CAPSTONE PROJECT ON

PREDICTING ACCIDENT SEVERITY USING MACHINE LEARNING MODELS

APPLIED DATA SCIENCE SPECIALIZATION

SUBMITTED BY

KARTHIKEYA VADADA

INDEX

INTRODUCTION AND BUSINESS UNDERSTANDING

1.1 DESCRIPTION OF PROBLEM

All around the world, roads are shared by many motorized vehicles that have made transportation faster and more comfortable while supporting many countries' economic and social development. Approximately 1.35 million people die each year because of road traffic crashes.

Almost 3,700 people are killed globally in road traffic crashes, where more than half of those killed are pedestrians, motorcyclists, and cyclists. The main business problem here is the severity of these accidents, which can sometimes be fatal and critical for pedestrians, bicycles, or vehicles. The project's objective is to get people to safety, and the best way to solve this problem is to prevent those accidents of happening. Imagine being able to predict in advance the probability of an accident happening depending on the weather, the type of road, light conditions, junction type, driver's influence of drugs before stepping inside a vehicle. This would help save many lives and help drivers get to their destination in the safest and fastest route possible.

This model is meant to alert drivers to remind them to be a little bit more careful, to switch roads or postpone their car rides.

There is another possible targeted audience for this study, the local police, health institutes, insurance companies etc. They can make good use of this model to know when to be fully ready to receive bad news about a specific road, and more importantly take prevention measures to avoid accidents on certain ones.

1.2 DISCUSSION OF THE BACKGROUND

In most cases, carelessness while driving, using drugs and alcohol, or driving too fast are some of the main causes of accidents that can be avoided by implementing stronger regulations.

Besides the above reasons, weather, visibility, or road conditions and road types might be the major uncontrollable factors which can be avoided by uncovering patterns hidden in the data and declaring a warning to local government, police and drivers on the roads. targeted routes or alerting the drivers before the road trips.

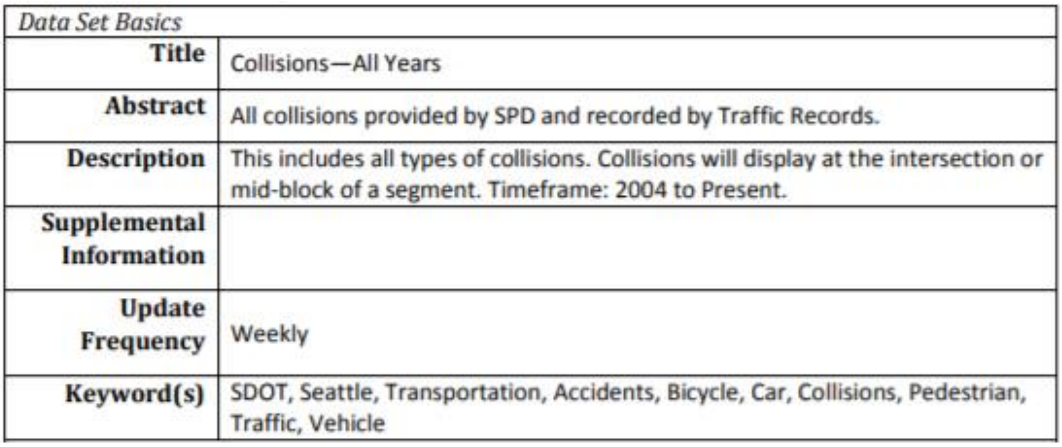
DATA UNDERSTANDING

2.1 SOURCE OF DATA

For this project, the Dataset was shared on Coursera as a csv file

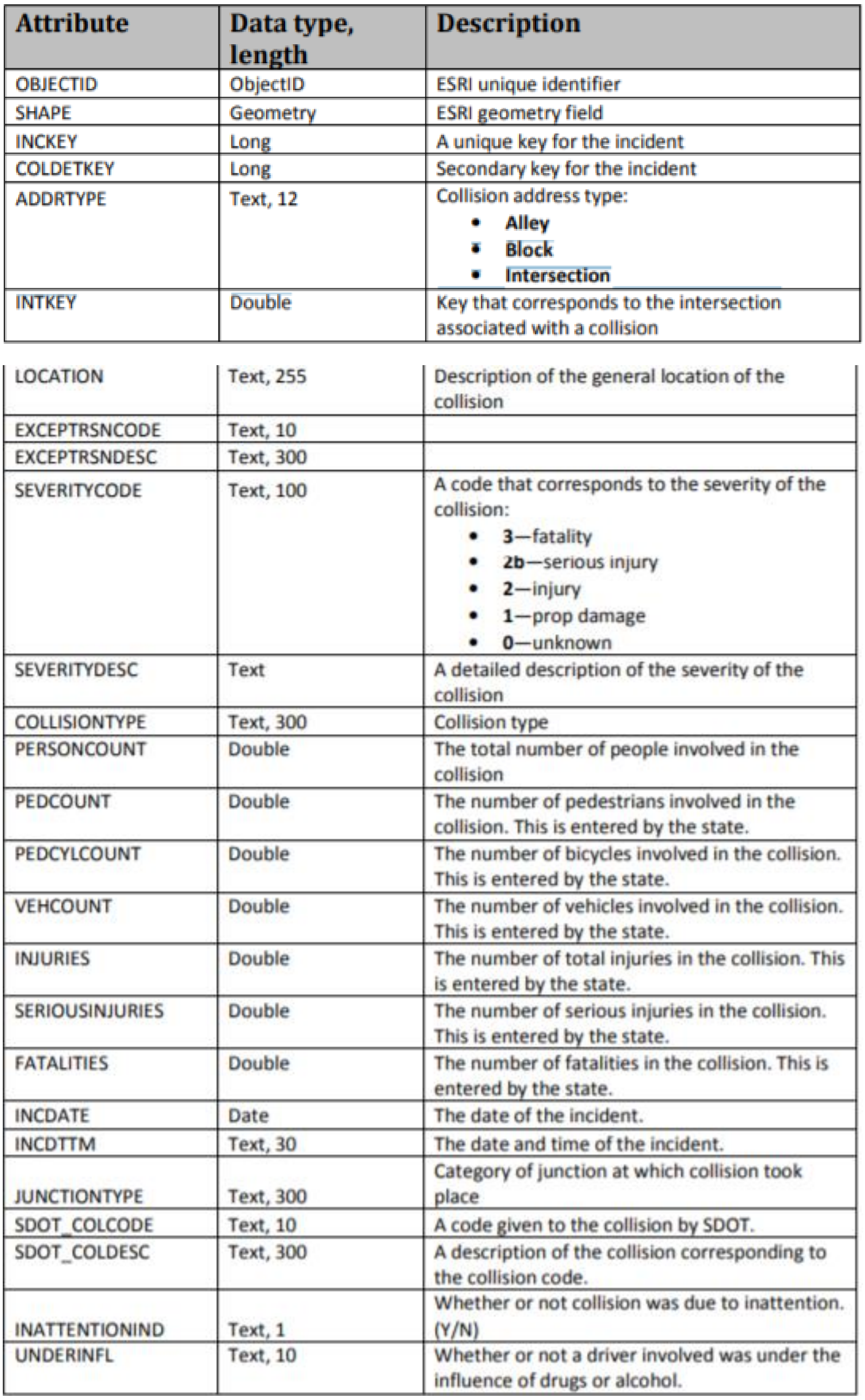
It concerns the city of Seattle, WA and it is provided by Seattle Police Dept. and recorded by Traffic Records, with a timeframe from 2004 to Present.

The data is composed of 38 features that accurately describe each car accident that happened in Seattle. They are classified in terms of severity, type of weather and road condition, location, address, influence of drugs, light conditions, fatalities etc during an accident



Data Set Basics	
Title	Collisions—All Years
Abstract	All collisions provided by SPD and recorded by Traffic Records.
Description	This includes all types of collisions. Collisions will display at the intersection or mid-block of a segment. Timeframe: 2004 to Present.
Supplemental Information	
Update Frequency	Weekly
Keyword(s)	SDOT, Seattle, Transportation, Accidents, Bicycle, Car, Collisions, Pedestrian, Traffic, Vehicle

ATTRIBUTE INFORMATION



Attribute	Data type, length	Description
OBJECTID	ObjectID	ESRI unique identifier
SHAPE	Geometry	ESRI geometry field
INCKEY	Long	A unique key for the incident
COLDETKEY	Long	Secondary key for the incident
ADDRTYPE	Text, 12	Collision address type: • **Alley** • **Block** • **Intersection**
INTKEY	Double	Key that corresponds to the intersection associated with a collision
LOCATION	Text, 255	Description of the general location of the collision
EXCEPTRSNCODE	Text, 10	
EXCEPTRSNDESC	Text, 300	
SEVERITYCODE	Text, 100	A code that corresponds to the severity of the collision: • **3**—fatality • **2b**—serious injury • **2**—injury • **1**—prop damage • **0**—unknown
SEVERITYDESC	Text	A detailed description of the severity of the collision
COLLISIONTYPE	Text, 300	Collision type
PERSONCOUNT	Double	The total number of people involved in the collision
PEDCOUNT	Double	The number of pedestrians involved in the collision. This is entered by the state.
PEDCYLCOUNT	Double	The number of bicycles involved in the collision. This is entered by the state.
VEHCOUNT	Double	The number of vehicles involved in the collision. This is entered by the state.
INJURIES	Double	The number of total injuries in the collision. This is entered by the state.
SERIOUSINJURIES	Double	The number of serious injuries in the collision. This is entered by the state.
FATALITIES	Double	The number of fatalities in the collision. This is entered by the state.
INCDATE	Date	The date of the incident.
INCDTTM	Text, 30	The date and time of the incident.
JUNCTIONTYPE	Text, 300	Category of junction at which collision took place
SDOT_COLCODE	Text, 10	A code given to the collision by SDOT.
SDOT_COLDESC	Text, 300	A description of the collision corresponding to the collision code.
INATTENTIONIND	Text, 1	Whether or not collision was due to inattention. (Y/N)
UNDERINFL	Text, 10	Whether or not a driver involved was under the influence of drugs or alcohol.

2.2 INFORMATION ABOUT DATA

The data consists of 37 independent variables and 194,673 rows. The dependent variable, “SEVERITYCODE”, contains numbers that correspond to different levels of severity caused by an accident from 0 to 4.

Severity codes are as follows:

0: Little to no Probability (Clear Conditions)

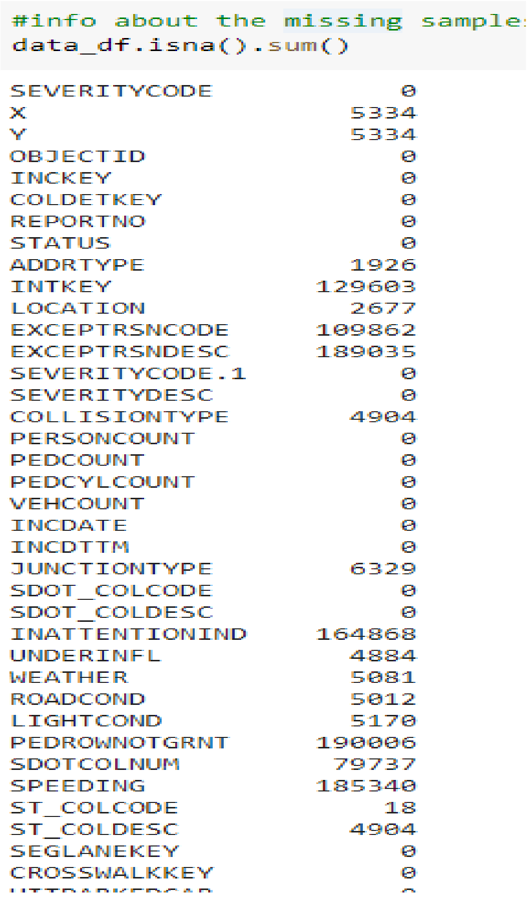
1: Very Low Probability — Chance or Property Damage

2: Low Probability — Chance of Injury

3: Mild Probability — Chance of Serious Injury

4: High Probability — Chance of Fatality

Furthermore, because of the existence of a huge unbalance in some attributes occurrences and the existence of null values in many records, the data needs to be pre-processed, cleaned, resampled, and balanced before any further processing. Total missing samples in each feature



```
#info about the missing sample
data_df.isna().sum()
```

```
SEVERITYCODE             0
X                     5334
Y                     5334
OBJECTID                 0
INCKEY                   0
COLDETKEY                0
REPORTNO                 0
STATUS                   0
ADDRTYPE              1926
INTKEY              129603
LOCATION              2677
EXCEPTRSNCODE       109862
EXCEPTRSNDESC       189035
SEVERITYCODE.1           0
SEVERITYDESC             0
COLLISIONTYPE         4904
PERSONCOUNT              0
PEDCOUNT                 0
PEDCYLCOUNT              0
VEHCOUNT                 0
INCDATE                  0
INCDTTM                  0
JUNCTIONTYPE          6329
SDOT_COLCODE             0
SDOT_COLDESC             0
INATTENTIONIND      164868
UNDERINFL             4884
WEATHER               5081
ROADCOND              5012
LIGHTCOND             5170
PEDROWNOTGRNT       190006
SDOTCOLNUM           79737
SPEEDING            185340
ST_COLCODE              18
ST_COLDESC            4904
SEGLANEKEY               0
CROSSWALKKEY             0
```

2.3 USE OF RELEVANT DATA TO SOLVE THE PROBLEM

for this project, not all the attributes are useful as the main objective is to predict an accident's probability and severity.

Therefore, the Dataset needs deep understanding and analysis before choosing the right attributes to reach goal.

For example, SDOTCOLNUM, X, Y, LOCATION, INCDTTM, INCDATE, REPORTNO, COLDETKEY, INCKEY and OBJECTID are features that give descriptive and detailed information about an accident, and are then not relevant to predict the severity of an accident in general.

Moreover, EXCEPTRSNCODE, EXCEPTRSNDESC, PEDROWNOTGRNT, SPEEDING, INATTENTIONIND and INTKEY have a high number of missing data that would skew and bias our predictive model.

We must select the most important features to weigh the severity of accidents in Seattle. Among all the features, the following features have the most influence in the accuracy of the predictions:

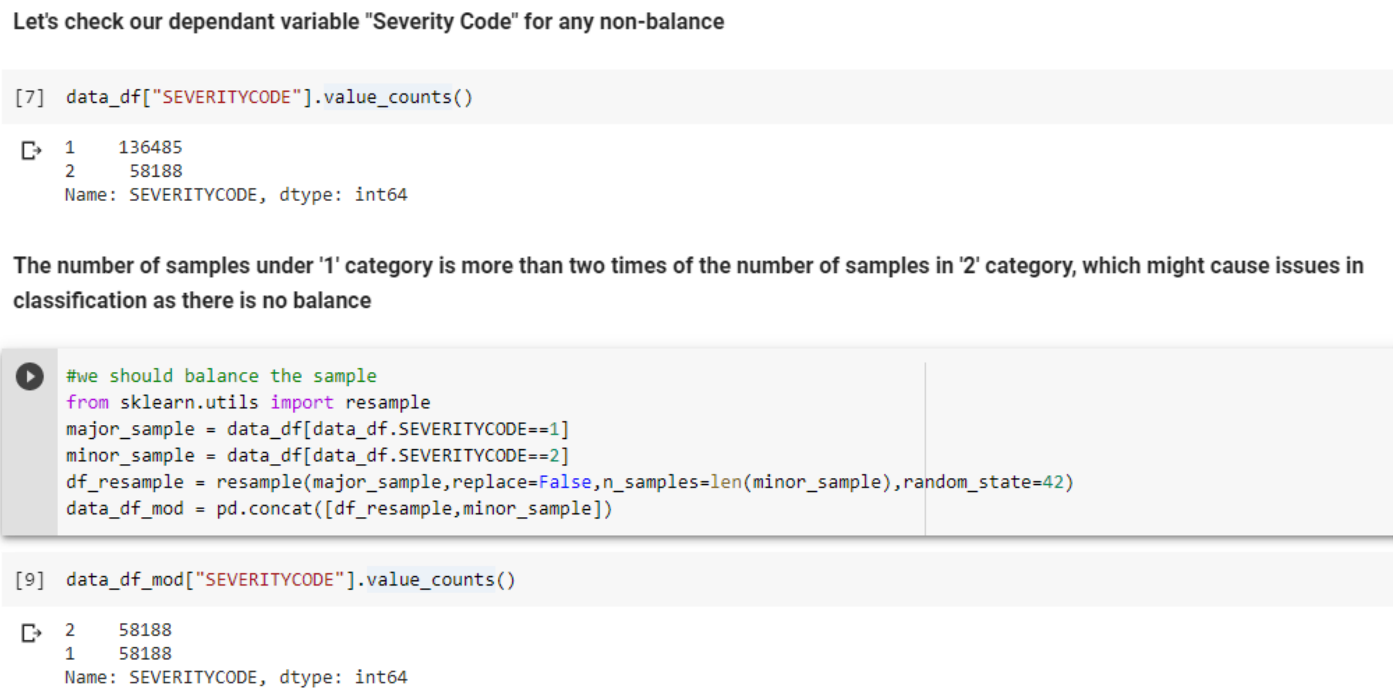
The **'WEATHER', 'ROADCOND'**,**'LIGHTCOND', 'UNDERINFL' and 'ADDRTYPE'** attributes to predict **'SEVERITYCODE'** attribute.

After selecting the appropriate features, the new Dataset is balanced and preprocessed before feeding it to a supervised machine learning model that will learn to predict in the future the probability of a car accident.

2.4 PREPROCESSING OF DATA

The null values from the new feature set are dropped and the Target variable balance is checked to avoid any bias for modelling

Let us study the balance of our dataset, let's check if the samples in the SEVERITYCODE column, have nearly equal value count.



Let's check our dependant variable "Severity Code" for any non-balance

```
[7]  data_df["SEVERITYCODE"].value_counts()
```

```
1     136485
2      58188
Name: SEVERITYCODE, dtype: int64
```

The number of samples under '1' category is more than two times of the number of samples in '2' category, which might cause issues in classification as there is no balance

```
#we should balance the sample
from sklearn.utils import resample
major_sample = data_df[data_df.SEVERITYCODE==1]
minor_sample = data_df[data_df.SEVERITYCODE==2]
df_resample = resample(major_sample,replace=False,n_samples=len(minor_sample),random_state=42)
data_df_mod = pd.concat([df_resample,minor_sample])
```

```
[9]  data_df_mod["SEVERITYCODE"].value_counts()
```

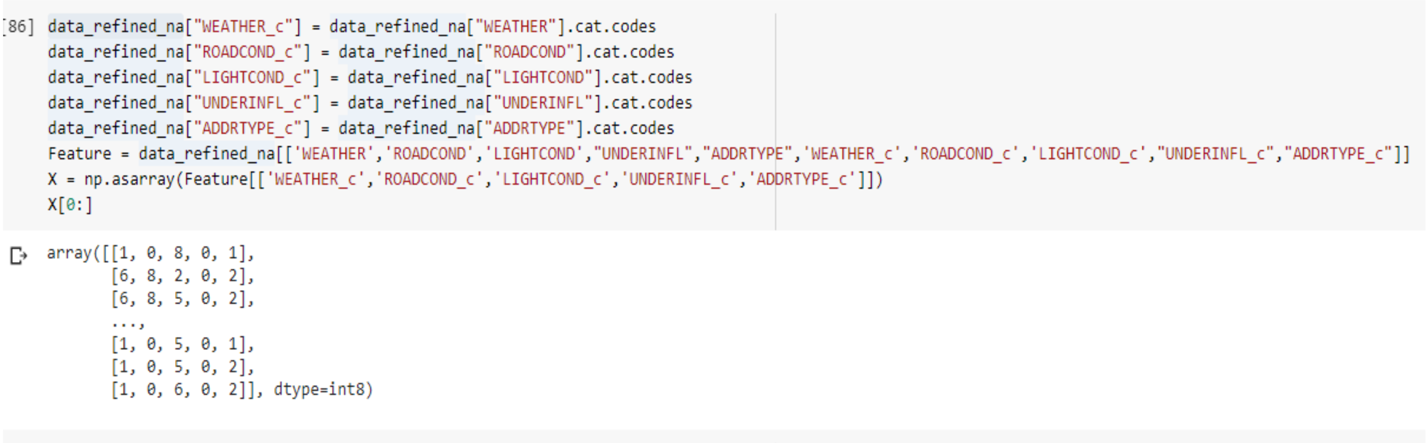
```
2     58188
1     58188
Name: SEVERITYCODE, dtype: int64
```

The sample is balanced after resampling and will have minimum skewness for modelling

To work with our features as categorical values, we should change the type of the columns for feature variable, and we will add 5 other columns containing the label encoding of categories in the 5 column attributes.

After type changing the category variables, we label encode the data and normalize the data to feed into models

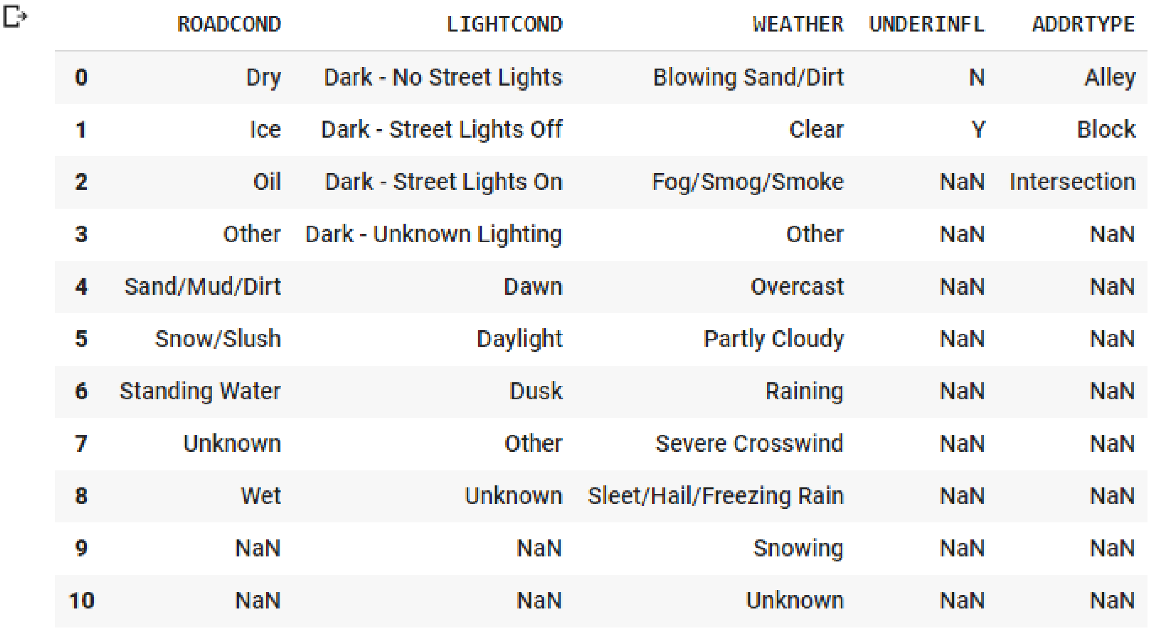
LABEL ENCODING



```
[86] data_refined_na["WEATHER_c"] = data_refined_na["WEATHER"].cat.codes
     data_refined_na["ROADCOND_c"] = data_refined_na["ROADCOND"].cat.codes
     data_refined_na["LIGHTCOND_c"] = data_refined_na["LIGHTCOND"].cat.codes
     data_refined_na["UNDERINFL_c"] = data_refined_na["UNDERINFL"].cat.codes
     data_refined_na["ADDRTYPE_c"] = data_refined_na["ADDRTYPE"].cat.codes
     Feature = data_refined_na[['WEATHER','ROADCOND','LIGHTCOND',"UNDERINFL","ADDRTYPE",'WEATHER_c','ROADCOND_c','LIGHTCOND_c',"UNDERINFL_c","ADDRTYPE_c"]]
     X = np.asarray(Feature[['WEATHER_c','ROADCOND_c','LIGHTCOND_c','UNDERINFL_c','ADDRTYPE_c']])
     X[0:]
```

```
array([[1, 0, 8, 0, 1],
       [6, 8, 2, 0, 2],
       [6, 8, 5, 0, 2],
       ...,
       [1, 0, 5, 0, 1],
       [1, 0, 5, 0, 2],
       [1, 0, 6, 0, 2]], dtype=int8)
```

Categorization of feature data



	ROADCOND	LIGHTCOND	WEATHER	UNDERINFL	ADDRTYPE
0	Dry	Dark - No Street Lights	Blowing Sand/Dirt	N	Alley
1	Ice	Dark - Street Lights Off	Clear	Y	Block
2	Oil	Dark - Street Lights On	Fog/Smog/Smoke	NaN	Intersection
3	Other	Dark - Unknown Lighting	Other	NaN	NaN
4	Sand/Mud/Dirt	Dawn	Overcast	NaN	NaN
5	Snow/Slush	Daylight	Partly Cloudy	NaN	NaN
6	Standing Water	Dusk	Raining	NaN	NaN
7	Unknown	Other	Severe Crosswind	NaN	NaN
8	Wet	Unknown	Sleet/Hail/Freezing Rain	NaN	NaN
9	NaN	NaN	Snowing	NaN	NaN
10	NaN	NaN	Unknown	NaN	NaN

METHODOLOGY

3.1 Tools and Technologies:

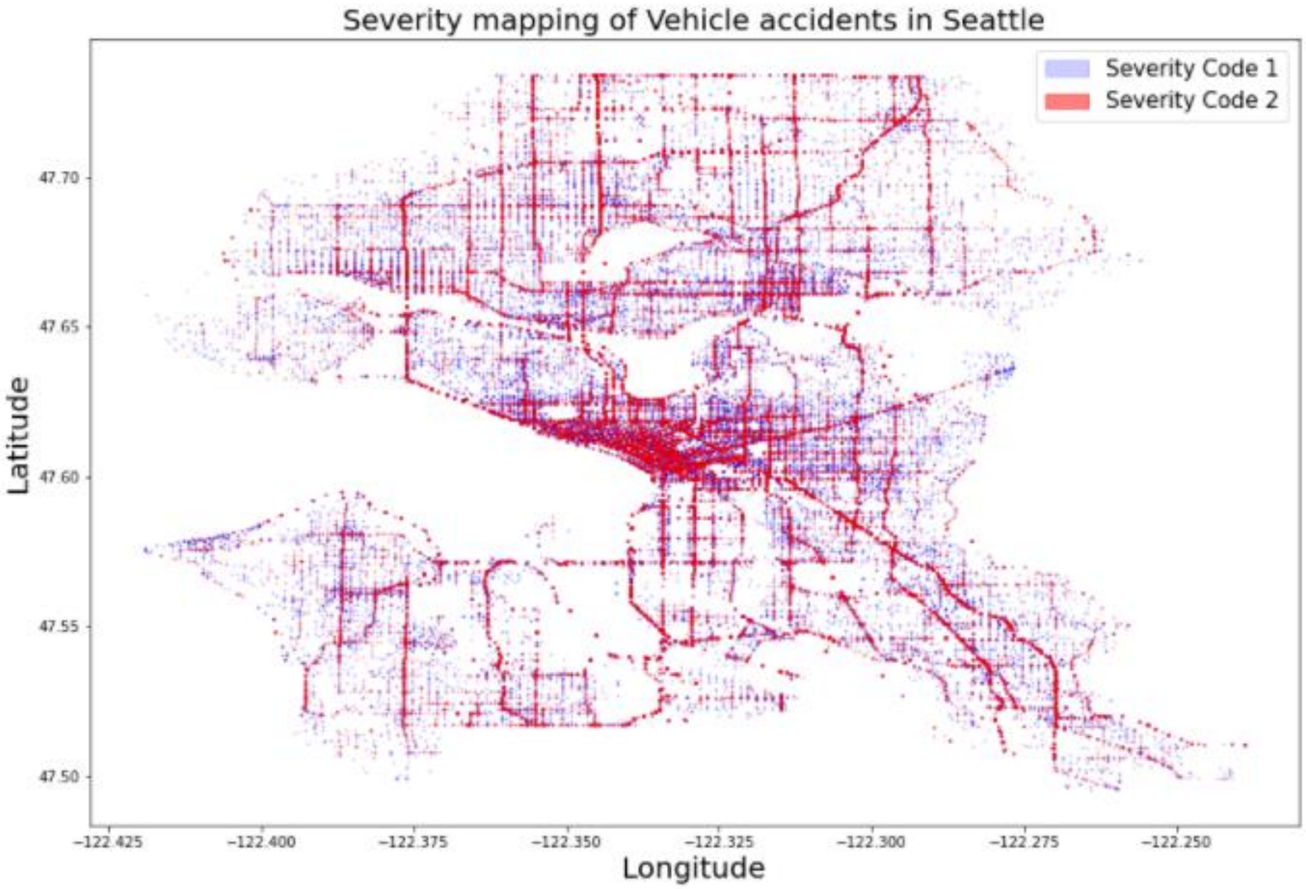
To help implement the solution, I have used a Github repository and Google Collab to pre-process data and build Machine Learning models.

3.2 EXPLORATORY DATA ANALYSIS

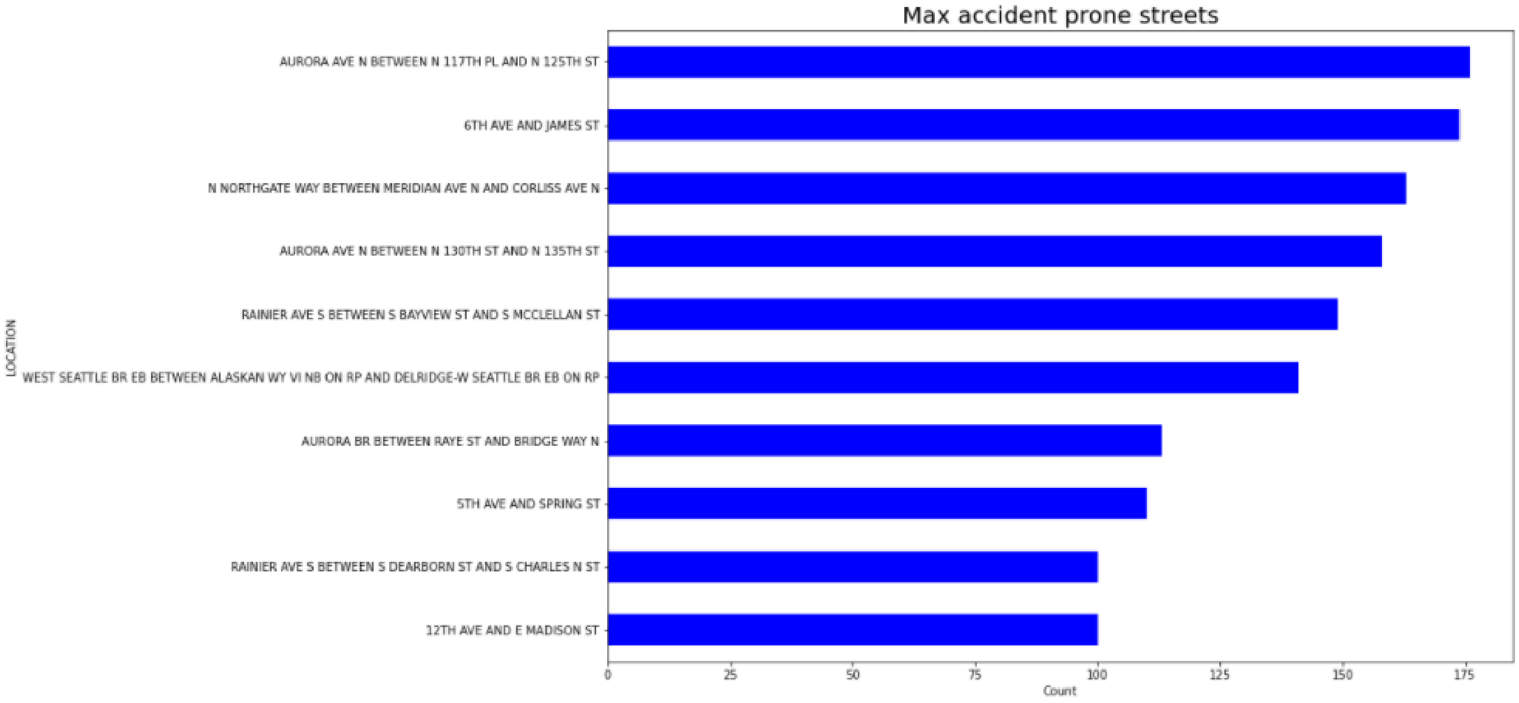
To clearly visualize the dataset location attribute. I created a map of Seattle with markers in scatterplot with latitude and longitudinal data that indicate the location of each accident and its severity. The markers in Blue represent the accidents of type 1 severity, where property damage occurred, and the markers in Red represent accidents of type 2 severity, in place of injury.

The severity of accidents is indicated in the scatter plot and most of these accidents seem to be located at the centre of Seattle for both types.

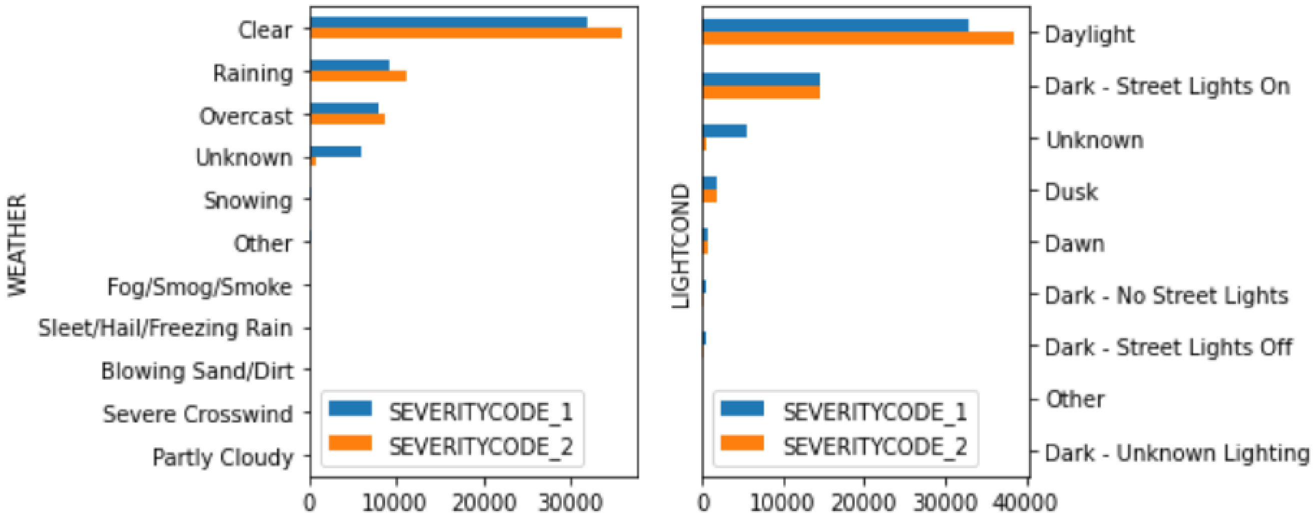
The most common location for both type I and type II accidents is also shown with a barh chart, which indicates the central streets of Seattle to be more accident prone



```
Text(0.5, 0, 'Count')
```

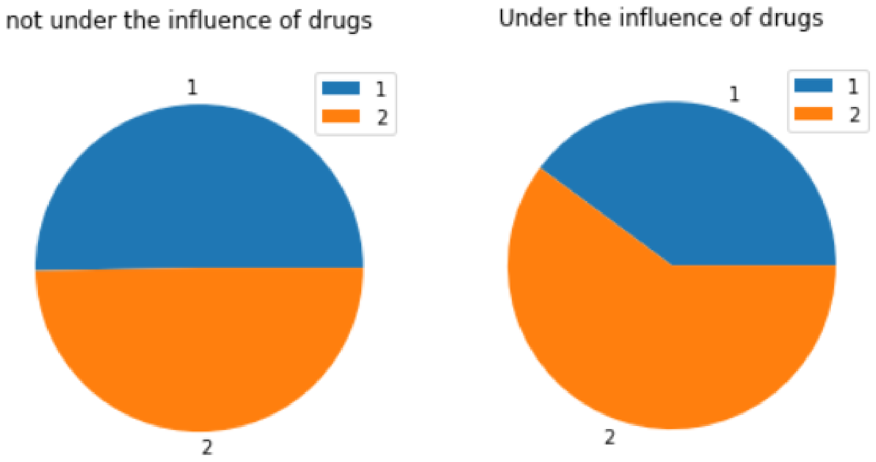


Next, we will explore the feature categorical data and see how they affect the severity of the accident



Upon analysing correlation between accidents (both type I and type II), "Clear" weather conditions and "Daylight" light conditions resulted in a greater number of type 1 and type 2 severity accidents, infers that accidents took place in safest of conditions possible

It appears that type 2 severity accidents have a higher proportion when the driver is under the influence of any drug or alcohol leading to a higher probability of injury than when the driver is sober.

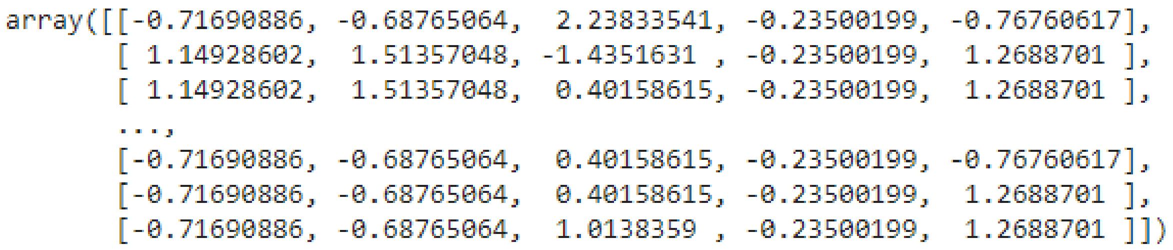


3.3 NORMALIZATION, SPLITTING AND CLASSIFICATION MODELLLING

After balancing SEVERITYCODE feature, and standardizing the input feature, the data has been ready for building machine learning models.

Normalizing the data

Standard scaler is used to normalize the feature set



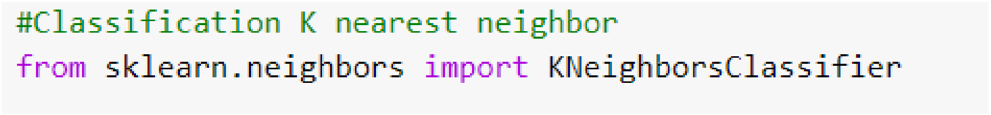
```
array([[-0.71690886, -0.68765064,  2.23833541, -0.23500199, -0.76760617],
       [ 1.14928602,  1.51357048, -1.4351631 , -0.23500199,  1.2688701 ],
       [ 1.14928602,  1.51357048,  0.40158615, -0.23500199,  1.2688701 ],
       ...,
       [-0.71690886, -0.68765064,  0.40158615, -0.23500199, -0.76760617],
       [-0.71690886, -0.68765064,  0.40158615, -0.23500199,  1.2688701 ],
       [-0.71690886, -0.68765064,  1.0138359 , -0.23500199,  1.2688701 ]])
```

Classification Models:

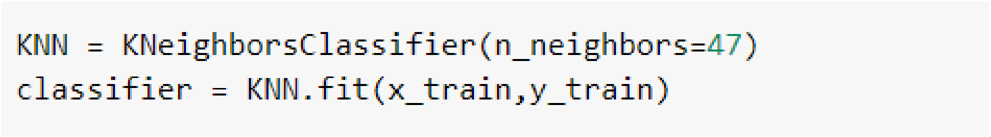
Having our data set ready we can now implement different classification problems:

- K-Nearest Neighbors (KNN)
- Decision Tree
- Logistic Regression

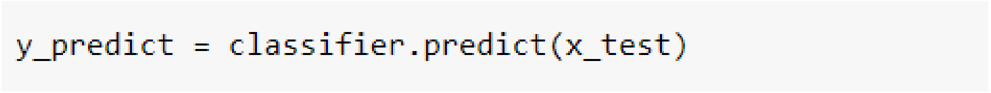
K-Nearest Neighbors :



```
#Classification K nearest neighbor
from sklearn.neighbors import KNeighborsClassifier
```



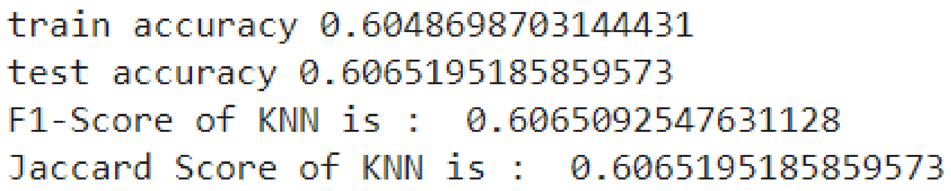
```
KNN = KNeighborsClassifier(n_neighbors=47)
classifier = KNN.fit(x_train,y_train)
```



```
y_predict = classifier.predict(x_test)
```

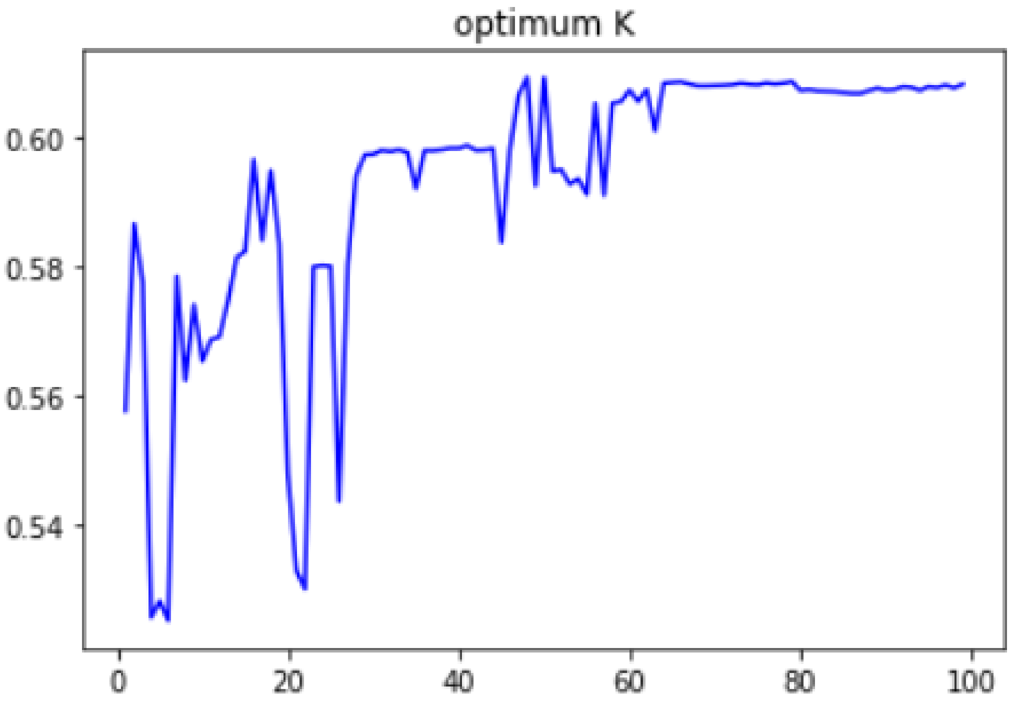
K=47 is based on the optimum plot to find maximize score

Result OF KNN



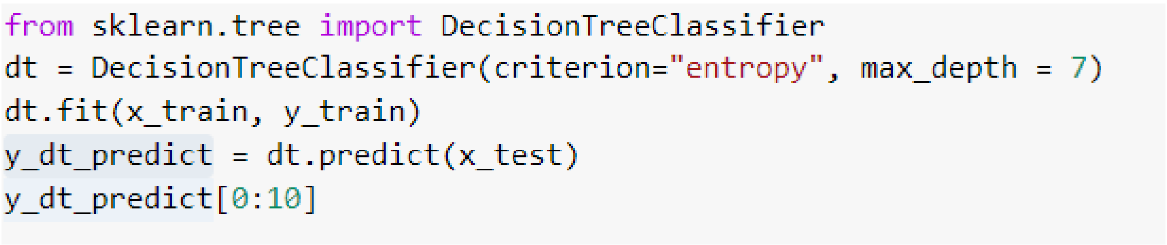
```
train accuracy 0.6048698703144431
test accuracy 0.6065195185859573
F1-Score of KNN is :  0.6065092547631128
Jaccard Score of KNN is :  0.6065195185859573
```

Optimum plot to find K



K in the range of 47-50 has given optimum score

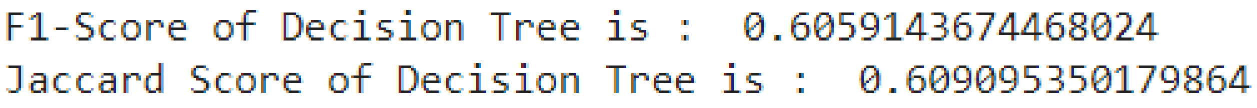
Decision Tree:



```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(criterion="entropy", max_depth = 7)
dt.fit(x_train, y_train)
y_dt_predict = dt.predict(x_test)
y_dt_predict[0:10]
```

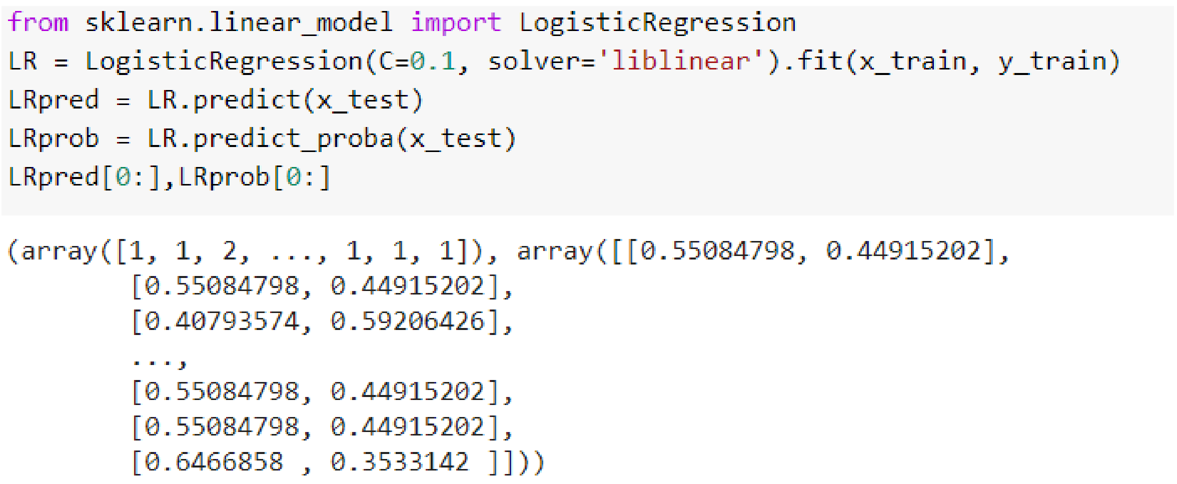
```
array([1, 1, 2, 1, 2, 1, 1, 2, 1, 1])
```

Result of Decision Tree:



```
F1-Score of Decision Tree is :  0.6059143674468024
Jaccard Score of Decision Tree is :  0.609095350179864
```

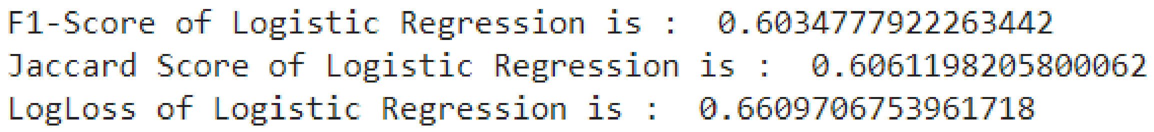
Logistic Regression:



```python
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(C=0.1, solver='liblinear').fit(x_train, y_train)
LRpred = LR.predict(x_test)
LRprob = LR.predict_proba(x_test)
LRpred[0:],LRprob[0:]
```

```
(array([1, 1, 2, ..., 1, 1, 1]), array([[0.55084798, 0.44915202],
        [0.55084798, 0.44915202],
        [0.40793574, 0.59206426],
        ...,
        [0.55084798, 0.44915202],
        [0.55084798, 0.44915202],
        [0.6466858 , 0.3533142 ]]))
```

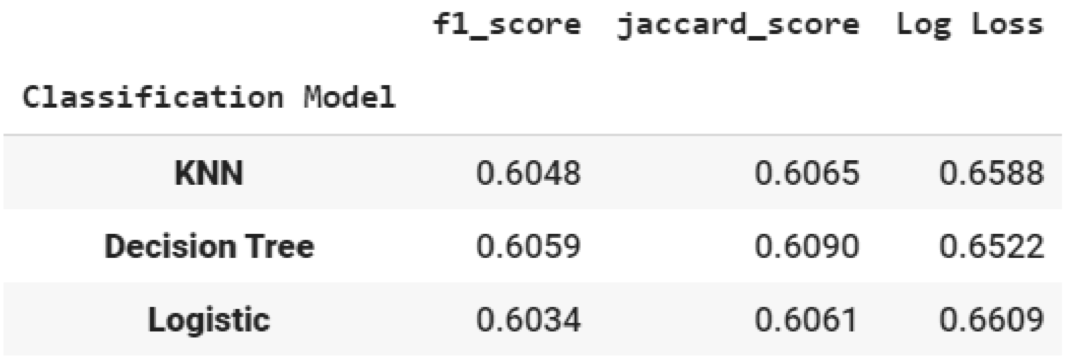
Result of Logistic Regression



```
F1-Score of Logistic Regression is :  0.6034777922263442
Jaccard Score of Logistic Regression is :  0.6061198205800062
LogLoss of Logistic Regression is :  0.6609706753961718
```

DISCUSSION

I applied classification models like logistic regression, decision trees and K-Nearest Neighbors to the dataset, using precision, recall, f1-score and mean accuracy as evaluation metric.



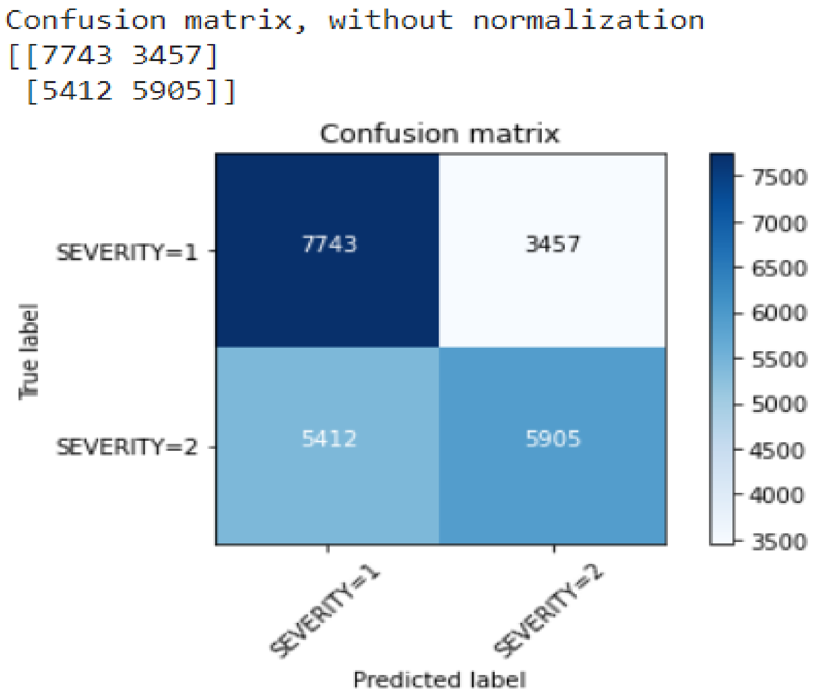
	f1_score	jaccard_score	Log Loss
Classification Model			
KNN	0.6048	0.6065	0.6588
Decision Tree	0.6059	0.6090	0.6522
Logistic	0.6034	0.6061	0.6609

These are the results of the inferential statistical testing performed using Jaccard score, f1 score and log loss as evaluation metrics. All the three models performed on almost the same accuracy. The model with the highest performance is the Decision Tree with a log loss of 65.2% and f1 score approximately like its Jaccard score with a value of 60.9%. Overall, there was still significant variance that could not be predicted by the models in this study, the algorithms have low scores and need more pre-processing.

Due to the binary aspect of the ‘SEVERITYCODE’ attribute we want to predict(only classes 1 & 2 were in this dataset), a Logistic Regression model was the first intuitive solution, but we also built the K-Nearest Neighbors and Decision Tree models to have more results, contrary to what we expected, the Decision Tree model had better F1-score and Jaccard-score.

This trend in the results may be due to a low tuning of the model’s hyperparameters or to the need to pre-process more the features beforehand.

Logistic regression confusion matrix shows that the model is good at predicting type 1 severity more than that of type 2



```
Confusion matrix, without normalization
[[7743 3457]
 [5412 5905]]
```

Another reason for this is possibly the similarity of the features for both type of accidents. In effect, the highest type 2 accident happen in the same light, road, weather condition of a type 1 severity accident. This similarity can cause the models to not be able to clearly define the strong attributes as most of them have the same tendency for both labels. This resemblance cause difficulty for the model to clearly define and classify a type of accident. Another source for this problem is the absence of other type of severity that would help the models better rank the accident.

Another factor can also be the big amount of missing data present in the given data, as most of the missing data correspond to very useful attributes that would have helped defining the two types of accidents. For example, the 'SPEEDING' attribute have 185,340 missing values of the 194,673 available, meaning more than 95.2% of the accident didn't specify if the driver was speeding or no. Theoretically, the faster the car, the biggest the damage and the higher the probability of injuries/fatalities. This argument would have a great impact on defining the type of accident if they were no missing value.

We can still improve the above models, by better tuning of the hyperparameters like the "k" in KNN, the "max_depth" in the Decision Tree, and the "C" parameter in the Logistic Regression.

CONCLUSION

In this study, our goal was to predict accurately the severity type of an accident depending on the selected features. The results can have a better performance, a lot of improvement can be done on class 1 and 2 predictions. These models can be very useful in helping weather stations or news program alert drivers of the probabilities of car crashes and its type of severity , we can also conclude that particular weather, road ,light ,drug influence and road intersection types have an impact on whether or not the car ride could result in one of the two classes property damage (class 1) or injury (class 2).