Predicting Car accident Severity

Andres Michel
IBM Data Science Professional Certification

September 14, 2020

Abstract

Vehicle collisions are a leading cause of death both in the World and in the U.S. Therefore, being able to understand the variables directly related to an increase in the odds of having a Severe car crash, over a non Severe one is of high importance. In this document we'll evaluate several ML Predictive models aiming to predict the severity of a given car crash based on several parameters from a database of more than 150k car crashes registered in Seattle, since 2004.

Contents

1 Introduction

During 2019, almost 40 thousand people lost their lives to car crashes solely in the U.S. and even though in recent years the mortality rate has been declining, it is still a leading cause of death. The odds of dying in a car crash in the US are of one in 103.

Being able to identify and predict when car crashes will occur, allows people to avoid certain roads, take extra precautions when driving under certain weather conditions and even alert authorities and manufacturers of trends and risk factors that could translate into policies and design improvements to help save lives.

1.1 Problem

Motor vehicle road accidents is a leading cause of death, and having enough data accounting for location, vehicle characteristics, road and weather conditions as well as the outcome (severity of car crash) can be used to build a model that predicts the chances of a high severity car crash to occur under certain conditions.

2 Data

We'll be using the Seattle GIS collision data, provided by the City of Seattle. All data was obtained by SPD and recorded by Traffic Records. This includes all types of collisions. Collisions will display at the intersection or mid-block of a segment. Time frame: 2004 to Present.

The data consists of 37 attributes and 194,673 rows (or accidents), with information ranging from Incident ID to Weather and Light conditions. Thus, in order to work with the data we've proceeded to clean it up, removing rows with missing data and removing redundant columns. After cleansing the data, we ended up with 182,660 rows.

3 Methodology

3.1 Data preparation

After Importing of the Collision.csv Data set it is important to first clean and prepare the data for the initial analysis. The first step is to understand the information contained in each column of the dataset. To do this, we need to fully understand the attribute description provided along with the data, as this is the document that explains the relevance that each attribute has for our predictive analysis.

Upon completion of this assessment we proceeded to remove several columns that were not important to our analysis. Such as: SPD reference codes, long string-type addresses, Collision Identifier, etc. Similarly, we performed another analysis, where we identified all the rows with missing values "NaN" or "Null" values, and then proceeded

to remove said rows. We decided to follow this approach instead of "filling" them with average or trend values, because we sufficient data, and it is better to have accurate information.

```
In [92]: df=df.drop(["ST_COLCODE","PEDCOUNT","PEDCYLCOUNT",
                     "SEVERITYCODE.1","STATUS","COLDETKEY",
                     "SEVERITYDESC","ST_COLDESC","HITPARKEDCAR",
                     "CROSSWALKKEY","INATTENTIONIND","SPEEDING",
                     "SEGLANEKEY","SDOTCOLNUM","PEDROWNOTGRNT",
                     "SDOT_COLDESC","SDOT_COLCODE","EXCEPTRSNDESC",
                     "OBJECTID","EXCEPTRSNCODE","INTKEY","INCKEY",
                     "REPORTNO"],axis=1)

         df_complete=df.dropna(subset=["X","Y","COLLISIONTYPE",
                                       "JUNCTIONTYPE","WEATHER",
                                       "ROADCOND","LIGHTCOND",
                                       "LOCATION","ADDRTYPE"])
         df_complete.describe()
```

Figure 1: Code used to clean Data

The next step prior to an EDA is to prepare the data, this activity consisted in transforming the data type of the columns depending on the values it contained. For instance Time related columns were transformed to Timestamps or Date values. Similarly, we created columns to contain the day of week for each record, as well as Hour of the day and Month. We also unified the data encoding for columns such as UNDERINFL that had a disparity of values. Finally, for the Target column, we transformed it into binary (1,0) values.

```
In [93]: #prepare timestamp and date related information
         Days=["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]
         df_complete['INCDATE'] = pd.to_datetime(df_complete['INCDATE'])
         df_complete['timestamp'] = pd.to_datetime(df_complete['INCDTTM'])
         df_complete['dayofweek'] = df_complete['timestamp'].dt.dayofweek.astype('int32')
         df_complete['Dayname']= df_complete['dayofweek'].apply(lambda x: Days[x])
         df_complete["UNDERINFL"]=df_complete['UNDERINFL'].apply(lambda x: 0 if (x=="0"or x==0 or x=="N")  else 1)
         df_complete['month'] = df_complete['timestamp'].dt.month
         df_complete['day'] = df_complete['timestamp'].dt.dayofyear
         df_complete['year'] = df_complete['timestamp'].dt.year
         df_complete['hour'] = df_complete['timestamp'].dt.hour
         df_complete['IsSevere'] = df_complete['SEVERITYCODE'].apply(lambda x: 1 if (x>1)  else 0)
         df_complete.head()
```

Figure 2: Code used to prepare Data

3.2 Exploratory Data Analysis

In this section we performed a series of analysis, mainly visuals, to understand the behaviour of the data depending on certain parameters. For instance, we did several groupBy() functions, in order to make some of the plots shown below, as this helped us to understand how each attribute had an influence in the severity of an accident. Although at this point we have not performed the encoding of the categorical variables, this was on done on purpose, as this way it is more easier to plot and understand the very nature of those categorical variables.

We proceeded to make some sense of the data by trying to identify the seasonality of the severe collisions. For instance to note if there is a pattern of collision increase during certain hours of the day, as well as day of the week:

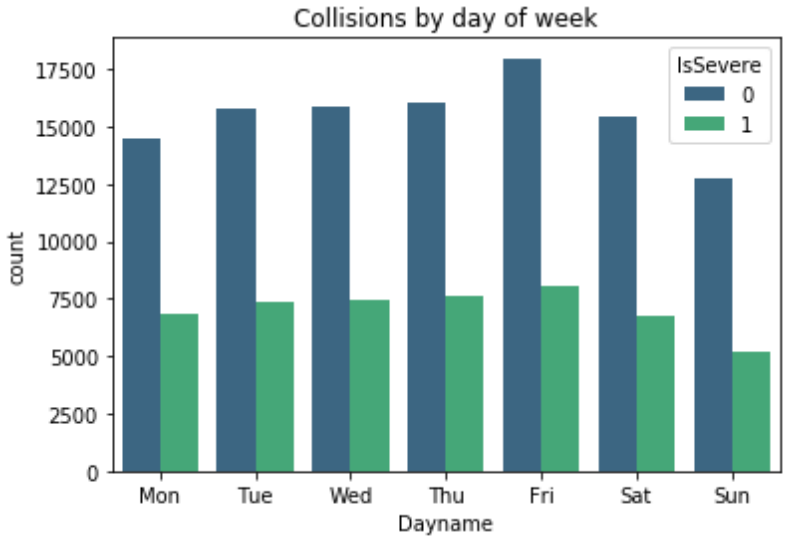

Figure 3: Collisions by day of week

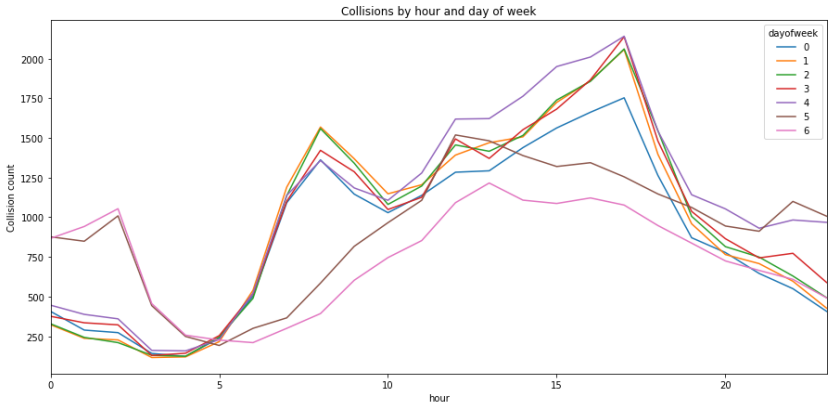

Figure 4: Collisions by hour and day of week

Given these two graphs we can identify a common trend, as more collisions are registered during Friday, and less on Sunday. Same can be said from Hour of day, where most accidents occur during midnight as well as in the afternoon.

In relationship to the weather conditions, we can note that the chances of a Collision increases with a cloudy sky:

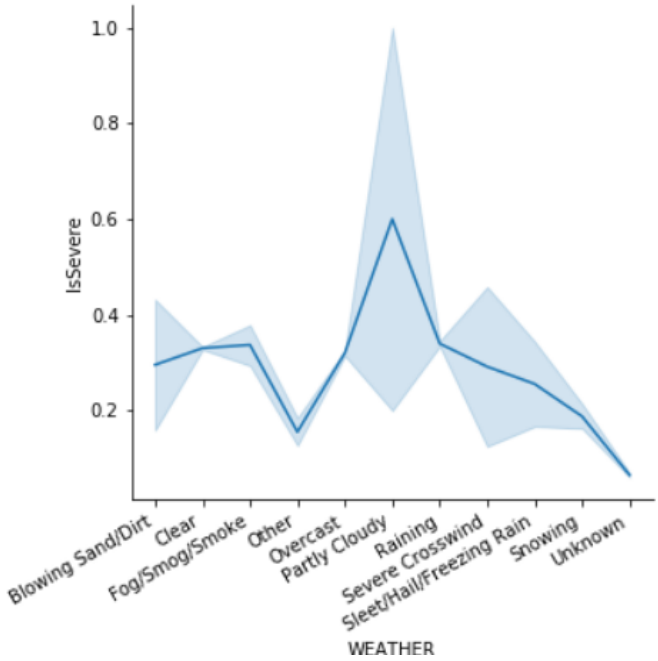

Figure 5: Severity of accidents depending on Weather

A similar assessment can be made for the condition of the road:

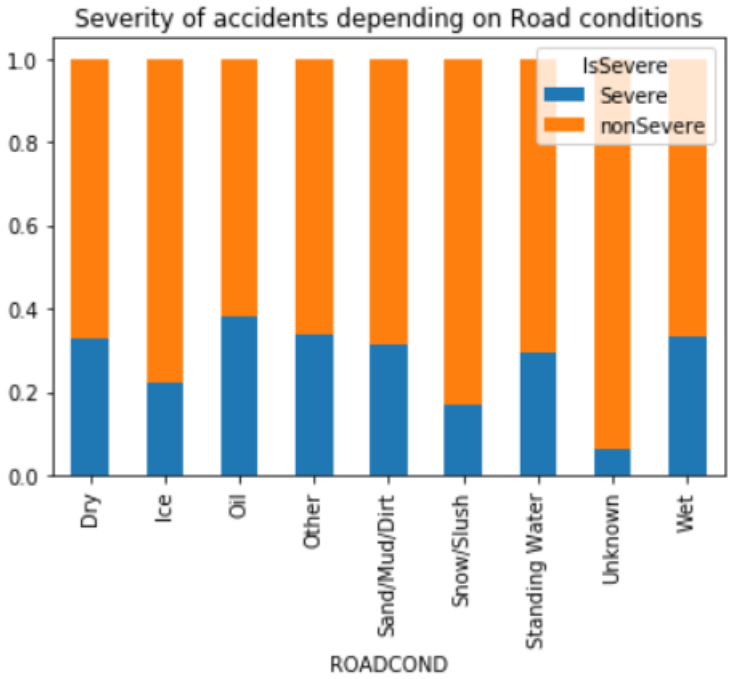

Figure 6: Severity of accidents depending on the Road conditions

Analyzing the information by Year, we notice that the total number of car accidents has been in decline over the last five years.

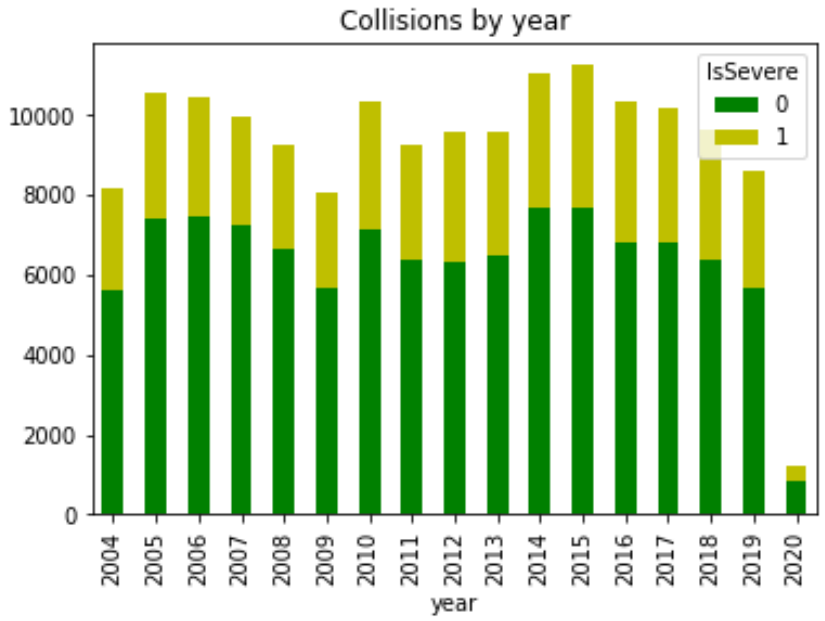

Figure 7: Total collisions by year

We also noted relevant increase of accidents during the month of October:

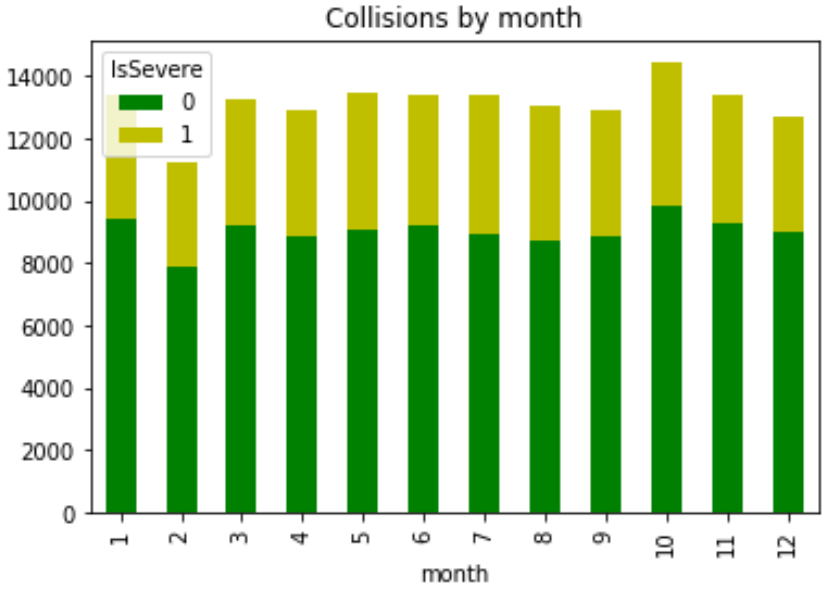

Figure 8: Total collisions by month

To conclude with the EDA analysis, we also explored the relationship between the severity of the accidents and the location (Junction type)

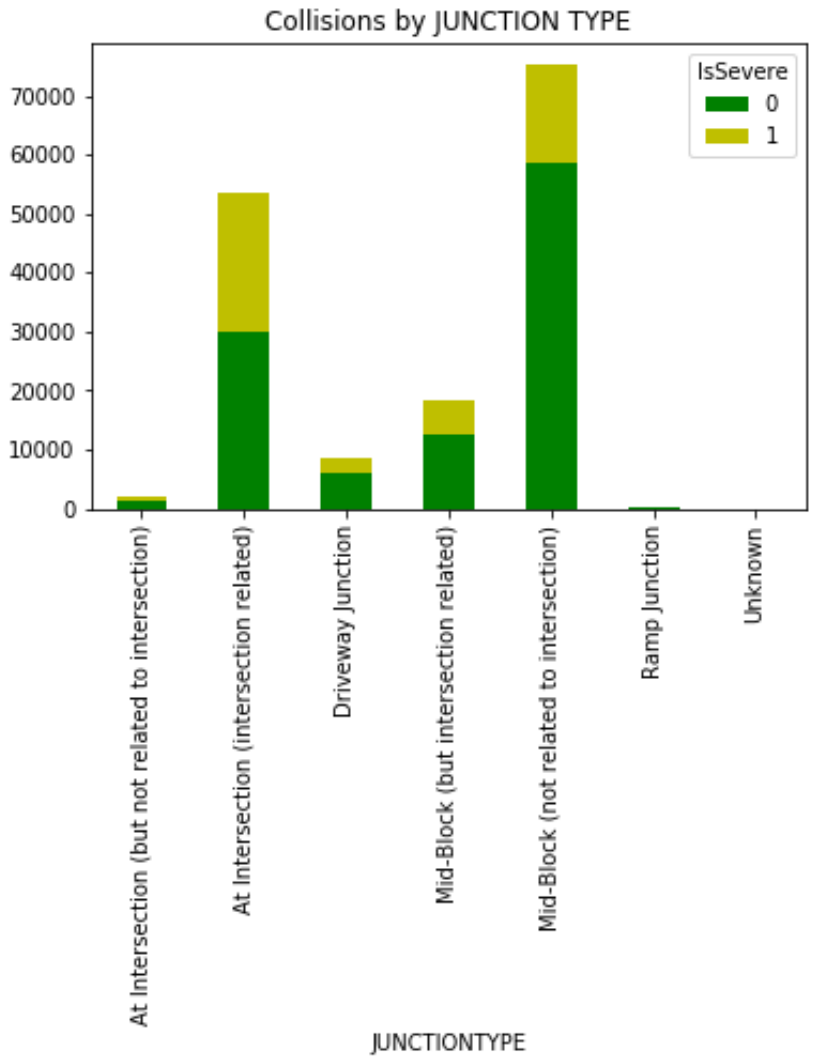

Figure 9: Total collisions by Junction type

3.3 Feature selection

After concluding with the exploratory analysis we ended up with a set of features as follows, where PPerVeh stands for the number of people per vehicle:

	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	dayofweek	month	hour	PPerVeh	IsSevere
0	0	Overcast	Wet	Daylight	2	3	14	1.000000	1
1	0	Raining	Wet	Dark - Street Lights On	2	12	18	1.000000	0
2	0	Overcast	Dry	Daylight	3	11	10	1.333333	0
3	0	Clear	Dry	Daylight	4	3	9	1.000000	0
4	0	Raining	Wet	Daylight	2	1	8	1.000000	1

Figure 10: Features

Here is a simple correlation heat map plot of the quantitative features

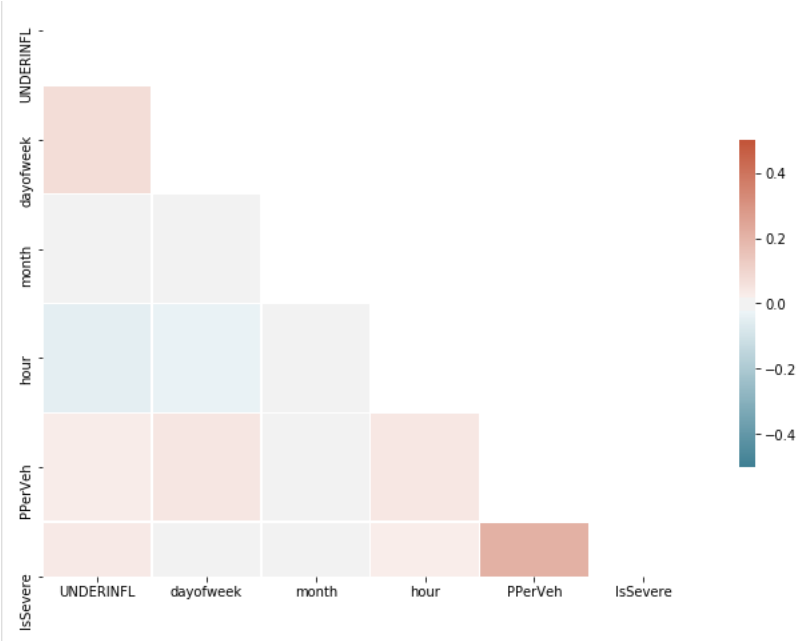

Figure 11: Correlation plot

After having identified the attributes that will be used as features, we proceeded to perform the "one-hot encoding" of all categorical variables, such as Weather and Road condition.

3.4 Test and training set selection

For the dataset splitting, we used the Sklearn train-test-split library, as it simplifies our task to randomly splitting the data into both train and test sets.

```
In [176]: #Split dataset
          from sklearn.model_selection import train_test_split

          X_train, X_test, y_train, y_test = train_test_split( X, y,
                                                               test_size=0.2,
                                                               random_state=6)
          print ('Train set:', X_train.shape,  y_train.shape)
          print ('Test set:', X_test.shape,  y_test.shape)

          Train set: (125919, 34) (125919,)
          Test set: (31480, 34) (31480,)
```

Figure 12: Data Split

3.5 Predictive Modeling

In order to predicting the chances of a Severe collision to occur, we decided to use classification algorithms. We had a particular interest in trying out the Logistic regression approach as it allowed us to actually calculate probabilities. However, we also tried out several other models

such as SVM, Decision Trees and KNN, in order to make a comparison between the outcome of these models.

4 Results

Below we have a comparison of the algorithms used to predict the Severity of a Vehicle Collision under circumstances that occur regularly in the Seattle area:

	Algorithm	Accuracy	F1-score	LogLoss
0	KNN	0.689988	0.644820	NA
1	Decision Tree	0.265019	0.694656	NA
2	SVM	0.690697	0.580981	NA
3	Logistic Regression	0.694498	0.616489	0.581436

Figure 13: Comparison table

As it can be seen, the algorithm that obtained the best performance was the Logistic Regression, though the KNN model also had a good performance. It is important to note that the Decision Tree algorithm, suffered in the accuracy score due to a high discrepancy in the rate of False positives / True positives.
Also we can see a similar result in the ROC curves:

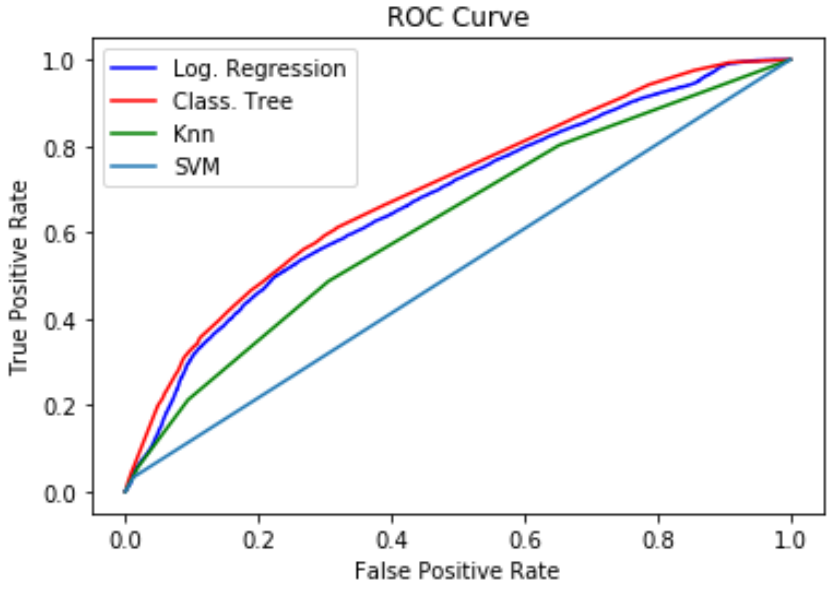

Figure 14: ROC curves

5 Discussion

Throughout the implementation and execution of several algorithms we had some issues regarding performance, this was particularly true for KNN and SVM algorithms. Upon doing some research we found that these algorithms do not perform well with large datasets (over 100k samples), this is due to the high complexity of the calculations being done, and that the temp results need to be stored in cache memory while the calculations are being performed. Therefore, this affects the viability of using this algorithms, however we were able to make some changes in the models and use, in the case of SVM some linear representations, also provided in the Scikit-learn, that employ stochastic gradient descent learning. This implementation obtained better results in terms of performance, but further tuning might've been able to improve the accuracy results, but due to time constrains we couldn't dive into such endeavour.

Similarly, in order to reduce variance bias we could've implemented the K-fold cross validation method, however due to the performance constrains already mentioned above, we would haven't been able to test all algorithms and make a proper comparison between them, that's why we opted for a more straight forward training approach.

As for the outcome, I believe the model can be further improved, and surely this will produce much better predictive capabilities. Its is also important to note that the data set itself, although it is a very impressive set of data, it requires a deep feature-engineering dive, in order to produce an even better feature / training dataset. This would require a better understanding of the codes being used, and possibly to further reduce the scope of the predictive outcome, meaning make it more specific, in order to have a better predictive model.

6 Conclusion

Working with real world data is complex problem. The mastery of a Data scientist is mostly determined by his ability to produce useful information, from a large data set, to be used by the models. Meaning that the hard work of cleaning, preparing and making sense of the data is most of what is required to build a useful, reliable predictive model.
In the case of predicting the severity of a Car crash, I believe that this is achievable to some extent, based on the dataset available, however further feature analysis and model tuning needs to be performed, as it is clear that the model will have a certain bias related to the data itself.