# IBM Capstone Project — Car accident severity

# **Business Understanding**

A number of factors contribute to the risk of collisions, in the majority of the cases is related to driver factors, road and weather conditions. Traffic collisions often result in injury, disability, death, property damage, financial costs as well as terrible traffic jams.In an effort to avoid and reduce the frequency of these type of accidents, I will build a model to predict the severity of an accident given the weather and the road conditions. This way we would be able to bring awareness to the drivers and warn people about the possibility of getting into a car accident and its severity if it happens. This way people would drive more carefully or even change the travel if able to.The big question to be answered is: Knowing the weather and road conditions, how severe would be the accident if it happens?

# **Data Understanding**

The raw data we will use is provided by the SDOT Traffic Management Division and contains data of all types of collisions that happened in Seattle city from 2004 to May/2020.The data contains 194,673 samples and have 37 features that covers the weather and road conditions, collision factors and fatality.Let’s have a look on the data and understand better how to find the answer to this problem.

## **Do we have missing values? How many?**

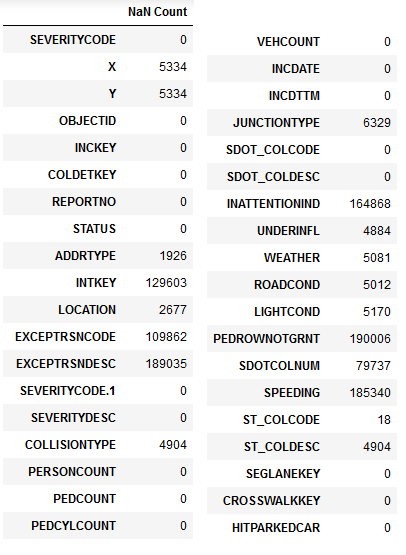


Figure 1 — Nan Value Count

There are missing values on part of the data, some features have over 40% of missing data for that I’ll not consider them to my model. Removing the irrelevant data attributes away, the variables I will use to classify the severity of the accidents are:

* COLLISIONTYPE: Collision type
* WEATHER: Weather conditions during the time of the collision.
* ROADCOND: The condition of the road during the collision.
* LIGHTCOND: The light conditions during the collision.
* UNDERINFL: Whether or not a driver involved was under the influence of drugs or alcohol.

These features contains missing values but its below 3% of the total amount of samples.

## **What is our target variable?**

Our target variable SEVERITYCODE that corresponds to the severity of the collision:

1: Property Damage only collision which is the same as Not injured collis ion

2: Injury collision By looking to the target variable I know it’s a binary classification problem.

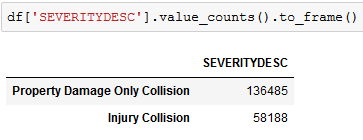


Figure 2 — Collision occurrence by Severity type

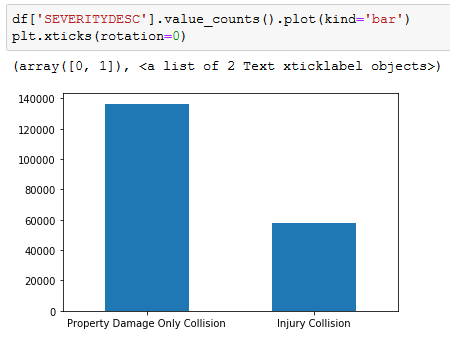


Figure 3 — Collision occurrence by Severity type Plot

## **Annual amount of traffic incidents in Seattle**

We notice there is a considerably high amount of incidents only discrepancy is from 2020 as it was recorded incidents that occured till May/2020 not a whole year like the others. We can also infer from the plots that no injury collisions are always more likely to happen.

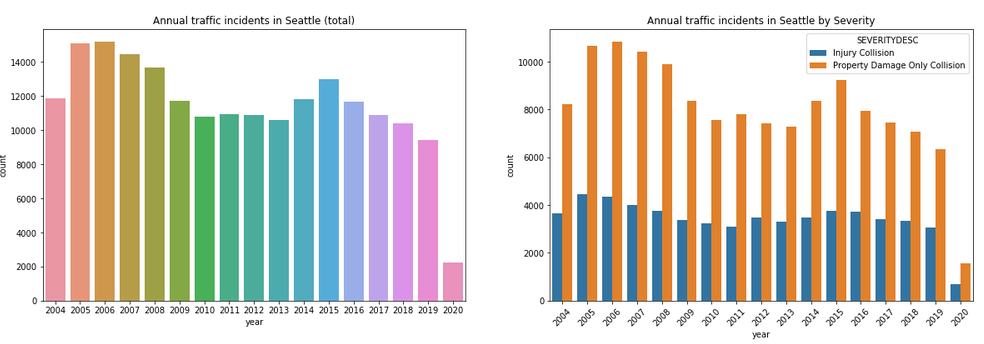


Figure 4 — Annual traffic incident occurrence in Seattle

## **Collision types**

There is a considerable difference on the collision occurences according to collision types. Being the most recurrent accidents with parked cars,angles and rear ended.

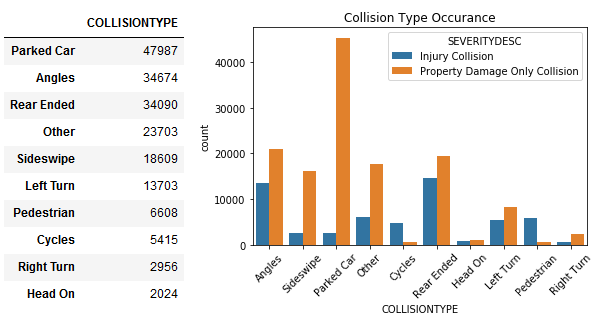


Figure 5 — Collision Type by Severity

## **Weather condition**

Considering Seattle weather conditions, we notice most incidents happened in a Clear weather. That could be because drivers are less careful when there is no harsh weather condition. It would be interesting to check the correlation between WEATHER and INATTENTIONIND(whether or not collision was due to inattention), but there are too many missing values, 85% of the data is missing.

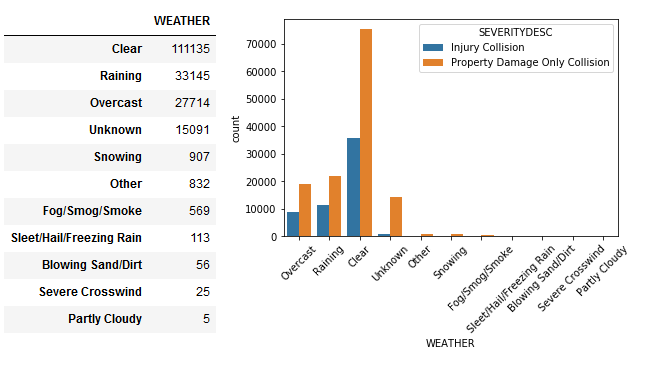


Figure 6 — Weather by Severity

## **The condition of the road during the collision**

More occurrences in normal road conditions.

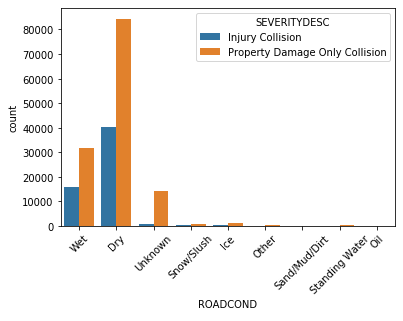


Figure 7 — Road Condition by Severity

## **The light conditions during the collision**

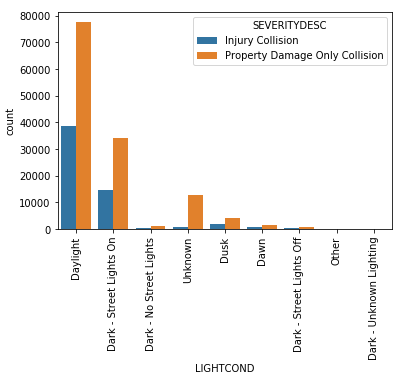


Figure 8 — Light Conditions by Severity

## **Driver under influence of drugs or alcohol**

In most incidents drivers were not under any influence.

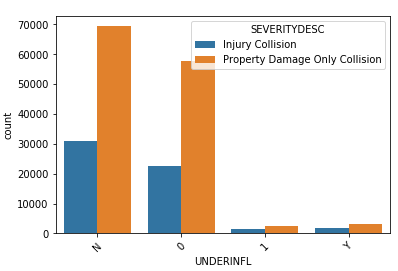


Figure 9 — Driver under influence by Severity

Each feature have a different weight of influence on the severity of the collision. Overall, all of them are consistently infering that no-injury accidents in normal driving conditions are more recurrent.

We will use COLLISIONTYPE, WEATHER, ROADCOND, LIGHTCOND and UNDERINFL as attributes to classify SEVERITYCODE. For that we will need to prepare this features so it is suitable for a binary classification model. I’ll use some popular machine learning algorithms (SVM, Logistic Regression, Naive Bayes and KNN) build up models to analyze their performance and predict the collision severity.

# **Methodology**

# **1. Data preparation and cleaning**

Data cleaning procedure to make the dataset readable and suitable to the machine learning algorithms.

## **Dropping all the irrelevant variables and attributes**

Out of the 37 attributes, I will not consider the features with over 40% of missing data, other unclear and irrelevant/noisy variables to our problem. I’ll use COLLISIONTYPE, WEATHER, ROADCOND, LIGHTCOND and UNDERINFL as attributes to classify SEVERITYCODE.

## **Dealing with missing values**

As my chosen attributes have about 3% of missing data I’ll just drop them. I’ll still have a considerable amount of data.

## **Treating the categorical variables**

In my case, all attributes are categorical. In this step, I will apply label encoding technique for all of them.

## **Train/Test split and data normalization**

Now that I treated all my variables I’ll separate my independent variables to dataset A and dependent variable ‘SEVERITYCODE’ to dataset B. After, I’ll use this data to randomly pick samples and split in this ratio:

* 70% to train my model
* 30% to test my model Following the split I’ll normalize all data to make sure my features are on a similar scale.

# **2. Classification: Modeling and Evaluation**

The prepared dataset will be used to model 3 classification models.

* Logistic Regression: Classifies data by estimating the probability of classes.
* Decision Tree: Classifies by breaking down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
* KNN: Classifies unseen data through the majority of its ‘neighbours’. In this case we already know K=2 (2 classes of SEVERITY CODES). After obtaining each model’s predictions we will evaluate their accuracy, precison, f1-score, log-loss and compare and discuss the results.

# **3. Discussion and Conclusion**

After obtaining the results and evaluating them, in this section I will brief any observations noted based on the results. Finally, will conclude the results of this analysis.

# **Model Evaluation using Test set**

For all three models Jaccard score, which measures accuracy is above 70%. With a 5% better accuracy the highest accuracy model is the Decision Tree Classifier. The same model also presents the best F1\_score and Recall(True positive rate).

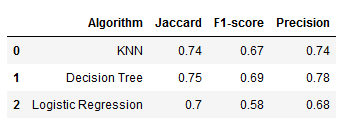


Figure 10 — Evaluation

From the Confusion Matrices we can see also exactly the amount of samples that were classified rightfully and wrongfully. Its noticeable the variation jump that happens when comparing false positives and true positives while true negatives and false negatives are quite stable.

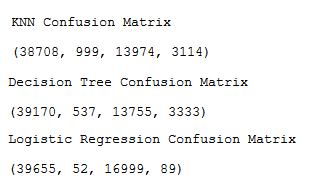


Figure 11 — Confusion matrix results

**Discussion and Conclusion**

In this analysis we evaluated the performance of 3 machine learning algorithms on the Seattle Collision dateset to predict the severity of an accident knowing the weather and road conditions.  
The three models performed very similarly, but Decision Tree stood out with a difference of 1% compared to KNN and 5% compared to Logistic Regression when we evaluate with the model’s accuracy.

Although I hand picked just 5 features out of 37, it showed to be a reasonable choice to find the answer we were searching for. But there is always room for improvement! In a future analysis I could investigate further the remaining features and the features chosen for this analysis to make sophisticated new features and extract information that could possibly contribute to my model. Also, I could think further the classification behavior specifically with true/false positives and understand better what my model is missing. Much more could be done, as use hyper parametrize to set my model better.