Car Accident Severity Report

(IBM Coursera Applied Data Science Capstone Project)

Introduction

According to World Health Organization (https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death), road traffic injuries were the tenth leading cause of death globally in 2010. With 1.35 million road traffic deaths globally in 2016, and millions more sustaining serious injuries and living with long-term adverse health consequences, it was not surprising that in 2016, road traffic injuries were ranked as the eighth leading cause, up by two places.

It is also anticipated that by 2030, death by road traffic injuries will be ranked as the seventh leading cause. Globally, road traffic crashes are a leading cause of death among young people, and the main cause of death among those aged 15–29 years.

To minimize such accidents, it is important to implement preventative measures.

This project aims to determine if the severity of accidents can be predicted using the following factors:

- Location (general location of the collision)
- Collision Address Type (whether alley, block or intersection)
- Day of the Incident
- Time of the Incident
- Weather Conditions
- Road Conditions
- Light Conditions

Target Audience

Drivers

This will be particularly helpful for drivers, particularly those in the 15 – 29 years age group. These young drivers may not be experienced to deal with driving under certain conditions and the outcome of this project could help them in applying safe driving techniques.

Law Enforcement

The information can be used in a road safety campaign. It could also be used by law enforcement in setting up patrols at “hotspots” based on the various factors indicated above.

Hospitals

Knowing the factors that contribute to the severity of road traffic injuries could aid in reducing preventable deaths.

Governments / Local Authorities

Authorities could re-look at infrastructure design to minimize collisions. Also, reduction in severe injuries from traffic collisions could lead to savings in medical and rehabilitation spending.

Data

The data used in this project was collected by the Seattle Police Department, recorded by Traffic Records, and provided by Coursera via this link. The time period for this data is from 2004 - 2020 and the dataset contains information on 194,673 car related accidents in the state of Seattle. There are also 37 different attributes including severity, location of the collision, collision type, date and time of the accident, the type of junction where the collision took place, weather conditions, road conditions, and light conditions.

The target attribute is severity and the Seattle Police Department records accident severity according to the following schema:

- 0: Unknown/no data
- 1: Property damage only
- 2: Minor injury collision
- 2b: Major injury collision
- 3: Fatality collision

This dataset only had entries for two levels – 1 (property damage only) and 2 (minor collision only).

The below attributes were selected as the independent variables:

- Collision Address Type (whether alley, block or intersection)
- Day of the Incident
- Time of the Incident
- Weather Conditions
- Road Conditions
- Light Conditions

More about the metadata can be found here.

Data Preprocessing

Rows with missing values were dropped.

Dataset was unbalanced. Unbalanced datasets are a special case for classification problem. An unbalanced dataset can lead to inaccurate results. If the data is biased, the results will also be biased.

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

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

Data was balanced using down-sampling

Balancing the Data using down-sampling

```
df_sev_majority = Car_Accidents[Car_Accidents['SEVERITYCODE'] == 1]
df_sev_minority = Car_Accidents[Car_Accidents['SEVERITYCODE'] == 2]
```

```
df_majority_downsampled = resample(df_sev_majority,
                                   replace=False,      # sample without replacement
                                   n_samples=58188,     # to match minority class
                                   random_state=123)   # reproducible results
```

```
# Combine minority class with downsampled majority class
df_result = pd.concat([df_majority_downsampled, df_sev_minority])
```

The Dataset is now Balanced

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

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

Create a Car_Accidents dataframe with the attributes we are interested in - SeverityCode, Location, ADDRTYPE, INCDATE, INCDTTM, WEATHER, ROADCOND, LIGHTCOND

```
Car_Accidents = df[['SEVERITYCODE','ADDRTYPE', 'INCDATE', 'INCDTTM', 'WEATHER', 'ROADCOND', 'LIGHTCOND']]
```

```
#Display the first 5 rows of the new Dataframe
Car_Accidents.head()
```

	SEVERITYCODE	ADDRTYPE	INCDATE	INCDTTM	WEATHER	ROADCOND	LIGHTCOND
0	2	Intersection	2013/03/27 00:00:00+00	3/27/2013 2:54:00 PM	Overcast	Wet	Daylight
1	1	Block	2006/12/20 00:00:00+00	12/20/2006 6:55:00 PM	Raining	Wet	Dark - Street Lights On
2	1	Block	2004/11/18 00:00:00+00	11/18/2004 10:20:00 AM	Overcast	Dry	Daylight
3	1	Block	2013/03/29 00:00:00+00	3/29/2013 9:26:00 AM	Clear	Dry	Daylight
4	2	Intersection	2004/01/28 00:00:00+00	1/28/2004 8:04:00 AM	Raining	Wet	Daylight

Methodology

Randomly split data into training and testing data using the function train_test_split. 80% for training, 20% for testing

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)


print("Number of test samples:", x_test.shape[0])
print("Number of training samples:",x_train.shape[0])
print("Test set:", x_test.shape, y_test.shape)
print("Training set:", x_train.shape, y_train.shape)
```

```
Number of test samples: 20877
Number of training samples: 83504
Test set: (20877, 6) (20877,)
Training set: (83504, 6) (83504,)
```

Three algorithms were trained on the pre-processed dataset and their accuracies compared:

K Nearest Neighbor, Decision Trees and Support Vector Machine (SVM)

- K- Nearest Neighbor
  The results, confusion matrix, classification report and accuracy are:

```
CONFUSION_MATRIX :

[[4581 3985]
 [5046 7265]]


REPORT :

              precision    recall  f1-score   support

           1       0.48      0.53      0.50      8566
           2       0.65      0.59      0.62     12311

   micro avg       0.57      0.57      0.57     20877
   macro avg       0.56      0.56      0.56     20877
weighted avg       0.58      0.57      0.57     20877

ACCURACY:


0.5674186904248695
```

- Decision Trees

  The results, confusion matrix, classification report and accuracy are:

```
CONFUSION_MATRIX:

[[5316 4666]
 [4311 6584]]
REPORT:

              precision    recall  f1-score   support

           1       0.55      0.53      0.54      9982
           2       0.59      0.60      0.59     10895

   micro avg       0.57      0.57      0.57     20877
   macro avg       0.57      0.57      0.57     20877
weighted avg       0.57      0.57      0.57     20877

ACCURACY:


0.5700052689562677
```

- Support Vector Machine

The results, confusion matrix, classification report and accuracy are:

```
CLASSIFICATION REPORT:

              precision    recall  f1-score   support

           1       0.68      0.54      0.60     12214
           2       0.50      0.64      0.56      8663

   micro avg       0.58      0.58      0.58     20877
   macro avg       0.59      0.59      0.58     20877
weighted avg       0.60      0.58      0.58     20877

CONFUSION MATRIX:

[[6537 5677]
 [3090 5573]]
ACCURACY:

0.5800641854672606
```

Jaccard Index and F1-score comparison

```
KNN Jaccard index: 0.57
KNN F1-score: 0.57
Decision Trees Jaccard index: 0.57
Decision Trees F1-score: 0.57
SVM Jaccard index: 0.58
SVM F1-score: 0.58
```

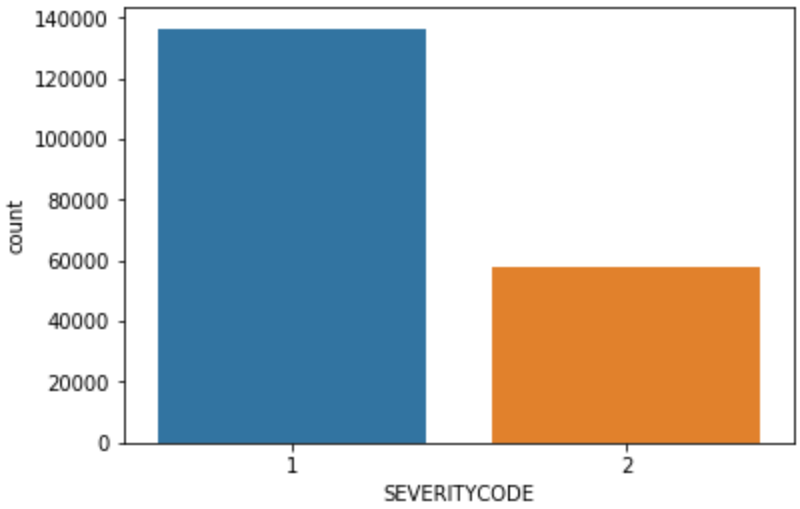
Results

There were twice as much property damage vs minor injuries:

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

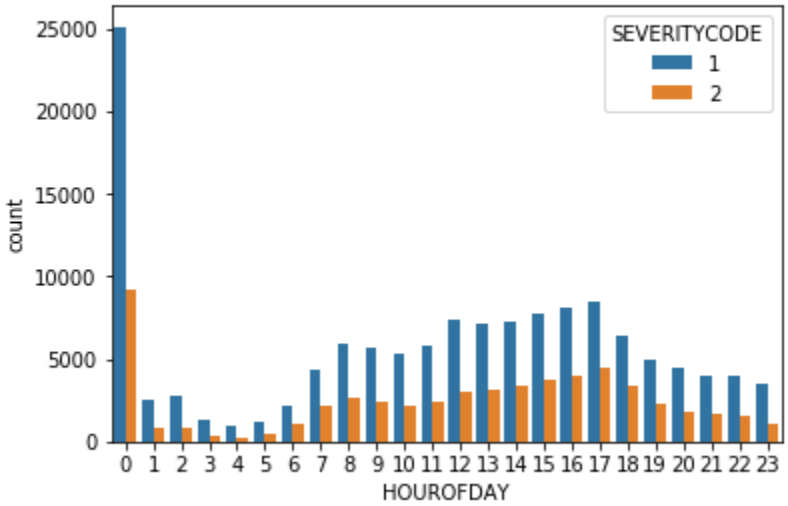
```
1    136485
2     58188
Name: SEVERITYCODE, dtype: int64
```

```
#Visually compare number of level 1 severity vs level 2 severity
ax=sns.countplot(x="SEVERITYCODE",data=Car_Accidents)
```

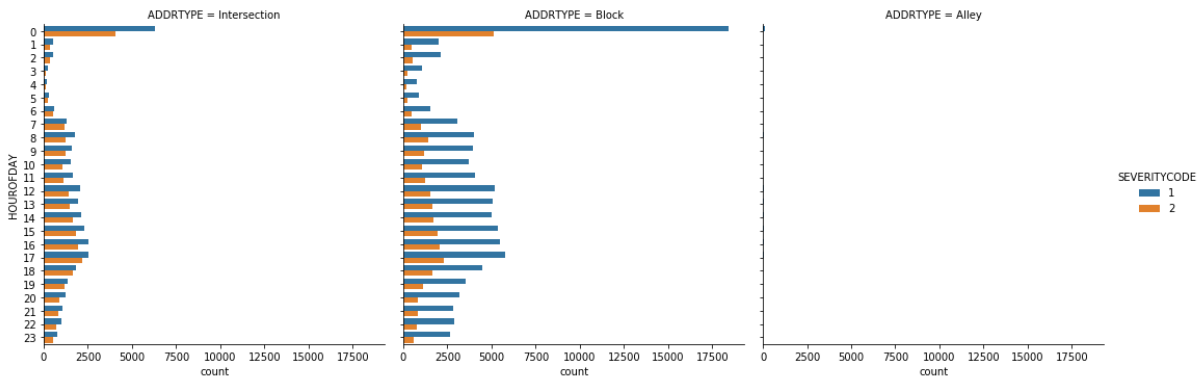


Most accidents occur around midnight. Both level 1 and 2 are at their highest at this time

```
ax=sns.countplot(x="HOUROFDAY",hue="SEVERITYCODE", data=Car_Accidents)
```

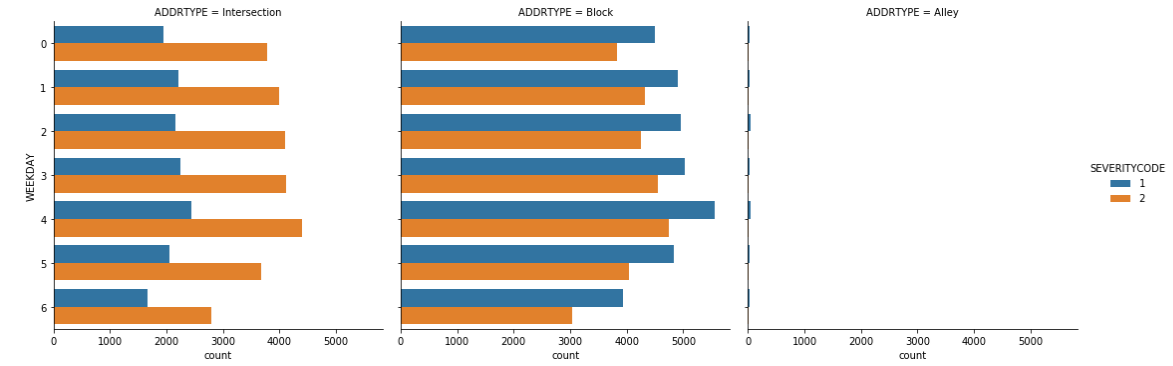


Most property damage collisions occur at Blocks at around midnight. The number of minor injuries is almost equal at intersections and blocks at midnight



The day of the week doesn't seem to impact the number of accidents, but there are more minor injuries at intersections vs property damage at intersections.

```
cp = sns.catplot(y="WEEKDAY", hue="SEVERITYCODE", col="ADDRTYPE", data=df_clean, kind="count")
```



Discussion

The algorithms used above gave accuracy scores of either 0.57 or 0.58, meaning that these models can predict the severity code of an accident with an accuracy of 57% - 58%.

Conclusion

The accuracy of the classifiers is not great, the highest being 58%. This usually means that the model is under fitted or needs to be trained in more data. Though the dataset used had a variety of attributes, most were not used due to the large amounts of missing data. This may have resulted in important correlation being overlooked.

The below attributes could therefore be useful in predicting car accident severity:

- Collision Address Type (whether alley, block or intersection)
- Day of the Incident
- Time of the Incident
- Weather Conditions
- Road Conditions

- Light Conditions