Car Accident Severity Analysis

(Applied Data Science Capstone)

The project aims to study factors which play a role in severity of accidents using Machine Learning Models

1. Introduction

1.1 Background

Seattle, also known as the Emerald city, is Washington State's largest city, with home to a large tech industry with Microsoft and Amazon headquartered in its metropolitan area. As of 2020, it has a total metro area population of 3.4 million (www.macrotrends.net). The total number of personal vehicles in Seattle in the year 2016 hit a new high of nearly 444,000 vehicles. In one South Lake Union census tract, the car population has more than doubled since 2010 (www.seattletimes.com). The increase in car ownership rates can lead to higher numbers of accidents on the road because of a simple probability. Worldwide, approximately 1.35 million people die in road crashes each year, on average 3,700 people lose their lives every day on the roads and an additional 20-50 million suffer non-fatal injuries, often resulting in long-term disabilities.

1.2 Problem

The world suffers due to car accidents, including the USA. National Highway Traffic Safety Administration of the USA suggests that the economical and societal harm from car accidents can cost up to $871 billion in a single year. According to 2017 WSDOT data, a car accident occurs every 4 minutes and a person dies due to a car crash every 20 hours in the state of Washington while Fatal crashes went from 508 in 2016 to 525 in 2017, resulting in the death of 555 people. The project aims to predict how severity of accidents can be reduced based on a few factors.

1.3 Stakeholders

The reduction in severity of accidents can be beneficial to the Public Development Authority of Seattle which works towards improving those road factors and the car drivers themselves who may take precaution to reduce the severity of accidents.

2.0 Understanding Data

2.1 Data Analysis

Dataset provided has 194673 responses and 38 variables. Below is the basic information about data, and each variable data type. Second figure shows variable with blanks in them, many of them have large no of blanks.

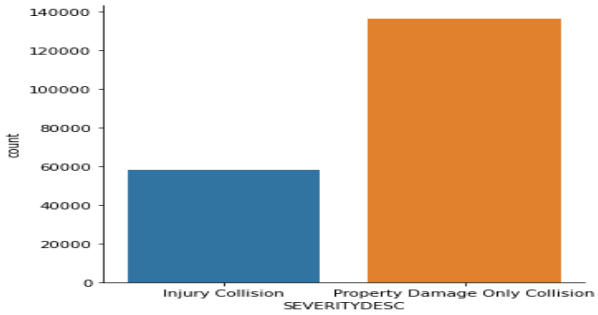
```
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   SEVERITYCODE    194673 non-null  int64
 1   X               189339 non-null  float64
 2   Y               189339 non-null  float64
 3   OBJECTID        194673 non-null  int64
 4   INCKEY          194673 non-null  int64
 5   COLDETKEY       194673 non-null  int64
 6   REPORTNO        194673 non-null  object
 7   STATUS          194673 non-null  object
 8   ADDRTYPE        192747 non-null  object
 9   INTKEY          65070 non-null   float64
 10  LOCATION        191996 non-null  object
 11  EXCEPTRSNCODE   84811 non-null   object
 12  EXCEPTRSNDESC   5638 non-null    object
 13  SEVERITYCODE.1  194673 non-null  int64
 14  SEVERITYDESC    194673 non-null  object
 15  COLLISIONTYPE   189769 non-null  object
 16  PERSONCOUNT     194673 non-null  int64
 17  PEDCOUNT        194673 non-null  int64
 18  PEDCYLCOUNT     194673 non-null  int64
 19  VEHCOUNT        194673 non-null  int64
 20  INCDATE         194673 non-null  object
 21  INCDTTM         194673 non-null  object
 22  JUNCTIONTYPE    188344 non-null  object
 23  SDOT_COLCODE    194673 non-null  int64
 24  SDOT_COLDESC    194673 non-null  object
 25  INATTENTIONIND  29805 non-null   object
 26  UNDERINFL       189789 non-null  object
 27  WEATHER         189592 non-null  object
 28  ROADCOND        189661 non-null  object
 29  LIGHTCOND       189503 non-null  object
 30  PEDROWNOTGRNT   4667 non-null    object
 31  SDOTCOLNUM      114936 non-null  float64
 32  SPEEDING        9333 non-null    object
 33  ST_COLCODE      194655 non-null  object
 34  ST_COLDESC      189769 non-null  object
 35  SEGLANEKEY      194673 non-null  int64
 36  CROSSWALKKEY    194673 non-null  int64
 37  HITPARKEDCAR    194673 non-null  object
dtypes: float64(4), int64(12), object(22)
memory usage: 56.4+ MB
```

```
X                 5334
Y                 5334
ADDRTYPE          1926
INTKEY          129603
LOCATION          2677
EXCEPTRSNCODE   109862
EXCEPTRSNDESC   189035
COLLISIONTYPE     4904
JUNCTIONTYPE      6329
INATTENTIONIND  164868
UNDERINFL         4884
WEATHER           5081
ROADCOND          5012
LIGHTCOND         5170
PEDROWNOTGRNT   190006
SDOTCOLNUM       79737
SPEEDING        185340
ST_COLCODE          18
ST_COLDESC        4904
dtype: int64
```

We need to predict severity of accident, using ‘SEVERITYCODE’. However, dataset is unbalanced. We could use technique like SMOT to balance the data.

```
accident_df['SEVERITYCODE'].value_counts()
```

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



The models aim was to predict the severity of an accident, considering that, the variable of Severity Code was in the form of 1 (Property Damage Only) and 2 (Injury Collision) which were encoded to the form of 0 (Property Damage Only) and 1 (Injury Collision). Furthermore, the Y was given value of 1 whereas N and no value was given 0 for the variables Inattention, Speeding and Under the influence. For lighting condition, Light was given 0 along with Medium as 1 and Dark as 2. For Road Condition, Dry was

assigned 0, Mushy was assigned 1 and Wet was given 2. As for Weather Condition, 0 is Clear, Overcast is 1, Windy is 2 and Rain and Snow was given 3. 0 was assigned to the element of each variable which can be the least probable cause of severe accident whereas a high number represented adverse condition which can lead to a higher accident severity.

2.2 Feature Selection

Finally selecting the below variables in the analysis along with SEVERITYCODE, which is our target variable.

Feature Variables	Description
ADDRTYPE	Collision address type: • Alley • Block • Intersection
JUNCTIONTYPE	Category of junction at which collision took place
INATTENTIONIND	Whether or not collision was due to inattention. (Y/N
UNDERINFL	Whether or not a driver involved was under the influence of drugs or alcohol.
WEATHER	A description of the weather conditions duringthe time of the collision.
ROADCOND	The condition of the road during the collision.
LIGHTCOND	The light conditions during the collision.
PEDROWNOTGRNT	Whether or not the pedestrian right of way was not granted. (Y/N)
SDOT_COLCODE	A description of the collision corresponding tothe collision code.
SPEEDING	Whether or not speeding was a factor in the collision. (Y/N)

Post selecting the variables and then checking for blanks. Approximately 6% of data is lost in the process to remove blanks.

3. Methodology

3.1 Data Collection

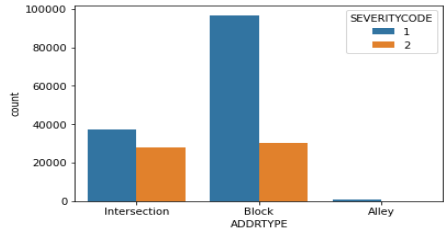
The dataset used for this project is based on car accidents which have taken place within the city of Seattle, Washington from the year 2004 to 2020. This data is regarding car accidents the severity of each car accidents along with the time and conditions under which each accident occurred. The data set used for this project can be found here.

3.2 Exploratory Analysis

- ADDRTYPE – Data shows Block has most collisions. Intersection has very high Injury Collision with respect to number of collisions occur in Intersection.

```
#Intersection more percentage of INJURY accident, block has very number of property damage accidents
sns.countplot(x='ADDRTYPE',hue='SEVERITYCODE',data=accident_df)
pd.pivot_table(accident_df,index=['SEVERITYDESC'],columns=['ADDRTYPE'],values='SEVERITYCODE',aggfunc='count')
```

SEVERITYDESC	ADDRTYPE Alley	Block	Intersection
Injury Collision	82	30096	27819
Property Damage Only Collision	669	96830	37251

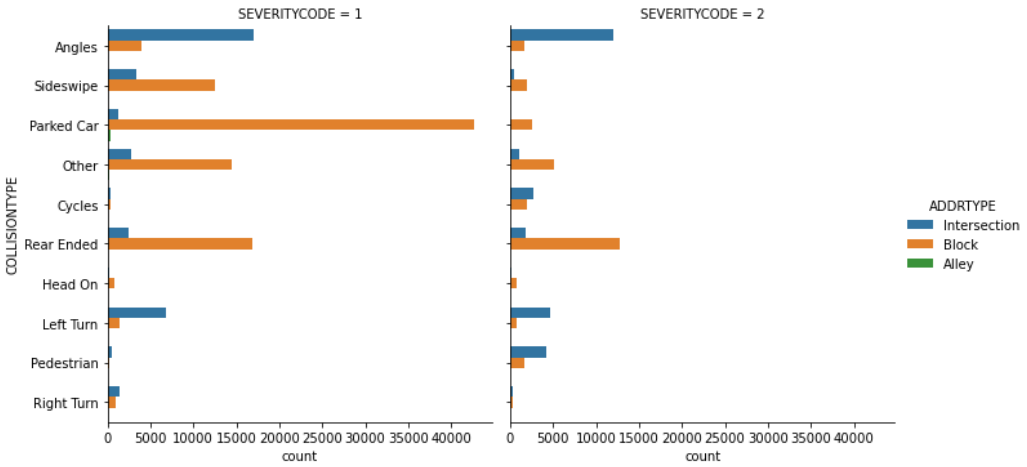


- COLLISIONTYPE – Parked car face most collisions, which mostly occur in Block. Rear ended have high number of injury collision. Cyclist and Pedestrian have injury collisions

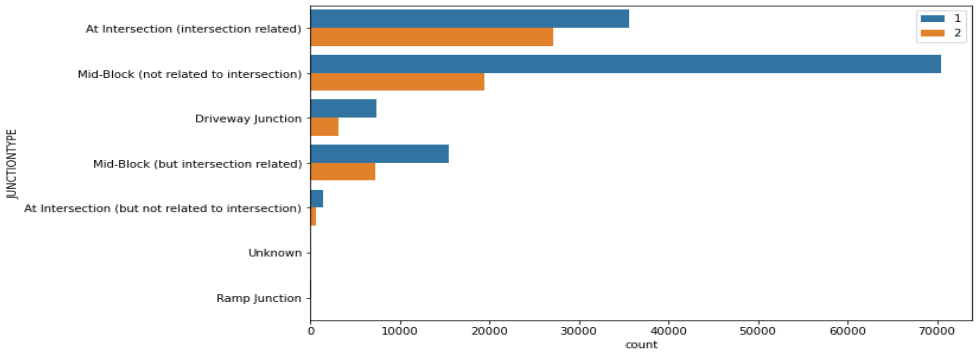
```
#Cyclist and Pedestrian have injury collisions
#Parked Car collosion with property damage mostly occur in block
#Rear ended have high number of injury collision
plt.figure(figsize=(10,10))
sns.catplot(y='COLLISIONTYPE',col='SEVERITYCODE',hue='ADDRTYPE',data=accident_df,kind='count',aspect=1)
```

```
<seaborn.axisgrid.FacetGrid at 0x1d85cb99be0>

<Figure size 720x720 with 0 Axes>
```



- JUNCTIONTYPE – “Mid-Block (not related to intersection)” and “At Intersection (intersection related)” have high number of collisions.



- SDOT_COLDESC – PEDACYCLIST/PEDESTRIAN face mostly Injury Collision. Trying group as category list is large.

SEVERITYDESC SDOT_COLCODE	Injury Collision	Property Damage Only Collision
0	708	9079
1	43079	115253
2	8836	1108
3	3872	10472
4	21	251
5	19	100
6	1481	216
7	172	6

```
accident_df['SDOT_COLCODE_New']=accident_df['SDOT_COLCODE'].copy()
accident_df['SDOT_COLCODE'].replace([11,12,13,14,15,16],1,inplace=True) #MOTOR VEHICLE STRUCK MOTOR VEHICLE
accident_df['SDOT_COLCODE'].replace([18,21,22,23,24],2,inplace=True)    #MOTOR VEHICLE STRUCK PEDALCYCLIST/PEDESTRIAN
accident_df['SDOT_COLCODE'].replace([25,26,27,28,29],3,inplace=True)    #MOTOR VEHICLE SELF
accident_df['SDOT_COLCODE'].replace([31,32,33,34,35,36],4,inplace=True) #DRIVERLESS VEHICLE STRUCK MOTOR
accident_df['SDOT_COLCODE'].replace([44,46,47,48],5,inplace=True)       #DRIVERLESS VEHICLE SELF
accident_df['SDOT_COLCODE'].replace([51,52,53,54,55,56,58],6,inplace=True) #PEDALCYCLIST STRUCK
accident_df['SDOT_COLCODE'].replace([61,64,66,68,69],7,inplace=True)     #PEDALCYCLIST SELF
accident_df['SDOT_COLCODE'].replace(0,0,inplace=True)                    #NOT APPLICABLE
```

- WEATHER – Mostly collisions occur when weather is clear.

SEVERITYDESC WEATHER	Injury Collision	Property Damage Only Collision
Blowing Sand/Dirt	15	41
Clear	35840	75295
Fog/Smog/Smoke	187	382
Other	116	716
Overcast	8745	18969
Partly Cloudy	3	2
Raining	11176	21969
Severe Crosswind	7	18
Sleet/Hail/Freezing Rain	28	85
Snowing	171	736
Unknown	816	14275

- ROADCOND – Most collisions take place when road condition is Dry.

```
Dry               124510
Wet                47474
4                  15210
Ice                 1209
Snow/Slush          1004
Standing Water       115
Sand/Mud/Dirt         75
Oil                   64
Name: ROADCOND, dtype: int64
```

- LIGHTCOND – Most collisions take place in Daylight condition.

```
Daylight                    116137
Dark - Street Lights On      48507
4                            13708
Dusk                          5902
Dawn                          2502
Dark - No Street Lights       1537
Dark - Street Lights Off      1199
Dark - Unknown Lighting         11
Name: LIGHTCOND, dtype: int64
```

- INATTENTIONIND, PEDROWNOTGRNT, SPEEDING, UNDERINFL – Have Y and blank row.

  Replacing blank with N. Replacing Yes with 1 and No with 0.

```
accident_df['INATTENTIONIND'].value_counts()
```

```
0.0     164868
1.0      29805
Name: INATTENTIONIND, dtype: int64
```

```
accident_df['PEDROWNOTGRNT'].value_counts()
```

```
0.0     190006
1.0       4667
Name: PEDROWNOTGRNT, dtype: int64
```

```
accident_df['SPEEDING'].value_counts()
```

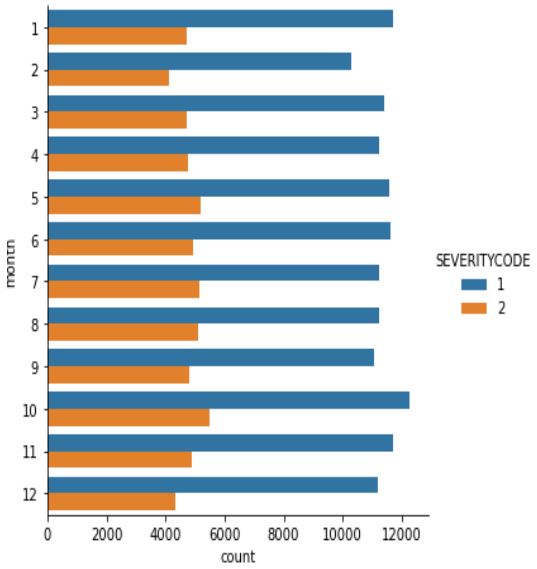
```
0.0     185340
1.0       9333
Name: SPEEDING, dtype: int64
```

```
accident_df['UNDERINFL'].value_counts()
```

```
0     185552
1       9121
Name: UNDERINFL, dtype: int64
```

- INCDTTM – Converting it to Datetime, to run additional date time analysis. Like checking Month, Year, Hour of the day, week of the day with collision to find any pattern.

SEVERITYDESC	Injury Collision	Property Damage Only Collision
year		
2004	3647	8218
2005	4450	10665
2006	4350	10838
2007	4017	10439
2008	3767	9893
2009	3378	8356
2010	3245	7563
2011	3099	7820
2012	3467	7440
2013	3290	7287
2014	3490	8351
2015	3752	9243
2016	3714	7945
2017	3419	7454
2018	3358	7061
2019	3062	6350
2020	683	1562



3.3 Machine Learning Model Selection

The machine learning models used are Logistic Regression, Decision Tree Analysis, k-Nearest Neighbor (KNN) and Random Forest. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The Decision Tree Analysis breaks down a data set into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Random forest is a technique used in modeling predictions and behavior analysis and is built on decision trees. It contains many decision trees that represent a distinct instance of the classification of data input into the random forest. The random forest technique takes consideration of the instances individually, taking the one with the majority of votes as the selected prediction. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (based on distance). The reason why Decision Tree Analysis, Logistic Regression, KNN and Random Forest methods were chosen is because the Support Vector Machine (SVM) model is inaccurate for large data sets, while this data set has more than 180,000 rows filled with data. Furthermore, SVM works best with dataset filled with text and images.

4. Results

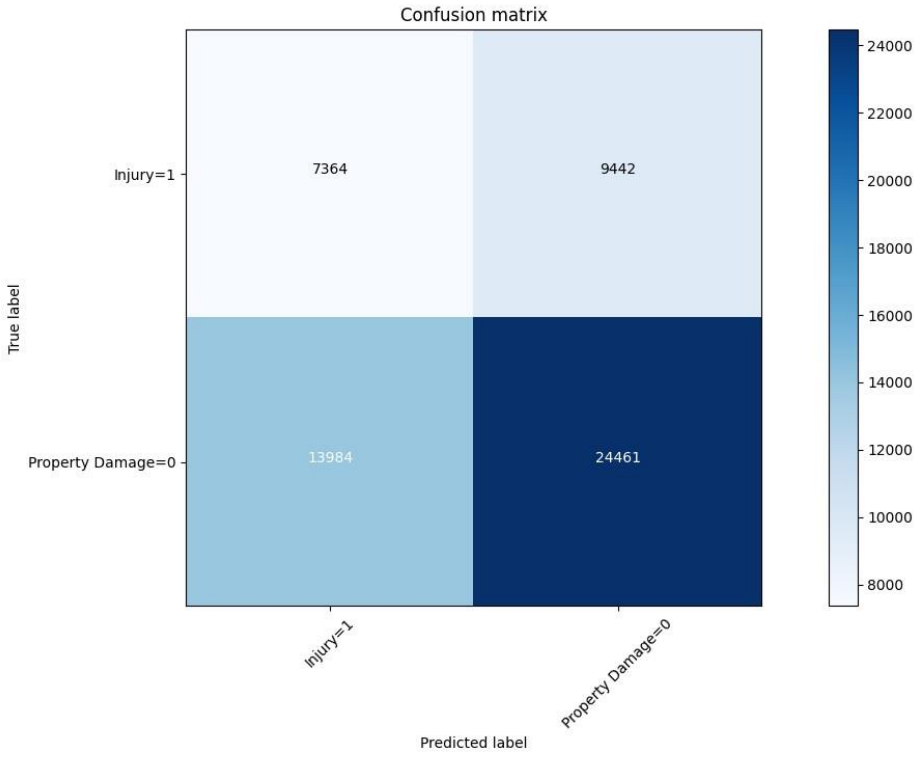
4.1 Decision Tree Analysis

Decision Tree Classifier from the scikit-learn library was used to run the Decision Tree Classification model on the Car Accident Severity data. The criterion chosen for the classifier was 'entropy' and the max depth was '6'. The post-SMOTE balanced data was used to predict and fit the Decision Tree Classifier.

4.1.1 Classification Report

	Precision	Recall	f1-score
0	0.64	0.72	0.68
1	0.44	0.34	0.39
Accuracy	0.58		
Macro Avg	0.54	0.53	0.53
Weighted Avg	0.56	0.58	0.56

4.1.2 Confusion Matrix



4.2 Logistic Regression

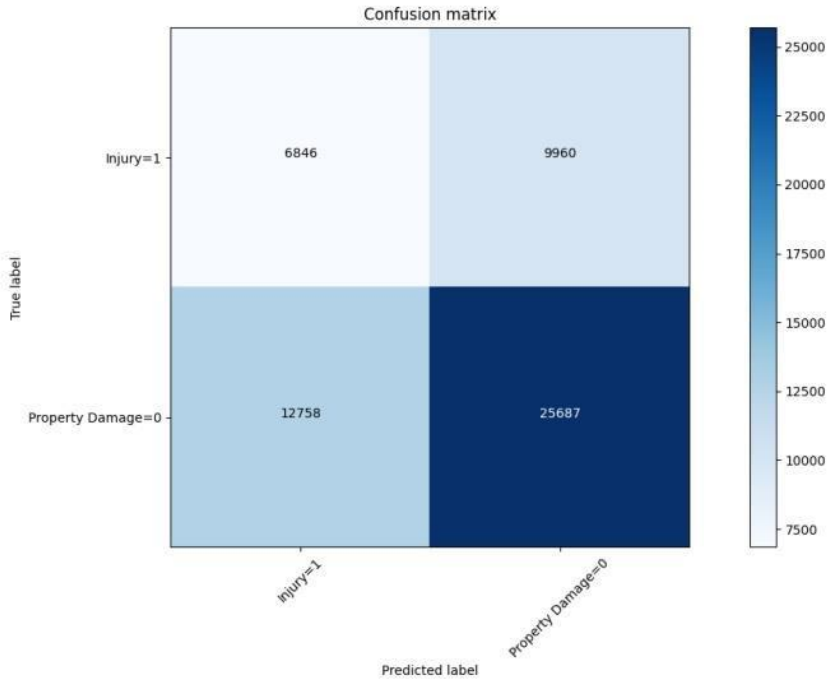
Logistic Regression from the scikit-learn library was used to run the Logistic Regression Classification model on the Car Accident Severity data. The C used for regularization strength was '0.01' whereas the solver used was 'liblinear'. The post-SMOTE balanced data was used to predict and fit the Logistic Regression Classifier.

4.2.1 Classification Report

	Precision	Recall	f1-score
0	0.72	0.67	0.69
1	0.35	0.41	0.38

Accuracy	0.59		
Macro Avg	0.53	0.54	0.53
Weighted Avg	0.61	0.59	0.60
Log Loss	0.68		

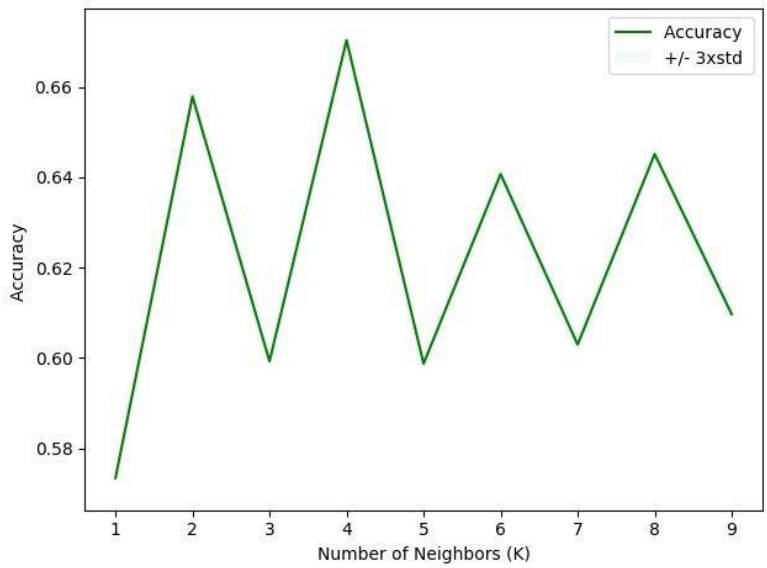
4.2.2 Confusion Matrix



4.3 k-Nearest Neighbor

k-Nearest Neighbor classifier was used from the scikit-learn library to run the k-Nearest Neighbor machine learning classifier on the Car Accident Severity data. The best K, as shown below, for the model where the highest elbow bend exists is at 4. The post-SMOTE balanced data was used to predict and fit the k-Nearest Neighbor classifier.

4.3.1 Best kNN value



4.3.2 Classification Report

	Precision	Recall	f1-score
0	0.93	0.70	0.80
1	0.08	0.32	0.13
Accuracy	0.67		
Macro Avg	0.50	0.51	0.46
Weighted Avg	0.86	0.67	0.75

5. Discussion

Algorithm	Average f1-Score	Property Damage (0) vs Injury (1)	Precision	Recall
Decision Tree	0.56	0	0.64	0.72
		1	0.44	0.34
Logistic Regression	0.60	0	0.72	0.67
		1	0.35	0.41
k-Nearest Neighbor	0.75	0	0.93	0.70
		1	0.08	0.32

5.1 Average f1-Score

f1-score is a measure of accuracy of the model, which is the harmonic mean of the model's precision and recall. Perfect precision and recall is shown by the f1-score as 1, which is the highest value for the f1-score, whereas the lowest possible value is 0 which means that either precision or recall is 0. The f1-score shown above is the average of the individual f1-scores of the two elements of the target variable i.e. Property Damage and Injury. When comparing the f1-scores of the three models, we can see that k-Nearest Neighbor has the highest f1-score meaning that it has a higher precision and recall of the other two models. Whereas, the Decision Tree model's f1-score is the lowest of the three at 0.56. Lastly, the f1-score of the Logistic Regression is at 0.60 which can be considered as an above average score. However, the average f1-score doesn't depict the true picture of the models accuracy because of the different precision and recall of the model for both the elements of the target variable. Hence, it is biased more towards the precision and recall of Property Damage due to its weightage in the model.

5.2 Precision

Precision refers to the percentage of results which are relevant, in simpler terms it can be seen as how many of the selected items from the model are relevant. Mathematically, it is calculated by dividing true positives by true positive and false positive. The highest precision for Property Damage is for Logistic Regression, whereas for Injury it is the Decision Tree. The Precision is calculated individually above in order to understand how accurate the model is at predicting Property Damage and Injury individually. For the Decision Tree the precision of 0 is 0.64 and for 1 it is 0.44 which is fairly good. As for the Logistic Regression model, for 0 it is at 0.72 and for 1 it is 0.35. Lastly, for the k-Nearest Neighbor at 0 it is 0.93, which is highly accurate, however for 1 it is 0.08, extremely low. In terms of precision, the best performing model is the decision tree.

5.3 Recall

Recall refers to the percentage of total relevant results correctly classified by the algorithm. In simpler terms, it tells how many relevant items were selected. It is calculated by dividing true positives by true positive and false negative. The highest precision for 0 is when using the k-Nearest Neighbor model at 0.70 as for 1 it is the Logistic Regression model at 0.41. The recall for both Property Damage and Injury is almost identical for the Decision Tree and k-Nearest Neighbor model. As for the Logistic Regression, the recall for Property Damage is 0.67 and for Injury it is 0.41. The recall for Property Damage and Injury is the most balanced in terms of being good for both the outputs of the target variable.

6. Conclusion

When comparing all the models by their f1-scores, Precision and Recall, we can have a clearer picture in terms of the accuracy of the three models individually as a whole and how well they perform for each output of the target variable. When comparing these scores, we can see that the f1-score is highest for k-Nearest Neighbor at 0.75. However, later when we compare the precision and recall for each of the model, we can see that the k-Nearest Neighbor model performs poorly in the precision of 1 at 0.08. The variance is too high for the model to be selected as a viable option. When looking at the other two models, we can see that the Decision Tree has a more balanced precision for 0 and 1. Whereas, the Logistic Regression is more balanced when it comes to recall of 0 and 1. Furthermore, the

average f1-score of the two models are very close but for the Logistic Regression it is higher by 0.04. It can be concluded that the both the models can be used side by side for the best performance.

In retrospect, when comparing these scores to the benchmarks within the industry, it can be seen that they perform well but not as good as the benchmarks. These models could have performed better if a few more things were present and possible.

A balanced dataset for the target variable

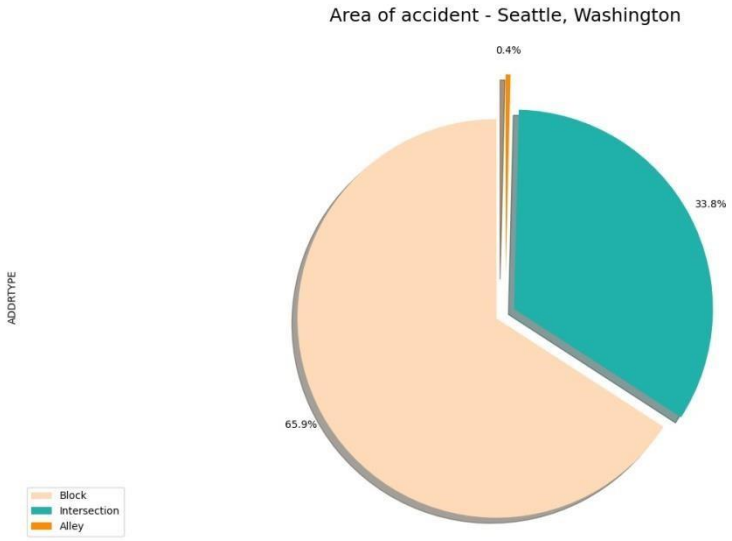
More instances recorded of all the accidents taken place in Seattle, Washington

Less missing values within the dataset for variables such as Speeding and Under the influence More factors, such as precautionary measures taken when driving, etc.

7. Recommendations

After assessing the data and the output of the Machine Learning models, a few recommendations can be made for the stakeholders. The developmental body for Seattle city can assess how much of these accidents have occurred in a place where road or light conditions were not ideal for that specific area and could launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors. Whereas, the car drivers could also use this data to assess when to take extra precautions on the road under the given circumstances of light condition, road condition and weather, in order to avoid a severe accident, if any.

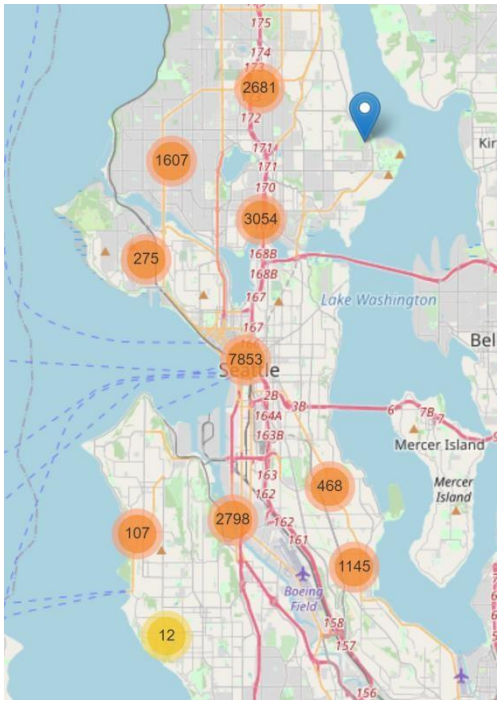
7.1 Public Development Authority of Seattle (PDAS)



Almost all of the accidents recorded have occurred on either a block or an intersection, the PDAS can take the following measures in response car accidents:

- Launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors
- Increased investment towards improving lighting and road conditions of the area which have high instances recorded
- Install safety signs on the roads and ensure that all precautions are being taken by people within the area

7.2 Car Drivers



A higher concentration of accidents can be mostly seen on the main roads of the city, specifically near the highway in the city center. The following steps can be taken by car drivers to avoid severe accidents:

- Be extra careful around the I-5 highway which goes through the city center since it has the highest proportion of accidents recorded of total seattle
- Most incidents occur under adverse weather, road and light conditions. Precautions should be taken under such circumstances, for e.g. driving slow on a wet road which may lead to loss of control