CAR ACCIDENT SEVERITY PREDICTOR

Report submitted in partial fulfillment of the requirement of

B.Tech.

In

Computer Science & Engineering/Information Technology

Under the Supervision of

Mr. Shailender Gaur

Asst. Professor, Information Technology Dept.

By

Name : **Anany Gupta**

Batch : **2018-2022**

Roll no. : **3-IT-18**

Enrolment no.: **00320803118**

Department of Information Technology

Bhagwan Parshuram Institute of Technology

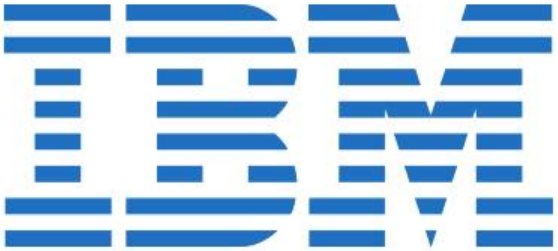
PSP-4, Sector-17, Delhi – 89

DECLARATION

This is to certify that Report titled “Car Accident Severity Predictor”, is submitted by us in partial fulfillment of the requirement for the award of degree of B.Tech. in Information Technology to BPIT Rohini Delhi affiliated to GGSIP University, Dwarka, Delhi. It comprises of our original work. The due acknowledgement has been made in the report for using other’s work.

Date: 20 October 2020 **Anany Gupta, 00320803118**

Course Certificates



05/28/2020

Anany Gupta

has successfully completed

Python for Data Science and AI

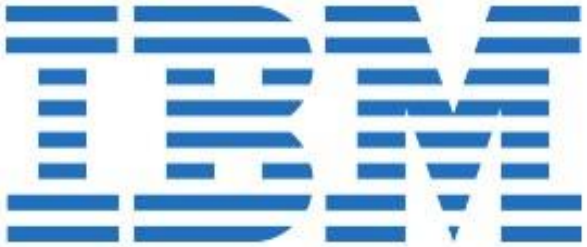
an online non-credit course authorized by IBM and offered through Coursera

Joseph Santarcangelo
Senior Data Scientist
IBM

COURSE CERTIFICATE

Verify at coursera.org/verify/5KBVKFKNKHW7

Coursera has confirmed the identity of this individual and their participation in the course.



08/09/2020

Anany Gupta

has successfully completed

Databases and SQL for Data Science

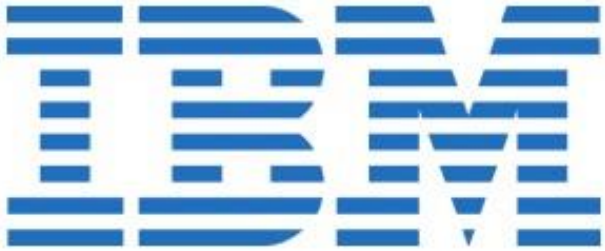
an online non-credit course authorized by IBM and offered through Coursera

Rav Ahuja
AI & Data Science Program Director
IBM Skills Network

COURSE CERTIFICATE

Verify at coursera.org/verify/ADND7NZE46GY

Coursera has confirmed the identity of this individual and their participation in the course.



09/06/2020

Anany Gupta

has successfully completed

Data Visualization with Python

an online non-credit course authorized by IBM and offered through Coursera

Alex Aklson, Ph.D.
Data Scientist

COURSE CERTIFICATE

Verify at coursera.org/verify/WFGBYG4PC9WW

Coursera has confirmed the identity of this individual and their participation in the course.

IBM

COURSE CERTIFICATE

05.10.2020

Anany Gupta

has successfully completed

Machine Learning with Python

an online non-credit course authorized by IBM and offered through Coursera

Saeed Aghabozorgi
Sr. Data Scientist
IBM

Joseph Santarcangelo
Senior Data Scientist
IBM

Verify at coursera.org/verify/BWA4TVF497PN

Coursera has confirmed the identity of this individual and their participation in the course.

IBM

COURSE CERTIFICATE

04.10.2020

Anany Gupta

has successfully completed

Applied Data Science Capstone

an online non-credit course authorized by IBM and offered through Coursera

Alex Aklson, Ph.D.
Data Scientist

Verify at coursera.org/verify/ST5MQ2M6WUJG

Coursera has confirmed the identity of this individual and their participation in the course.

9 Courses

- What is Data Science?
- Tools for Data Science
- Data Science Methodology
- Python for Data Science and AI
- Databases and SQL for Data Science
- Data Analysis with Python
- Data Visualization with Python
- Machine Learning with Python
- Applied Data Science Capstone

06.10.2020

Anany Gupta

has successfully completed the online, non-credit Professional Certificate

IBM Data Science

In this Professional Certificate learners developed and honed hands-on skills in Data Science and Machine Learning. Learners started with an orientation of Data Science and its Methodology, became familiar and used a variety of data science tools, learned Python and SQL, performed Data Visualization and Analysis, and created Machine Learning models. In the process they completed several labs and assignments on the cloud including a Capstone Project at the end to apply and demonstrate their knowledge and skills.

Rav Ahuja
AI & Data Science
Program Director
IBM Skills Network

Verify this certificate at:
coursera.org/verify/professional-cert/DFUHL2D5TGNE

IBM

CERTIFICATE
OF PARTICIPATION

THIS IS TO CERTIFY THAT Anany Gupta

ATTENDED THE IBM Hack Challenge 2020

CONDUCTED BY IBM INDIA UNIVERSITY RELATIONS.

Mona Bharadwaj
Head of University Relations,
IBM India

Date: April 2020 - August 2020

Certificate by Supervisor

This is to certify that Report titled “Car Accident Severity Predictor” is submitted by Anany Gupta (00320803118) in partial fulfilment of the requirement for the award of degree of B. Tech in Information Technology to BPIT Rohini affiliated to GGSIP University, Dwarka, Delhi. It is a record of the candidates own work carried out by them under my supervision. The matter embodied in this Report is original and has not been submitted for the award of any other degree.

Date: **Signature**

(Supervisor)

Certificate by HOD

This is to certify that Report titled “Car Accident Severity Predictor” is submitted by Anany Gupta (00320803118) in partial fulfillment of the requirement for the award of degree of B. Tech in Information Technology to BPIT Rohini affiliated to GGSIP University, Dwarka, Delhi. The matter embodied in this Report is original and has been dully approved for the submission.

Date: **Signature**

Dr. Abhishek Swaroop

ACKNOWLEDGEMENT

Every work accomplished is a pleasure – a sense of satisfaction. However, it would not have been possible without the kind support and help of many individuals in the organization. I would like to extend my sincere thanks to all of them.

In preparing the project report, I had to take the help and guideline of some respected persons, who deserve my greatest gratitude. The completion of this project gives me much pleasure. I would like to show my gratitude to Mr. Shailender Gaur, for cooperating with me and helping me successfully complete my project.

(Signature of the students with Date)

Table of Content

Table of Figures

Abstract

Throughout the world, roads are shared by cars, buses, trucks, motorcycles, mopeds, pedestrians, animals, taxis, and other travellers. Travel made possible by motor vehicles supports economic and social development in many countries. Yet each year, vehicles are involved in crashes that are responsible for millions of deaths and injuries. The number of road traffic deaths continues to rise steadily, reaching 1.35 million in 2016. However, the rate of death relative to the size of the world's population has remained constant. When considered in the context of the increasing global population and rapid motorization that has taken place over the same period, this suggests that existing road safety efforts may have mitigated the situation from getting worse. However, it also indicates that progress to realise Sustainable Development Goal (SDG) target 3.6 – which calls for a 50% reduction in the number of road traffic deaths by 2020 – remains far from sufficient.

Predicting crash severity is a crucial constituent of reducing the consequences of a car accident. This study developed machine learning models to predict car accident severity using accident related parameters. Three models were developed: K Nearest Neighbour (kNN), Decision Tree and Logistic Regression. Features that were easily identified with a little efforts on crash site investigation were used as an input so as to predict the severity of an accident and accordingly prepare for the help and treatment of the victims. The crash dataset of Seattle from 2004 to 2019 was used, mainly focusing on attributes related to weather, vehicular and road conditions. A random part of dataset was used to train the data set and the rest was used to test the developed model. The developed models were then compared on basis of 3 evaluation matrices: F1 score, accuracy and the Jaccard similarity score for classifying the accident into chance of property damage and chance of injury. This study concluded that the Decision Tree model proved to be the most accurate and reliable model among the three models.

Chapter 1

Introduction

1.1 Existing problem

As long as there have been roads there have been crashes. As vehicle technology improved and speeds increased, these crashes became more and more destructive. While the implementation of crumple zones, driver cages, steel bar doors, and airbags have served to lower the cost in human loss to these crashes, a more comprehensive vision is needed, a view that looks to avoid the crash altogether. Antilock brake systems, four wheel steering, better roadway lighting, and strictly enforced geometric design are certainly a step in the right direction. Each only concentrates on prevention at the driver level attempting to affect how individuals drive. While this paradigm has yielded many advances, different approaches maybe equally effective.

Road accidents in India claimed over 1.5 lakh lives in 2018. The ministry of road transport and highways issued a report on Road accidents in India in 2018, which showed that road accidents last year increased by 0.46% as compared to 2017.A total of 4,67,044 road accidents have been reported by States and Union Territories (UTs) in the calendar year 2018, claiming 1,51,417 lives and causing injuries to 4,69,418 persons. Over-speeding accounted for 64.4% of the persons killed. India, ranks 1st in the number of road accident deaths across the 199 countries reported in the World Road Statistics, 2018 followed by China and US. As per the WHO Global Report on Road Safety 2018, India accounts for almost 11% of the accident related deaths in the World. National Highways which comprise of 1.94 percent of total road network, accounted for 30.2 per cent of total road accidents and 35.7 per cent of deaths in 2018.State Highways which account for 2.97% of the road length accounted for 25.2 percent and 26.8 percent of accidents and deaths respectively. The major cause of such high numbers and percentages is the poor safety measures taken by the people and Govt. of India.

Supervised Machine Learning Algorithms are being used in various field be in Manufacturing Industries from optimising work to setting up tasks, in Metrological science for forecasting weather, Agricultural industry where it is used to improve the productivity and quality of the crops in the agriculture sector. The seed retailers use this agriculture technology to churn the data to create better crops, and also in Medical Science where it can be used to identify various diseases by analysing pattern in the occurrence of symptoms or health conditions of an individual to get an accurate diagnosis of problem.

As progress is made in the prevention and control of infectious diseases, the relative contribution of deaths from non-communicable diseases and injuries has increased. Road traffic injuries are the eighth leading cause of death for all age groups. More people now die as a result of road traffic injuries than from HIV/AIDS, tuberculosis or diarrhoeal diseases. Road traffic injuries are currently the leading cause of death for children and young adults aged 5–29 years, signalling a need for a shift in the current child and adolescent health agenda which, to date, has largely neglected road safety.

This makes it necessary to have a system that could predict the severity of a crash and its frequency along with identifying the patterns in the variable affecting/ causing an accident so that those reasons can be avoided and necessary steps for emergency situations can be deployed to reduce the chances and severity of car crashes.

1.2Proposed Solution

My approach was to use a number of different models to classify the car accident to different severity codes and then choose the one which provides the most accurate and reliable results.

In my presented prototype I have predicted the classifications for the region of Seattle USA, I build classification models to predict accident severity on basis of 5 conditions

1. Road Conditions: whether the road were dry or not, whether they were muddy or had stagnant water or snowy,

2. Weather Conditions: describing the weather of the seen whether it was cloudy/rainy/clear/sunny/stormy etc.

3. Light Conditions: It describes whether it was day or dark, whether the street lights were on or not etc.

4. Address Type: It is the location where it happened whether it was a block or an ally or an intersection.

5. Collision Type: It describes the nature of accident.

Chapter 2

Literature review & Scope of the work

2.1Literature review

The occurrence of a car accident can be related to various factors, it may be the conditions related to road or the vehicle or the condition of person inside the vehicle each contribute in some form or another. Hence it is crucial to take such factors into account to predict a road accident's severity and taking necessary steps to prevent it from happening and steps that would be necessary if the certain accident occurs.

A reliable and accurate prediction model is required so as to gain correct knowledge of occurrence of certain accident at a particular coordinate to apply appropriate safety measures and emergency rescues.

2.1.1 Traffic accident severity prediction using a novel multi-objective genetic algorithm.

Here the author has discussed about the disadvantages of using conventional methods for classification of data and has proposed the use of novel multi-objective genetic algorithm which can predict traffic accident severity according to user's preferences removing most of the disadvantages of the conventional models.

2.1.2 Classification of Vehicle Collision Patterns in Road Accidents using Data Mining Algorithms.

This paper emphasizes the importance of Data Mining classification algorithms in predicting the vehicle collision patterns occurred in training accident data set. This paper is aimed at deriving classification rules which can be used for the prediction of manner of collision.

2.1.3 Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data.

This study aims to classify the injury severity in motor-vehicle crashes with both high accuracy and sensitivity rates. Accident severity datasets are typically imbalanced, with the non-fatal class containing disproportionally more data points compared to the fatal class, which can lead to an unreliable model and weak model. This study shows how we can tackle this imbalanced dataset to develop a strong model.

2.1.4Using Machine Learning to Predict Car Accident Risk.

In this paper the author has discussed about building a ML model to predict the count of car accident before it happens using a dataset collected from different regions the author has constructed a Gradient Boosting model to generate high quality results.

2.2 Scope of Work

The Seattle government is going to prevent avoidable car accidents by employing methods that alert drivers, health system, and police to remind them to be more careful in critical situations.

In most cases, not paying enough attention during driving, abusing drugs and alcohol or driving at very high speed are the main causes of occurring accidents that can be prevented by enacting harsher regulations. Besides the aforementioned reasons, weather, visibility, or road conditions are the major uncontrollable factors that can be prevented by revealing hidden patterns in the data and announcing warning to the local government, police and drivers on the targeted roads.

The target audience of the project is local Seattle government, police, rescue groups, and last but not least, car insurance institutes. The model and its results are going to provide some advice for the target audience to make insightful decisions for reducing the number of accidents and injuries for the city.

Chapter 3

Data Analysis

3.1Data Collection

The data was collected by the Seattle Police Department and Accident Traffic Records Department from 2004 to 2019.

The dataset consists of 37 independent variables describing the details of each accident including the weather conditions, collision type, date/time of accident and location (latitude and longitude) and 194,673 rows.

The dependent variable, “SEVERITYCODE”, contains numbers that correspond to 2 different levels of severity caused by an accident.

Severity codes are as follows:

1: Chance of Property Damage

2: Chance of Injury

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight
...	...	...	...	...	...	...	...	...	...	...	...	...	...
194668	2	-122.290826	47.565408	219543	309534	310814	E871089	Matched	Block	NaN	...	Dry	Daylight
194669	1	-122.344526	47.690924	219544	309085	310365	E876731	Matched	Block	NaN	...	Wet	Daylight
194670	2	-122.306689	47.683047	219545	311280	312640	3809984	Matched	Intersection	24760.0	...	Dry	Daylight
194671	2	-122.355317	47.678734	219546	309514	310794	3810083	Matched	Intersection	24349.0	...	Dry	Dusk
194672	1	-122.289360	47.611017	219547	308220	309500	E868008	Matched	Block	NaN	...	Wet	Daylight

Figure 1: Data Set

```
data.dtypes
```

```
SEVERITYCODE        int64
X                 float64
Y                 float64
OBJECTID            int64
INCKEY              int64
COLDETKEY           int64
REPORTNO           object
STATUS             object
ADDRTYPE           object
INTKEY            float64
LOCATION           object
EXCEPTRSNCODE      object
EXCEPTRSNDESC      object
SEVERITYCODE.1      int64
SEVERITYDESC       object
COLLISIONTYPE      object
PERSONCOUNT         int64
PEDCOUNT            int64
PEDCYLCOUNT         int64
VEHCOUNT            int64
INCDATE            object
INCDTTM            object
JUNCTIONTYPE       object
SDOT_COLCODE        int64
SDOT_COLDESC       object
INATTENTIONIND     object
UNDERINFL          object
WEATHER            object
ROADCOND           object
LIGHTCOND          object
PEDROWNOTGRNT      object
SDOTCOLNUM        float64
SPEEDING           object
ST_COLCODE         object
ST_COLDESC         object
SEGLANEKEY          int64
CROSSWALKKEY        int64
HITPARKEDCAR       object
dtype: object
```

Figure 2: Data Variables

3.2 Data Pre-processing

3.2.1. Dataset includes unnecessary/redundant columns

The accident dataset includes many columns of metadata (such as incident report numbers) and columns which duplicate information which is already included in other columns (such as a text field "SEVERITYDESC" which provides a written definition of the accompanying accident severity code, the target variable). Columns which include unnecessary/redundant to the dataset.

After analysing the data set, I have decided to focus on only six features Severity Code, Address Type, Collision Type, Weather, Road Conditions and Light Conditions.

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND
0	2	Intersection	Angles	Overcast	Wet	Daylight
1	1	Block	Sideswipe	Raining	Wet	Dark - Street Lights On
2	1	Block	Parked Car	Overcast	Dry	Daylight
3	1	Block	Other	Clear	Dry	Daylight
4	2	Intersection	Angles	Raining	Wet	Daylight
...	...	...	...	...	...	...
194668	2	Block	Head On	Clear	Dry	Daylight
194669	1	Block	Rear Ended	Raining	Wet	Daylight
194670	2	Intersection	Left Turn	Clear	Dry	Daylight
194671	2	Intersection	Cycles	Clear	Dry	Dusk
194672	1	Block	Rear Ended	Clear	Wet	Daylight

194673 rows × 6 columns

Figure 3: Choosen Attriutes

3.2.2. Data incompleteness

Around 4% of the accidents in the refined dataset are missing one or more key features, including in some cases the target variable (accident severity code) and in others, are missing information about weather or road conditions. As the purpose of building the model is to see how these various features interact and influence the overall accident severity, data entries which are missing one or more of these key features are not useful, and were removed from the dataset.

```
In [66]: pro_data.shape

Out[66]: (194673, 6)

In [68]: pro_data=pro_data.dropna()
         pro_data.shape

Out[68]: (187504, 6)
```

Figure 4: Removal of missing values

3.2.3. Data Visualisation:

Visualising data is the most crucial part of data analysis as it is something that enables you to understand data in a better and an easy way as pictorial representations of data is the easiest representation to understand.

For data visualisation I have used the Matplotlib and Seaborn libraries to develop the histograms given below:-

```
pro_data1=pro_data[['SEVERITYCODE','ADDRTYPE']]
ax1=sns.catplot(x='ADDRTYPE',hue='SEVERITYCODE',kind='count',data=pro_data1,height=6)
plt.title("Number of car accidents by Collision Address Type")
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='large')
```

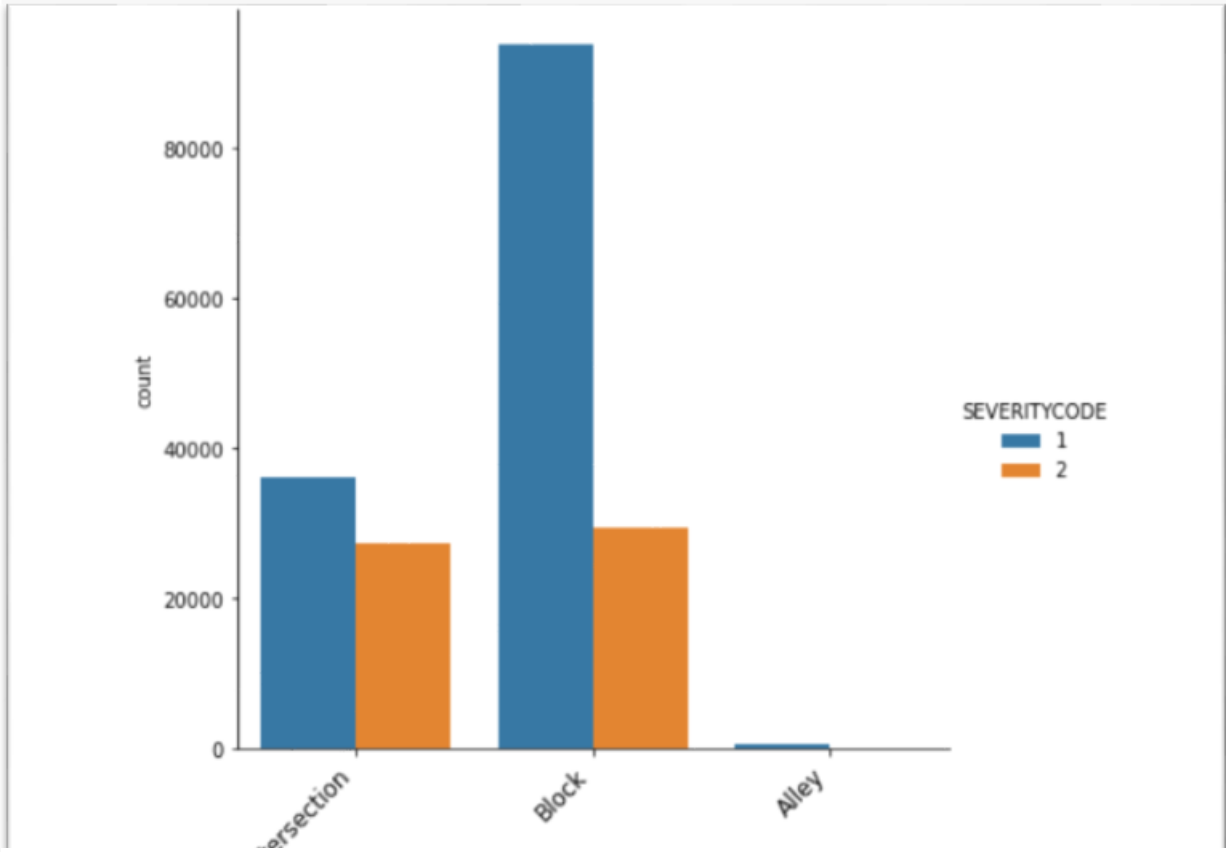


Figure 5: Address Type vs Number of Accidents

```
ready_2=ready_data[['SEVERITYCODE','ROADCOND']]
sns.catplot(x='ROADCOND',hue='SEVERITYCODE',kind='count',data=ready_2,height=6)
plt.title("Number of car accidents by Road Conditions Type")
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='large')
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7]), <a list of 8 Text major ticklabel objects>)
```

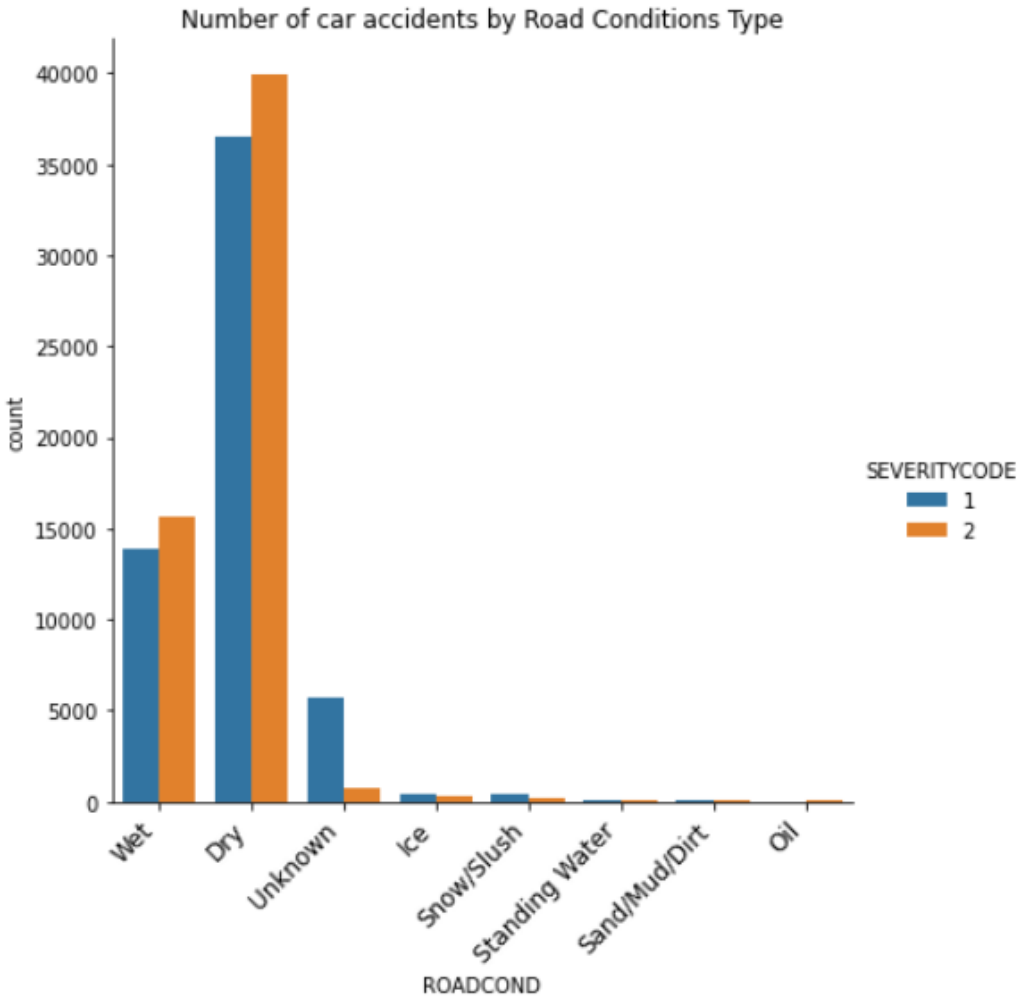


Figure 6 Number of Accidents distributed by Road Conditions

```
ready_3=ready_data[['SEVERITYCODE','LIGHTCOND']]
ax=sns.catplot(x='LIGHTCOND',hue='SEVERITYCODE',kind='count',data=ready_3,height=6)
plt.title("Number of car accidents by Light Conditions")
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='large')
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7]), <a list of 8 Text major ticklabel objects>)
```

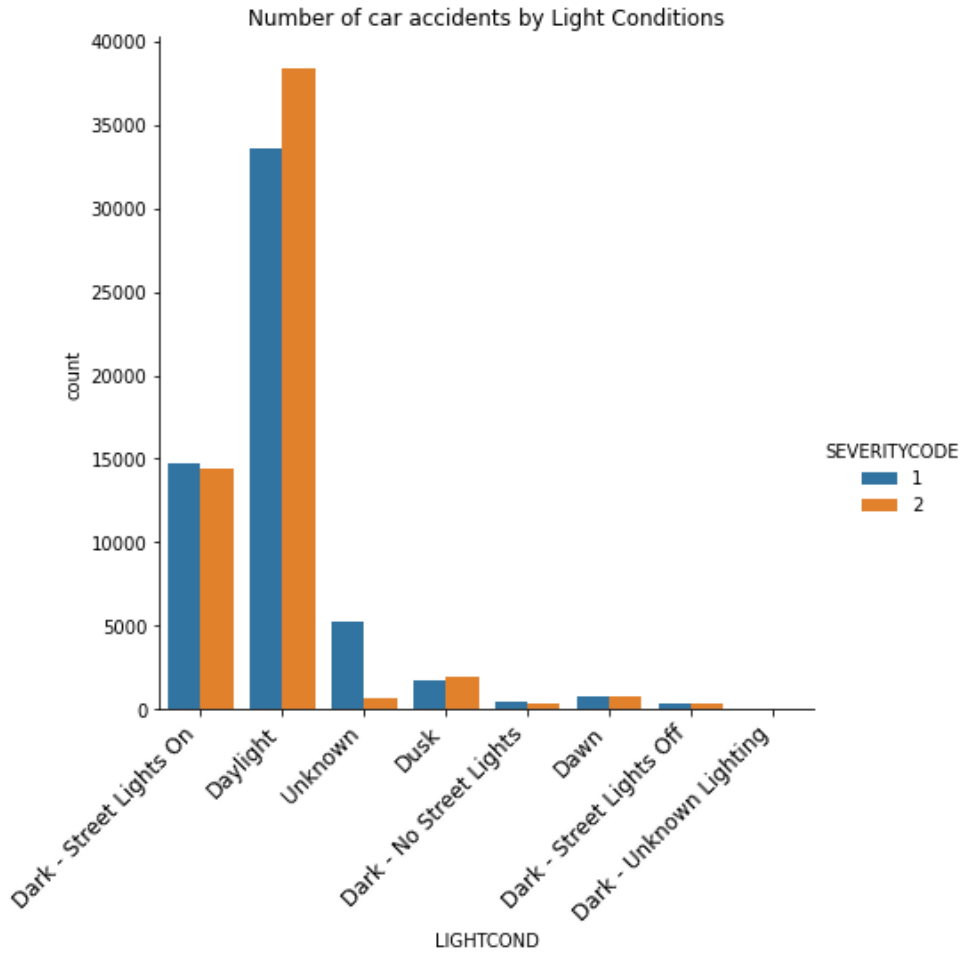


Figure 7: Number of Accidents distributed by Light Conditions

```
ready_3=ready_data[['SEVERITYCODE','WEATHER']]
sns.catplot(x='WEATHER',hue='SEVERITYCODE',kind='count',data=ready_3,height=6)
plt.title("Number of car accidents by Weather Conditions")
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='large')
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 <a list of 10 Text major ticklabel objects>)
```

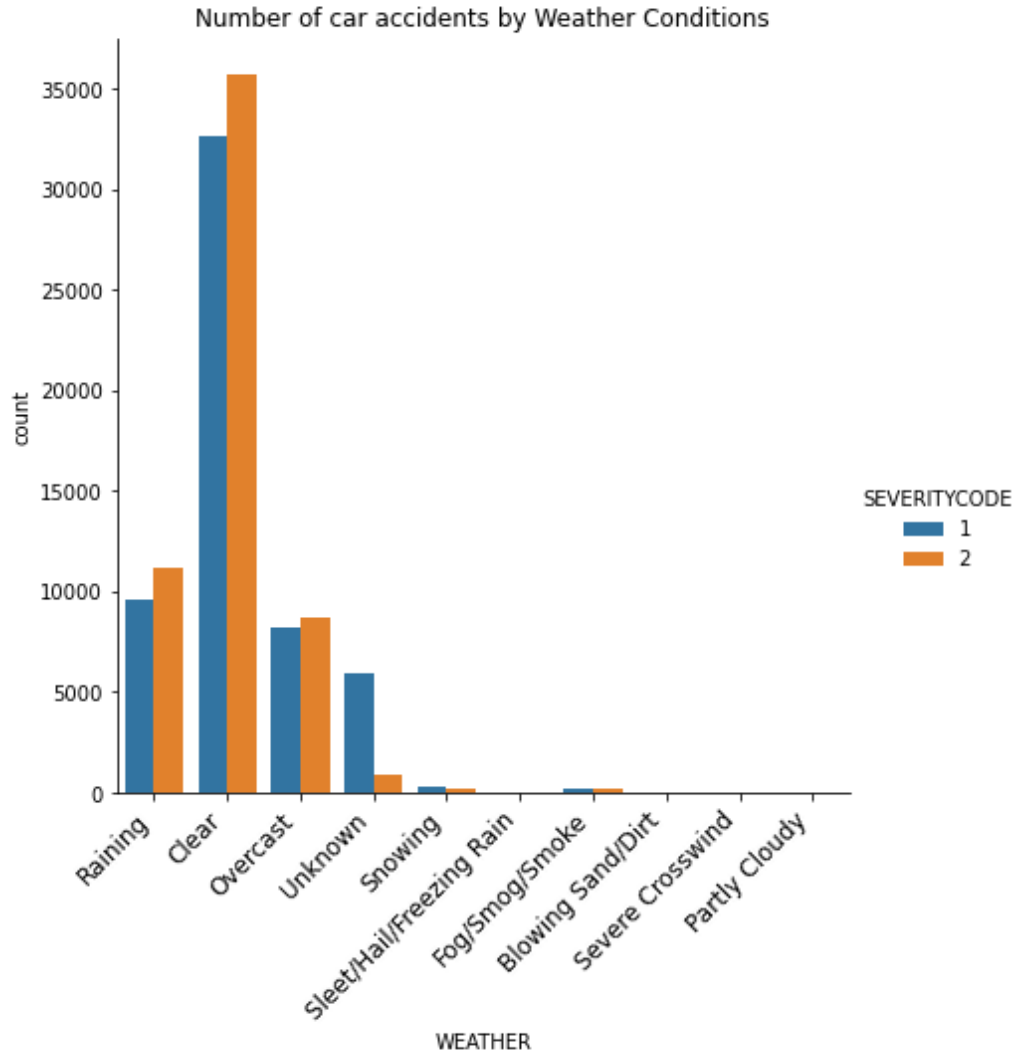


Figure 8: Number of Accidents Distributed by Weather

```
ready_5=ready_data[['SEVERITYCODE','COLLISIONTYPE']]
sns.catplot(x='COLLISIONTYPE',hue='SEVERITYCODE',kind='count',data=ready_5,height=6)
plt.title("Number of car accidents by Weather Conditions")
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='large')
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 <a list of 10 Text major ticklabel objects>)
```

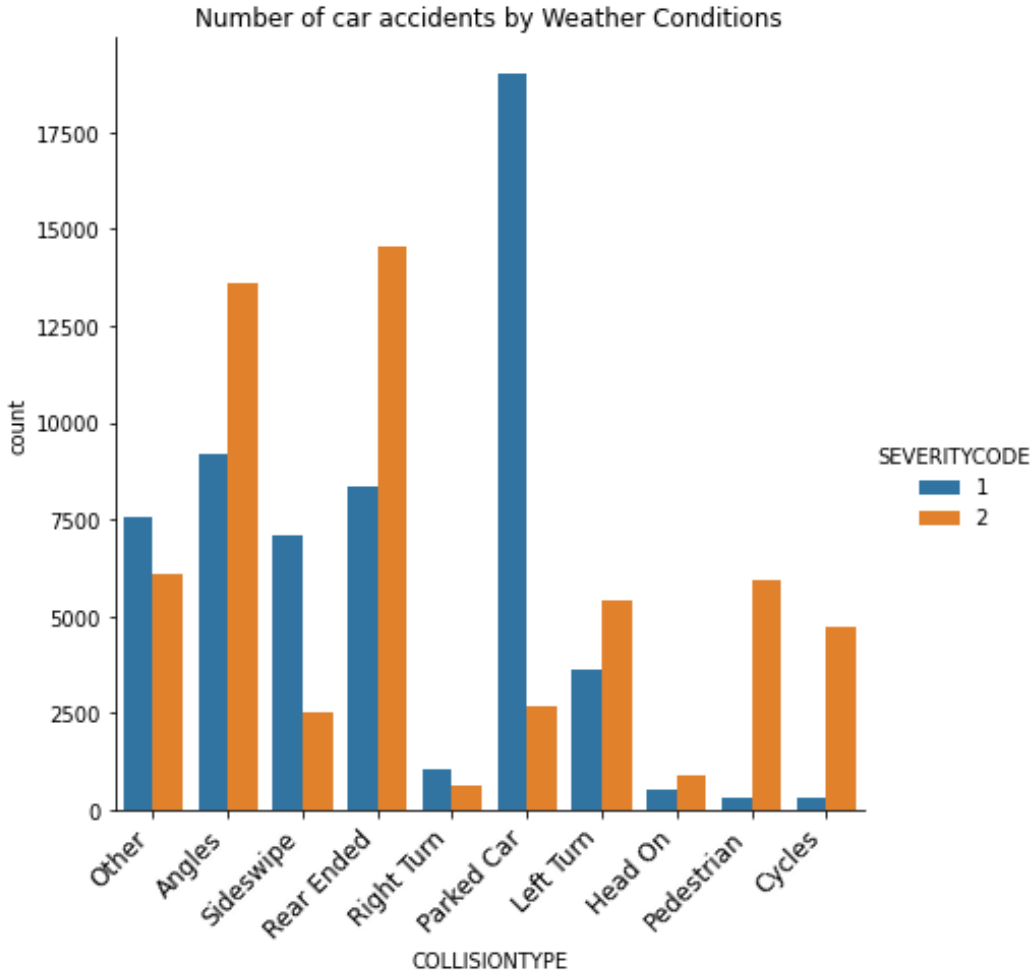


Figure 9: Number of Accidents Distributed by Collision Type

3.2.4 Data is imbalanced:

After visualising and analysing the data, I checked difference in values for different severity codes and the results show, the target feature that is imbalanced and needs to be balanced. Furthermore, the attributes Road Condition, Weather and Light Condition had values as 'Unknown' and 'Other' which basically meant the same so I changed the values for the 'Other' to 'Unknown'. And then resampled the data set so that the number of rows for Severity Code '1' and number of rows for Severity Code '2' are equal.

```
In [72]: ready_data['SEVERITYCODE'].value_counts()

Out[72]: 2    56870
         1    56870
         Name: SEVERITYCODE, dtype: int64
```

```
In [38]: df1=pro_data[pro_data.SEVERITYCODE==1]
         df2=pro_data[pro_data.SEVERITYCODE==2]
         df_resampled=resample(df1 , replace=False ,n_samples=56870,random_state=111)

         ready_data=pd.concat([df_resampled,df2])

         ready_data['SEVERITYCODE'].value_counts()
         #This is done so that the model is trained to predict the severity for any case with equal probability

Out[38]: 2    56870
         1    56870
         Name: SEVERITYCODE, dtype: int64
```

Figure 10: Resampling Dataset

Doing so resulted in the data frame index to become irregular, so the data frame index had to be reset. For this I used reset_index() function resetting the index starting from 0.

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND
67964	1	Block	Other	Raining	Wet	Dark - Street Lights On
171470	1	Block	Angles	Clear	Dry	Daylight
70537	1	Intersection	Angles	Overcast	Wet	Daylight
178911	1	Block	Sideswipe	Clear	Dry	Daylight
73038	1	Block	Rear Ended	Overcast	Dry	Daylight

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND
0	1	Block	Other	Raining	Wet	Dark - Street Lights On
1	1	Block	Angles	Clear	Dry	Daylight
2	1	Intersection	Angles	Overcast	Wet	Daylight
3	1	Block	Sideswipe	Clear	Dry	Daylight
4	1	Block	Rear Ended	Overcast	Dry	Daylight

Figure 11: Reindexing

3.2.5. Presence of categorical/non-numeric data:

Machine Learning models require numerical data, and cannot handle alphanumeric strings. For example, each entry in the "WEATHER" column contains a text string which takes one of eleven values (e.g. "Clear", "Rain", "Snow", etc.) which describes the prevailing weather conditions at the time of the accident.

So we first change all the object type data values to category and then use label encoding to encode these categories into numerical values which are stored in dataset as Variable_category.

Label Encoding: Label encoding is simply converting each value in a column to a number. Here each value is converted to a numeric label which can be used as a replacement of the original non numeric value as non-numeric values doesn't work with ML models.

```
In [49]: ready_data.dtypes

Out[49]: SEVERITYCODE      int64
         ADDRTYPE         object
         COLLISIONTYPE    object
         WEATHER          object
         ROADCOND         object
         LIGHTCOND        object
         dtype: object
```

```
In [50]: ready_data["COLLISIONTYPE"] = ready_data["COLLISIONTYPE"].astype('category')
         ready_data['WEATHER'] = ready_data['WEATHER'].astype('category')
         ready_data['ROADCOND'] = ready_data["ROADCOND"].astype('category')
         ready_data['LIGHTCOND'] = ready_data['LIGHTCOND'].astype('category')
         ready_data['ADDRTYPE'] = ready_data['ADDRTYPE'].astype('category')
         ready_data.dtypes

Out[50]: SEVERITYCODE        int64
         ADDRTYPE         category
         COLLISIONTYPE    category
         WEATHER          category
         ROADCOND         category
         LIGHTCOND        category
         dtype: object
```

Figure 13: Categorising Object data

```
ready_data['ADDRTYPE_Category'] = ready_data['ADDRTYPE'].cat.codes
ready_data["COLLISIONTYPE_Category"] = ready_data["COLLISIONTYPE"].cat.codes
ready_data['LIGHTCOND_Category'] = ready_data['LIGHTCOND'].cat.codes
ready_data['ROADCOND_Category'] = ready_data["ROADCOND"].cat.codes
ready_data['WEATHER_Category'] = ready_data['WEATHER'].cat.codes

ready_data.head()
```

TYPE	WEATHER	ROADCOND	LIGHTCOND	ADDRTYPE_Category	COLLISIONTYPE_Category	LIGHTCOND_Category	ROADCOND_Category	WEATHER_Category
Other	Raining	Wet	Dark - Street Lights On	1	4	2	7	5
ngles	Clear	Dry	Daylight	1	0	5	0	1
ngles	Overcast	Wet	Daylight	2	0	5	7	3
swipe	Clear	Dry	Daylight	1	9	5	0	1
nded	Overcast	Dry	Daylight	1	7	5	0	3

Figure 12: Label Encoding

```
r=ready_data[['COLLISIONTYPE','COLLISIONTYPE_Category']]
r = r.groupby('COLLISIONTYPE').apply(lambda x: x['COLLISIONTYPE_Category'].unique())
r
```

```
COLLISIONTYPE
Angles        [0]
Cycles        [1]
Head On       [2]
Left Turn     [3]
Other         [4]
Parked Car    [5]
Pedestrian    [6]
Rear Ended    [7]
Right Turn    [8]
Sideswipe     [9]
dtype: object
```

Figure 15Encoded Collisions

```
r=ready_data[['ADDRTYPE','ADDRTYPE_Category']]
r = r.groupby('ADDRTYPE').apply(lambda x: x['ADDRTYPE_Category'].unique())
r
```

```
ADDRTYPE
Alley           [0]
Block           [1]
Intersection    [2]
dtype: object
```

```
r=ready_data[['WEATHER','WEATHER_Category']]
r = r.groupby('WEATHER').apply(lambda x: x['WEATHER_Category'].unique())
r
```

```
WEATHER
Blowing Sand/Dirt           [0]
Clear                       [1]
Fog/Smog/Smoke              [2]
Overcast                    [3]
Partly Cloudy               [4]
Raining                     [5]
Severe Crosswind            [6]
Sleet/Hail/Freezing Rain    [7]
Snowing                     [8]
Unknown                     [9]
dtype: object
```

Figure 14: Encoded Address and Weather

```
r=ready_data[['ROADCOND','ROADCOND_Category']]
r = r.groupby('ROADCOND').apply(lambda x: x['ROADCOND_Category'].unique())
r
```

```
ROADCOND
Dry               [0]
Ice               [1]
Oil               [2]
Sand/Mud/Dirt     [3]
Snow/Slush        [4]
Standing Water    [5]
Unknown           [6]
Wet               [7]
dtype: object
```

```
r=ready_data[['LIGHTCOND','LIGHTCOND_Category']]
r = r.groupby('LIGHTCOND').apply(lambda x: x['LIGHTCOND_Category'].unique())
r
```

```
LIGHTCOND
Dark - No Street Lights     [0]
Dark - Street Lights Off    [1]
Dark - Street Lights On     [2]
Dark - Unknown Lighting     [3]
Dawn                        [4]
Daylight                    [5]
Dusk                        [6]
Unknown                     [7]
dtype: object
```

Figure 16: Encoded Road and Light Conditions

3.3 Final dataset

After all the pre-processing the final dataset I had contained 11 columns with the categories and their encoded labels.

```
ready_data.dtypes
```

```
SEVERITYCODE                int64
ADDRTYPE                 category
COLLISIONTYPE            category
WEATHER                  category
ROADCOND                 category
LIGHTCOND                category
ADDRTYPE_Category            int8
COLLISIONTYPE_Category       int8
LIGHTCOND_Category           int8
ROADCOND_Category            int8
WEATHER_Category             int8
```

Figure 17:Final Data Variables

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND	ADDRTYPE_Category	COLLISIONTYPE_Category	LIGHTCOND_Ca
0	1	Block	Other	Raining	Wet	Dark - Street Lights On	1	4	
1	1	Block	Angles	Clear	Dry	Daylight	1	0	
2	1	Intersection	Angles	Overcast	Wet	Daylight	2	0	
3	1	Block	Sideswipe	Clear	Dry	Daylight	1	9	
4	1	Block	Rear Ended	Overcast	Dry	Daylight	1	7	
...	...	...	...	...	...	...	...	...	
113735	2	Block	Angles	Raining	Wet	Daylight	1	0	
113736	2	Block	Angles	Clear	Wet	Daylight	1	0	
113737	2	Block	Head On	Clear	Dry	Daylight	1	2	
113738	2	Intersection	Left Turn	Clear	Dry	Daylight	2	3	
113739	2	Intersection	Cycles	Clear	Dry	Dusk	2	1	

113740 rows × 11 columns

Figure 18: Final Dataset

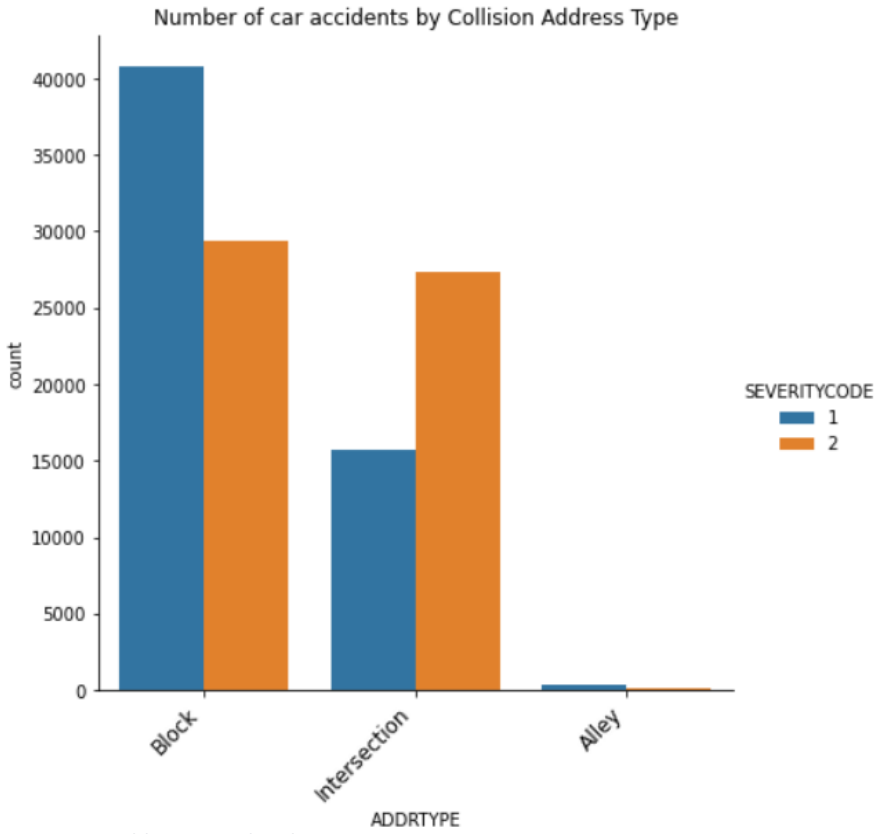


Figure 19: Address type distribution

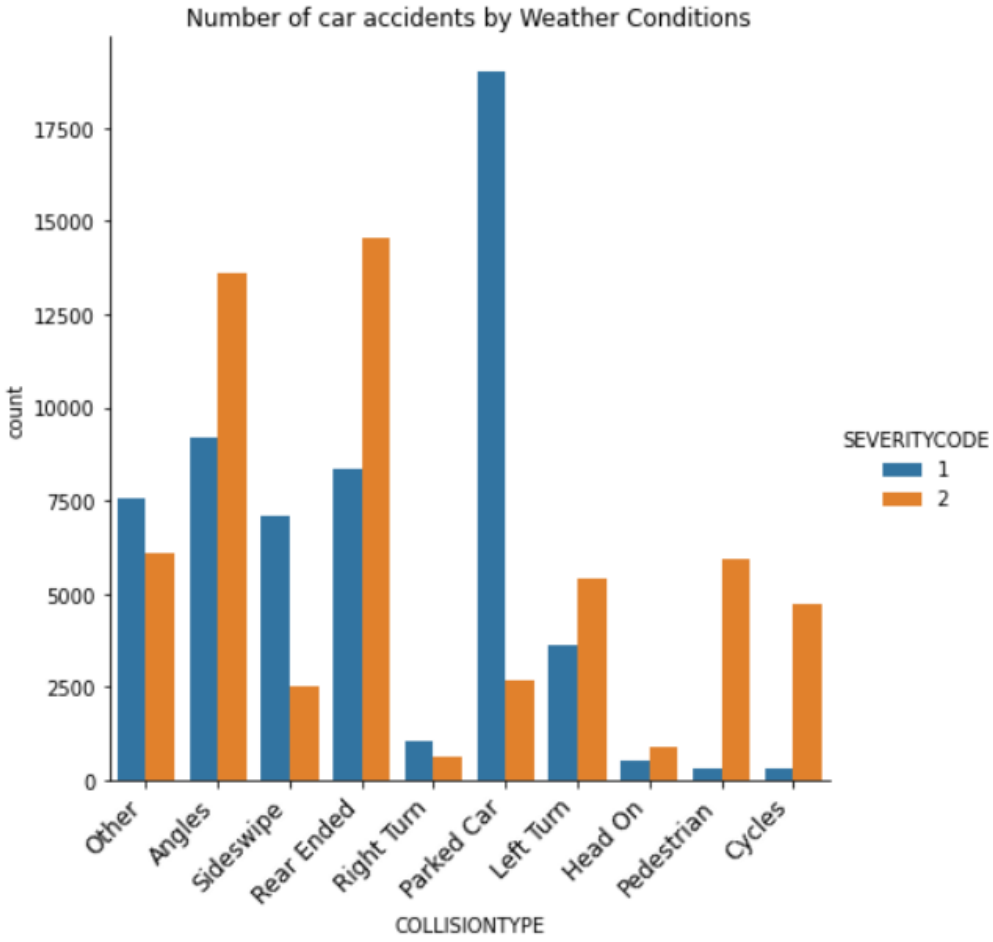


Figure 20: Collision type distribution

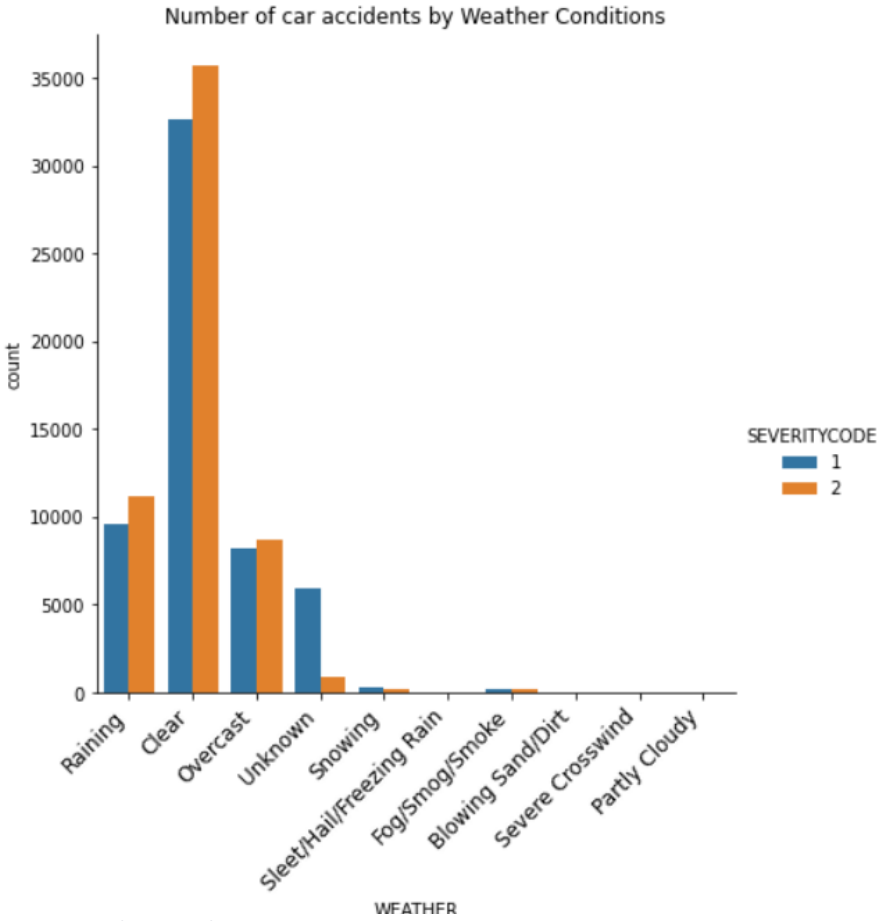


Figure 21: Weather Distribution

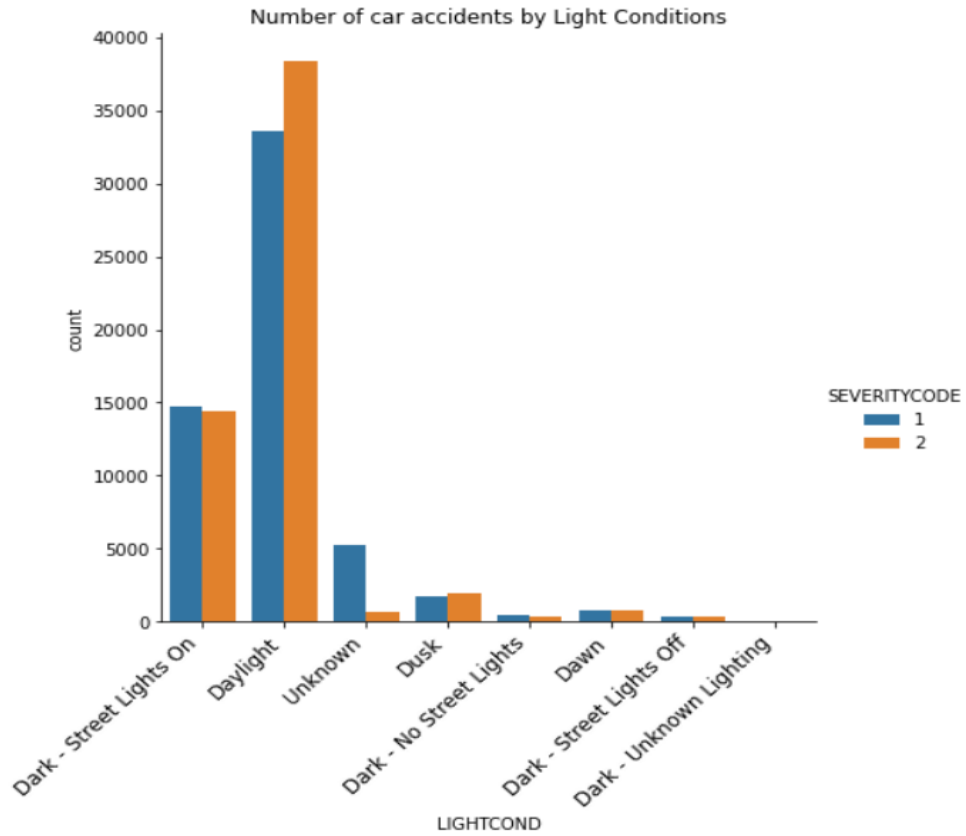


Figure 23: Lighting Conditions Distribution

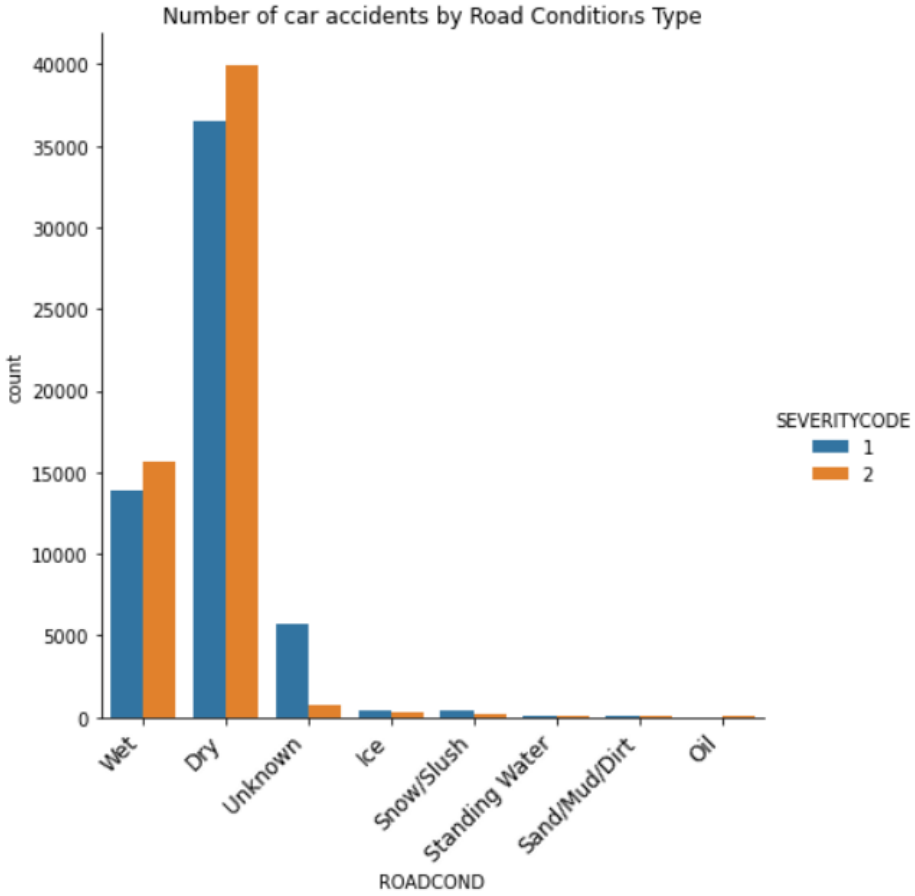


Figure 22: Road Conditions Distribution

Chapter 4

Model Development

In order to develop a model for predicting accident severity, the re-sampled, cleaned dataset was split in to testing and training sub-samples (containing 21% and 79% of the samples, respectively) using the scikit learn "train_test_split" method. In total, three models were trained and evaluated.

4.1 k-Nearest Neighbour Model (kNN model)

kNN models seek to categorise the outcome of an unknown data sample based on its proximity in the multi-dimensional hyperspace of the feature set to its "k" nearest neighbours, which have known outcomes.

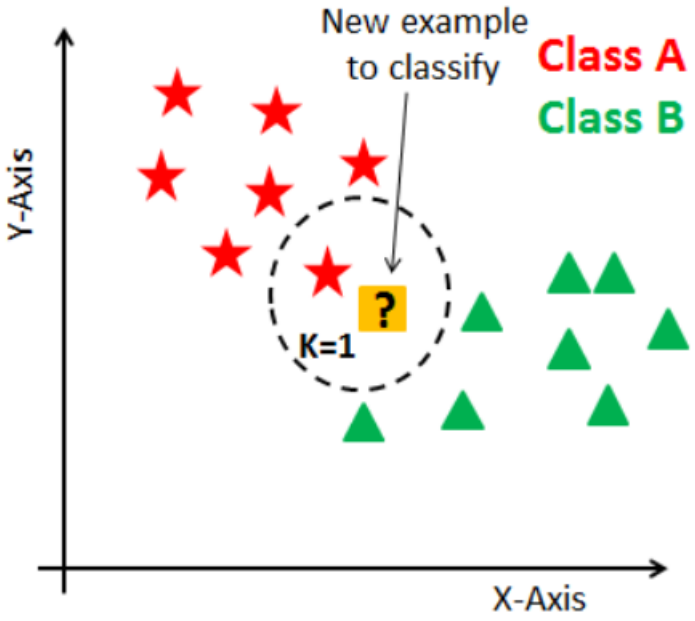


Figure 24: What is Knn model

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Establishing the value of "k" which optimises the model's accuracy (between 1 and the total number of samples in the dataset) is an empirical undertaking: if too-few neighbouring datapoints are used, the model is susceptible to being dominated by noise, however if too many neighbours are included in the classification, the model risks losing all diagnostic power completely.

NN models were built for k=1–100 using the kNeighborsClassifier function from scikit learn. The model is optimised at k=80.

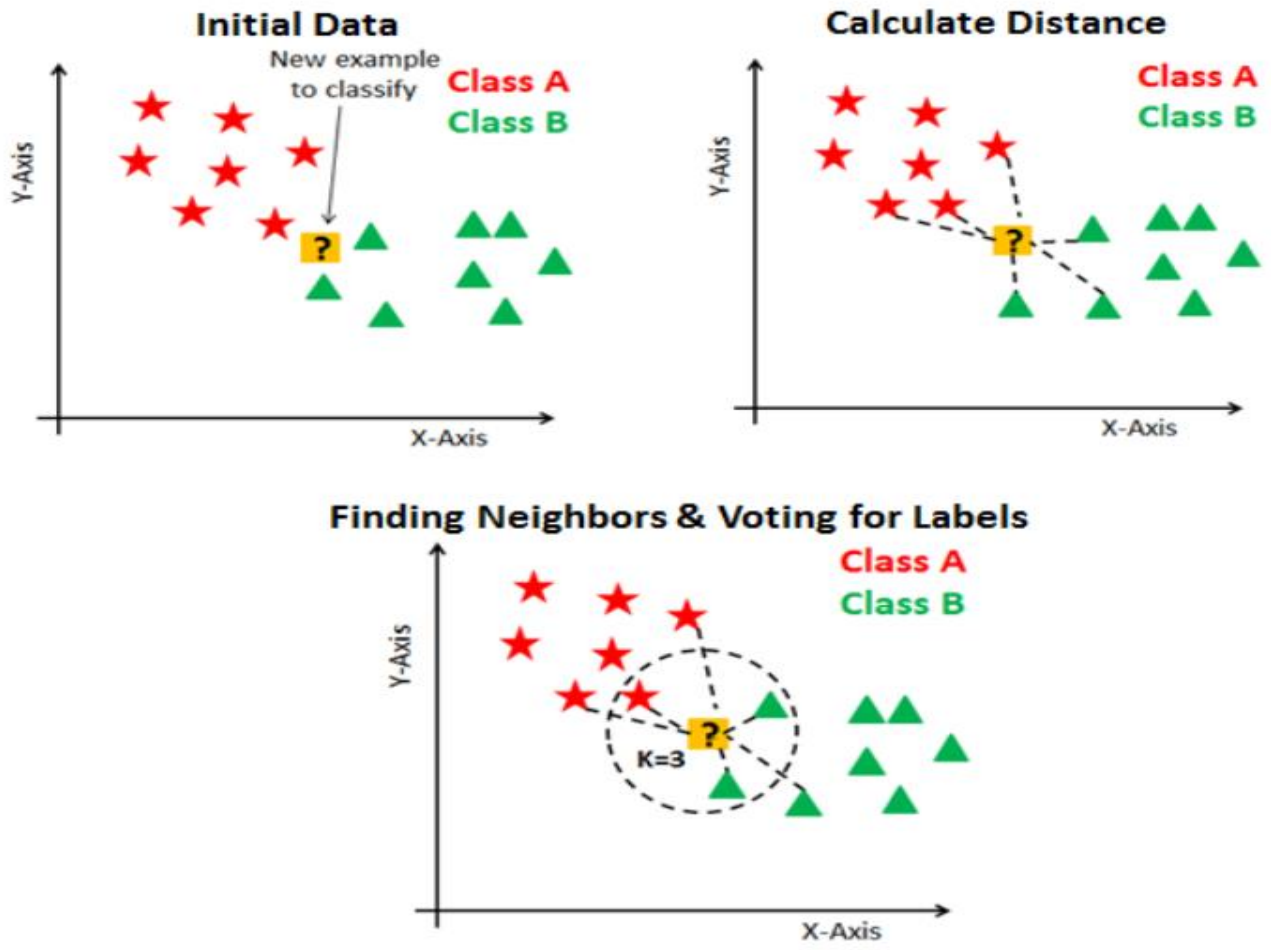


Figure 25: How KNN works

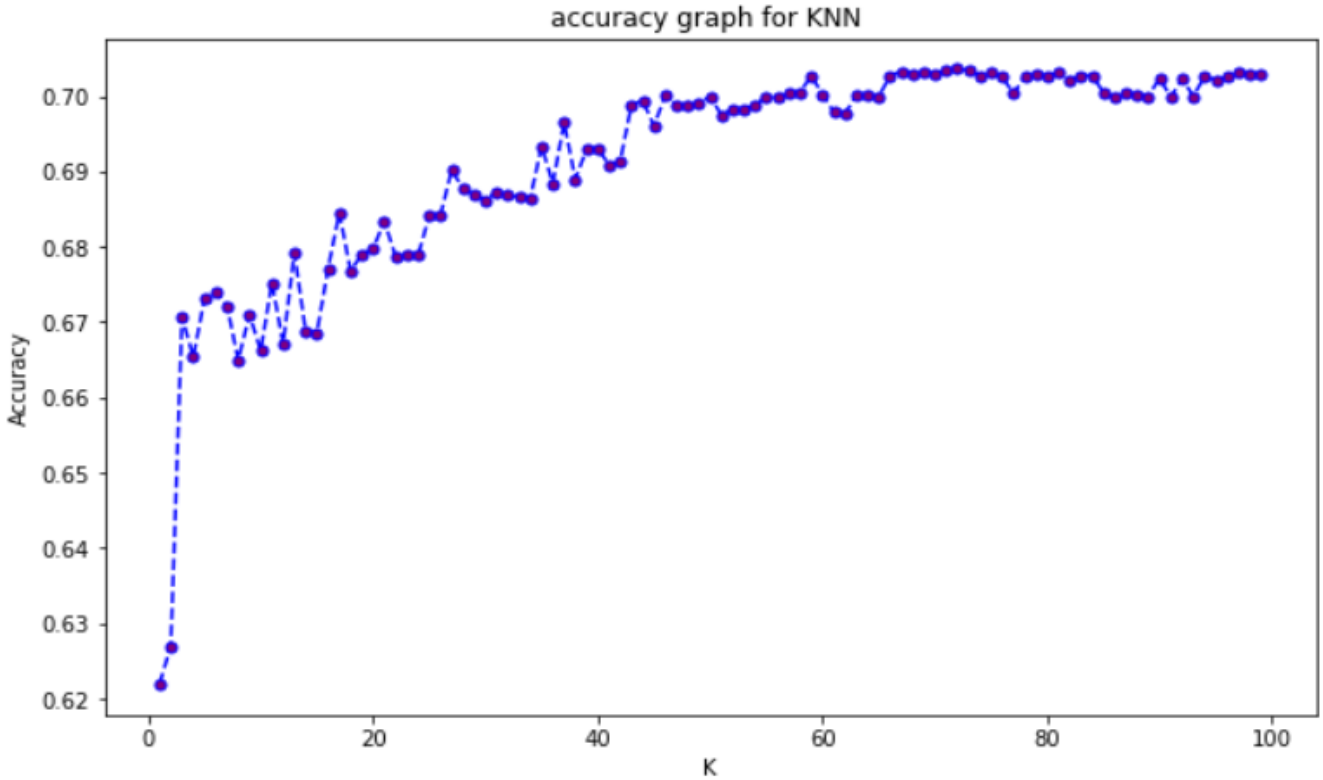


Figure 26: Accuracy Score for KNN model for different values of k

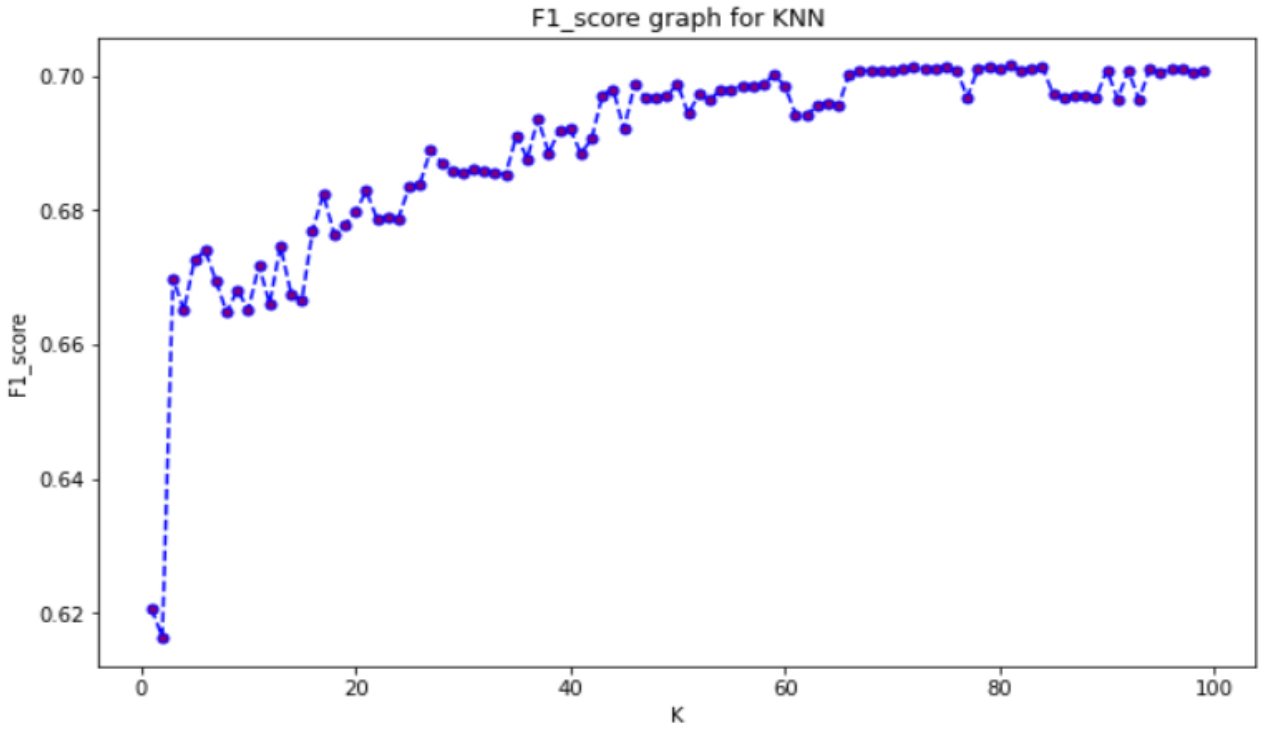


Figure 28F1 Score for different values of k

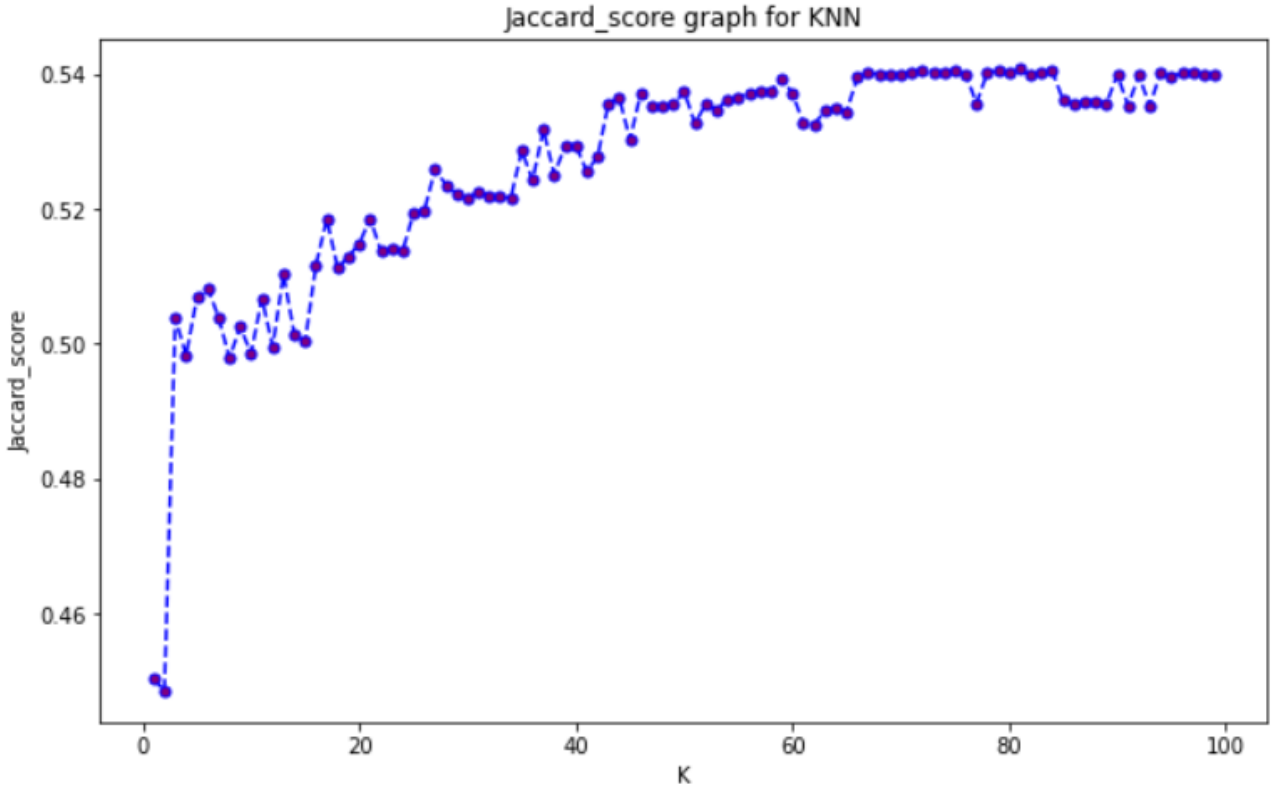


Figure 27: Jaccard Similarity for different values of K

4.2. Decision Tree Model

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

Decision tree models identify the key features on which the data can be partitioned (and the thresholds at which to partition the data) in the hope of arriving, after some iterations, at “leaves” which contain only accidents belonging to one target variable value (in this case, accident severity code).

Decision Trees were made for maxdepth in range for (1, 30) to find the depth with maximum accuracy which came out to be at max_depth=7.

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

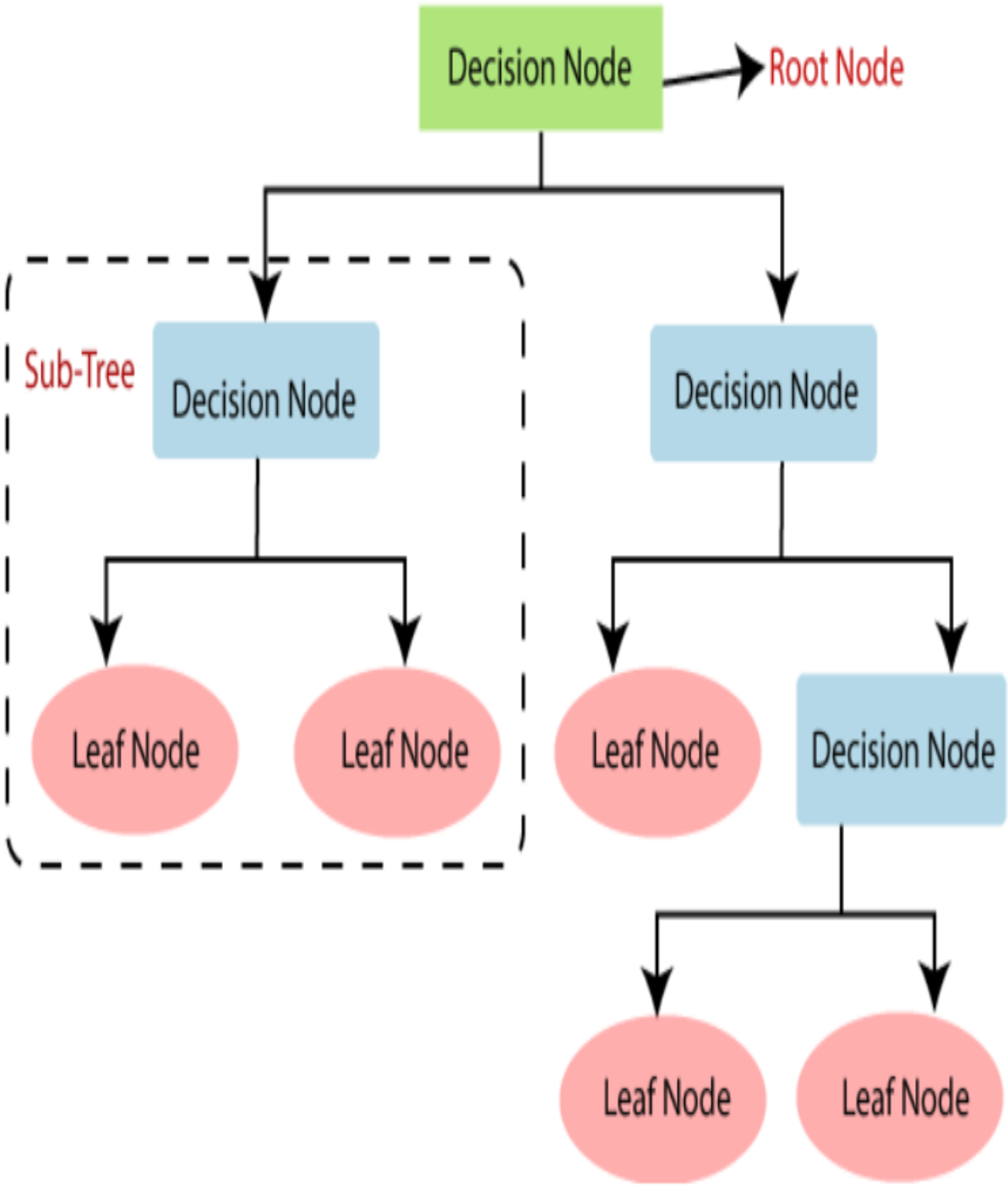


Figure 29Decision tree model

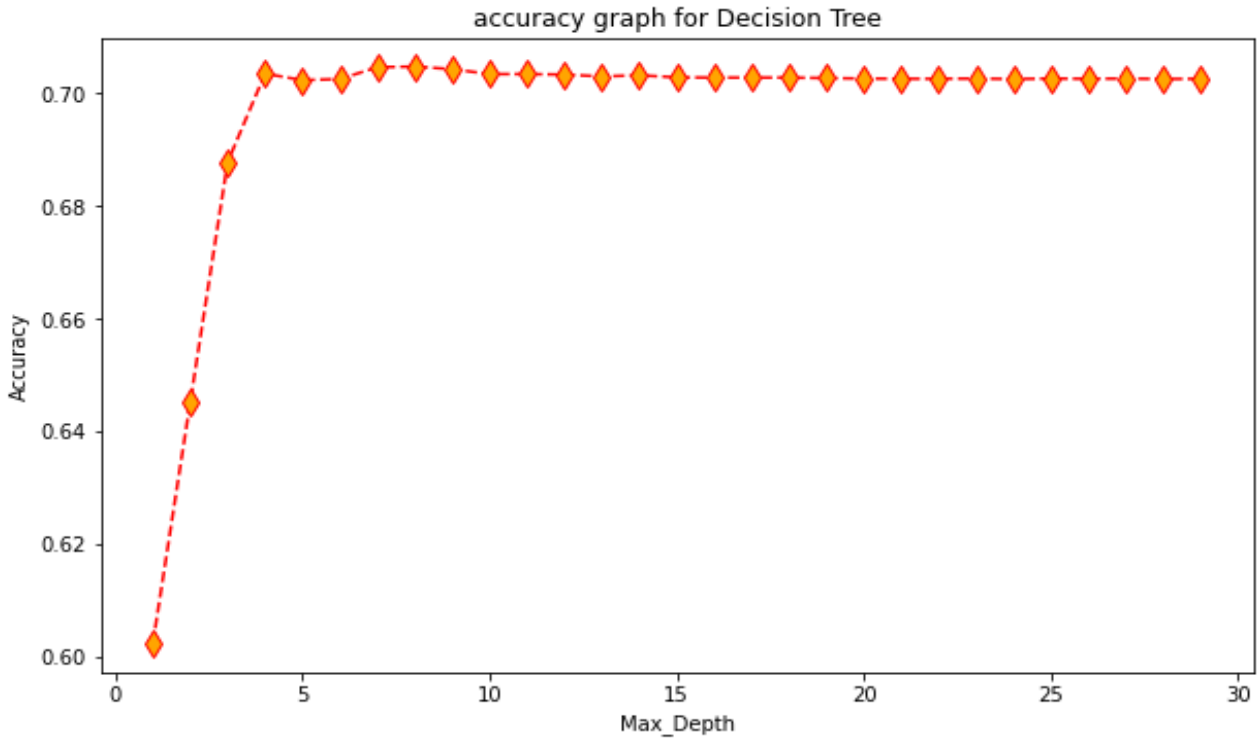


Figure 30: Accuracy vs. Max_Depth

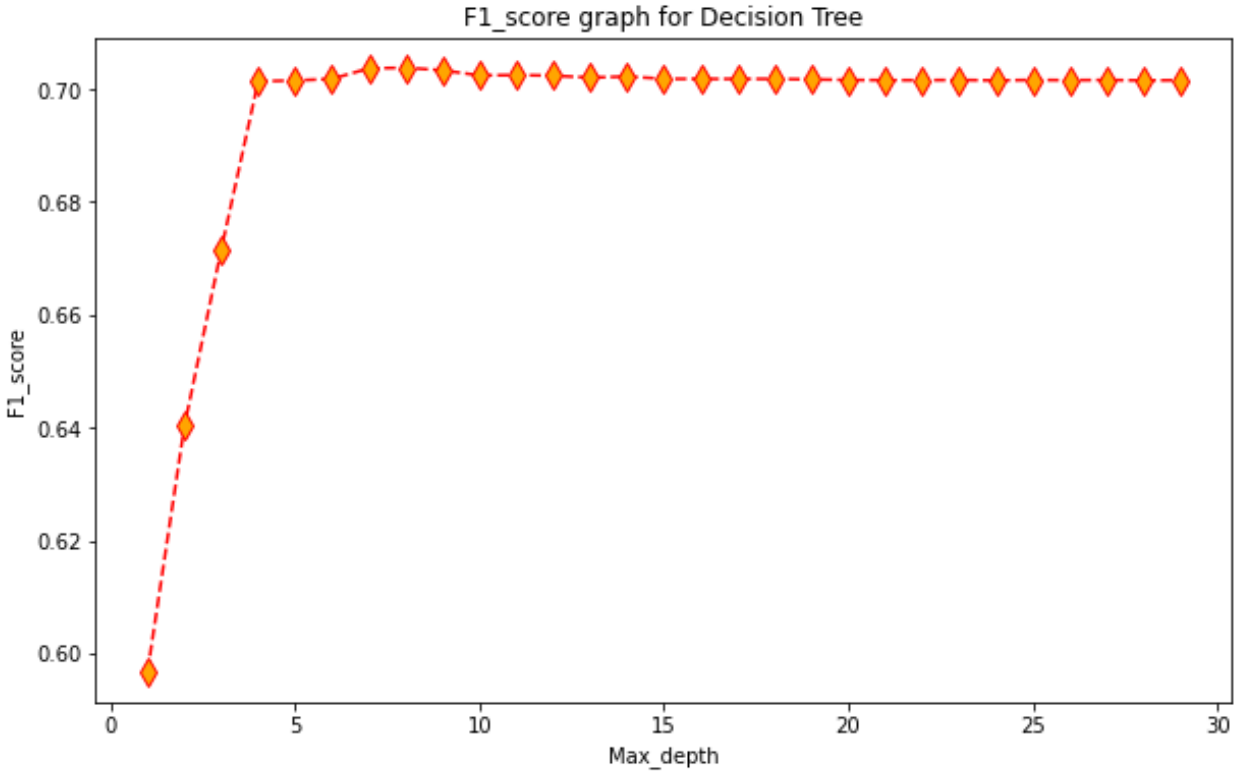


Figure 31: F1 Score vs. Max_depth

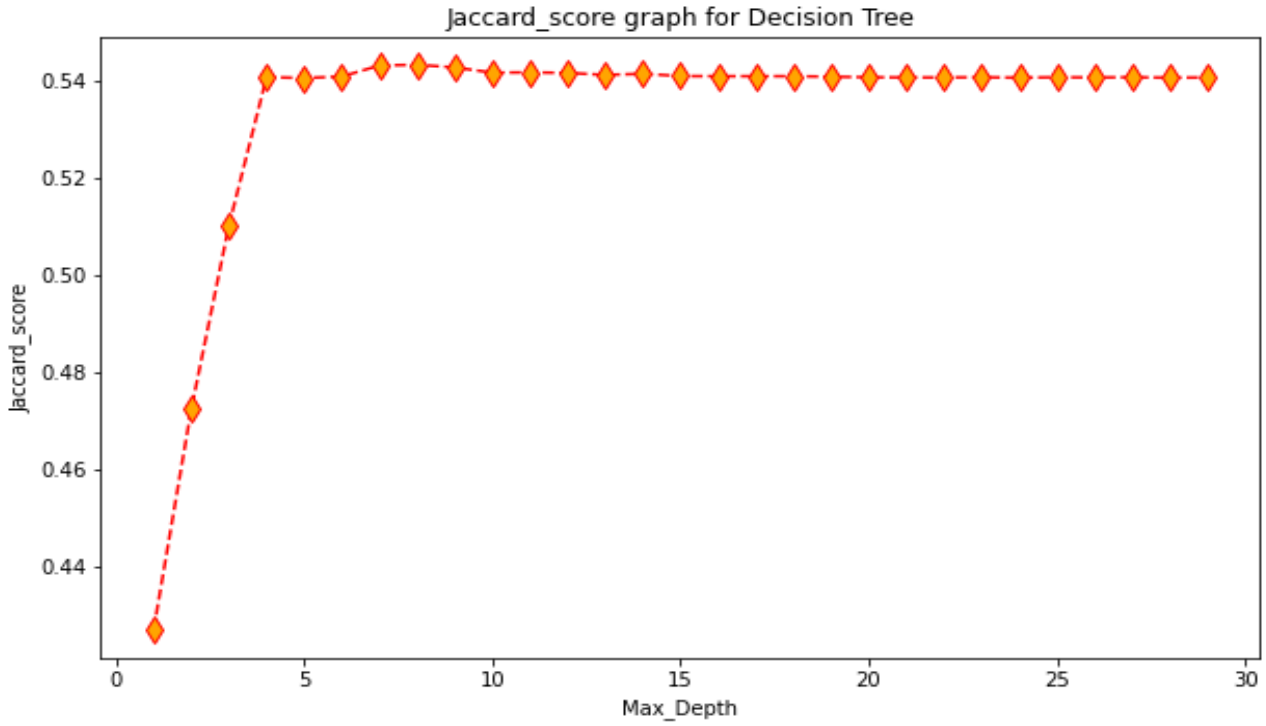


Figure 32: Jaccard similarity vs Max_depth

4.3Logistic Regression Model

Logistic regression (LR) is a statistical method similar to linear regression since LR finds an equation that predicts an outcome for a binary variable, *Y*, from one or more response variables, *X*. However, unlike linear regression the response variables can be categorical *or* continuous, as the model does not strictly require continuous data.

To predict group membership, LR uses the log odds ratio rather than probabilities and an iterative maximum likelihood method rather than a least squares to fit the final model.

Since the accident severity data is a binary variable (0 for no/minor injuries, 1 for major injuries/fatalities) we can employ Logistic Regression techniques to attempt to classify accident outcomes based on the properties in the feature set. A Logistic Regression model was trained using an inverse-regularisation strength C=0.01, and tested on the testing subset.

```
acc_lr=metrics.accuracy_score(y_test, yhat3)
f1_lr=f1_score(y_test, yhat3,average="macro")
j_lr=jaccard_score(y_test, yhat3, average='macro')
```

```
print ("Accuracy = %f\nF1_score = %f\nJaccard_score = %f" % (acc_lr , f1_lr , j_lr))
```

```
Accuracy = 0.605417
F1_score = 0.601163
Jaccard_score = 0.431000
```

Figure 33Accuracy, F1 score, Jaccard-Similarity sores for logistic regression

Chapter 5

Model Evaluation

The models were evaluated on 3 model evaluation scores:

5.1 Jaccard Similarity Score:

The Jaccard index, also known as Intersection over Union and the Jaccard similarity coefficient (originally given the French name *coefficient de communauté* by Paul Jaccard), is a statistic used for gauging the similarity and diversity of sample sets.

The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

(If A and B are both empty, define $J(A,B) = 1$.)

$$0 \leq J(A,B) \leq 1.$$

The Jaccard distance, which measures *dis*similarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union

5.2 Accuracy score:

It is the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 34: Confusion Matrix

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 35Formula for Calculating Accuracy

5.3 F1_score:

It is the Harmonic Sum of precision and recal (where the precision is the number of correctly identified positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of correctly identified positive results divided by the number of all samples that should have been identified as positive)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 36 How to Calculate Precision and Recall

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

Figure 37: Formula for F1 Score Calculation

Model/Score	Accuracy	F1_Score	Jaccard Score
KNN	0.702587	0.701023	0.540100
Decision Tree	0.704639	0.703682	0.543092
Logistic Regression	0.605417	0.601163	0.431000

Table 1 Results

From the above evaluation results we can see that the most reliable model for the given data set is the Decision Tree which is closely followed by the K Nearest Neighbour and the least accurate is the logistic regression.

Chapter 6

Conclusions and Future Work

The prediction of car accident severity is not completely finished. Based on the results, the dataset is under fitted, which means I will need to collect more data model. In addition, the dataset only contains binary data for severity, hence this model has a lot of scope of improvement. Also this project can further be extended and applied to accident databases of other regions/cities.

But the work highlights that machine learning techniques can be used to probe historical data in order to make reliable predictions about the outcome of road traffic accidents, given information which is available at the time when an accident is reported. The accuracy, F1 scores for the models developed are appreciable for an under fitted dataset, and hence these models can be relied upon to help the govt. authorities and people of country to reduce the number and severity of accidents. Although the Jaccard similarity score is low this can be changed by extending this project and adding more features and data to train the data set and increasing the accuracy and efficiency of the model. Attributes like drivers condition, narcotics level in a driver, number of passenger, type of vehicle and condition of vehicles are some of the attributes which can be added to this model to drastically improve the evaluation matrix.

By doing so, city planners can gain insight into the road conditions/features which are associated with high accident severity, and use this insight to improve road design. Additionally, by predicting accident severity as functions of weather, date, location and road conditions, this model may be able to help aid the decision making of emergency services call handlers, by allowing them to prioritise resources toward collisions with a greater likelihood of severe consequences.

Chapter 7

References

1. https://www.cdc.gov/injury/features/global-road-safety/index.html

2. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.673.2411&rep=rep1&type=pdf

3. https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

4. IBM Data Science Professional Certificate course (https://www.coursera.org/professional-certificates/ibm-data-science)

5. https://medium.com/geoai/using-machine-learning-to-predict-car-accident-risk-4d92c91a7d57

6. https://pbpython.com/categorical-encoding.html

7. https://sci-hub.se/https://www.sciencedirect.com/science/article/abs/pii/S0001457518305232

8. https://www.sciencedirect.com/topics/medicine-and-dentistry/logistic-regression-analysis

9. https://www.tandfonline.com/doi/abs/10.1080/13588265.2016.1275431

10. https://www.who.int/violence_injury_prevention/road_safety_status/2018/GSRRS2018_Summary_EN.pdf