Business Understanding

The problem consists on an analysis of data collected by the Seattle Department of Transportation from 2004 to present days. More than 200000 accidents have been happening since then. With the help of the dataset collected over the years, I want to build a machine learning model that could predict the severity of an accident giving particular external conditions, such as the type of location, the weather, the road conditions, or the light. Implementing such a model would allow for calculating the proabbility of an accident and alert drivers of dangerous conditions, with the goal of decrease the overall number of accidents.

The audience is therefore the Seattle Traffic Management Department, which could use the model to improve external conditions (for instance, taking care of damaged roads or increasing the lights on dark streets) and to alert the police or the drivers when dangerous conditions may lead to severe accidents.

Data Understanding

I decided to directly download the dataset from the official Seattle GeoData website (https://data-seattlecitygis.opendata.arcgis.com/). The advantage of this is that this dataset is updated weekly, so the model can be directly updated with the new entries each week. and continuously updated) reports all accidents that have been happening in Seattle from 2004 to present days.

The dataset is a csv file, consisting of 221738 rows and 40 columns at the time of writing (oct 7th, 2020). Each column, or attribute, is helpful in the overall description of the accident. For instance, they describe the address of the accident and the type of location (intersections or segments), whether the accident involved cars, bikers or pedestrians, and the total number of people involved in each accident. A number of attributes describes the external conditions surrounding the accident, such as the weather, the visibility, the road conditions, or the presence of drivers under influence.

An important column is the description of the severity of the accident, which has 5 possible outcomes: fatality, serious injury, inkury, or property damage only, plus a category for unknown severity. This will become the dependent variable (y) in our analysis, with other conditions being the independent variables (x): type of location, weather, light condition, road condition, type of collision. These are the attributes that I will use to train the machine learning models.

To help in this decision, during this phase I will use some simple plots to visualize the data and explore correlations: for example, plotting the number of people involved in accidents as a function of weather conditions.

METHODOLOGY

Before proceeding with the data analysis, the downloaded data needs to be prepared. In order to do so, we need to clean up the dataset by removing unnecessary columns and missing values, and then choose, build and evaluate machine learning models to make predictions.

- **Data preparation**

We start by reading the dataset into a panda dataframe. This dataframe contains 221738 rows and 40 columns. With the *info()* method, we find out that several columns have missing values. I used *drop()*, *replace()* and *dropna()* method to get rid of columns (attributes) that are not needed for our analysis, as well as missing values. Some of the attributes (columns) that we want to keep instead include the followings:

- ◊ ADDRTYPE: it described the type of location where the accident happened: an alley, a block or an intersection
- ◊ LOCATION: coordinate of the collision
- ◊ PERSONCOUNT: the number of people involved in each accident
- ◊ PEDCOUNT: number of pedestrians
- ◊ PEDCYLCOUNT: number of bicycles
- ◊ VEHCOUNT: number of vehicles
- ◊ INCDTTM : date and time of the accident
- ◊ JUNCTIONTYPE: type of junction where the accident took place
- ◊ WEATHER: a description of the weather conditions at the time of the accident
- ◊ ROADCOND: road conditions at the time of the accident
- ◊ LIGHTCOND: light conditions at the time of the accident
- ◊ UNDERINFL: whether or not the driver was driving under influence

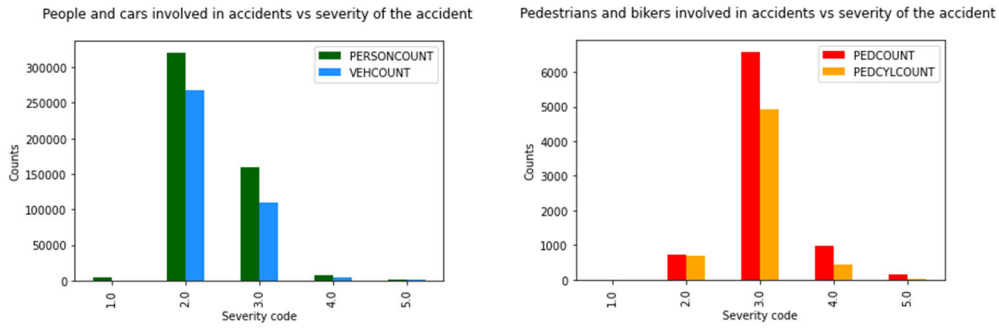
- **Data exploration**

After some clean up of the sataset, it is a good practice to visualize the data with some simple plots, in order to look at possible correlations between the attributes.

For instance, we can start by looking at the overall number of people involved in accidents duting the years, or by month, or by day of the week (0 to 6 meaning Monday to Sunday):

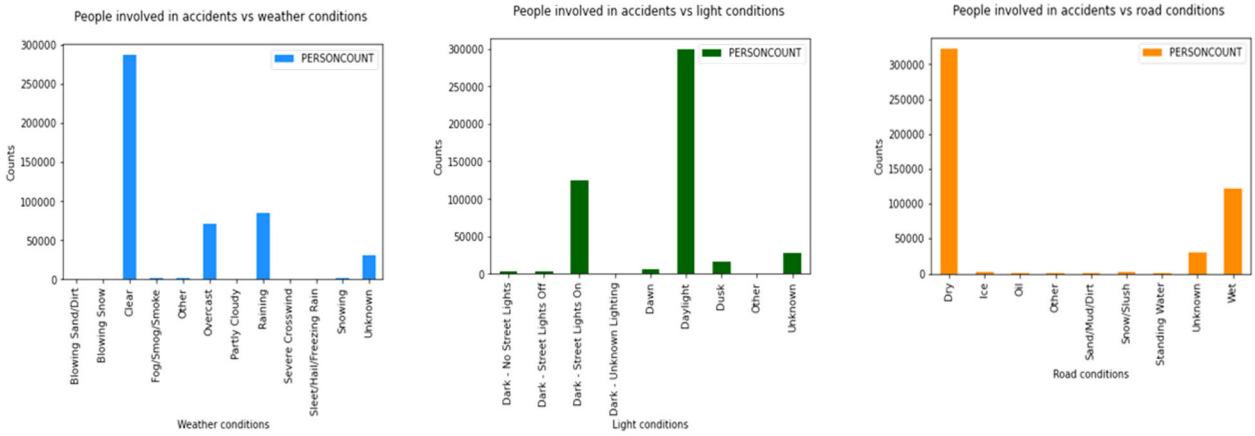
From here, we can see some trends already: accidents seem to decrease over the years, but they are still present during the lockdown months of 2020. The months do not really seem to have a trend, whereas there is an increase in the number of accident during Fridays.

Another useful piece of information could be the number of car drivers, bikers and pedestrians as a function of the severity of the accident (from 1 to 5 meaning from unknown, to damaged property, to injury, to severe injury, to fatality):



It is worth noticing that the severity is higher for bikers and pedestrians rather than for car drivers.

Finally, we can take a look at how the road, light and weather conditions affect the overall number of people involved in the accident:



The main takeway here is that indeed such conditions have an impact on the accident, and the majority of the accident are caused by a limited number of conditions. This information could be used, for instance, to further simplify the dataset and group the conditions with a very limited number of data into one.

- **Machine learning models**

Once we have an understanding of the dataset we are working with, we can start building the models to train. I started by defining a new dataset corresponding to the attributes that I want to use in the model as X (independent variables), such as weather, light and road conditions, etc.

As a dependent variable Y we will use the severity code.

We perform a train-test splitting of our dataset, keeping the test set size at 30%, and we proceed with the normalization of our X dataset.

I decided to train three different machine learning models: decision tree, k-nearest neighbors and logistic regression. After the training and the prediction step, we need to evaluate the models in roder to know how good they are.

I perform the evaluation with two different metrics, the Jaccard score and the F1 score.

Moreover, for the KNN model I run a for cycle that could explore different values of k. I found that the best accuracy is given by k=8, which I used to run the model again.

Results and Discussion

After the modeling section, I reported the final accuracy values as measured by Jaccard score and F1 score for the three algorhytms I used:

Algorithm	Jaccard	F1-score
Decision Tree	0.733372	0.684546
K-Nearest Neighbors	0.698035	0.684352
Logistic Regression	0.724192	0.659623

As we can see from the above table, the different models have a similar accuracy, whose values range between 66% and 73%. That means that none of them is extremely accurate, since the prediction they will make will only have a ~70% probability of actually happening. This may be caused by under fitting, that is, we do not have a large enough dataset to make more accurate predictions. On the other hand, it is clear that the overall combination of conditions has an impact on the accidents.

Conclusions

The results allow me to conclude that using machine learning is a good strategy to improve viability and halp preventing accidents. However, the dataset is still not large enough to be able to make predictions that are more than 70% accurate. Moreover, it would be helpful to have more accurate reports of the accidents, without missing values.

These conclusions could be extremely helpful for our stakeholder, that is the Seattle Department of Transportaiton and the Seattle Police. Officers in charge of taking the report could be trained to improve the recording of the accidents, and the model could be deployed to police and car drivers to advise when the external conditions mey lead to an accident.