Analyzing and Predicting Traffic Accident Severity in Seattle, Washington

Rohit Rajesh Dube

October 05, 2020

1. Introduction

1.1 Background

According to the World Health Organization (WHO), every year approximately 1.35 million people die from traffic accidents. Out of which 93% of the world's fatalities happen in low and middle-income countries which only posses 60% of world's vehicles. On top of it, traffic accidents are the leading causes of death among children and young adults aged 5-29 years. The seaport city of Seattle is the largest city in the state of Washington, as well as the largest in the Pacific Northwest. As of the latest census, there were 713,700 people living in Seattle. Seattle residents get around by car, trolley, streetcar, public bus, bicycle, on foot, and by rail. With such bustling streets, it is no surprise that Seattle sees car accidents every day. According to data from the Washington State Department of Transportation (WSDOT), last year, Seattle saw more than 10,315 crashes on the street.

1.2 Problem

In 2019 only, there were 22 fatal car accidents, 190 serious injury collisions, 834 minor injuries, 2612 possible injuries and 6657 apparent injuries in the Seattle. The goal of this project is to analyze the previously occurred traffic accidents' severity and predict severity of new accidents which will help first responders and medics prepare themselves to take care of any such disaster.

1.3 Interest

Authorities like local Seattle government, police, paramedics, and public development authority will be interested in the model and its result as it might help them reduce occurrence of accidents and save life of citizens. The model might also help private companies, which are working on products related to public safety.

2. Data

2.1 Data Sources

The dataset, we will be using in this project was downloaded from the City of Seattle Open Data portal website. Seattle Police Department and Accident Traffic Records Department collected and maintained data from 2004 to present. The data includes many columns of details of the accidents and the severity of each car accidents.

2.2 Data Cleaning

In our quest to predict the severity of an accident, we have come across a dataset which has lots of NaN values in its original form. It has 221,525 rows and 37 columns. After inspecting the dataset carefully using the metadata PDF, I have decided that only 6 columns can help us to make a proper prediction and they are - 'HITPARKEDCAR', 'LIGHTCOND', 'ROADCOND', 'WEATHER', 'UNDERINFL', 'SEVERITYCODE'. Out of which, 'SEVERITYCODE' is the target variable. A code that corresponds to the severity of the collision:

• 3—fatality

• 2b—serious injury

• 2—injury

• 1—prop damage

• 0—unknown

We will later convert the above to 0 and 1 where 0 - prop damage and 1 – injury.

After dropping the rows containing NaN values and unnecessary columns, we start label encoding the columns we have:

- 'HITPARKEDCAR' - {0: 'N', 1: 'Y'}
- 'LIGHTCOND' - {0: 'Dark - No Street Lights', 1: 'Dark - Street Lights Off', 2: 'Dark - Street Lights On', 3: 'Dark - Unknown Lighting', 4: 'Dawn', 5: 'Daylight', 6: 'Dusk', 7: 'Other', 8: 'Unknown'}

- 'ROADCOND' - {0: 'Dry', 1: 'Ice', 2: 'Oil', 3: 'Other', 4: 'Sand/Mud/Dirt', 5: 'Snow/Slush', 6: 'Standing Water', 7: 'Unknown', 8: 'Wet'}
- 'WEATHER' - {0: 'Blowing Sand/Dirt', 1: 'Clear', 2: 'Fog/Smog/Smoke', 3: 'Other', 4: 'Overcast', 5: 'Partly Cloudy', 6: 'Raining', 7: 'Severe Crosswind', 8: 'Sleet/Hail/Freezing Rain', 9: 'Snowing', 10: 'Unknown'}
- 'UNDERINFL' – {0: 'N', 1: 'Y'}

2.3 Feature Selection

A total of 5 features were selected to predict the target variable which is the 'SEVERITYCODE'.

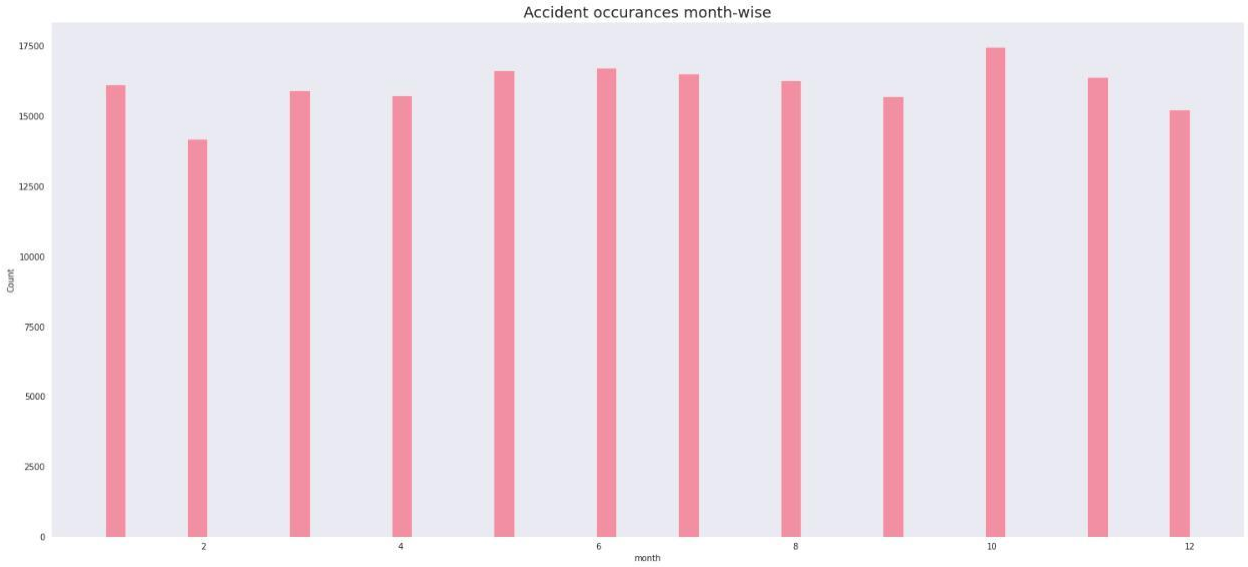
- 'HITPARKEDCAR' - Whether the collision involved hitting a parked car. (Y/N)
- 'LIGHTCOND' - The light conditions during the collision.
- 'ROADCOND' - The condition of the road during the collision.
- 'WEATHER' - A description of the weather conditions during the time of the collision.
- 'UNDERINFL' – Whether or not a driver involved was under the influence of drugs or alcohol.

3. Methodology

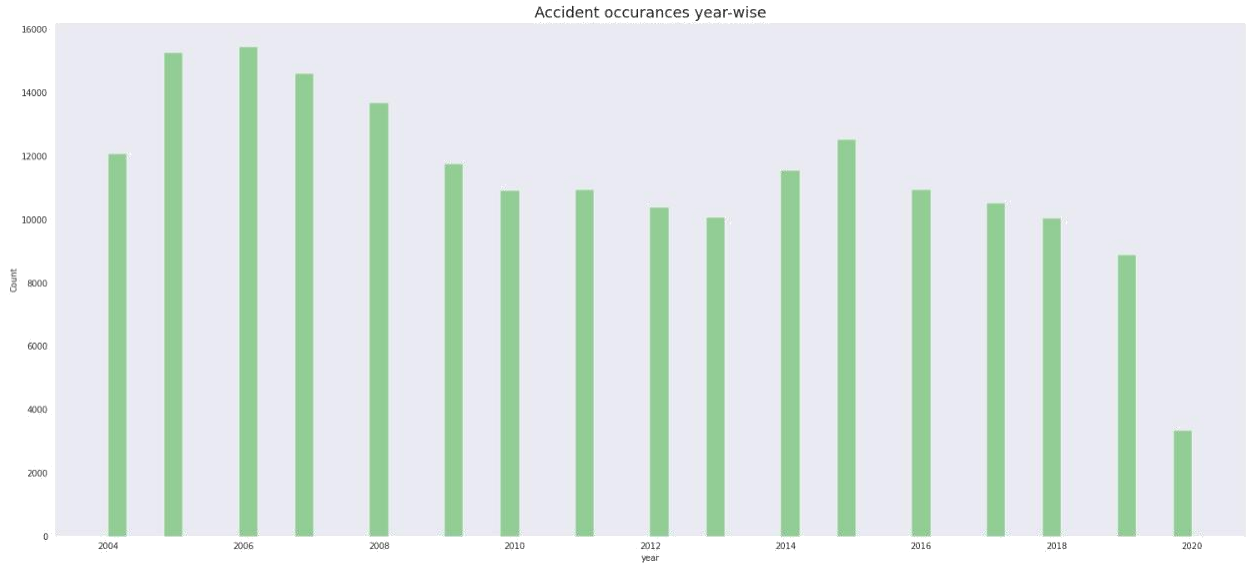
3.1 Exploratory Analysis

Before we head towards running our cleaned dataset through Machine Learning models, we need to visualize the data to better understand what is hidden within these data. Since we have mostly categorical values, we decided to create histograms to understand when and where and how these traffic accidents took place.

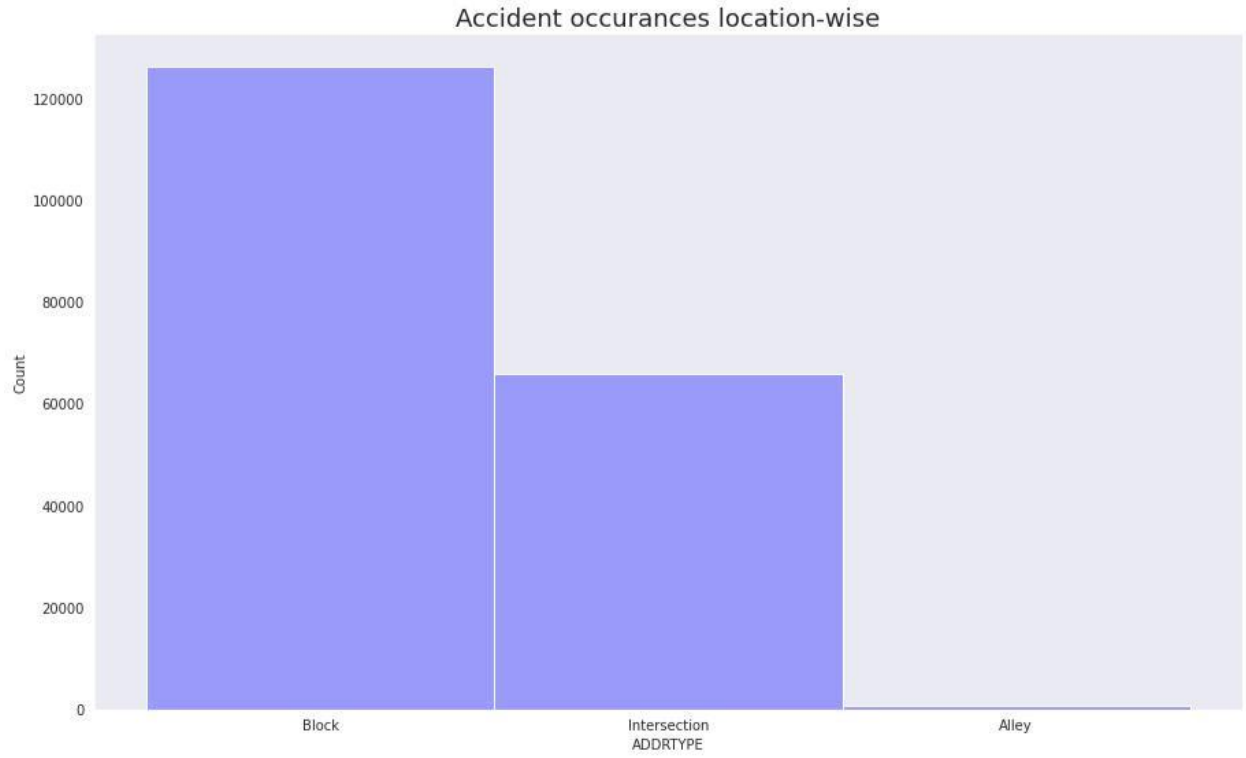
First, we look at how the incidents are spread across all the months. Immediately we see that number of traffic accidents are more or less same in all months. However, it does peak in October.



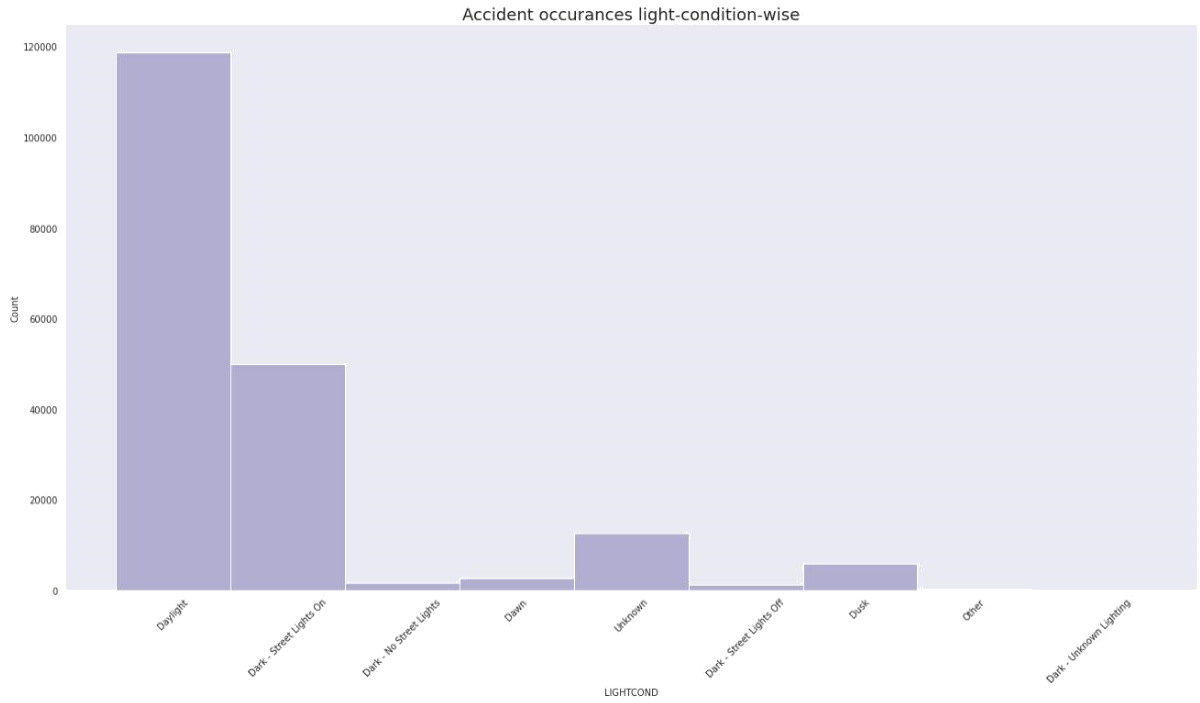
And then we look at the data year-wise. Since the data collection started at 2004, the number of traffic case incidents have been slowly decreasing which is a good sign.



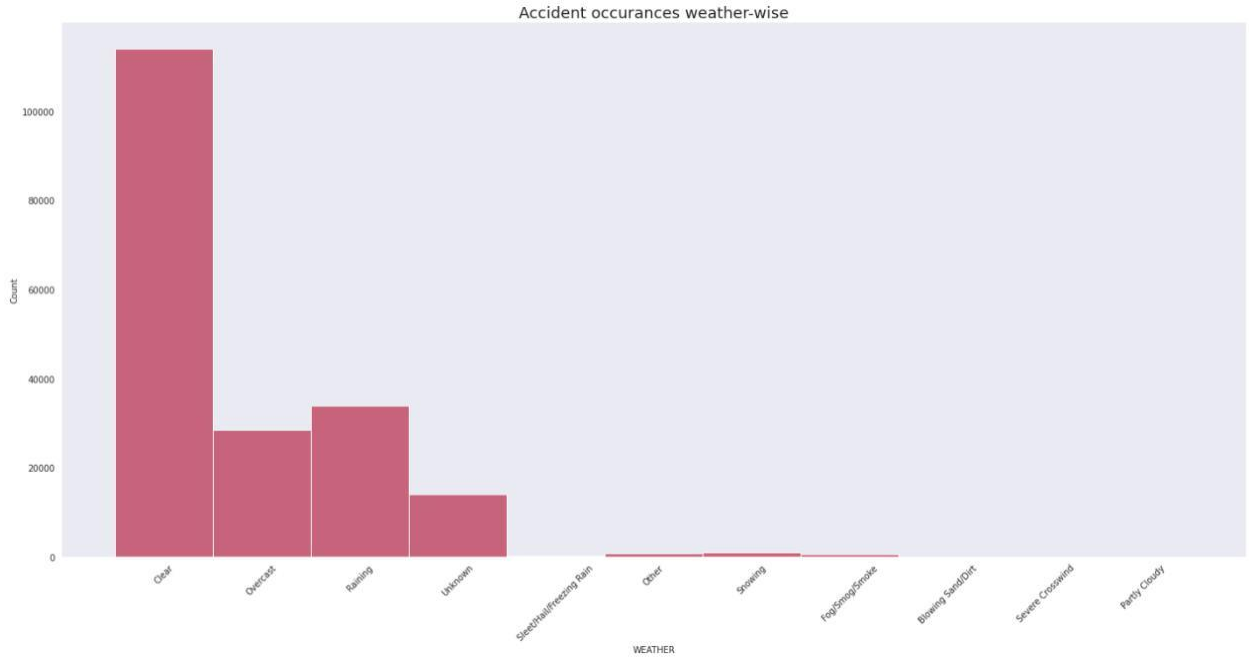
From the next histogram, we can conclude that most of the accidents take place in busy places as the number of accidents in Block area is way more than Intersections and alleys.



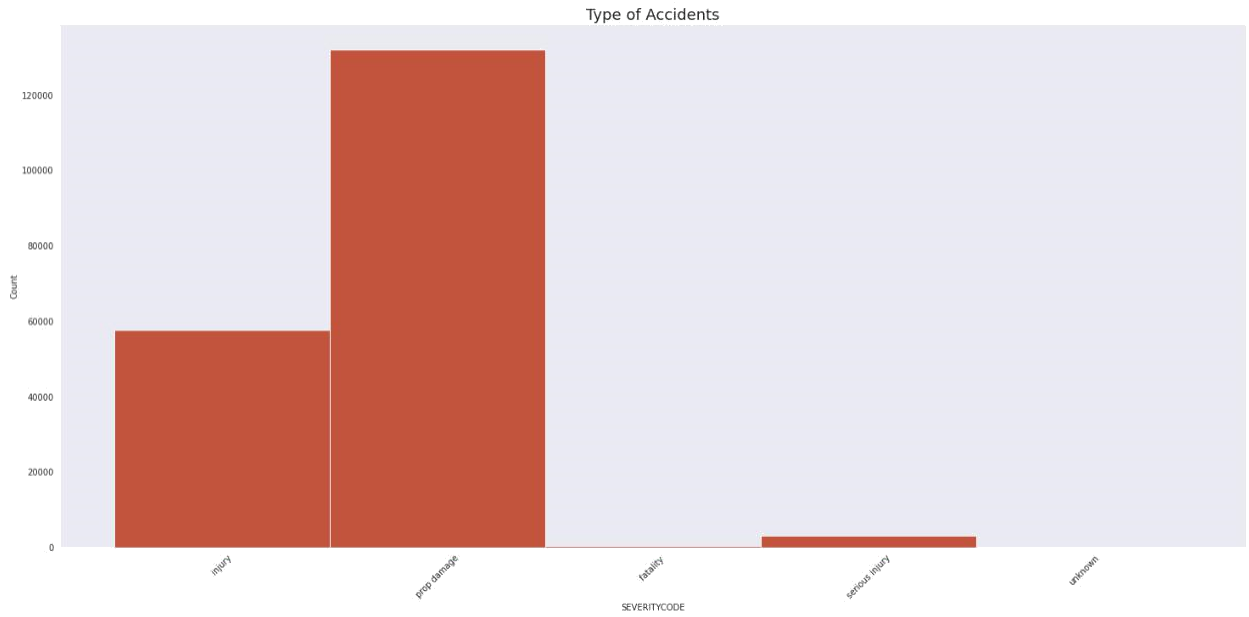
In our next visualization, we look at the count of accidents on basis of the time of the day it took place. Majority of the traffic accidents took place at daylight. Less than half of it took place at night, out of which most were in places where there are traffic lights. It rises a question that since most accidents are taking place in well lit areas, does that mean majority accidents are due to not following traffic rules rather than uncontrolled situation?



Then in the next graph, we look at the weather during the accidents. Majority of the traffic accidents took place on clear days. Only few cases took place in rainy days. This again rises the question, does that mean majority accidents are due to not following traffic rules rather than uncontrolled situation?



In the last visualization, we look at the severity of each accidents. Most of the accidents are usually just property damages and in half of that number, injuries took place. There is also extremely low number of fatality cases which is a good sign.



3.2 Machine Learning Model Selection

The Machine Learning models we will be using are –

- **Logistic regression -**
    - **Definition:** Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.
    - **Advantages:** Logistic regression is designed for this purpose (classification) and is most useful for understanding the influence of several independent variables on a single outcome variable.
    - **Disadvantages:** Works only when the predicted variable is binary, assumes all predictors are independent of each other and assumes data is free of missing values.
- **K-Nearest Neighbors –**
    - **Definition:** Neighbors based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point.
    - **Advantages:** This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.
    - **Disadvantages:** Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

- **Decision Tree -**
  - **Definition:** Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.
  - **Advantages:** Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.
  - **Disadvantages:** Decision tree can create complex trees that do not generalize well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

4. Results

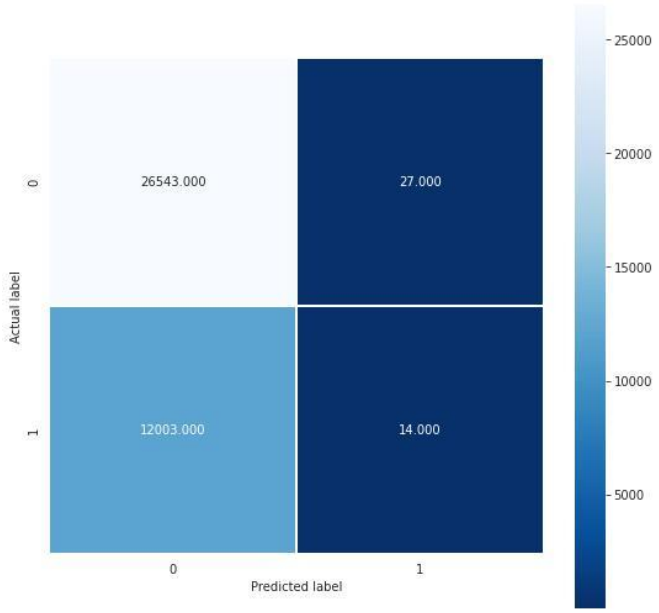
4.1 Logistic regression

We used Logistic Regression from the scikit-learn library to run the Logistic Regression Classification model on our cleaned dataset. Using log_loss metric, we were able to decide on the best available parameters for our model. We used 'liblinear' as our Solver and we took C (regularization strength) value as '0.001'.

4.1.1 Classification Report

Jaccard Similarity Score	F1 Score
0.69	**0.56**

4.1.2 Confusion Matrix



4.2 K-Nearest Neighbors

We used K-Nearest Neighbors from the scikit-learn library to run the K-Nearest Neighbors machine learning classifier on our cleaned dataset. Using accuracy_score metric, we were able to decide on the best available parameter for our model. We used 8 as K's value as the highest elbow bend exists at 8.

4.2.1 Classification Report

Jaccard Similarity Score	F1 Score
0.68	**0.58**

4.2.2 Best kNN value

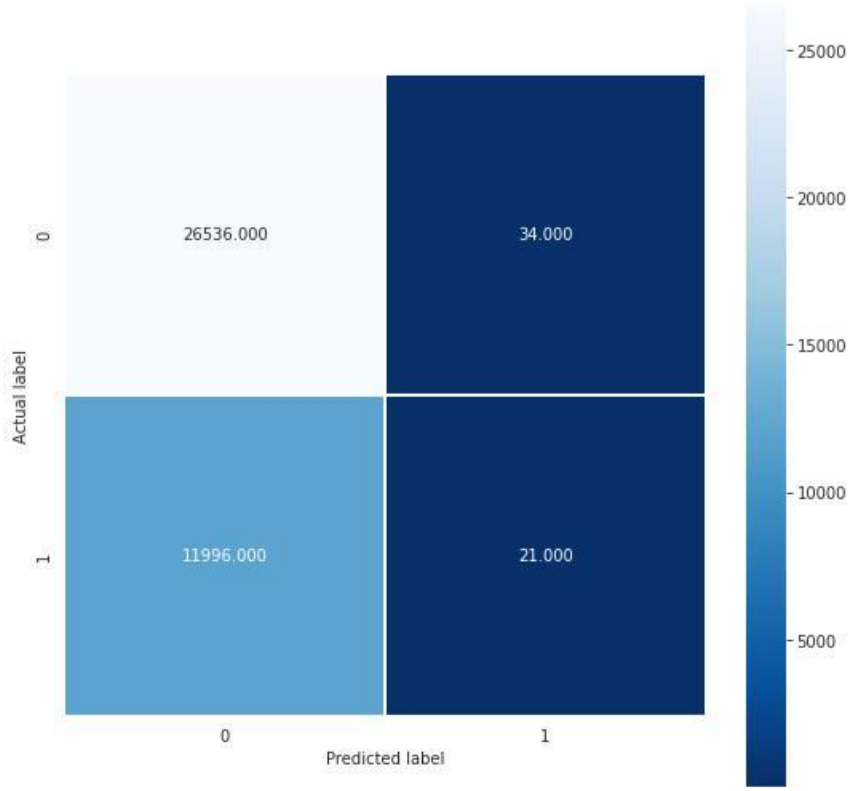
4.3 Decision Tree

We used Decision Tree Classifier from the scikit-learn library to run the Decision Tree Classification model on our cleaned dataset. Using jaccard_similarity_score and f1_score metric, we were able to decide on the best available parameter for our model. We found out; our model should have a depth of '3' to perform the best.

4.3.1 Classification Report

Jaccard Similarity Score	F1 Score
0.69	**0.56**

4.3.2 Confusion Matrix



5.Discussion

Algorithm	Jaccard Similarity Score	F1 Score
Logistic Regression	**0.69**	**0.56**
KNN	**0.68**	**0.58**
Decision Tree	**0.69**	**0.56**

The metrics we used to compare the accuracy of our models are the Jaccard Similarity Score and F1 Score.

The Jaccard similarity index (sometimes called the Jaccard similarity coefficient) compares members for two sets to see which members are shared and which are distinct. It is a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two populations. Although it is easy to interpret, it is extremely sensitive to small samples sizes and may give erroneous results, especially with exceedingly small samples or data sets with missing observations.

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It is calculated from the precision and recall of the test, 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.

While all our models performed almost equally in those tests, KNN took too much time to yield result (almost 2 mins). Also, when it came to model score, all the models performed closed to each other. But score of Decision Tree was the highest, 0.6886 compared to KNN’s 0.6803 and Logistic Regression’s 0.6882.

6.Conclusion

We achieved 68% accuracy using Logistic Regression, KNN, and Decision Tree. However, the performance is not up to the mark. It could have been better if we were able to use more features with more categorized values. Another thing I would say that it is not appropriate to simplify the severity values into binary classes. If we could create a range, then we might have been able to predict severity of traffic cases better using regression models.