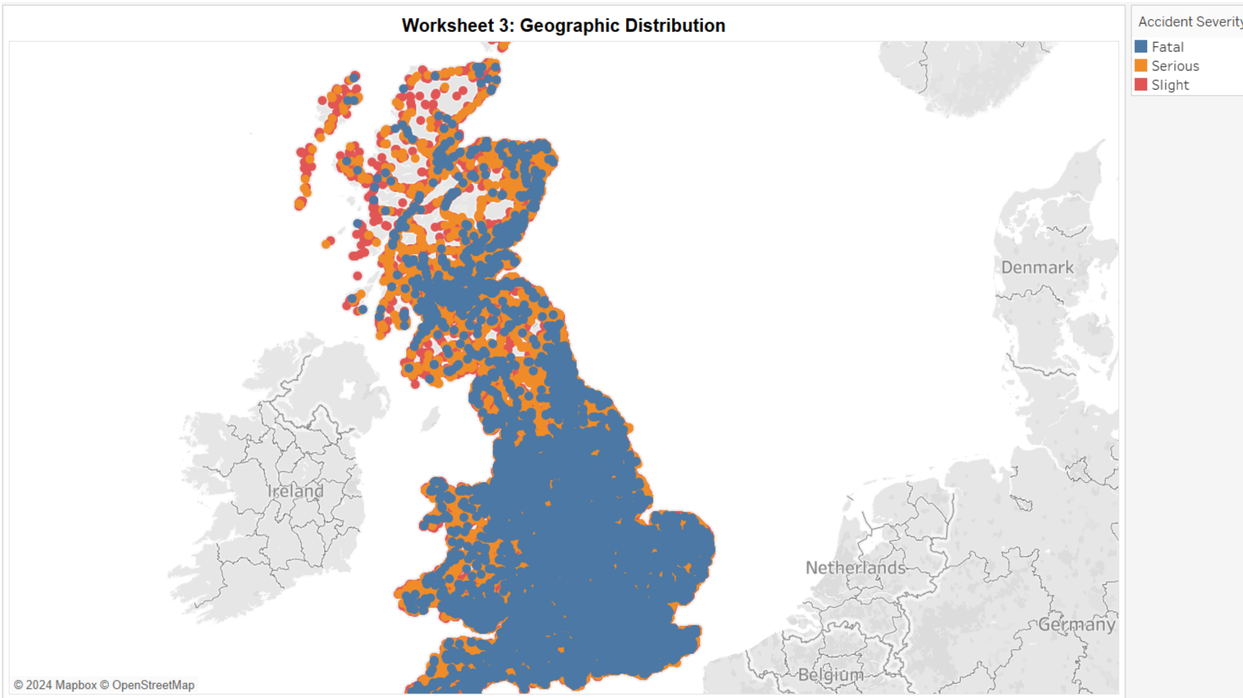
In Worksheet 1, the Time Trend Analysis graph presents a comprehensive overview of accident characteristics over the observed period. The x-axis represents the timeline, with years indicated for each point. The y-axes illustrate the total number of casualties and vehicles involved in accidents. Notably, both lines exhibit a gradual decrease over time, suggesting a positive trend in reducing the overall number of casualties and vehicles involved in accidents. This encouraging pattern may indicate successful interventions, improved safety measures, or changing traffic dynamics. Further analysis and correlation with external factors can provide deeper insights into the causes behind this positive trajectory, aiding in the formulation of effective accident prevention strategies.

3.2.2 Worksheet 2: Distribution in Urban and Rural Areas



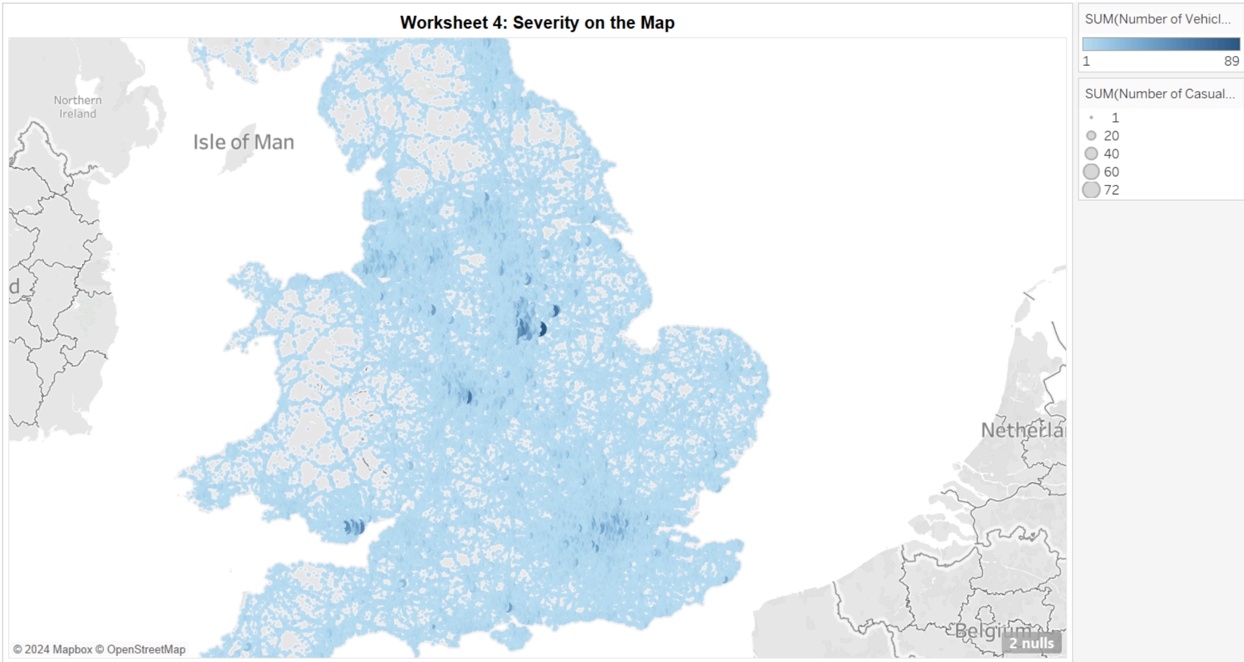
The Distribution in Urban and Rural Areas utilizes a clustered bar chart to highlight the impact of different weather conditions on accident characteristics. The x-axis displays various weather conditions, while the y-axis represents the average number of casualties and vehicles involved in accidents. The clustered bars showcase a clear distinction in the measure values, with an average of 1.4 for casualties and 1.8 for vehicles across different weather conditions. This chart allows for a quick comparison of the influence of weather on both casualties and vehicles, providing valuable insights for traffic management and safety measures.

3.2.3 Worksheet 3: Geographic Distribution



In Worksheet 3, the Geographic Distribution map employs longitude and latitude as columns and rows, respectively, creating a symbol map to visualize the spatial distribution of accidents across the United Kingdom. The color mark is used to represent the severity of accidents, with blue indicating fatal incidents, orange for serious, and red for slight. The map reveals a distinct pattern where fatal accidents are predominantly concentrated in the central to southern regions, extending towards the south, while the northern areas are coded with a higher prevalence of serious incidents.

3.2.4 Worksheet 4: Severity on the Map

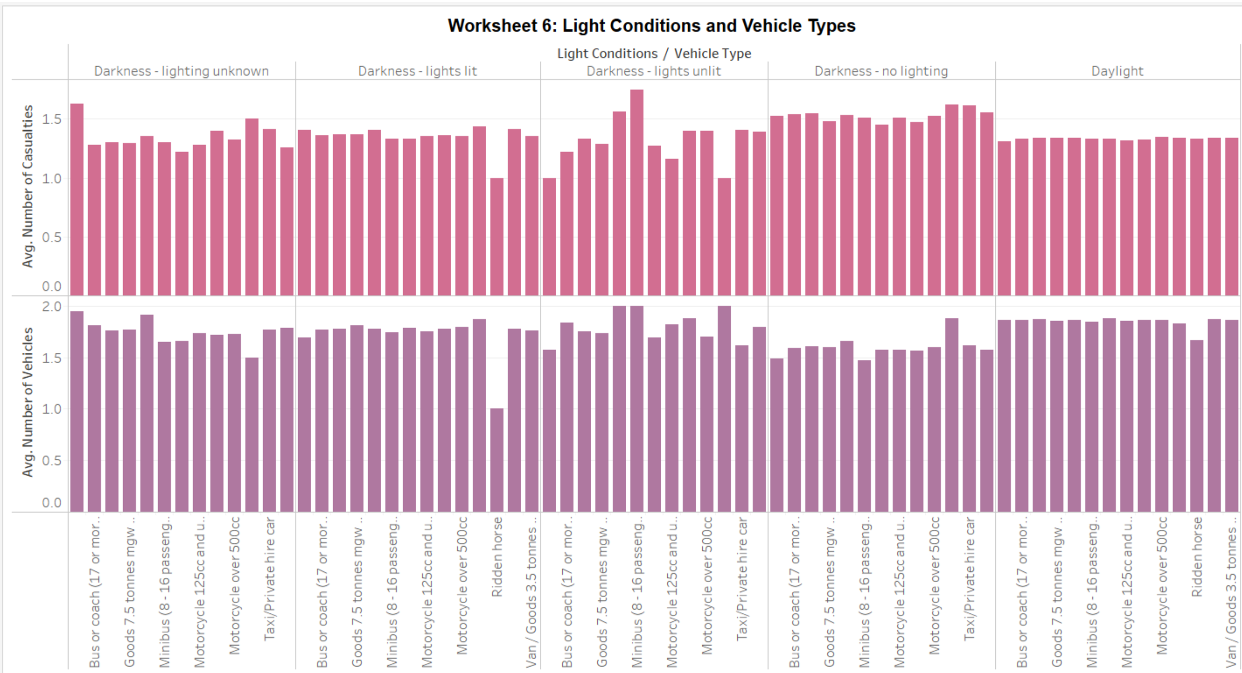


In Worksheet 4, the Severity on the Map graph utilizes latitude and longitude as rows and columns, respectively, creating a symbol map to visually represent the severity and impact of accidents. The filter allows users to explore accident severity levels. The color code, in blue, indicates the total number of vehicles involved in accidents, while the size mark represents the total number of casualties. Larger symbols denote a higher number of casualties, providing an immediate visual cue to the severity of accidents in specific geographic locations.

3.2.5 Worksheet 5: Factors by Severity

The Factors by Severity graph provides a comprehensive view of the distribution of accidents based on road surface conditions and road types. The graph employs stacked bars with AVG(Number of Casualties) and AVG(Number of Vehicles) as rows, and Road Surface Condition as columns. The filter allows users to explore the impact of different road types on accident severity levels. The color-coded bars vividly represent various road types, such as blue for dual carriageways, orange for one-way streets, red for roundabouts, light teal for single carriageways, and green for slip roads. The stacked structure of the bars allows for a clear comparison of the contribution of each road surface condition to different severity levels, offering insights into the factors influencing accident outcomes and facilitating targeted interventions for specific road types.

3.2.6 Worksheet 6: Light Conditions and Vehicle Types



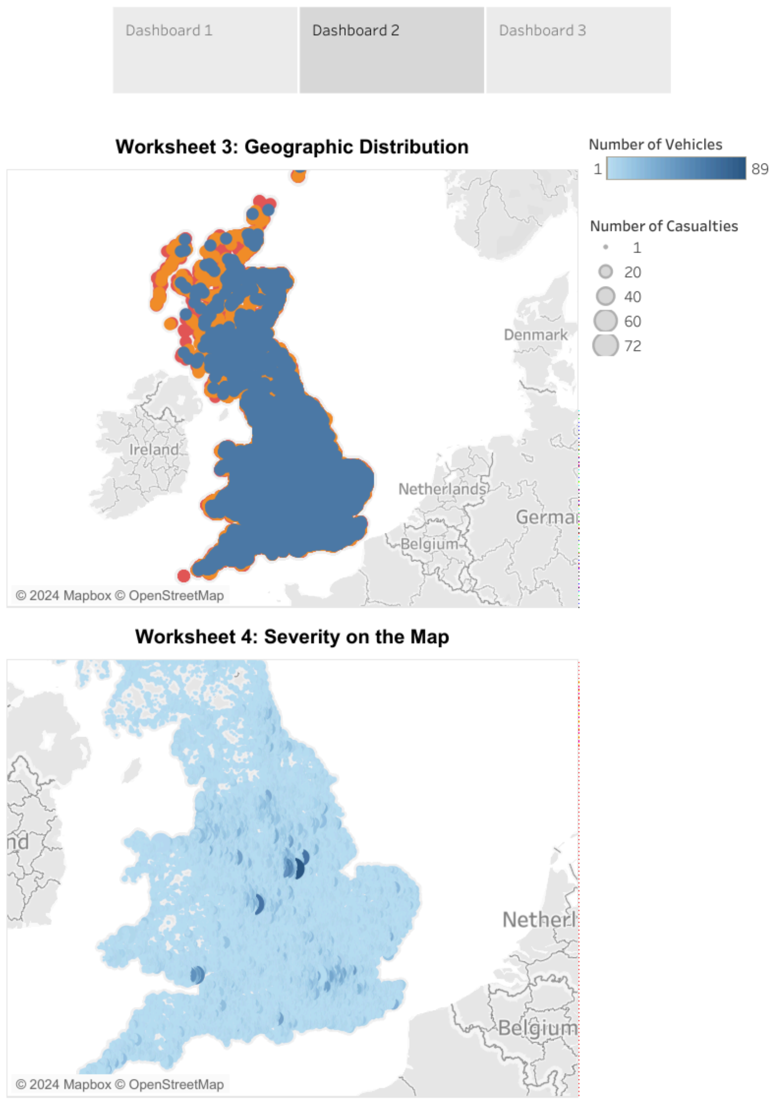
In Worksheet 6, the Light Conditions and Vehicle Types graph presents a detailed analysis of the distribution of accidents based on light conditions and vehicle types. The graph utilizes a side-by-side bar chart layout with AVG(Number of Casualties) and AVG(Number of Vehicles) as rows, Light Conditions as columns, and Vehicle Types as a filter. This visualization enables a nuanced exploration of how different light conditions impact accident severity and the involvement of various vehicle types. By segregating the data into distinct bars for each combination of light condition and vehicle type, the graph provides a comprehensive understanding of the relationships between these factors. Policymakers and safety experts can derive valuable insights from this visual representation to develop targeted strategies for improving road safety under specific conditions and vehicle scenarios.

3.3 Combination of Worksheets into a Single Story

3.3.1 Dashboard 1

3.3.2 Dashboard 2

Story 1



3.3.3 Dashboard 3

3.4 Story Publication in Tableau Public Page

- All Tableau worksheets were combined within a single Story, comprising 3 Dashboards. Each dashboard contains two worksheets. Link of Tableau publication: https://public.tableau.com/views/GroupProjectPart3_17043908217930/Story1?:language=en-US&:display_count=n&:origin=viz_share_link