2020.9.7

Huijie Hu

Report about Car-Accident-Severity on Data Science Basis

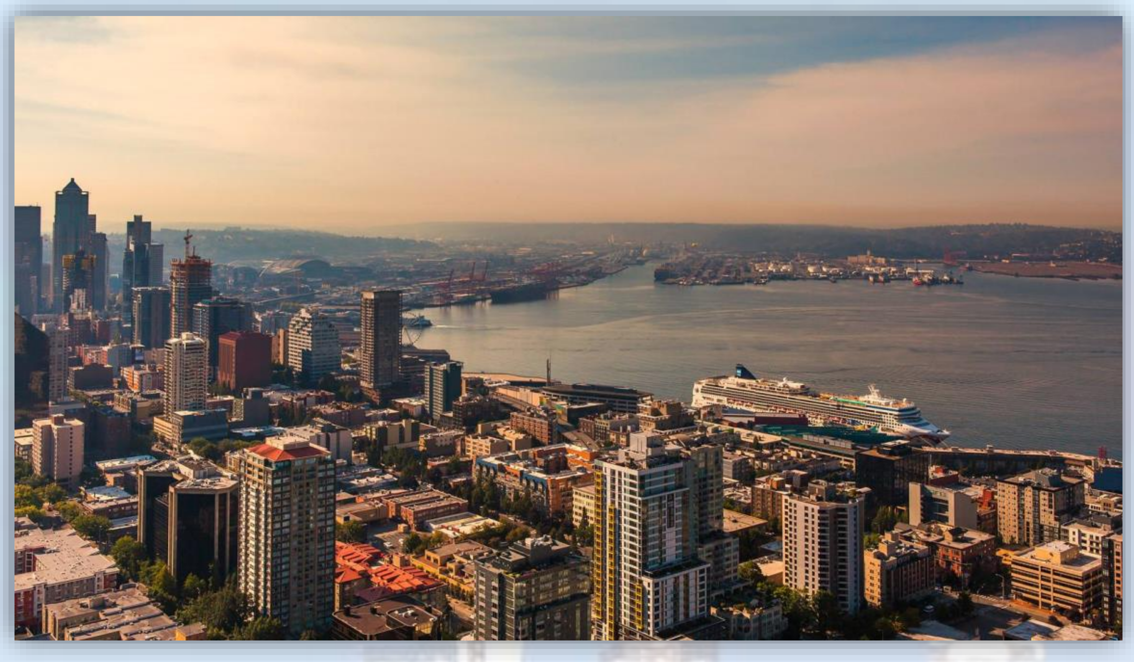
Applied Data Science IBM Capstone Project

Report about Car-Accident-Severity on Data Science Basis.

Content

1. Business Understanding

1.1 About Seattle



Seattle is a seaport city on the West Coast of the United States. It is the seat of King County, Washington. Seattle is the largest city in both the state of Washington and the Pacific Northwest region of North America. According to U.S. Census data released in 2019, the Seattle metropolitan area's population stands at 3.98 million, making it the 15th-largest in the United States. In July 2013, Seattle was the fastest-growing major city in the United States and remained in the top five in May 2015 with an annual growth rate of 2.1%. In July 2016, Seattle was again the fastest-growing major U.S. city, with a 3.1% annual growth rate.

Seattle is situated on an isthmus between Puget Sound (an inlet of the Pacific Ocean) and Lake Washington. It is the northernmost large city in the United States, located about 100 miles (160 km) south of the Canadian border. A major gateway for trade with Asia, Seattle is the fourth-largest port in North America in terms of container handling as of 2015.

1.2 About Car Accident

A traffic collision, also called a motor vehicle collision, car accident, or car crash, occurs when a vehicle collides with another vehicle, pedestrian, animal, road debris, or other stationary obstruction, such as a tree, pole or building. Traffic collisions often result in injury, disability, death, and property damage as well as financial costs to both society and the individuals involved.

Several factors contribute to the risk of collisions, including vehicle design, speed of operation, road design, road environment, driving skills, impairment due to alcohol or drugs, and behavior, notably distracted driving, speeding and street racing.

In 2013, 54 million people worldwide sustained injuries from traffic collisions. This resulted in 1.4 million deaths in 2013, up from 1.1 million deaths in 1990. About 68,000 of these occurred in children less than five years old.[2] Almost all high-income countries have decreasing death rates, while most low-income countries have increasing death rates due to traffic collisions. Middle-income countries have the highest rate with 20 deaths per 100,000 inhabitants, accounting for 80% of all road fatalities with 52% of all vehicles. While the death rate in Africa is the highest (24.1 per 100,000 inhabitants), the lowest rate is to be found in Europe (10.3 per 100,000 inhabitants).

1.3 Problem

Car accident is a worldwide problem that we are facing. It is crucial to know what are real reasons that cause the problem. Light, weather, drivers' focus, and humidity can all be factors. By analyzing the data provided by seattle_car_accident_severity, we can find out relation between different variables. People who are practitioners in car industry or buyers of cars, may be interested in this report.

2. Data Access & Data Wrangling

2.1 Import Libraries

2.2 File Handling

Open the file and read the raw data resources. As we can see, there's a lot of irregular data. Following steps that we'll take are aimed at wrangling data.

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEDROWNOTGRNT	SDOTCOLNUM	SPEEDING	ST_COLCODE	ST_COLDESC	SEGLANEKEY	CROSSWALKKEY	HITPARKEDCAR
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight	NaN	NaN	NaN	10	Entering at angle	0	0	N
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On	NaN	6354039.0	NaN	11	From same direction - both going straight - bo...	0	0	N
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight	NaN	4323031.0	NaN	32	One parked--one moving	0	0	N
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight	NaN	NaN	NaN	23	From same direction - all others	0	0	N
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight	NaN	4028032.0	NaN	10	Entering at angle	0	0	N

5 rows × 38 columns

2.3 Data Selection

There are a lot of unrelated data, such as: 'OBJECID', 'LOCATION','INTKEY', so we just drop them.

	SEVERITYCODE	X	Y	PERSONCOUNT	VEHCOUNT	JUNCTIONTYPE	INATTENTIONIND	WEATHER	ROADCOND	LIGHTCOND	SPEEDING
0	2	-122.323148	47.703140	2	2	At Intersection (intersection related)	NaN	Overcast	Wet	Daylight	NaN
1	1	-122.347294	47.647172	2	2	Mid-Block (not related to intersection)	NaN	Raining	Wet	Dark - Street Lights On	NaN
2	1	-122.334540	47.607871	4	3	Mid-Block (not related to intersection)	NaN	Overcast	Dry	Daylight	NaN
3	1	-122.334803	47.604803	3	3	Mid-Block (not related to intersection)	NaN	Clear	Dry	Daylight	NaN
4	2	-122.306426	47.545739	2	2	At Intersection (intersection related)	NaN	Raining	Wet	Daylight	NaN

2.4 Fix Missing Values

There are also a lot of 'NaN' or Null' values in this excel, so we need to fill them up. There are several ways to accomplish that and we chose to fill the blank with the most common values in this column.

For example:

```
# Replacing NaN value by Unknown
df1['ROADCOND'].replace(np.NaN, "Unknown", inplace=True)
df1['ROADCOND'].value_counts()
```

```
]: Dry                 124510
   Wet                  47474
   Unknown              20090
   Ice                   1209
   Snow/Slush            1004
   Other                  132
   Standing Water         115
   Sand/Mud/Dirt           75
   Oil                     64
   Name: ROADCOND, dtype: int64
```

```
# Replacing NaN value by Unknown
df1['WEATHER'].replace(np.NaN, "Unknown", inplace=True)
df1['WEATHER'].value_counts()
```

```
]: Clear                        111135
   Raining                       33145
   Overcast                      27714
   Unknown                       20172
   Snowing                         907
   Other                           832
   Fog/Smog/Smoke                  569
   Sleet/Hail/Freezing Rain        113
   Blowing Sand/Dirt                56
   Severe Crosswind                 25
   Partly Cloudy                     5
   Name: WEATHER, dtype: int64
```

2.5 Binary & Encoding

In data analysis processing, it is better for us to use 'int' or 'float' to analyze. However, as we can see, the dataset still contains a lot of 'str', like 'rainy', 'clean', etc.

What we are going to do is to encode them into numbers like ‘1’,’2’,’3’.

For example.

```
#Converting Severity Code from (1/2) tp (0/1)
severity_code = df1['SEVERITYCODE'].values
labels = preprocessing.LabelEncoder()
labels.fit([1, 2])
severity_code = labels.transform (severity_code)
df1["SEVERITYCODE"] = severity_code
```

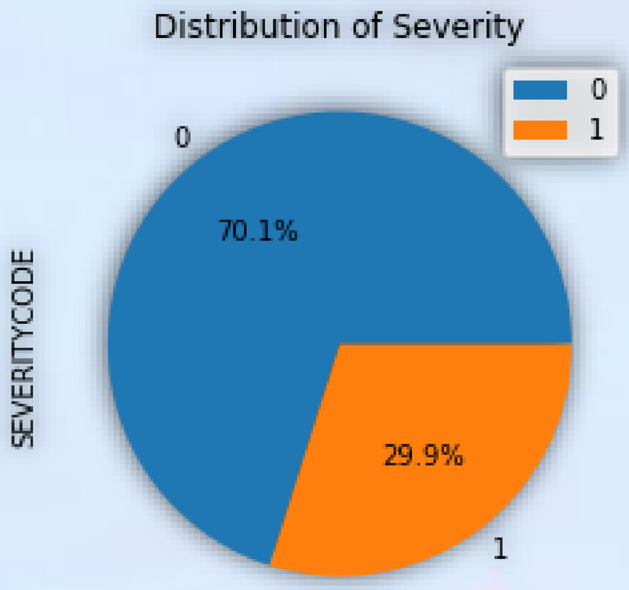
```
encoding_WEATHER = {"WEATHER":
                        {"Clear": 1,
                         "Unknown": 1,
                         "Other": 1,
                         "Raining": 2,
                         "Overcast": 3,
                         "Snowing": 4,
                         "Fog/Smog/Smoke": 5,
                         "Sleet/Hail/Freezing Rain": 6,
                         "Blowing Sand/Dirt": 7,
                         "Severe Crosswind": 8,
                         "Partly Cloudy": 9}}
df1.replace(encoding_WEATHER, inplace=True)
df1['WEATHER'].value_counts()
```

```
]:  1    132139
    2     33145
    3     27714
    4       907
    5       569
    6       113
    7        56
    8        25
    9         5
    Name: WEATHER, dtype: int64
```

3. Exploratory Data Analysis

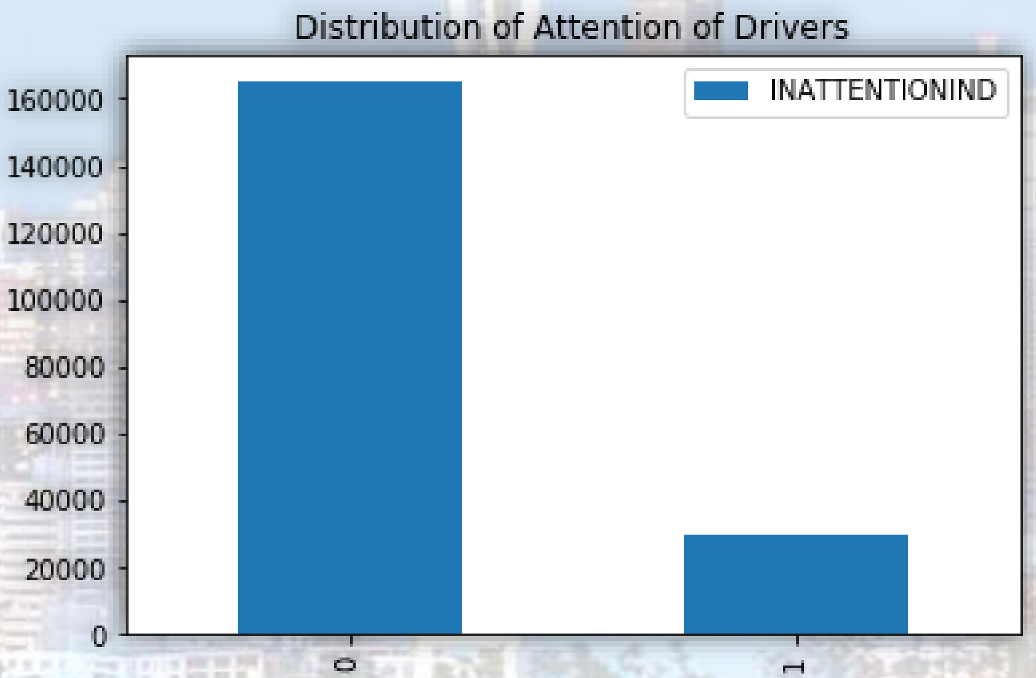
3.1 Basic Data Visualization

In this sector, we provide some basic data visualization. Because we just did encoding and binary firstly, there is no text in following charts.



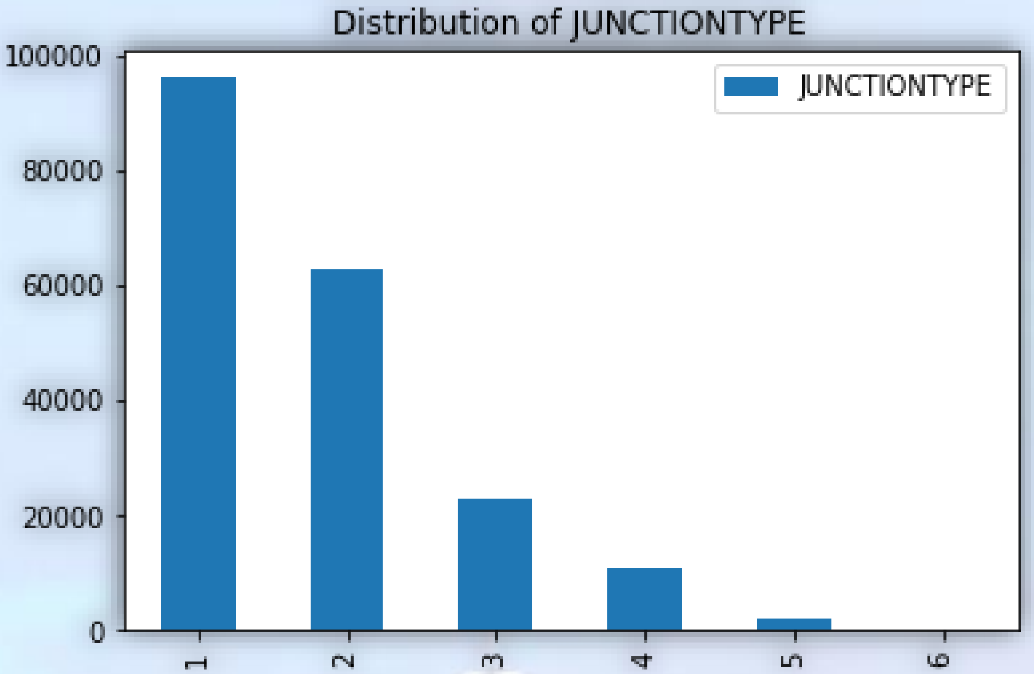
0: Prob damage

1: Injury



0: Attention

1:Inattention



1: Midblock/Unknown

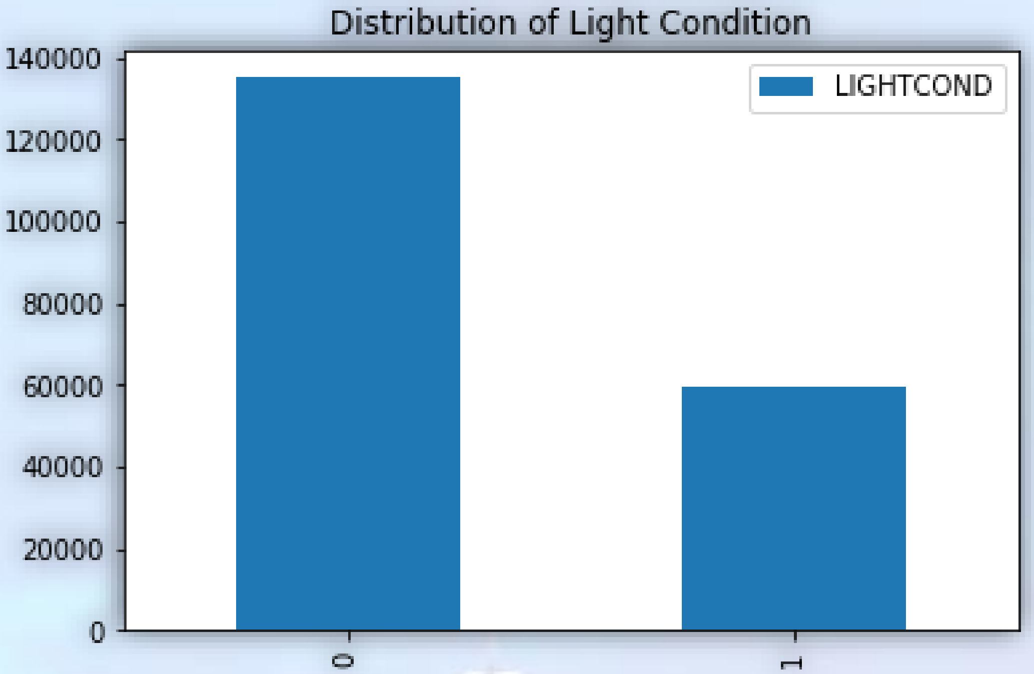
2: At intersection

3: Midblock related to intersection

4: Driveaway junction

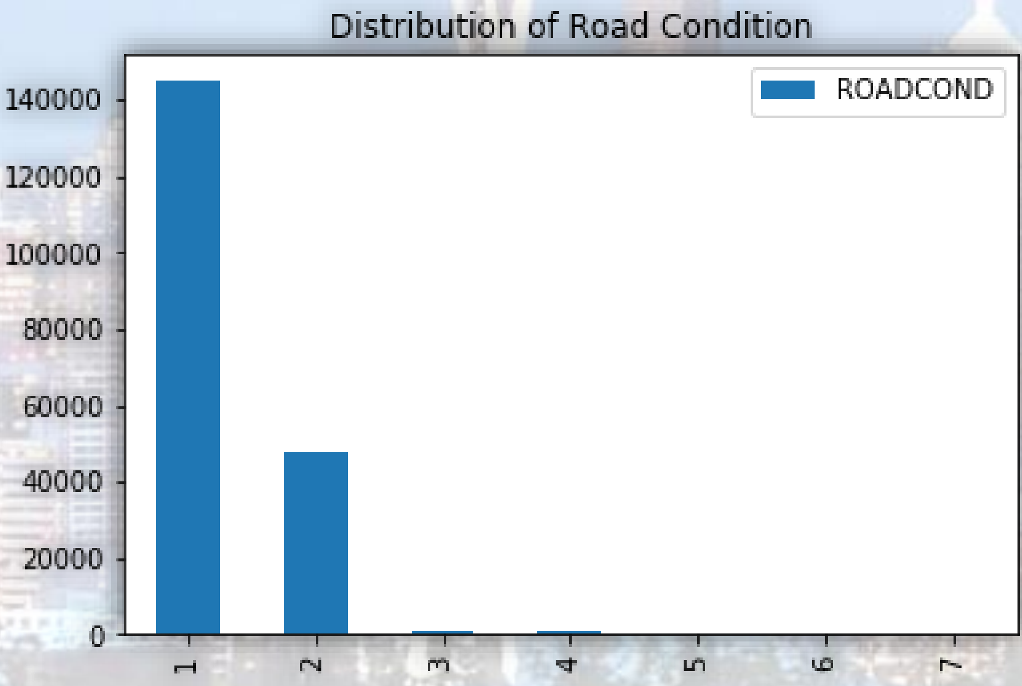
5: At intersection but not related to intersection

6: Ramp Junction



0: Good Light Condition

1: Bad Light Condition



1: Dry/Unknown/Other

2: Wet

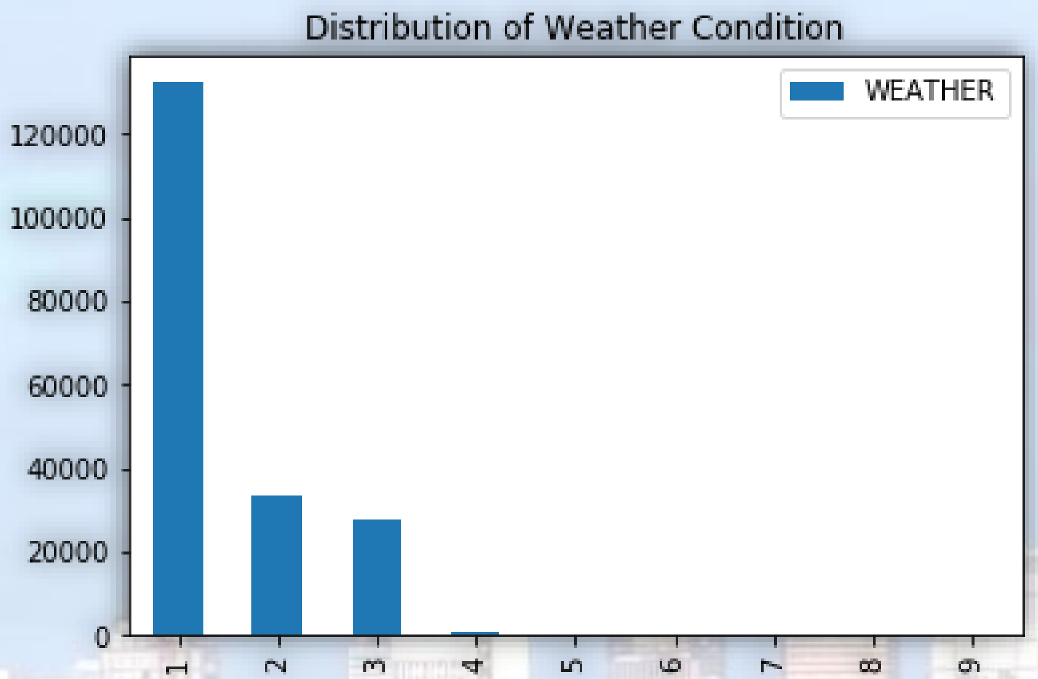
3: Ice

4: Snow/Slush

5: Standing Water

6: Sand/Mud/Dirt

7: Oil



1: Clear/Unknown/Other

2: Raining

3: Overcast

4: Snowing

5: Fog/Smog/Smoke

6: Sleet/Hail/Freezing Rain

7: Blowing Sand/Dirt

8: Severe Crosswind

9: Partly Cloudy

3.2 Map Visualization

4. Modeling & Evaluation

4.1 Machine Learning Preparation

Machine learning models here used are K-Nearest Neighbors, Decision Tree Analysis and Logistic Regression. These methods vary from each other.

Decision Tree Analysis Model use entropy to set the whole dataset to different branches based on different depths. It comes up with leaf nodes and decision nodes. KNN is a regression model based on the k-distance, which is the parameter set manually. Logistic Regression is a static model based on binary variables.

4.2 Data Normalization

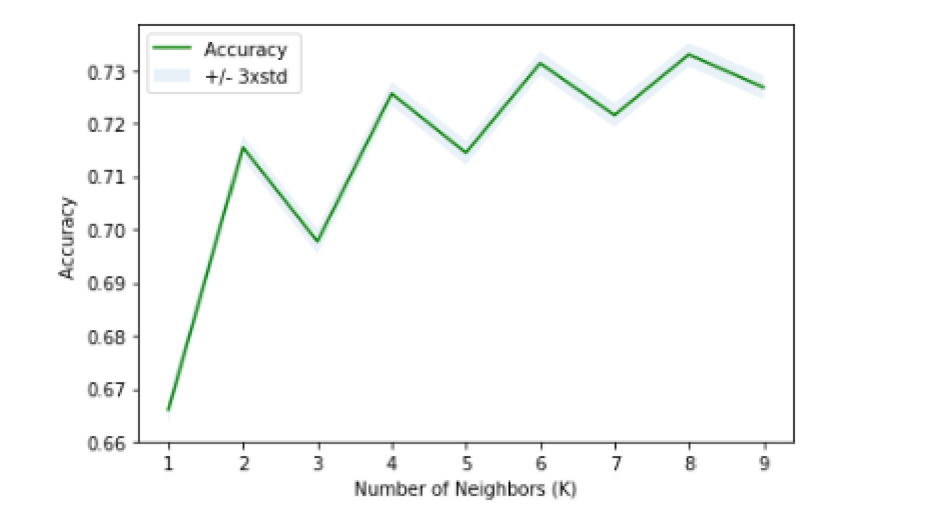
Before we do modeling and analyzing, we need to normalize data resources.

4.2 Data Normalization

```
X= preprocessing.StandardScaler().fit(X).transform(X)
X[0:5]
```

4.3 K-Nearest Neighbors

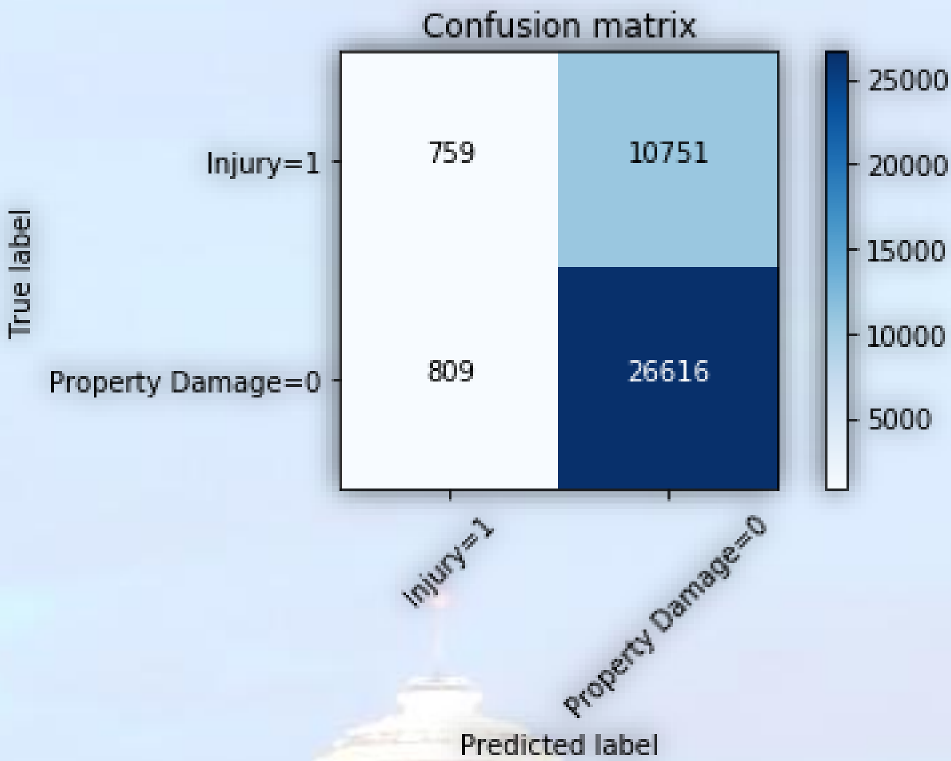
As we can see in next graph that the max of ‘Accuracy’ occurs where ‘k=8’




```
print( "Best k =", mean_acc.argmax()+1)
print("K-Nearest Neighbours Accuray: ", metrics.accuracy_score(Y_test, yhat))

    Best k = 8
    K-Nearest Neighbours Accuray:  0.7267753948889174
```

	Precision	Recall	f1-score	Support
0	0.71	0.97	0.82	27425
1	0.48	0.07	0.12	11510
Micro Avg	0.70	0.70	0.70	38935
Macro Avg	0.60	0.52	0.47	38935
Weighted Avg	0.64	0.70	0.61	38935

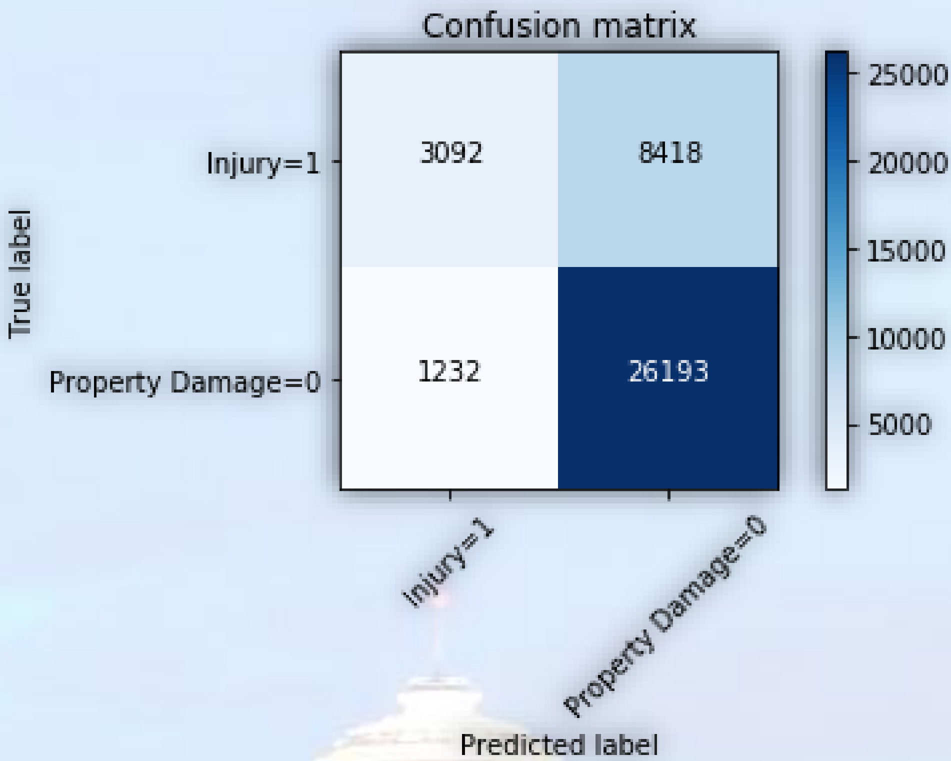


4.4 Decision Tree

In this report, the criterion classifier used is 'entropy' and the depth is 6

	Precision	Recall	f1-score	Support
0	0.96	0.76	0.84	34611
1	0.27	0.72	0.39	4324
Micro Avg	0.75	0.75	0.75	38935
Macro Avg	0.61	0.74	0.62	38935
Weighted	0.88	0.75	0.79	38935

```
Accuracy score for Decision Tree =  0.7521510209323231
```

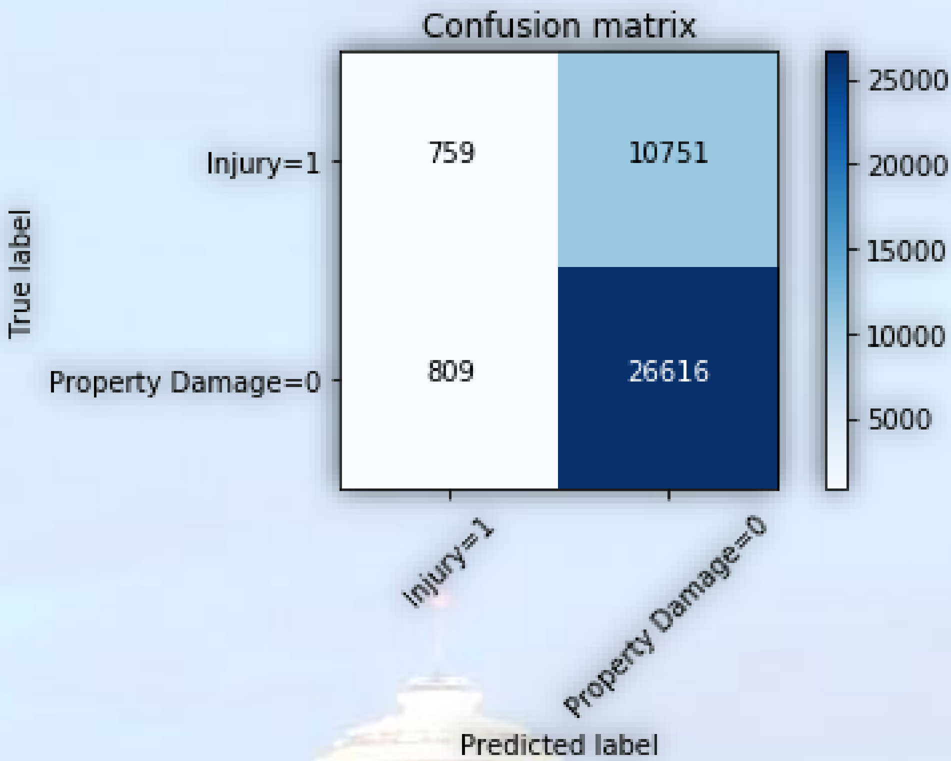


4.5 Logistic Regression

In this report, logistic regression applies ‘0.01’ as C for regularization strength and ‘liblinear’ as solver.

	Precision	Recall	f1-score	Support
0	0.71	0.97	0.82	27425
1	0.48	0.07	0.12	11510
Micro Avg	0.70	0.70	0.70	38935
Macro Avg	0.60	0.52	0.47	38935
Weighted	0.64	0.70	0.61	38935

```
0.5828257292180228
Accuracy 0.7030949017593425
```



5. Evaluation

5.1 f1-score

F1-score is a symbol of accuracy of the model that we deployed. The closer between 1 and f1-score, the more precise the model is.

As what we've mentioned above:

	f1-score
K-Nearest Neighbors	0.61
Decision Tree Decision	0.79
Logistic Regression	0.61

The f1-score we listed above are the weighted average values of 3 different models. DT decision model is the best model and has the highest f1-score.

5.2 Precision

This is an output that weighs the relevance. It is calculated by dividing true positives by true positives and false positives.

	Precision
K-Nearest Neighbors	0.64
Decision Tree Decision	0.88
Logistic Regression	0.64

As we can extract from above, DT decision has the highest precision values.

5.3 Recall

Recall gives us information about percentage of total relevant results correctly classified by our algorithm.

	Recall
K-Nearest Neighbors	0.70
Decision Tree Decision	0.75
Logistic Regression	0.70

As we can see, DT Decision Model has the highest recall values.

6. Conclusion

By comparing all 3 methods and 3 evaluation output, we can figure out that Decision Tree Decision Model fit this problem the best. By using parameters like light, weather, drivers' attention, and road condition, we can predict the possibility of severity of a car accident that may take place.