Car Accident Severity Prediction

IBM Data Science Capstone Project

By : Aimad GUALLAL

Introduction:

In order to reduce the frequency of car collisions in a city(we choose here a dataframe associated to Seattle city), a data sieniste have to use an algorithm which must be developed to predict the severity of an accident based on several facotrs such as : the weather, road and visibility conditions. When the conditions are bad, the model or algorithm must alert drivers to be carful. The data may also be given to specific teams like the sructural engineers and road disgners to predcit how danger the roads are before constructing them.

The target audience of the project is local government, police, rescue groups, road design teams, car insurance companies and last but not the least the drivers themselves. The model and its results are going to provide some key insights for the target audience to make impactful decisions in reducing the number of accidents and injuries in their localities.

Data description:

Firstly, we will use the dataframe found with the course material, data collision in Seattle City. Data provided by the Seattle Department of Transportation (SDOT) on vehicle accidents along with its severity. The dataset consists of 38 columns having different kinds of data like, collision severity, road conditions, number of people involved, location of collision, weather etc.

You will find the information of dataset in the link below, all the attribute information.

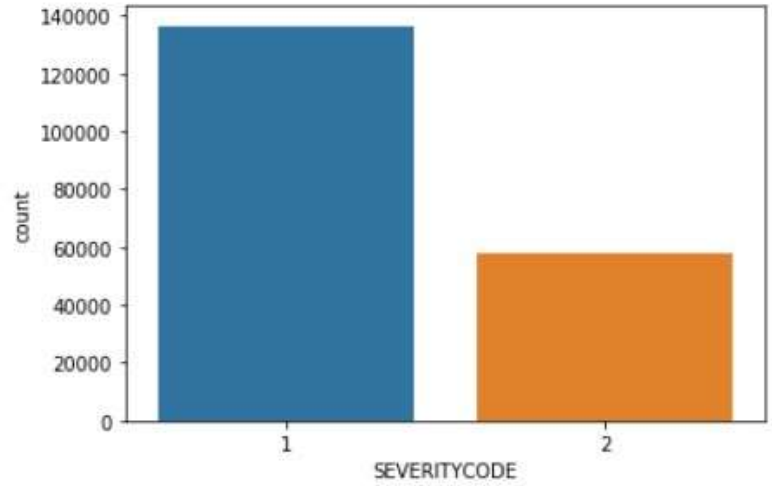
https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf

Our target will be the severity code, it takes two values 1 or 2, it's the measure of the severity of an accident.

1: refers to Property Damage Only Collision.
2: refers to Injury Collision.

```
import seaborn as sns
sns.countplot(df['SEVERITYCODE'],data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20f6020aa48>
```



The data frame is an original one, so we have to extract our dataset. To doing so, there are many columns that we won't use. Besides the severity code we will keep WEATHER, which describes the weather at the time of crash, ROADCOND, which describes the condition of the road at the time of crash, LIGHTCOND, which describes the light conditions at the time of crash.ADDRTYPE, for collision address type. JUNCTIONTYPE,Category of junction at which collision took place. There is also the speeding which is an important factor in the severity of accident, we attend to use it but the values of its column are inappropriate (we have Y and Nan values), so we will not use it. We use only what we've mentioned before.

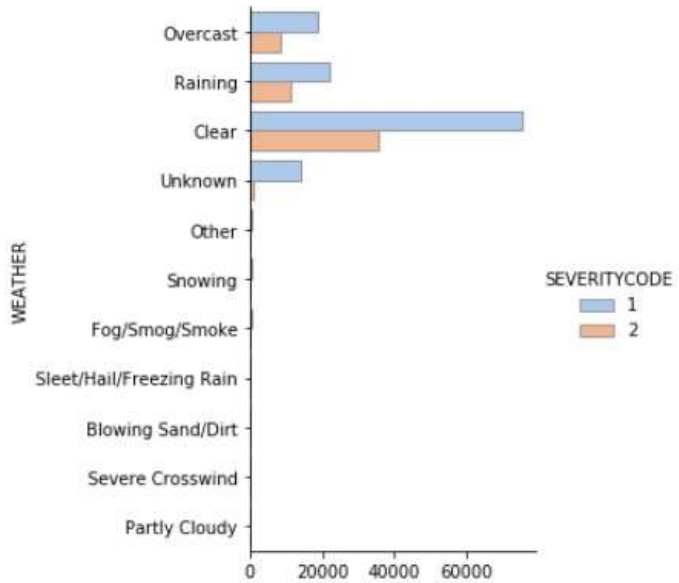
```
df_new=df[["ADDRTYPE","JUNCTIONTYPE","WEATHER","ROADCOND","LIGHTCOND","SEVERITYCODE"]]
df_new
```

	ADDRTYPE	JUNCTIONTYPE	WEATHER	ROADCOND	LIGHTCOND	SEVERITYCODE
0	Intersection	At Intersection (intersection related)	Overcast	Wet	Daylight	2
1	Block	Mid-Block (not related to intersection)	Raining	Wet	Dark - Street Lights On	1
2	Block	Mid-Block (not related to intersection)	Overcast	Dry	Daylight	1
3	Block	Mid-Block (not related to intersection)	Clear	Dry	Daylight	1
4	Intersection	At Intersection (intersection related)	Raining	Wet	Daylight	2
...	...	...	...	...	...	...
194668	Block	Mid-Block (not related to intersection)	Clear	Dry	Daylight	2
194669	Block	Mid-Block (not related to intersection)	Raining	Wet	Daylight	1
194670	Intersection	At Intersection (intersection related)	Clear	Dry	Daylight	2
194671	Intersection	At Intersection (intersection related)	Clear	Dry	Dusk	2
194672	Block	Mid-Block (not related to intersection)	Clear	Wet	Daylight	1

Let's see relation between one of the features and our target. Let's take the weather for example.

```
import numpy as np
sns.catplot(y="WEATHER", hue="SEVERITYCODE", kind="count",
            palette="pastel", edgecolor=".6",
            data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x20f659fda48>
```



Methodology:

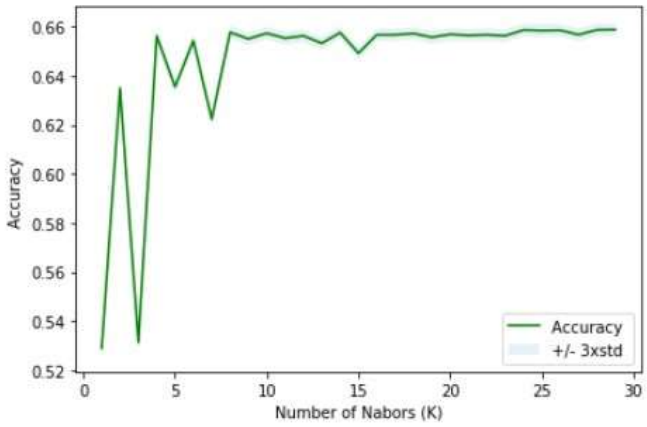
We constate that some attributes of our features have values like Unkown, and Nan values. So we have to drop them. After that we will transform all the features to numerical values since they are categorical ones.

	ADDRTYPE	JUNCTIONTYPE	WEATHER	ROADCOND	LIGHTCOND	SEVERITYCODE
0	2	1	4	7	5	2
1	1	4	6	7	2	1
2	1	4	4	0	5	1
3	1	4	1	0	5	1
4	2	1	6	7	5	2

Now, our data must be standarized to start building our model. The problem refers to a classification one. So we have to find the best classifer which can predict is the accident causes injury or just proprety damage. The classification methods to be used are :

- K Nearest Neighbor (KNN)
- Decision Tree
- Logistic Regression
- Support Victor Machine (SVM)

For implementing the Machine learning Predictive Modeling, I have used Github as a repository and Anaconda Navigator for running Jupyter Notebook to preprocess data and build Machine Learning models. Regarding coding, I have used Python and its popular packages such as Pandas, NumPy and Sklearn. First, we split our data to training and testing data. At the end of building the model, we use the evaluation metrics to find the best method in terms of accuracy. But first for the K Nearest Neighbor, we must find the best K.




```
The best accuracy was with 0.6587036816394615 with k= 29
```

Now, we can apply our classification methods and perform them by two evaluation metrics, specifically: Jaccard index, F1-score, and Log Loss. We define Jaccard as the size of the intersection divided by the size of the union of two label sets. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (which represents perfect precision and recall) and its worst at 0. It is a good way to show that a classifier has a good value for both recall and precision. Logarithmic loss (also known as Log loss) measures the performance of a classifier where the predicted output is a probability value between 0 and 1.

Results:

After performing our four algorithms, you find the results of our ML model in the table below:

Algorithm	Jaccard	F1-Score	Log Loss
KNN	0.6587	0.5729	-
Decision Tree	0.6652	0.5315	-
Logistic Regression	0.6650	0.5320	0.6222
SVM	0.6652	0.5315	-

We notice that the values of metrics, Jaccard for all algorithms, F1-Score for Decision Tree and Logistic Regression and SVM, are nearly the same. We can

conclude that all the algorithms have 0.66 for the Jaccard index. But the F1-Score of KNN is higher than others.

Discussion:

What I can say about our Machine Learning model is that we have in the majority of cases: accidents that causes only property damage and not an injury. All the four predictive algorithms confirm these results.

Now, as we have the higher values of F-Score associated to K Nearest Neighbor. We can assure that is more precise than other algorithms. So, it may be the best classifier to our Capstone project.
But since the metrics of the four algorithms were generally near to each other. We can use them all. And we will have approximatively the same predictions.

Conclusion:

Besides speeding and driving under the effect of alcohool,weather, visibility, or road conditions are the major uncontrollable factors that can be prevented by revealing hidden patterns in the data and announcing warning to the local government, police and drivers on the targeted roads.

After our ML modeling, we can conclude that particular conditions like weather, road, and light conditions have a somewhat impact on whether or not travel could result in property damage (severitycode = 1) or injury (severitycode = 2).

In the end, there's no perfect solution, only a solution that is good enough for the intended purpose.