Robin Kjär
25.10.2020
github.com/RobinKj/Coursera_Capstone

PREDICTING TRAFFIC ACCIDENT SEVERITY IN SEATTLE, WA

IBM Data Science Professional Certificate
Capstone Project

Table of Content

1. Introduction

1.1 Scope and Background

Road accidents are a growing issue worldwide, with millions of deaths and injuries every year all over the world. Road traffic injuries result in 1.35 million deaths on roadways each year globally. This effects people across all age groups with more than half of the victims being pedestrians, motorcyclists and cyclists. [1] This issue weighs even more heavily within the U.S. where people rely more on their cars, because of the limited public transport options.

Seattle is the **25th biggest city in the U.S.** and the biggest city in Washington State with almost 3,8 million people living in its metropolitan area. Its inhabitants like to sometimes call it rain city, as the Pacific Northwest is not particularly famous for its fair weather.

2019 the Seattle Times cite a Texas A&M report saying Seattle ranks **seventh** in the whole United States **in time stuck in traffic**. The report estimates that there were 167,384,000 hours of delay resulting from traffic in 2017. As the saying goes - time is money – meaning these hours also result in a total annual cost of $3.1 billion or $1,408 for each single commuter.[2]

The Washington State Department of Transportation (WSDOT) counted 6,252,554 registered vehicles in Washington State in 2015, with 59.7 billion vehicle miles being travelled within the state. There annual collision report emphasizes the incredible numbers of **a crash every 4.5 minutes and a fatal crash every 16 hours**.[3]

Sadly, these numbers don't seem to have been decreasing since.

1.2 The Objective

Which brings us to our objective: What if we could actively reduce the number of traffic accidents or their severity if we better understand the underlining factors that affect the severity of accidents.

We want to identify **which variables have the biggest impact on the severity of accidents**. Through analysis of the broad range of variables, from road and weather conditions to location data, we can train models to **accurately predict accident severity**.

The insight on which specific conditions can determine the severity can help improve vehicular and road safety, as well as potentially leading to early warning systems for traffic.

These findings gained through the research on Seattle data can then be used to study other cities within the U.S. and worldwide.

[1] https://www.cdc.gov/injury/features/global-road-safety/
[2] https://www.seattletimes.com/seattle-news/transportation/seattle-area-traffic-congestion-is-among-the-worst-in-the-country-study-shows/
[3] https://www.wsdot.wa.gov/mapsdata/crash/pdf/2015_Annual_Collision_Summary.pdf

1.3 Interest and Stakeholders

Learning more about what causes severe crashes should interest everyone, as it leads to making our roads safer.

Governments could use the findings to improve road safeness by counteracting the conditions affecting the severity of accidents. Installing early warning systems could inform drivers of dangerous conditions via radio, roadside signs or similar, leading to more careful driving. The police, ambulances and hospitals could be on alert near places where specific conditions are met.

Navigation apps leading through traffic could be enhanced by reacting to the underlying conditions found in our data analysis to avoid traffic and potential accidents. Even car manufactures and insurance companies could benefit by furthering their autonomous driving assistants and tailoring their products to the new information.

In general, the findings could save lives, time and money and therefore should concern everyone.

2. The Data

2.1 The Source

The data we used to study car accident severity in Seattle was provided through the course materials on Coursera[4], but can also be found on the open data platform of the city of Seattle[5].

It lists almost 200,000 samples of collisions between 2004 and today. The data comes from traffic records from the Seattle Department of Transportation (SDOT), containing all vehicular accidents between cars, pedestrians, cyclists and so forth.

It features the target variable *SEVERITYCODE* and 37 attributes describing all kinds of characteristics. From location data, type of surrounding of the accident, in which form the vehicles collided, to information about the state of the driver causing the accident and lighting, weather and road conditions.

The target variable has values between 1 and 2. The higher value is equivalent to a severe accident including injuries, while the value 0 equals an accident with only property damages.

We must take note of only 58,188 of the 194,673 samples being listed with a severity of 2, leading to the dataset being unbalanced.

[4] https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
[5] https://data-seattlecitygis.opendata.arcgis.com/datasets/

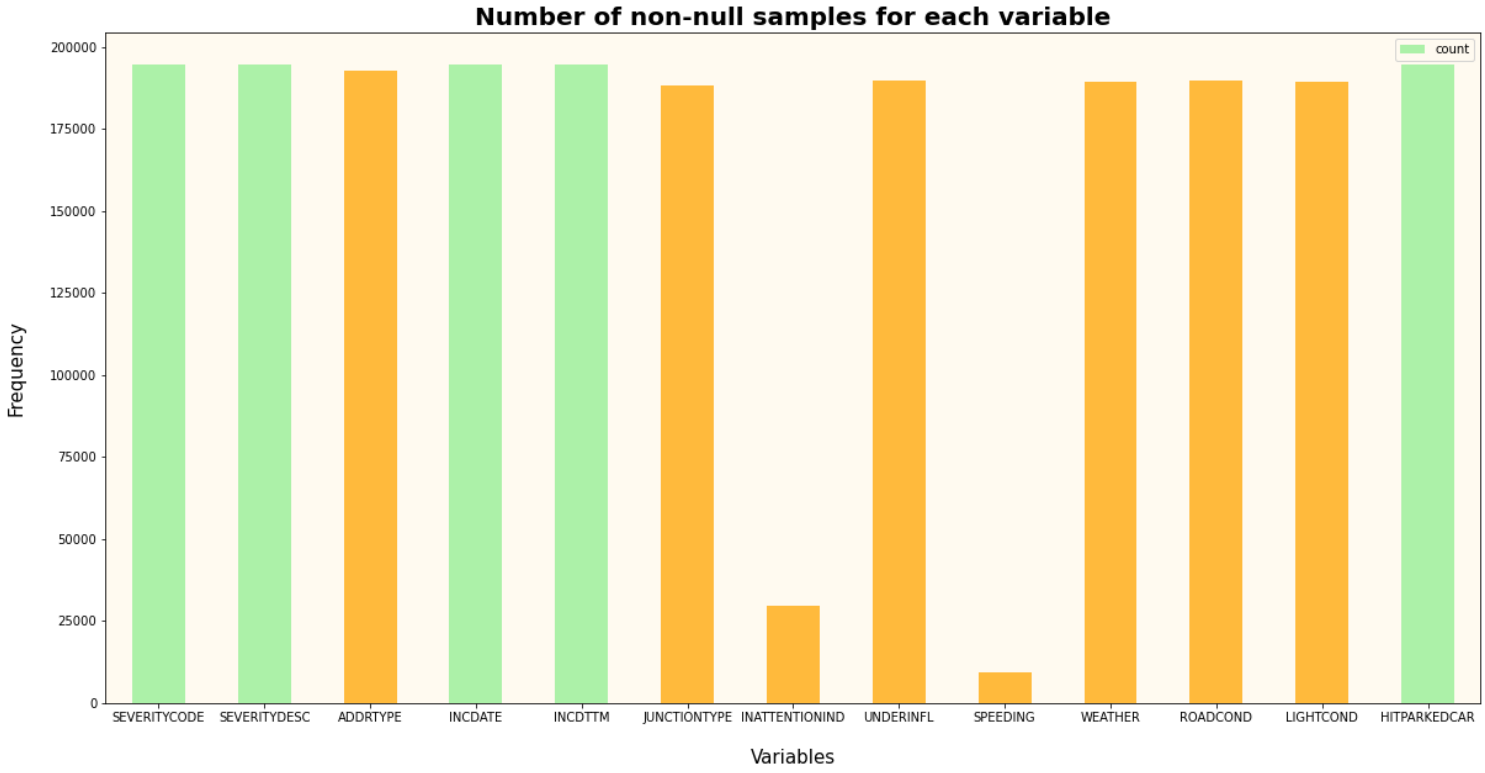
2.2 Data Cleaning

Our data has several problems regarding missing data as well as including a lot of unnecessary information for our specific purposes. We will replace missing values with the value of the highest frequency and eliminate some columns. Only then is the data suited for the machine learning algorithms.

The goal of our analysis and machine learning model is the prediction of the accident severity. This column is our target variable and for the sake of the machine learning algorithms we will transform the values from 1/2 to 0/1.

We also must react to the problem of an interchanging format in some columns. Those columns include values with the format Y/N, as well as values with the format 1/0. They contain the exact same information, but we must align the information to the same format. For easier computational practice, we chose the 0/1 format.

The biggest problem within the dataset is the big amount of missing data. The following graphic shows which of the columns we decided to work with are affected.



The green colored bars have zero missing data, while the orange bars are missing information. The variables about if the driver was speeding or paying attention are lacking a considerable amount of values, but we will optimize all variables listed in orange.

We replace missing values with the most common one, achieving a dataset with zero information left out. We also reduced the number of different categories in the categorical variables, by combing some categories to one. This makes it easier to plot information and get a clean overview of the dataset.

2.3 Feature Selection

As the dataset includes too much different variables, we decided on removing 25 columns that were deemed unnecessary for our work. This allows us to focus on the important factors in our models and graphical analysis. To utilize our planned machine learning algorithms, we need to reduce the number of variables again and do a feature selection. The features used in our machine learning models are therefore marked in orange.

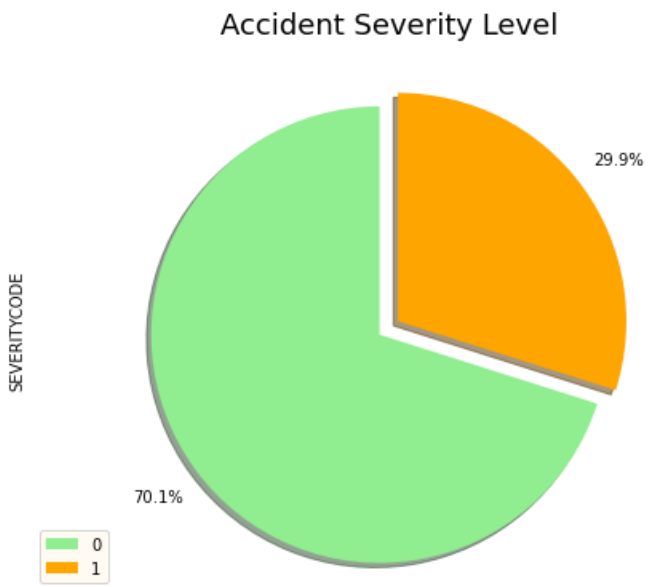
Variables	Description
SEVERITYCODE	The severity of the accident (1/0)
SEVERITYDESC	The severity of the accident (as a category)
ADDRTYPE	At what kind of location the accident took place (as a category)
INCDATE	Date of the accident
INCDTTM	Date of the accident
JUNCTIONTYPE	At what kind of junction the accident took place (as a category)
INATTENTIONIND	Whether or not the driver was paying attention (1/0)
UNDERINFL	Whether or not the driver was under the influence (1/0)
SPEEDING	Whether or not the driver was speeding (1/0)
WEATHER	Weather conditions during the accident
ROADCOND	Road conditions during the accident
LIGHTCOND	Light conditions during the accident
HITPARKEDCAR	Whether or not the accident involved a parked car (1/0)

3. Exploratory Data Analysis

We now have these 11 features left that can potentially have differently strong influence on the severity of each accident. Our goal in this chapter is to better understand each feature and how it's different values can affect the result. We will therefore use several plotting methods to visualize how each variable impacts the severity of the accident.

We also added four different temporal variables to analyze how the accidents' severity was influenced by the year, month, day of the week and hour of the day it happened on.

Before we begin with each single feature, let's get a graphical impression of the target variable:



As clearly visible, a big majority of all cases (70,1 %) are accidents with only property damage. We will react to the unbalanced dataset in the next step to prepare for them for our machine learning models. If we don't balance the dataset, the disproportionate distribution of samples will lead to biased models.

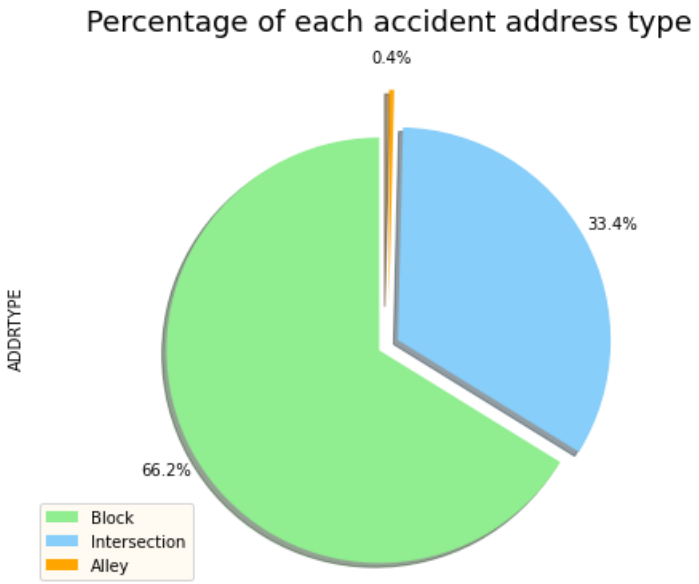
We don't need to change anything for this chapter yet, but keep in mind that the samples are unbalanced when judging the graphs.

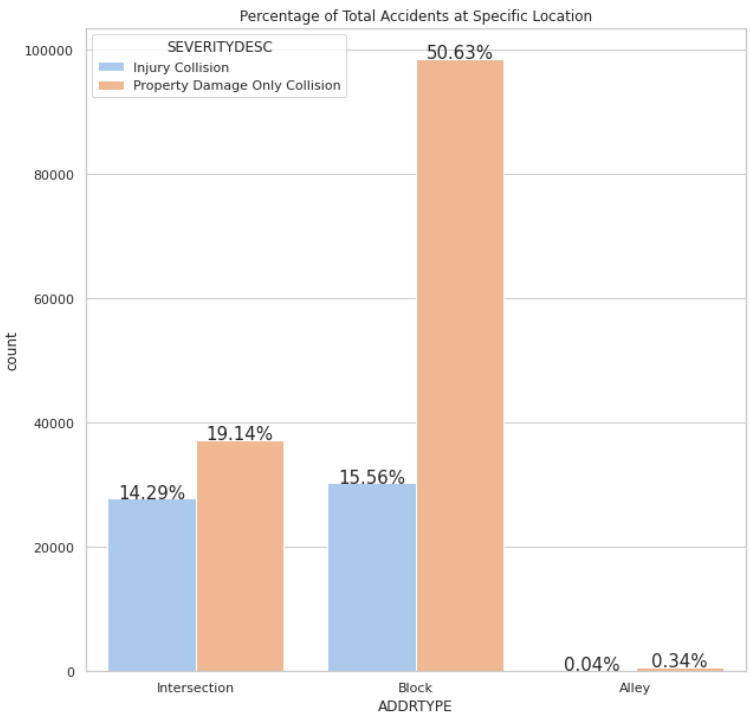
3.1 Categorical Variables

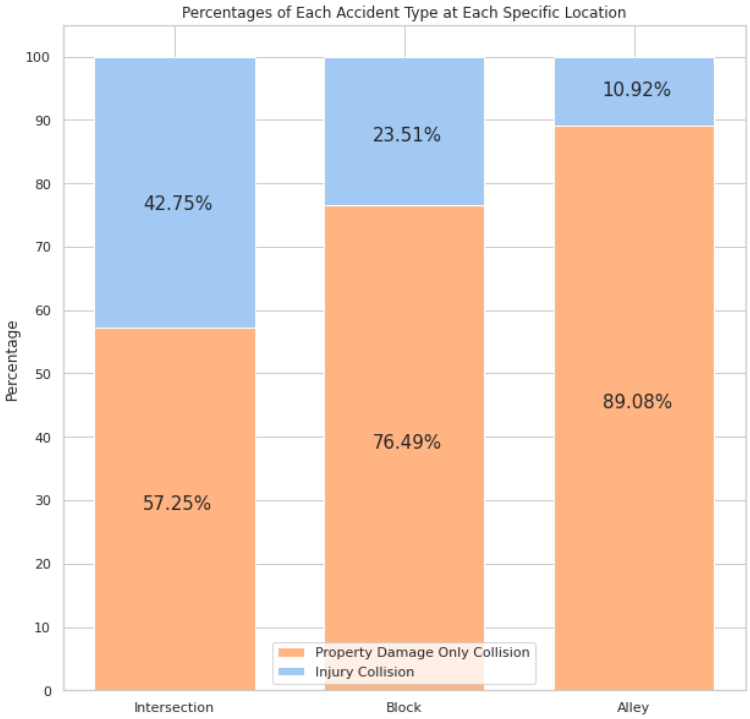
Let's now look at each of the five categorical features we have in our dataset. For each feature we will first visualize what the share of the total set each category has. The next graph shows how that individual share of the pie breaks up into severe and non-severe samples. With our last graph we will then see how each category type splits percentage wise between severe and non-severe cases.

3.1.1 Address Type

We will start with the analysis of how each different address type impacts the severity of an accident:



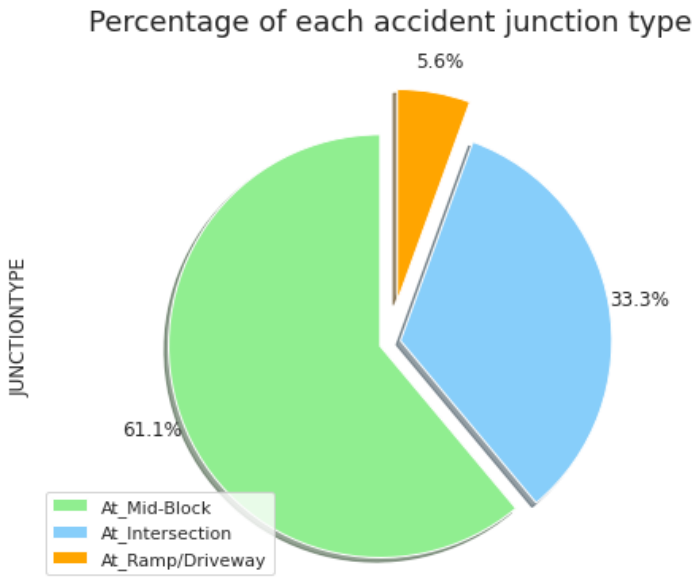


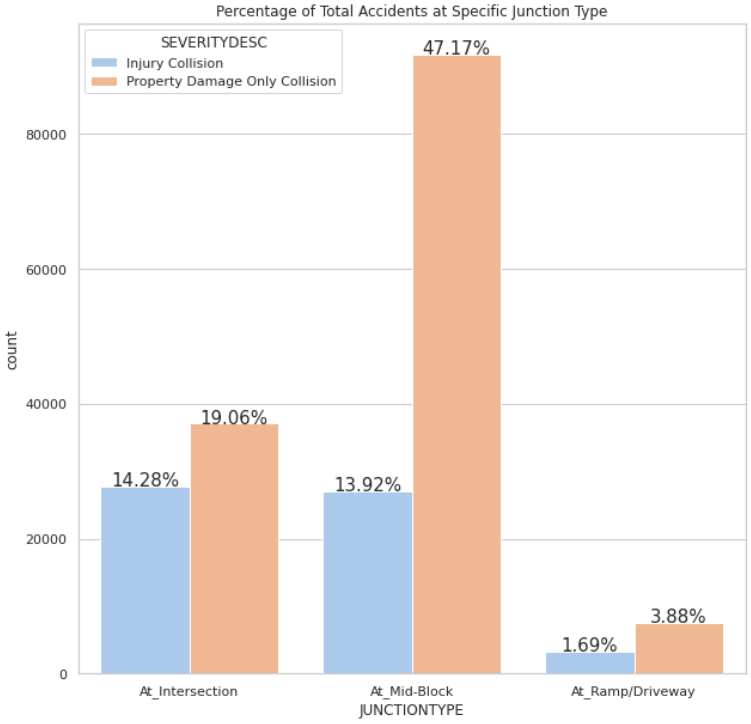


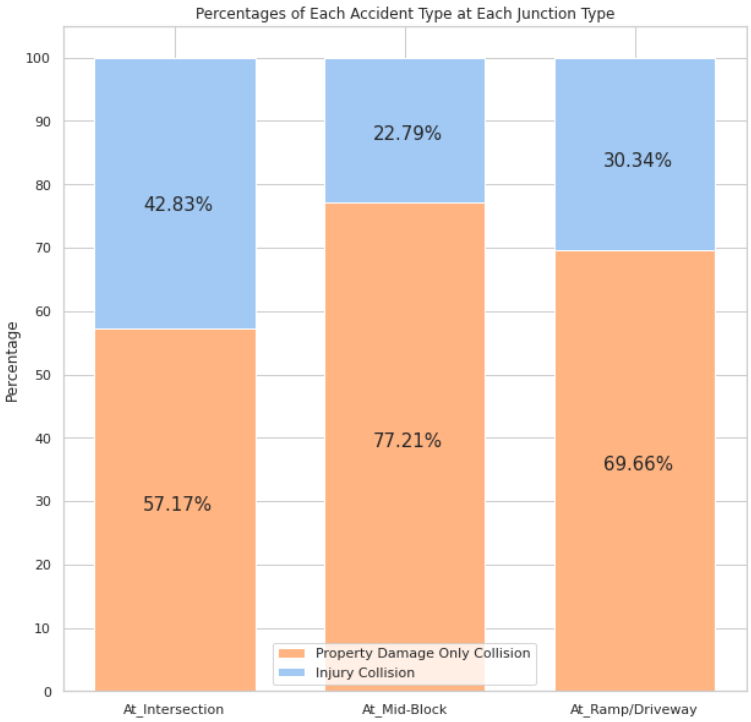
As visible from the first graph, around two thirds of all accidents happen between blocks, while about a third of all registered accidents happen at an intersection. Only a very small amount of all registered accidents happen in alleys, with the category also having the smallest percentage of the cases being severe (~10 %). This is understandable, as the average speed in alleys is very low and therefore doesn't create a big amount of accidents or severe accidents.

Almost the same amount of severe accidents come from intersections as from blocks, which gets even more emphasized by the high percentage of severe accidents in intersections visible in graph 3. That can be explained by cars crossing each other at intersections, which then leads to more dangerous situations and potential crashes.

3.1.2 Junction Type

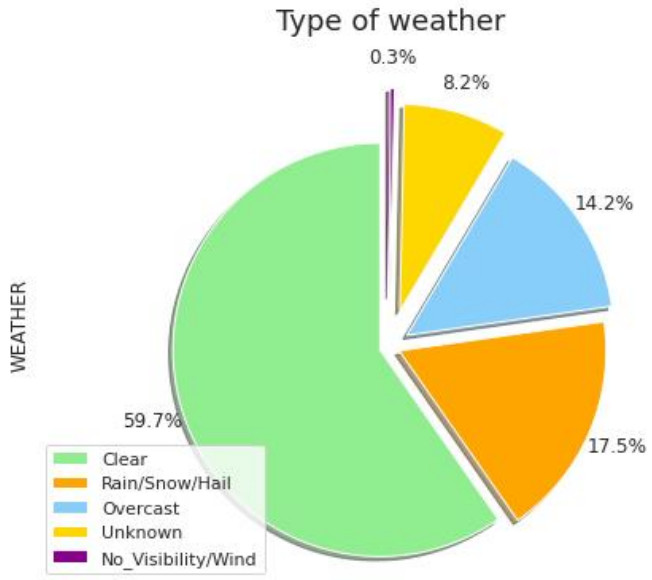


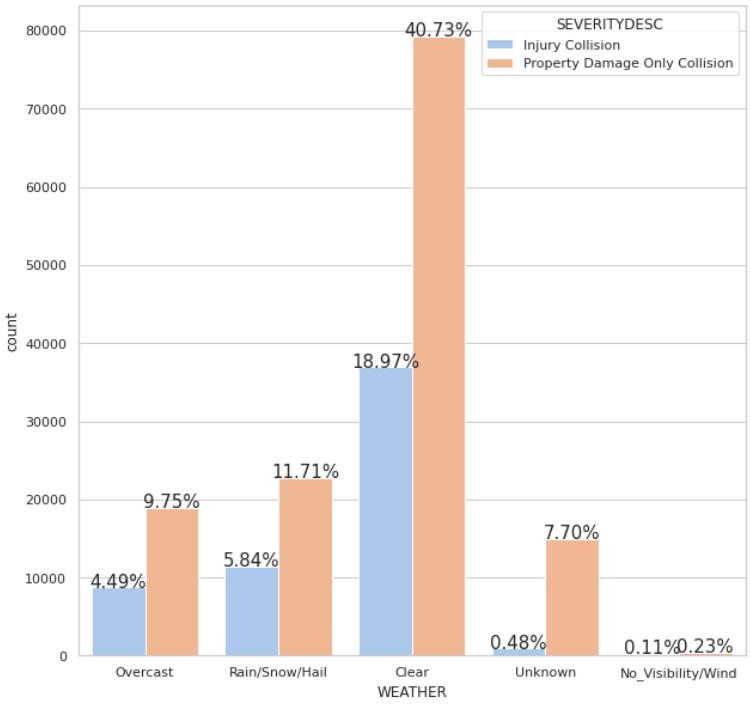


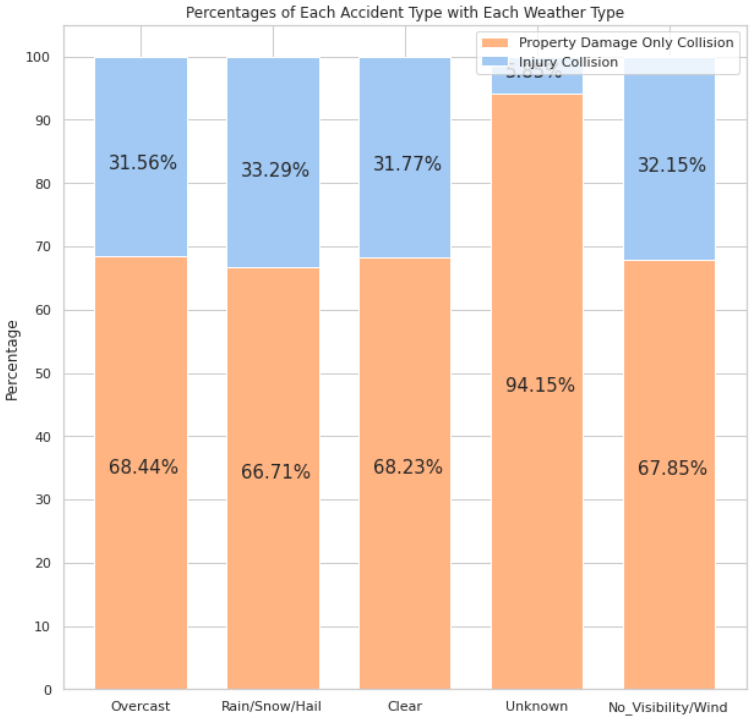


The shares from graph 1 are very similar to the graph 1 from the address types, with the "ramp/driveway" category having a bigger share in this category. Intersections are still the most dangerous for severe accidents, second place being ramp and driveways. The percentage of severe accidents at blocks seems to be nearly the same as with the address type category.

3.1.3 Weather Type



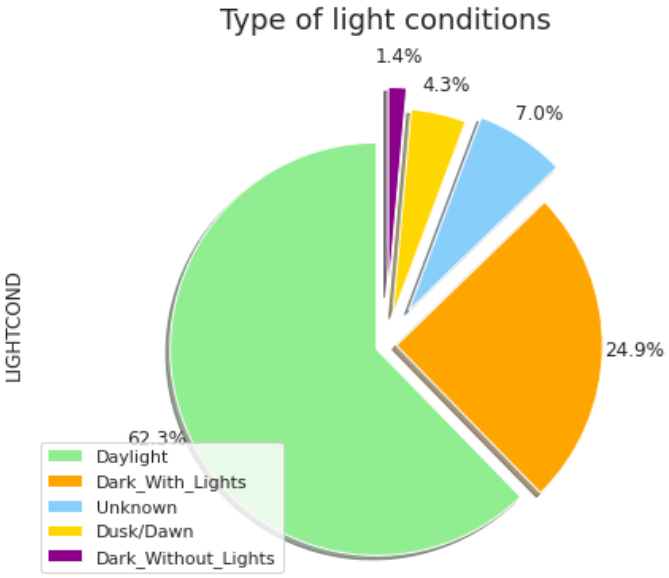


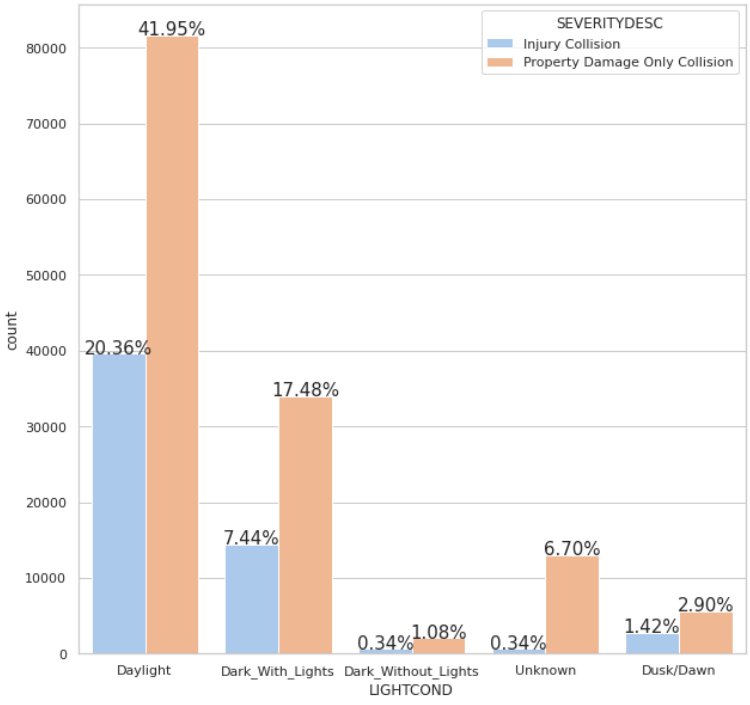


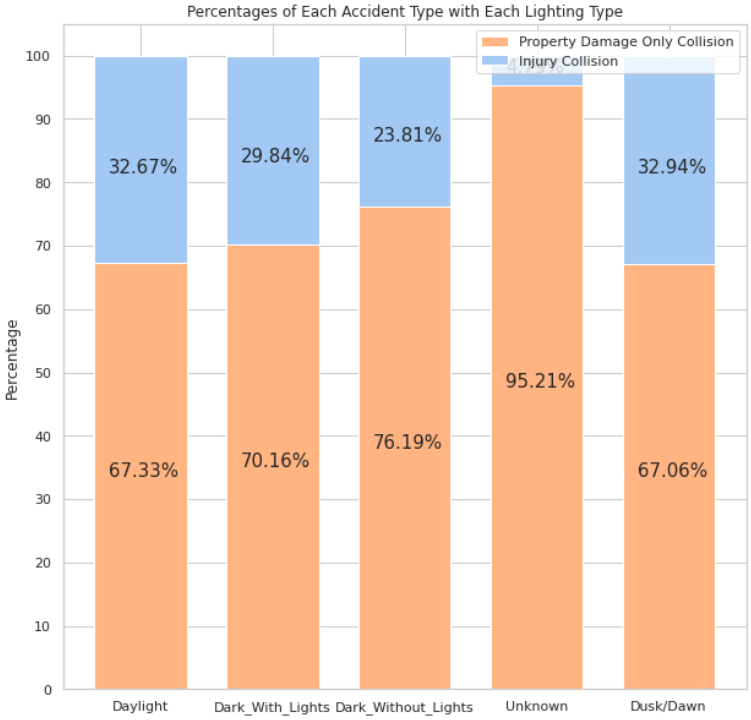
We want to now analyze the different weather categories and their impact on accident severity. More than half of all cases had clear weather, with the rest being split between rainy, snowy or cloudy conditions. Some cases have unknown weather conditions, while there also is a minute amount of cases with strong wind and no visibility as their weather type.

Looking at the spread percentages, graph 3 shows that all categories have the same percentage of severe accidents, except the category “Unknown”.

3.1.3 Light Conditions



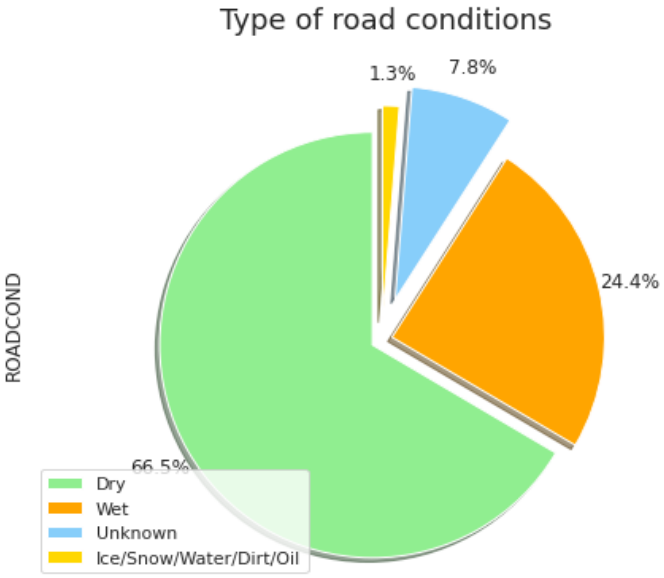


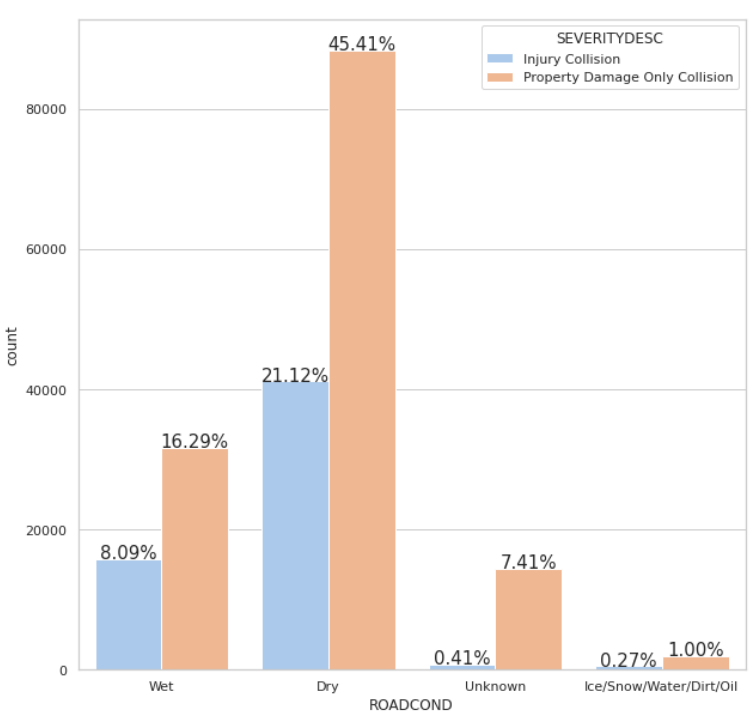


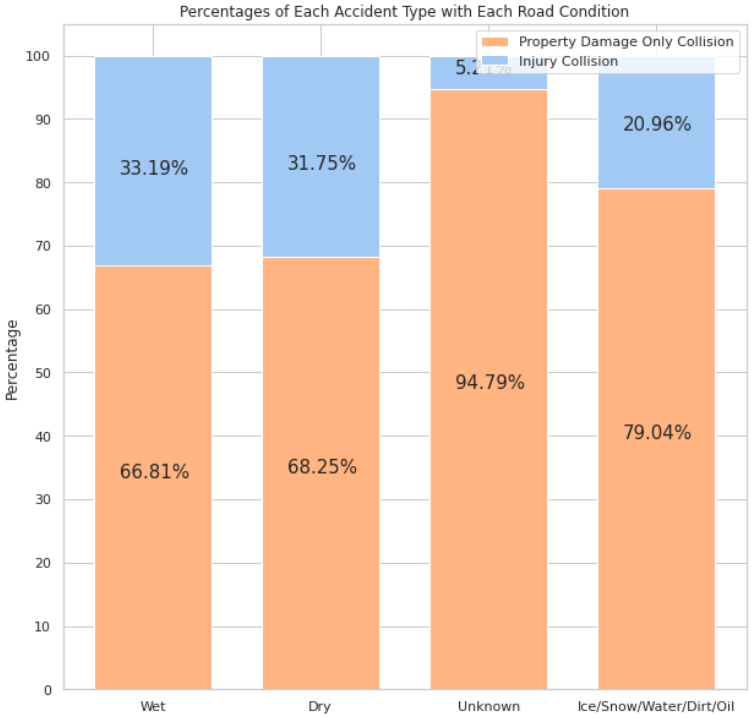
Looking at the different light conditions, we see that more than two thirds of all accidents happen during daylight, with another quarter happening during nighttime with streetlights around. The remaining ~12 percent are split between unknown light conditions, dusk and dawn as well as night without lights.

Looking at the severity percentages of graph 3 we see that daylight, dusk and dawn, as well as dark with lights are similarly dangerous. The most dangerous lighting condition is dusk and dawn, while the least dangerous is sadly the unknown category.

3.1.3 Road Conditions





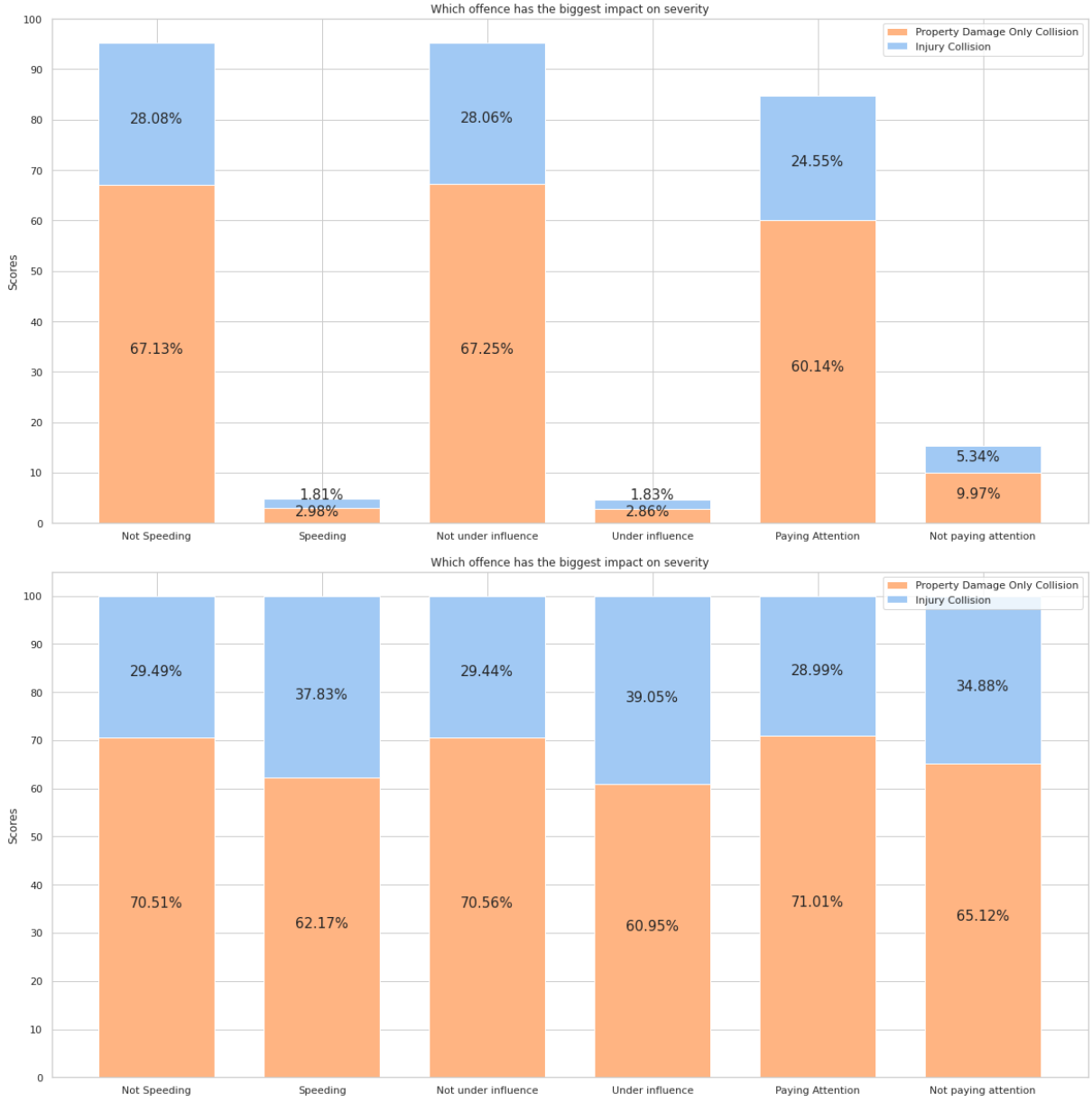


The final category to analyze is road condition types. We see that two third of all accidents registered happened with dry road conditions, with another quarter of all accidents happen when it was wet. The other types get a combined 10 percent. Wet and dry road conditions seem to have similar impact on severity, with wet conditions being the most dangerous.

After analyzing all categorical values, the most impactful feature seems to be the junction type variable, closely followed by the address type variable. The different categories in those two features have a big impact on severity, with the other three being less impactful.

3.2 True/False Variables

We created two graphs to analyze how each true-false variable impacts the accident severity.



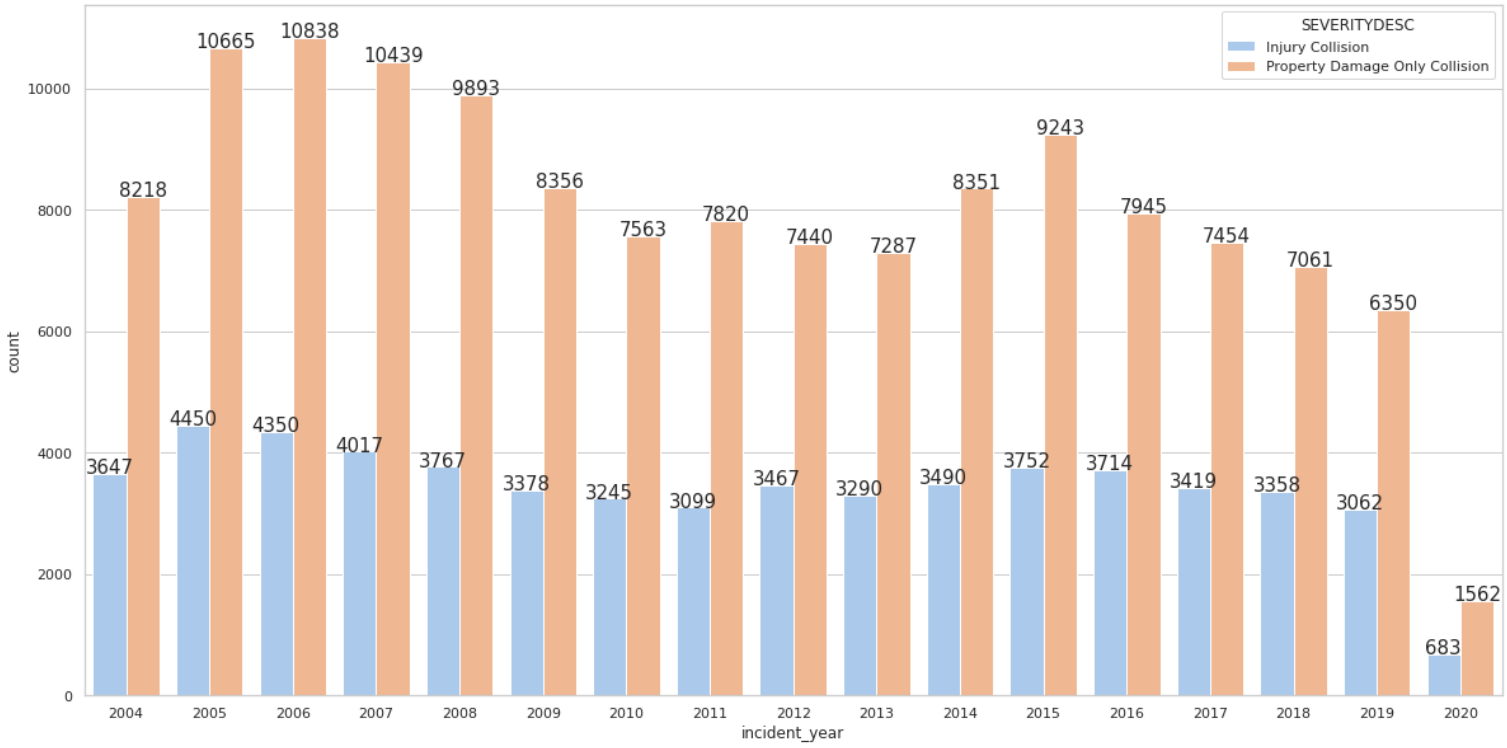
As we can see from graph 1 there is only a minimal number of samples in our dataset where the variables had the value “true”. Only around 4 percent of cases involved speeding or drivers under influence, with a slightly larger number of cases involving drivers not paying attention.

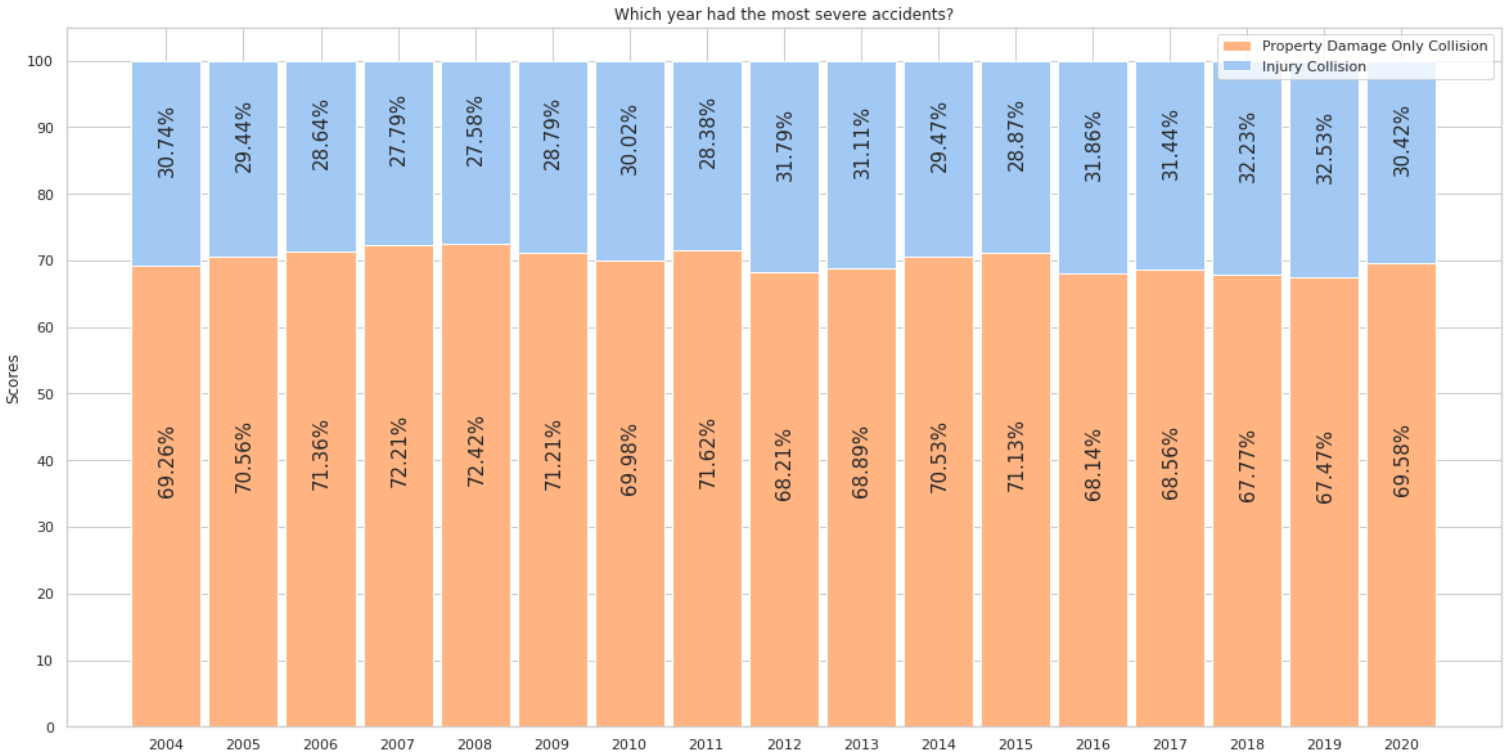
The impact of the variables on the severity becomes a lot clearer in graph 2. We can see big jumps in the percentage of severe accidents, if a driver involved in the crash was committing an offence. All in all, the true-false variables still don’t pay the picture on their own, which is why we need our machine learning models to accurately predict the severity of accidents.

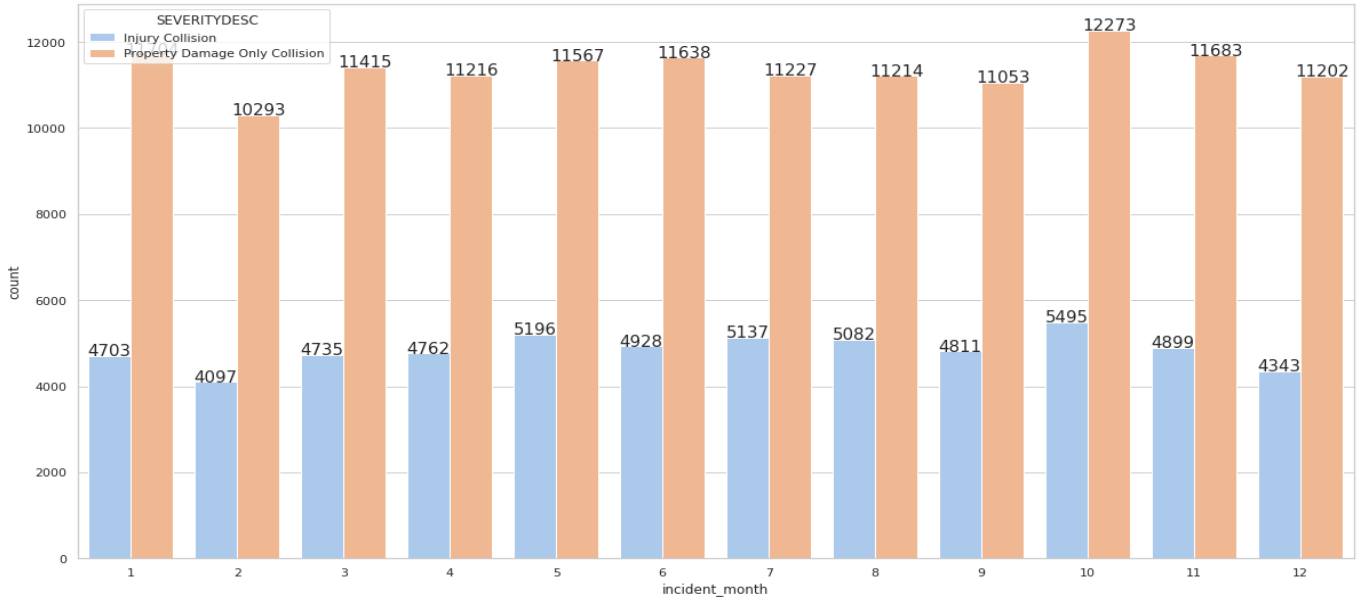
As a last exploratory data analysis, we will look at the temporal variables created in the step before. These don’t give insights in what creates a severe accident but lets us how the dataset and accident cases are structured and spread out.

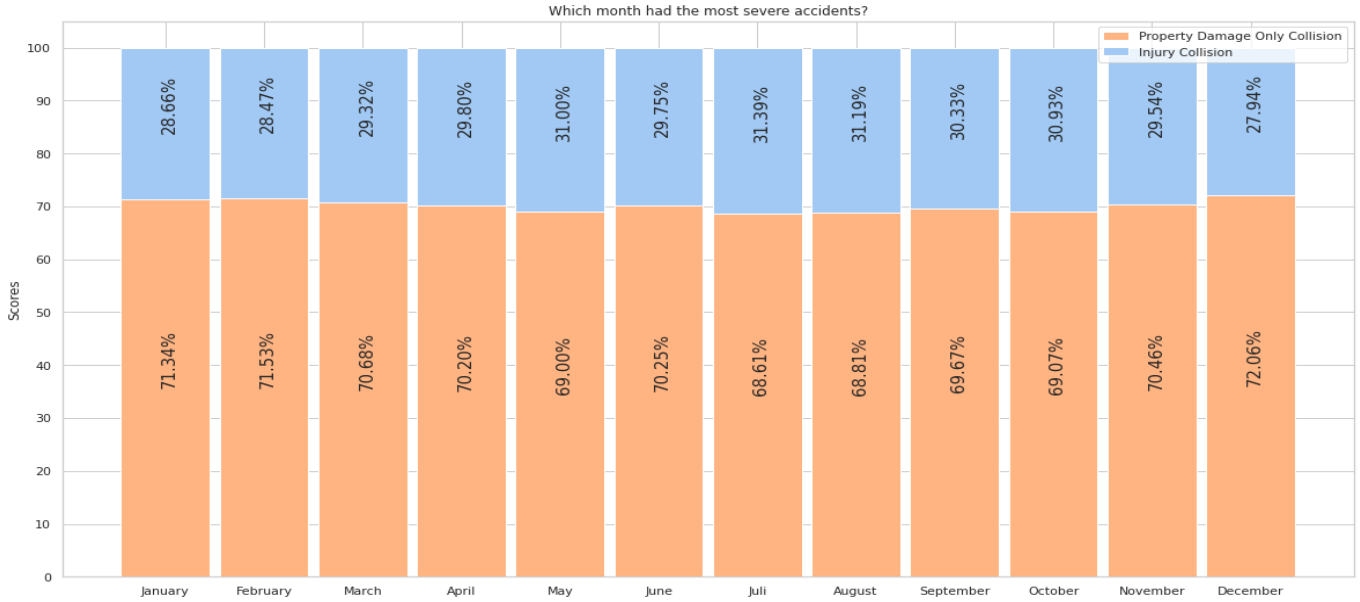
3.2 Temporal Variables

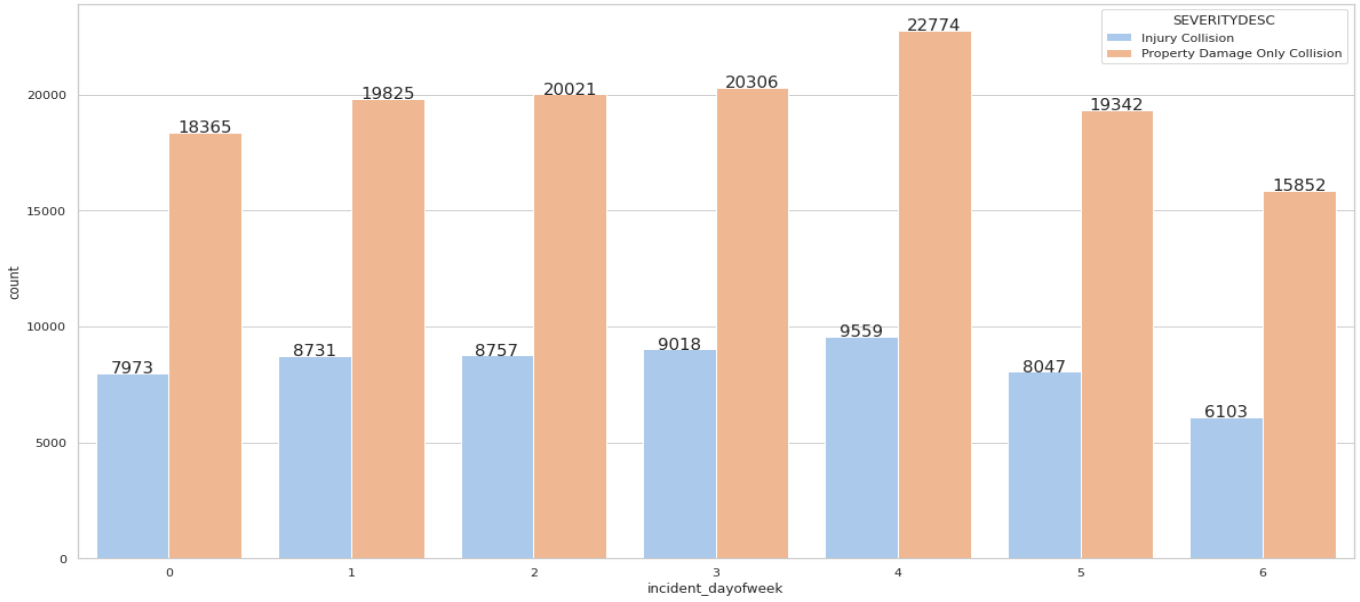
Let's look at our graphs analyzing the timing of accidents. We assessed the count of accidents, as well as the percentage of severe accidents happening in the given time frame.

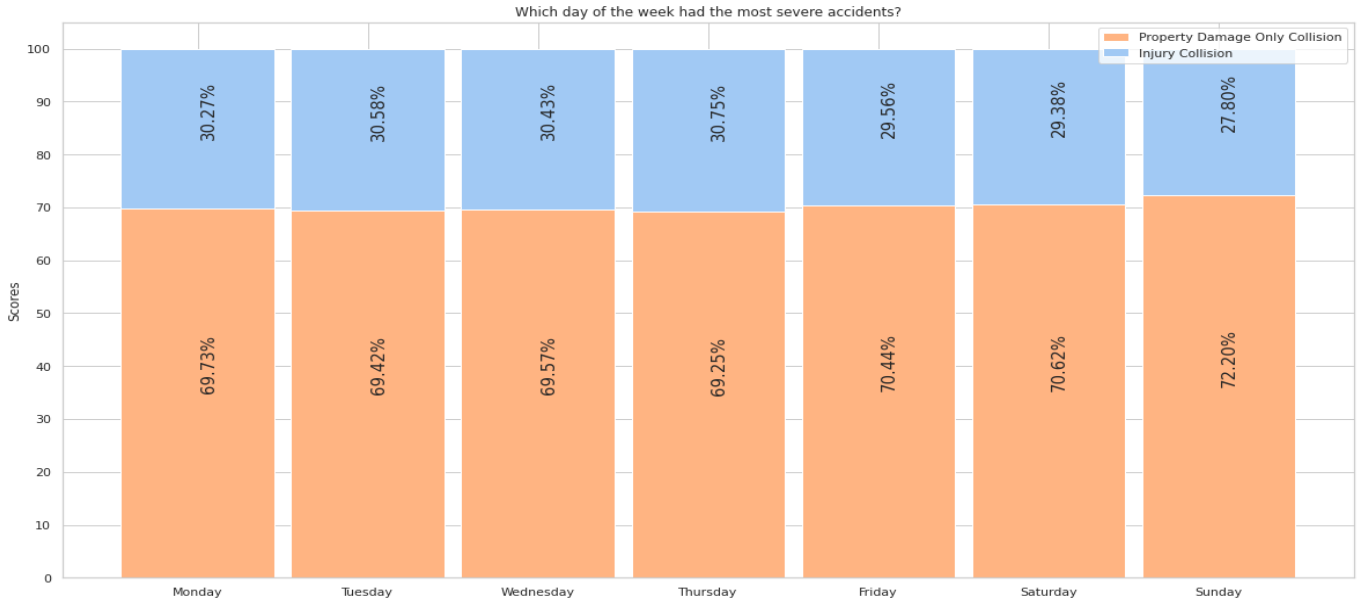


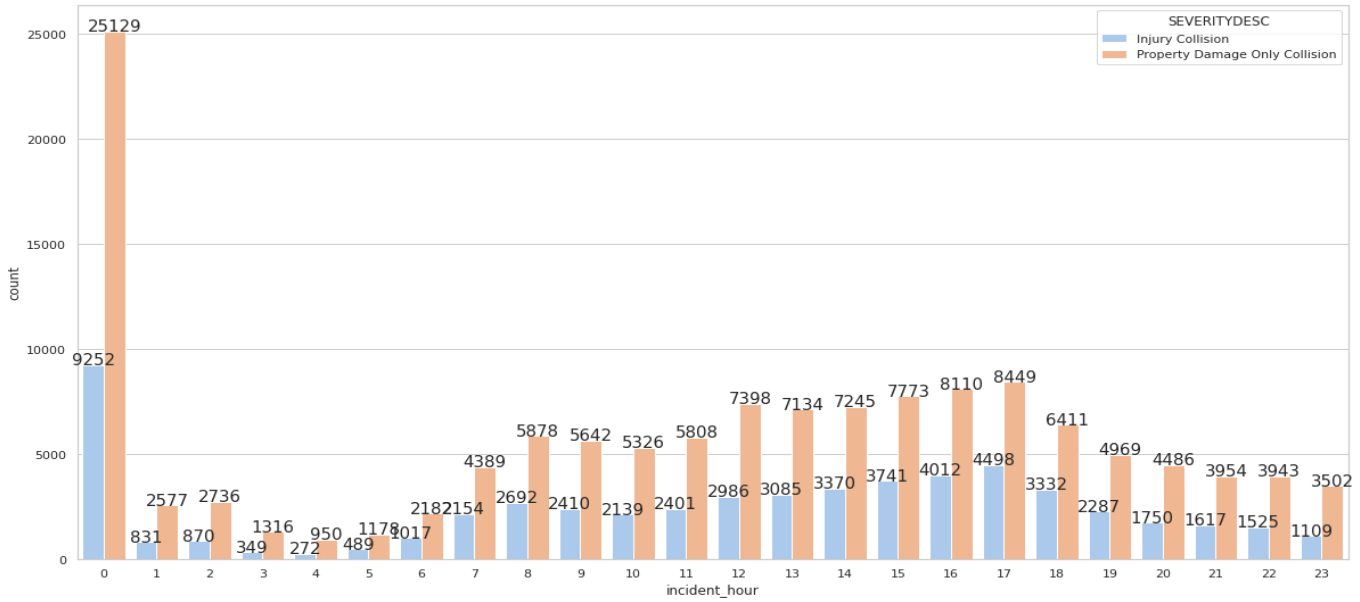


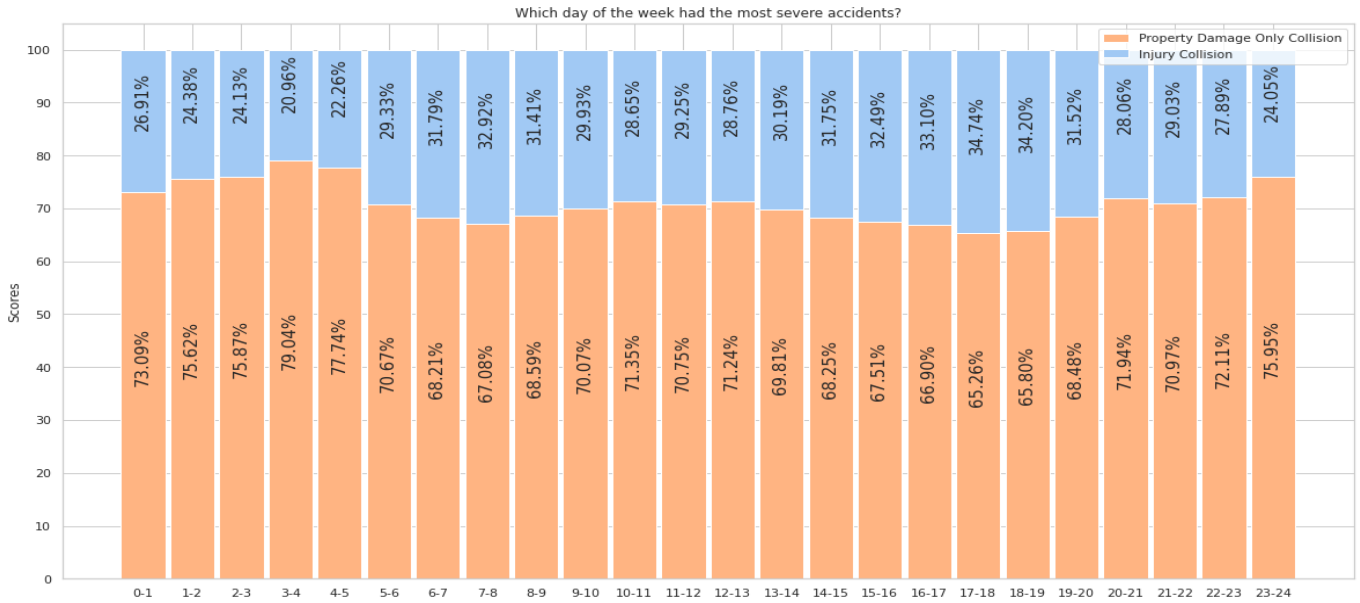

SEVERITYDESC
Injury Collision
Property Damage Only Collision
count
incident_month
4703
10293
11415
11216
11567
11638
11227
11214
11053
12273
11683
11202
4097
4735
4762
5196
4928
5137
5082
4811
5495
4899
4343


Which month had the most severe accidents?
Property Damage Only Collision
Injury Collision
Scores
January
February
March
April
May
June
Juli
August
September
October
November
December
28.66%
28.47%
29.32%
29.80%
31.00%
29.75%
31.39%
31.19%
30.33%
30.93%
29.54%
27.94%
71.34%
71.53%
70.68%
70.20%
69.00%
70.25%
68.61%
68.81%
69.67%
69.07%
70.46%
72.06%


SEVERITYDESC
Injury Collision
Property Damage Only Collision
count
incident_dayofweek
7973
18365
8731
19825
8757
20021
9018
20306
9559
22774
8047
19342
6103
15852


Which day of the week had the most severe accidents?
Property Damage Only Collision
Injury Collision
Scores
30.27%
30.58%
30.43%
30.75%
29.56%
29.38%
27.80%
69.73%
69.42%
69.57%
69.25%
70.44%
70.62%
72.20%
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


SEVERITYDESC
Injury Collision
Property Damage Only Collision
count
incident_hour
25129
9252
831
2577
870
2736
349
1316
272
950
489
1178
1017
2182
2154
4389
2692
5878
2410
5642
2139
5326
2401
5808
2986
7398
3085
7134
3370
7245
3741
7773
4012
8110
4498
8449
3332
6411
2287
4969
1750
4486
1617
3954
1525
3943
1109
3502


Which day of the week had the most severe accidents?
Property Damage Only Collision
Injury Collision
Scores
26.91%
24.38%
24.13%
20.96%
22.26%
29.33%
31.79%
32.92%
31.41%
29.93%
28.65%
29.25%
28.76%
30.19%
31.75%
32.49%
33.10%
34.74%
34.20%
31.52%
28.06%
29.03%
27.89%
24.05%
73.09%
75.62%
75.87%
79.04%
77.74%
70.67%
68.21%
67.08%
68.59%
70.07%
71.35%
70.75%
71.24%
69.81%
68.25%
67.51%
66.90%
65.26%
65.80%
68.48%
71.94%
70.97%
72.11%
75.95%
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10-11
11-12
12-13
13-14
14-15
15-16
16-17
17-18
18-19
19-20
20-21
21-22
22-23
23-24

As we can see in the first graph, the total amount of cases has been consistently decreasing over the years. Yet at the same time the number of severe accidents has been around the same each year. Therefore, as visualized in graph 2, the percentage of severe accidents has been higher in the last five years than in the years before.

Looking at the graphical representation of accidents each month, we see that there doesn't seem to be a difference in the amount of accidents per month. The percentage of severe accidents also stays roughly the same around 30 %, with May being the most dangerous month and December the least dangerous month.

Our next graph shows us how the accidents spread throughout the days of the week. We can easily recognize that most accidents happen on Fridays, with the least amount of accidents happening on Sundays. This can be attributed to people going out more on Fridays, while staying home more on Sundays. Accidents look slightly more likely to be severe during the week, with the lowest likelihood of an severe accident happening on Sundays.

One problem with the dataset for our last graph showing of the hourly number and likelihood of accidents is that cases without an information for the hour they took place on, get placed in the time frame from 0 to 1. This missing data skews the first graph a bit, but we can see that most of the total accidents happen during daytime. That makes sense, as there is a higher number of drivers in total on the roads in those hours.

Looking at hour data we see that a big majority of cases happen between 0 and 1, which could also be because of a tracking error/error in data, as cases without a correctly listed time will be put in this time frame. The last graph shows us that the most severe accidents also happen during daytime, with the number peeking between 17-18. Rush hour at that time can potentially explain the high percentage of severe accidents, while people driving more safely during the night leads to a low percentage of severe accidents during nighttime.

4. The Machine Learning Models

4.1 Preparation

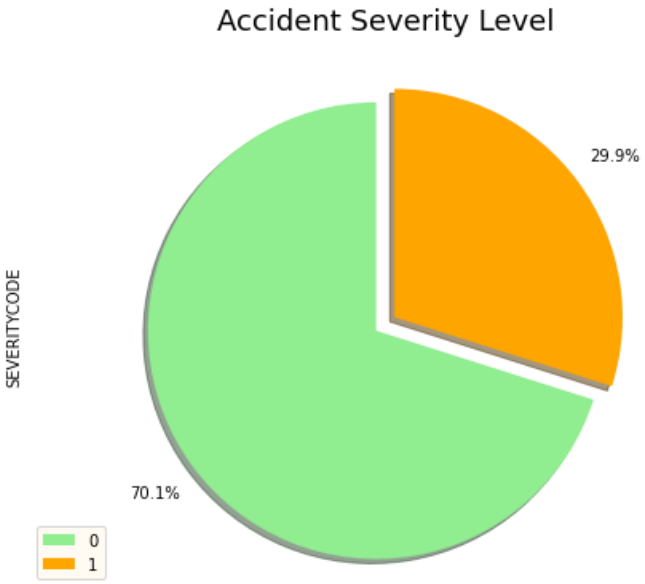
We downloaded a variety of different algorithms from the **scikit-learn** machine learning library. We imported some pre-processing tools to normalize and prepare the data four our machine learning models. Next to the machine learning models we imported, we also decided on multiple evaluation metrics to use. Another library we need and therefore imported is the **imblearn** library, which we need to use methods to balance our dataset and remove bias.

We did another feature selection to reduce the number of factors going into our machine learning models. We chose the features that we deemed to have a significant impact on accident severity based on our exploratory data analysis. The target variable is not an input, so it doesn't appear is this list.

Variables	Description
ADDRTYPE	At what kind of location the accident took place (as a category)
JUNCTIONTYPE	At what kind of junction the accident took place (as a category)
INATTENTIONIND	Whether or not the driver was paying attention (1/0)
UNDERINFL	Whether or not the driver was under the influence (1/0)
SPEEDING	Whether or not the driver was speeding (1/0)
WEATHER	Weather conditions during the accident
ROADCOND	Road conditions during the accident
LIGHTCOND	Light conditions during the accident
HITPARKEDCAR	Whether or not the accident involved a parked car (1/0)

It was necessary to use one-hot encoding to turn our dataset's categorical variables into variables with values of 0 and 1. For example the column **ADDRTYPE** got turned into three individual columns **ADDRTYPE_Alley**, **ADDRTYPE_Block** and **ADDRYTPE_Intersection** with corresponding values of 0 and 1 depending on the original address type. After the one-hot encoding process our whole feature set doesn't include any object types, but only variables of the integer type.

As we discovered in our exploratory data analysis, the dataset has a imbalance with around 70% of the samples being non-severe accident.



This imbalance in the dataset will lead to bias in our machine learning models towards non-severe cases, which is why we want to use the SMOTE (Synthetic Minority Over-sampling Technique) method from the imblearn library to create more severe accident samples based on existing severe accident samples and thus create a balanced dataset. After using SMOTE the samples will be spread 50-50 between severe and non-severe accidents. This will lead to our machine learning models making more reliable and accurate predictions.

4.2 Building and Analyzing our Models

We selected three machine learning models to use in our analysis. We decided on Decision Tree Analysis, Logistic Regression, and K-Nearest Neighbor.

The Decision Tree Analysis works by breaking down the dataset into smaller subsets while the connected decision tree is incrementally developed at the same time. The result will be the decision tree with all the decision nodes and leaf nodes.

Logistic Regression is like Linear Regression a binary classification model that in its basic form uses a logistic function to model the binary dependent variable.

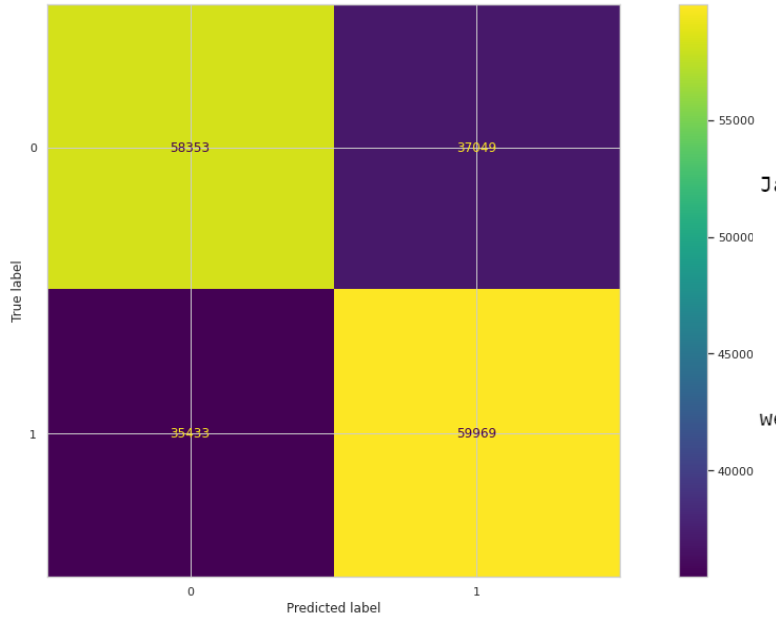
K-Nearest-Neighbor is one of the simplest classification algorithms and one of the most used ones. It stores all available cases and classifies new cases based on a similarity measure being the distance to the center of a class.

We decided on not using the support Support Vector Machine model, as its inaccurate for large data sets and this dataset has around 200.000 rows. All these algorithms deliver a supervised learning approach and we will later assess the prediction accuracy using a range of metrics. Before we set up our first model, we used a 70-30 split to break up the dataset into a training and testing set.

4.2.1 Decision Tree Analysis

Using our post-SMOTE balanced data, we used the scikit-learn library's Decision Tree Classifier to build our model. We chose entropy as the classifier and a max depth of 6 for our decision tree.

It took the system only around 0,32 seconds to build the model and the resulting accuracy was around ~0,615. You can evaluate the results more accurately with the following images.



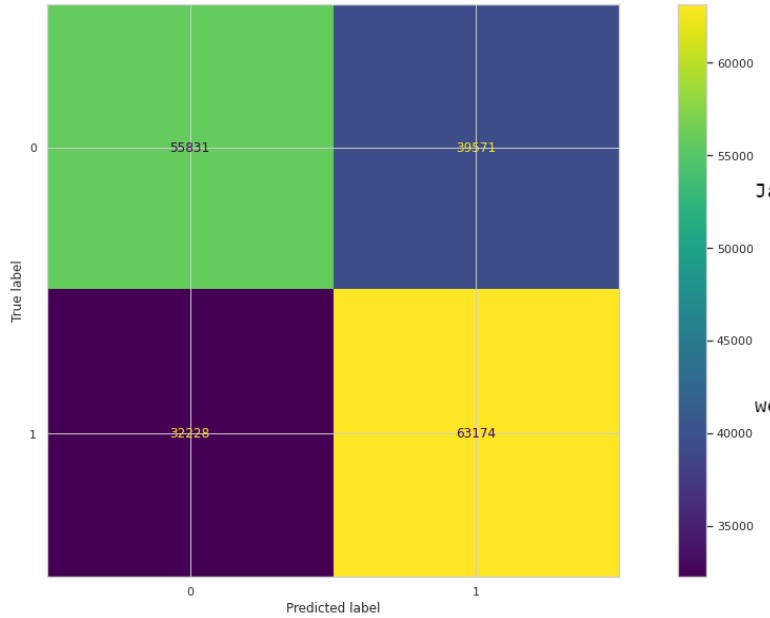
Jaccard score for the decision tree model: 0.3277469080480373

	precision	recall	f1-score	support
0	0.80	0.61	0.69	41083
1	0.40	0.63	0.49	17319
accuracy			0.61	58402
macro avg	0.60	0.62	0.59	58402
weighted avg	0.68	0.61	0.63	58402

4.2.2 Logistic Regression

Using our post-SMOTE balanced data, we used the scikit-learn library's Logistic Regression to build our model. We chose a regularization strength of "0.01", as well as the "liblinear" solver.

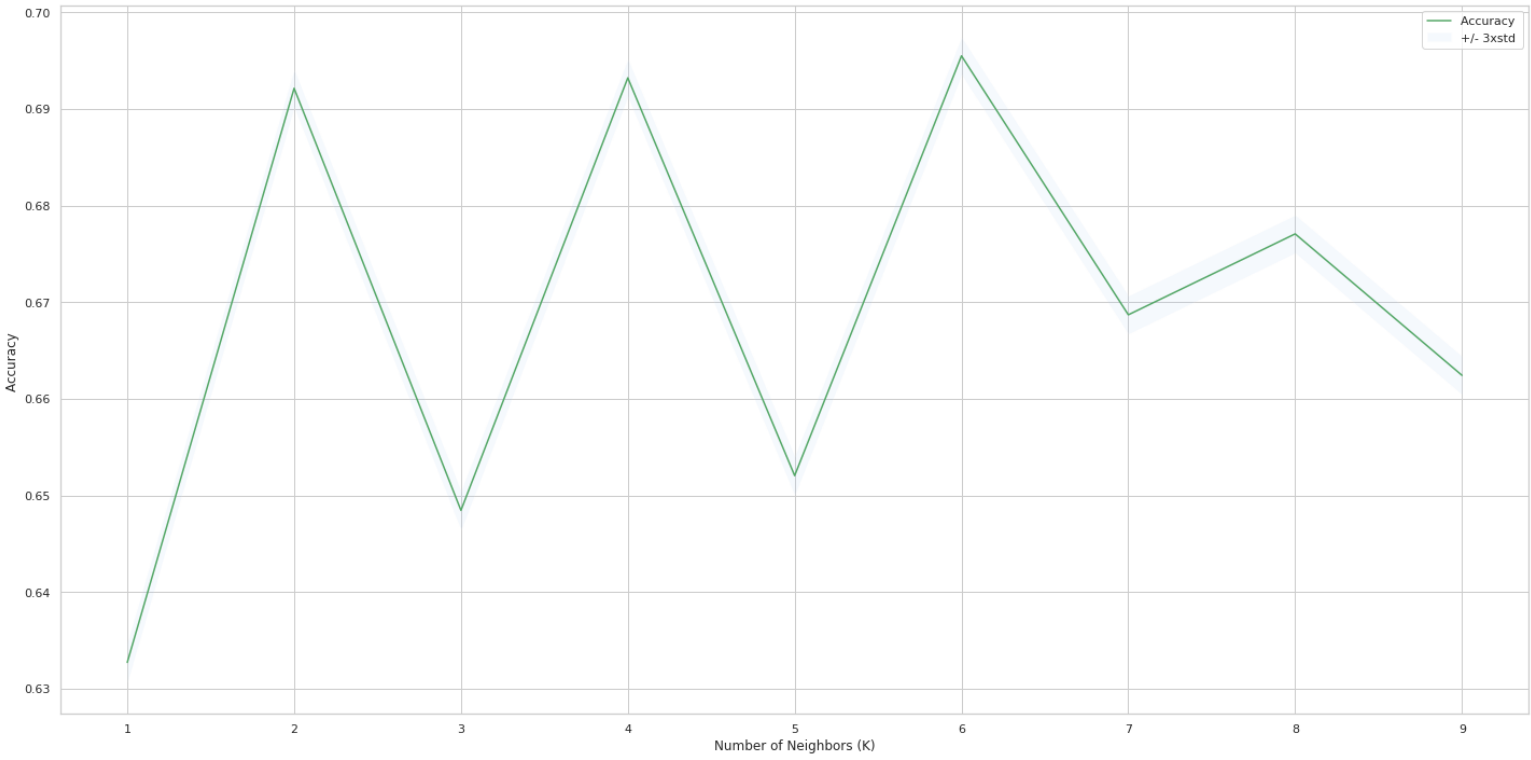
It took the system around 0,55 seconds to build the model and the resulting accuracy was around ~0,606. Additionaly the logistic loss of the model was around 0,645. You can evaluate the results more accurately with the following images.



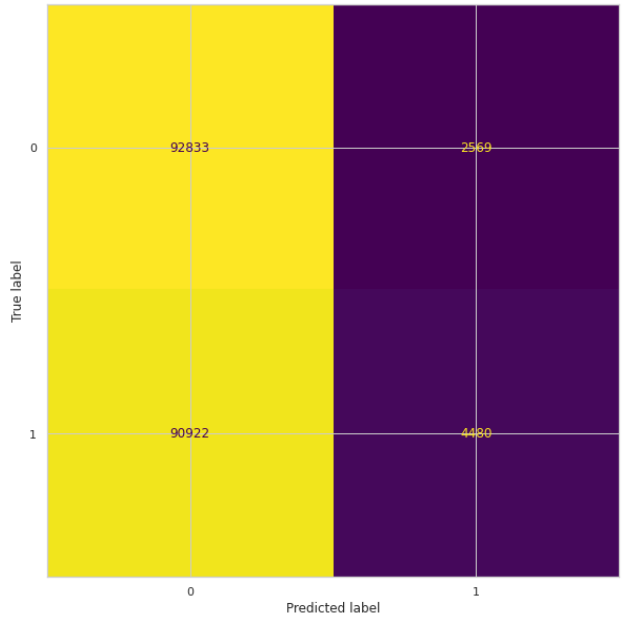

```
Jaccard score for the logistic regression model: 0.3338735818476499
              precision    recall  f1-score   support

           0       0.80      0.58      0.67     41083
           1       0.40      0.67      0.50     17319

    accuracy                           0.61     58402
   macro avg       0.60      0.62      0.59     58402
weighted avg       0.69      0.61      0.62     58402
```

4.2.2 K Nearest Neighbor

Before we started building our model, we wanted to determine the best K to use for the model. As visible in the graph, 6 was determined to be the best K with an accuracy score of ~0,695, slightly ahead of 2 and 4.



Using our post-SMOTE balanced data, we used the scikit-learn library's k-Nearest Neighbor to then build our model. It then took the system around 94,96 seconds to build the model and the resulting accuracy was around ~0,695. You can evaluate the results more accurately with the following images.




```
Jaccard score for the logistic regression model: 0.0425325724130505
              precision    recall  f1-score   support

           0       0.71      0.97      0.82     41083
           1       0.39      0.05      0.08     17319

    accuracy                           0.70     58402
   macro avg       0.55      0.51      0.45     58402
weighted avg       0.61      0.70      0.60     58402
```

5. Results and Discussion

5.1 Results

We decided on using the following metrics to evaluate and compare each model's accuracy. Jaccard Score, f1-Score, Precision and Recall will be the main comparing factors, as well as including the Logistic Loss for the Logistic Regression model and the time needed to build each model.

Algorithm	Accuracy	Jaccard	F1-score	Precision	Recall	Time (s)	LogLoss
Decision Tree	0.614688	0.327747	0.631087	0.404446	0.633466	0.324444	NA
Logistic Regression	0.605904	0.333874	0.622953	0.400987	0.666089	0.552270	0.644885
KNN	0.695490	0.042533	0.599260	0.386308	0.045615	94.956030	NA

5.2 Discussion

Looking at the accuracy score, it looks like KNN is the best model by having a big advantage with its score of 0,695 being higher than 0,615 and 0,606 respectively. On the other hand, KNN's Jaccard score is much worse than the other two models' score. Additionally, it took substantially longer to build the KNN model. That's why we should take a closer look at the remaining metrics and factor in which metrics are most important for our use case problem.

5.2.1 F1-Score

F1-Score measures the accuracy of a model by calculating the harmonic mean of the model's precision and recall. If both precision and recall are perfect (both being 1), then the f1-score would be a perfect 1. The lowest possible value is 0 which happens if either recall or precision are 0.

The F1-score in our table above is averaged out of the two F1-Scores for each target variable type (0 & 1). That means that it's biased towards property damage because of the imbalance in the dataset.

Decision Tree Analysis has the best F1-Score with 0,631, ahead of Logistic Regression (0,623) and KNN (0,599).

5.2.2 Precision

The precision score relays how accurately the models predicts true positives without predicting false positives. The percentage is calculated by dividing the number of true positives by the sum of true positives and false positives.

For our car accident severity problem, the precision value allows us to evaluate how many of our machine learning model's predicted severe accidents were severe and not predicted as severe and then actually non-severe.

The Decision Tree Analysis model has the best score again with 0,404, a nudge ahead of the Logistic Regression Model with 0,401. The KNN model is a bit behind with 0,386, putting it in current last place.

5.2.3 Recall

The recall score also gives a percentage of how many relevant results were correctly classified by the algorithm. In this case it measures how accurately the models predicts true severe cases and how many severe cases it didn't classify correctly. The percentage is calculated by dividing the number of true positives by the sum of true positives and false negatives.

The recall value allows us to evaluate how accurate our predictions were regarding the percentage of severe cases predicted correctly and how many were missed.

The best model in this category was the Logistic Regression model with a score of 0,666. Closely behind is the Decision Tree Analysis Model with a score of 0,633. The KNN model has very poor result, getting only a score of 0,005, which is caused by the big amount of cases the model falsely predicted as non-severe but being severe.

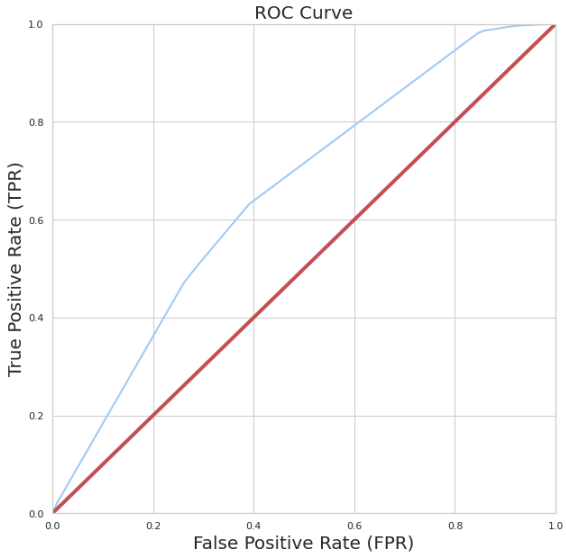
Regarding the significance of each metric for our specific case study, the recall value has the highest high importance. A high recall equals our model accurately recognize the factors leading to a severe accident and not missing any severe cases. This can substantially help with drivers being warned and reacting to situations that could lead to severe accidents. The precision score can tell us how many times we reacted to a predicted severe case

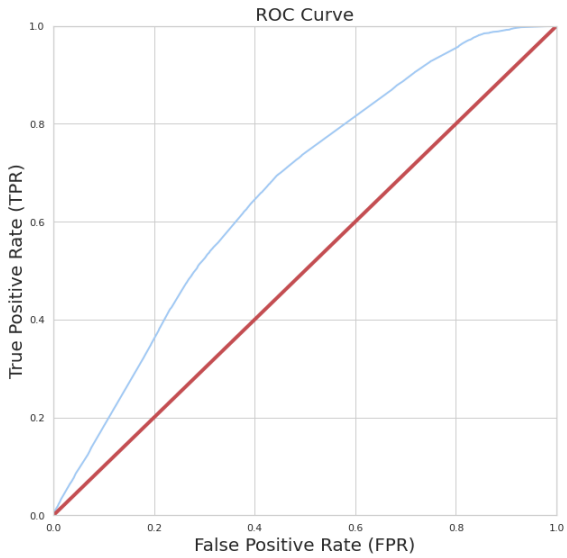
unnecessarily, but in the accident severity case study we prefer to be more cautious than missing any potential severe accident.

That is why our leading model is now the Logistic Regression model, which is only slightly ahead of the Decision Tree Analysis model because of the Logistic Regression model's higher recall score. The other metrics are similarly scored for both models, so the recall score is the decider in this case.

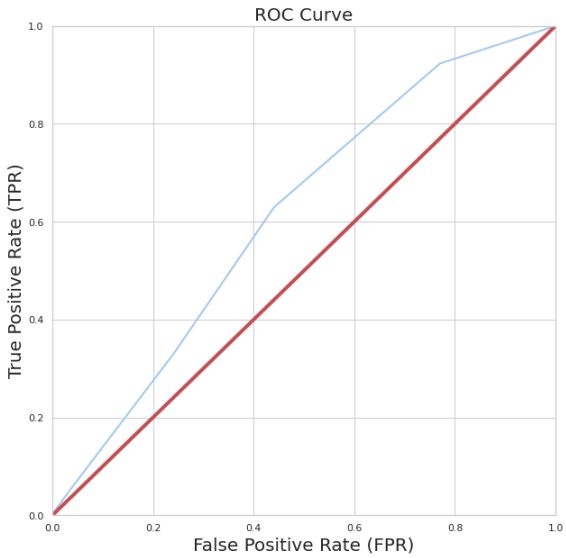
5.3 ROC Curve

As an additional evaluation method, we used the ROC (Receiver Operating Characteristic) Curve. This highly regarded metric evaluates the success of the predictions in a graphical way, showing the relationship between the true positive rate and the false positive rate. A very good model should have a steep curve around the top left corner.





Graph 1 shows the ROC curve for the Decision Tree Analysis model, graph 2 the same for the Logistic Regression model. We see how close the 2 different models are in their evaluation, with a slight edge gained by the Logistic Regression model. The following graph 3 shows that the KNN model isn't as accurate in this specific metric.



6. Conclusion

After assessing all different evaluation metrics, we can confirm that the Logistic Regression model is the best machine learning model in our case problem to accurately predict severe accidents. That's why we recommend using this modeling algorithm, as it also leads in the evaluation metrics most important to our problem case.

Our evaluation metric scores for the model were far from perfect however, which can be improved upon by implementing the following potential improvements to our dataset and model:

- A bigger sample size would lead to a more refined model. Especially increasing the amount of severe accident samples would reduce the bias in the model and allow to gain more insights on which factors impact the severity the most.
- Another improvement would be to reduce the number of missing values in our dataset that we needed to clean up, replace and fill with useful information. The "unknown" type in our categorical variables should also be eliminated from the future dataset.
- Adding more features to the dataset like information about the vehicles involved, how the accident between transpired and more could lead to a more refined model.

All in all, we are happy with the result of our machine learning models and think that they can already be helpful to our stakeholders in several ways:

- Drivers can be taught about which factors lead to severe accidents and can then adjust their driving appropriately.
- Hospitals, police and ambulances can prepare and be on alert for potential severe accidents happening, if the features values predict them at a current time.
- Governmental agencies can use the model to decrease the rate of severe negating the impact of impactful features by improving locations where accidents happen, or improving conditions on their streets
- Navigation app developers can include live information about factors increasing the probability of severe accidents and influencing drivers to drive more safely

Another potential future use case for this analysis and model is to evolve the model to also forecast accidents. Hospitals, ambulances, police and health organizations can be more prepared when they can accurately expect severe accidents to happen during specific time, locations and conditions. This can also lead to the prevention of a big amount of severe cases by implementing road signs informing drivers of the high likelihood of severe accidents happening and thus reduce the total number of accidents and vehicular deaths happening.