Predicting the Car accident severity

1. Introduction

1.1 Problem description

Throughout the world, roads are shared by cars, buses, trucks, motorcycles, mopeds, pedestrians, animals, taxis, and other travelers. Travel made possible by motor vehicles supports economic and social development in many countries.

Nowadays vehicles are involved in crashes that are responsible for millions of deaths and injuries. According to Centers for Disease Control and Prevention (CDC), National Center for Injury Prevention and Control (NCIPC). Web-based Injury Statistics Query and Reporting System (WISQARS): https://www.cdc.gov/injury/wisqars/LeadingCauses.html road traffic crashes are a leading cause of death in the world and the leading cause of non-natural death or healthy citizens for all age groups.

Low- and middle-income countries are most affected. According World Health Organization (WHO) Global Status Report on Road Safety 2018: https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/ the road traffic crash death rate is over three times higher in low-income countries than in high-income countries.

Road traffic injuries place a huge economic burden on low- and middle-income countries. Each year, according to the latest available cost estimate (1998), road traffic injuries cost 518 billion dollars worldwide and $65 billion USD in low- and middle-income countries, which exceeds the total amount that these countries receive in development assistance (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.174.5207&rep=rep1&type=pdf).

According to Eurostat statistic "Road accident fatalities by vehicles" car drivers and passengers represented the largest category of road traffic deaths in the EU in 2018, with 44.8% of all road traffic fatalities (Figure 1). Therefore, it is important to search new ways to reduce road traffic accidents.

The most effective ways today are the development of infrastructure, setting speed limits, and the development of smart driver assistance systems. However, low-income countries cannot always afford to build safe and reliable infrastructure. Also, not everyone can afford or want to buy expensive cars that will be equipped with all modern safety systems.

Therefore, it makes sense to develop accident prevention capabilities that can be used by most people around the world.

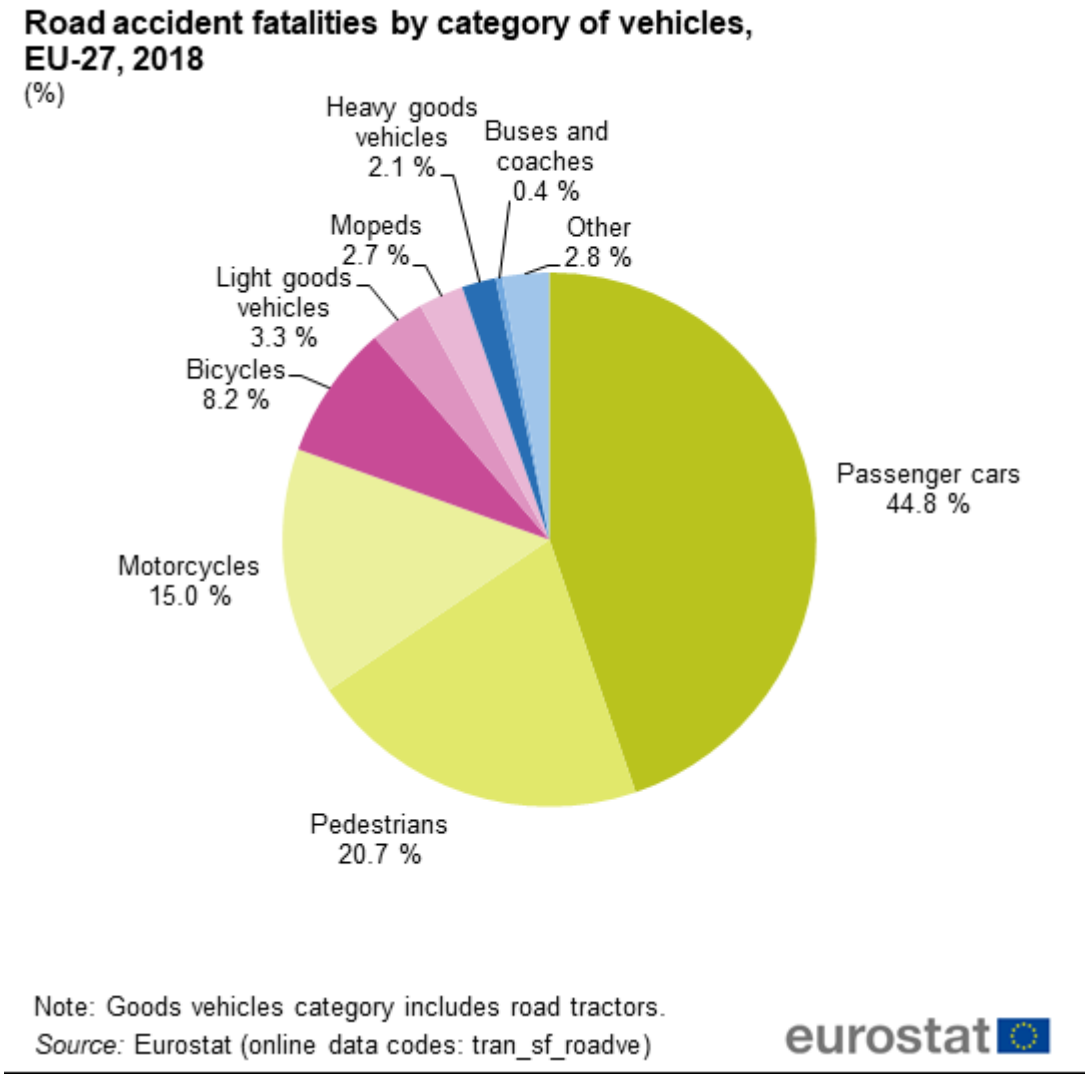

Figure 1. Road accident fatalities by category of vehicles in Europe

1.2 Objective

To develop the navigation app that could warn the car drivers, given the weather, the road conditions and some other parameters about the possibility of getting into a car accident on the chosen route and how severe it would be. So, this app would offer to change the travel route if it is possible or the car drivers can simply take into account the warn message and drive more carefully.

1.3 The target audience

The target audience for this app are the car drivers with smartphones.

1.4 Stakeholders

The main stakeholders are the governments of the countries that are interested in the reducing of car accidents on the roads.

1.5 Question

Can we predict for given route the severity of possible car accident in real time for any region?

2. Data

2.1 Data source

In order to create the required prediction model, it is necessary to find the dependence of the severity of the accident on parameters that can be collected in real time, such as weather conditions.

First, I used statistics about road traffic accidents to determine the factors that influence their severity. For these purposes, data from SDOT Traffic Management Division were used.

All collisions provided by SPD and recorded by Traffic Records with weekly update frequency. This includes all types of collisions in timeframe from 2004 to present. This dataset includes many attributes that describe all the circumstances of the accident, the number of victims and their severity.

2.2 Data acquisition

At this stage, I uploaded a .csv file from the Internet source to my Python environment and created a data frame using pandas to perform data analysis and derive some additional info from raw data to define, which attributes could be potentially useful for future prediction model.

I noticed that the names of the attributes in the dataset are written using abbreviations. To decrypt them, in addition to the dataset, it is also necessary to download a file with metadata using a following link https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf, which contains descriptions of the attributes. This will allow to understand what information I have.

2.3 Data understanding

2.3.1 Defining a target function

First of all, I determined which of these attributes I will predict as a target function in our model. Task is to predict the severity of a road traffic accident, so I used SEVERITYCODE as a target function. It corresponds to the severity of the collision and is a discrete value. Therefore, I will use classification model for prediction.

2.3.2 Selecting the required attributes

Based on Metadata, we can preliminarily analyze the attributes and filter out those that will not be useful for creating a prediction model.

At first, I deleted the specific codes and definitions that SDOT uses for its reports, since this information cannot in any way be used for data analysis. This includes the following attributes:

- INCKEY - A unique key for the incident
- COLDETKEY - Secondary key for the incident
- SDOT_COLCODE - A code given to the collision by SDOT
- SDOT_COLDESC - A description of the collision corresponding to the collision code
- SDOTCOLNUM - A number given to the collision by SDOT
- REPORTNO - A number of report
- STATUS
- EXCEPTRSNCODE
- EXCEPTRSNDESC

Since for the task it is necessary that the algorithm be universal and can be applied in different regions, I filtered out those attributes that are relevant only for Seattle. This includes all attributes that contain specific location data:

- OBJECTID - ESRI unique identifier
- SHAPE - ESRI geometry field
- INTKEY - Key that corresponds to the intersection associated with a collision
- LOCATION - Description of the general location of the collision
- SEGLANEKEY - A key for the lane segment in which the collision occurred
- CROSSWALKKEY - A key for the crosswalk at which the collision occurred
- ADDRTYPE - Collision address type

This report contains attributes that contain information about the number of people and vehicles involved in the incident, as well as information about the type of incident. Although they give a

more complete understanding of the severity of the incident, they cannot be used as input parameters for the model, since this data was obtained after the incident.

- COLLISIONTYPE - Collision type
- PERSONCOUNT - The total number of people involved in the collision
- PEDCOUNT - The number of pedestrians involved in the collision.
- PEDCYLCOUNT - The number of bicycles involved in the collision.
- VEHCOUNT - The number of vehicles involved in the collision
- ST_COLCODE - A code provided by the state that describes the collision
- ST_COLDESC - A description that corresponds to the state's coding designation
- HITPARKEDCAR - Whether or not the collision involved hitting a parked car
- JUNCTIONTYPE - Category of junction at which collision took place
- PEDROWNOTGRNT - Whether or not the pedestrian right of way was not granted

I deleted also attributes containing date and time stamps of incidents, because based on this information, it is not possible to make assumptions about an accident in the future and its severity:

- INCDATE - The date of the incident
- INCDTTM - The date and time of the incident

This dataset has two SEVERITYCODE columns that duplicate each other. Therefore, I removed one of them.

There are factors that led to the accident, such as the driver being under the influence of alcohol or drugs, speeding or inattention, affect the severity of the accident. But unfortunately, these factors cannot be predicted in advance when applying a route for our users and they cannot be used for our model. Therefore, I removed all cases of accidents that were caused by one of these factors, so as not to distort our statistics. After all cases caused by these factors have been deleted, I deleted the corresponding columns.

As a result, only those parameters remained that could potentially affect the severity of the accident and which can be collected in real time in order to always give the user an up-to-date prediction. These include:

- WEATHER - weather conditions during the time of the collision
- LIGHTCOND - The light conditions during the collision
- ROADCOND - The condition of the road during the collision

2.4 Data pre-processing

2.4.1 Dealing with missing values

Typically, datasets may contain missing values (no data value is stored for feature for a particular observation). There are several ways to solve this problem in order to analyze the data, and therefore use it for machine learning algorithms:

- Drop the missing values (drop the variable, drop the data entry)
- Replace the missing values (with an average, by frequency, based on other functions)

Since there is a lot of data in our dataset, it will be permissible to drop some data entries wirh missing values.

After checking, it turned out that some records contain missing values in one of the selected columns or in all at once. It was also found that some conditions were flagged as unknown. Therefore, it was decided to delete the corresponding rows from the dataset.

2.4.2 Exploratory data analysis

At the beginning of the analysis, it is necessary to understand the distribution of data by class (Figure 2).

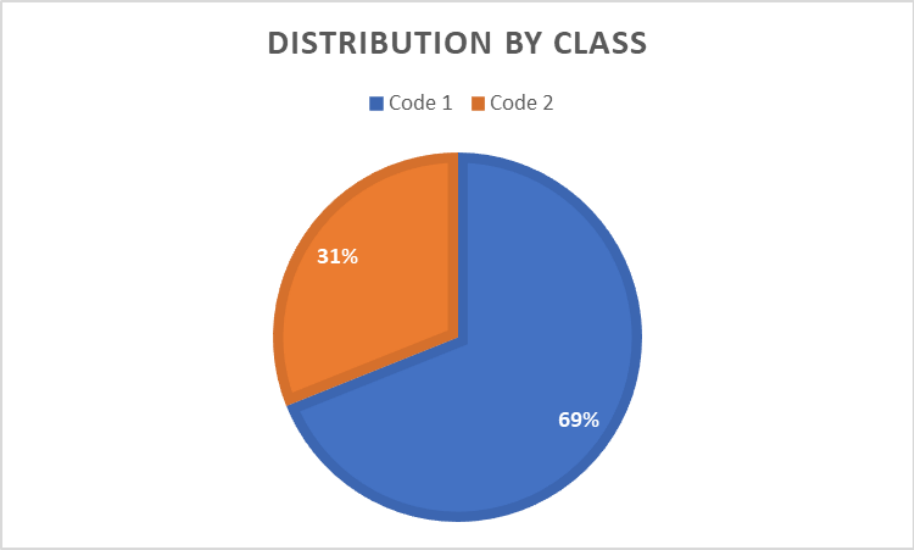

Figure 2. Distribution of all data by SEVERITYCODE

There are 90851 cases of property damage only and 40925 cases of injury collisions. This means that the distribution is uneven, which can negatively affect the model. No fatalities accidents left after pre-processing the data. Thus, I was unable to investigate the conditions that could lead to a fatal accident.

Then I investigated how many accidents occurred in each type of weather (Figure 3), road (Figure 4) and light condition (Figure 5).

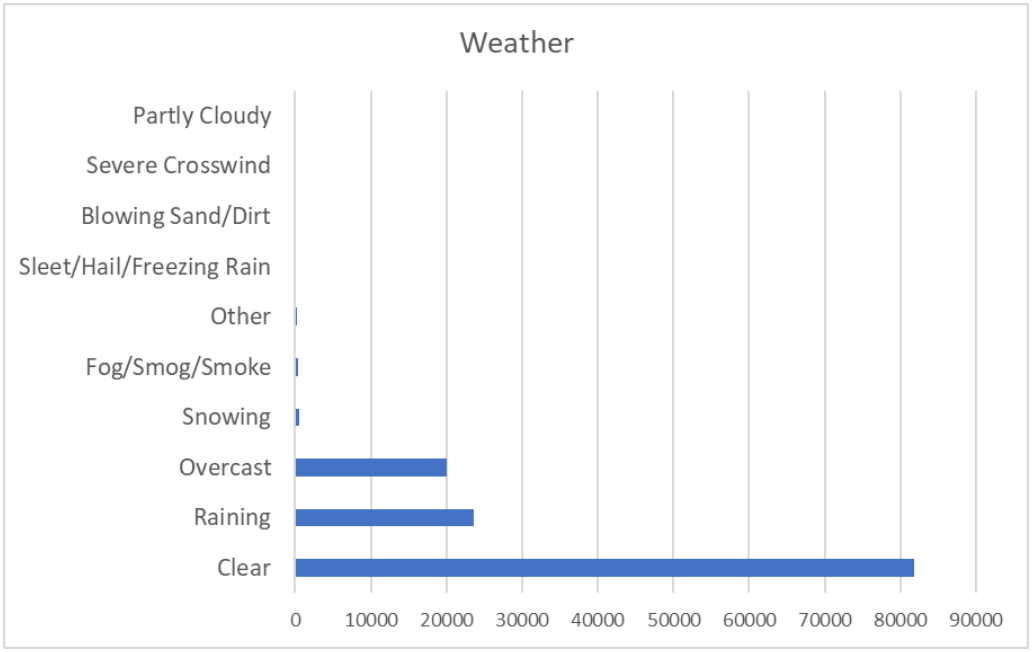

Figure 3. Distribution of data by WEATHER

Most accidents happened in clear weather. Also, many accidents occurred in overcast and during rain. These are the most frequent weather conditions that can be encountered on the road. Least of all accidents happened during partly cloudy weather and during a severe crosswind. These are the most unlikely weather conditions that can be encountered on the road.

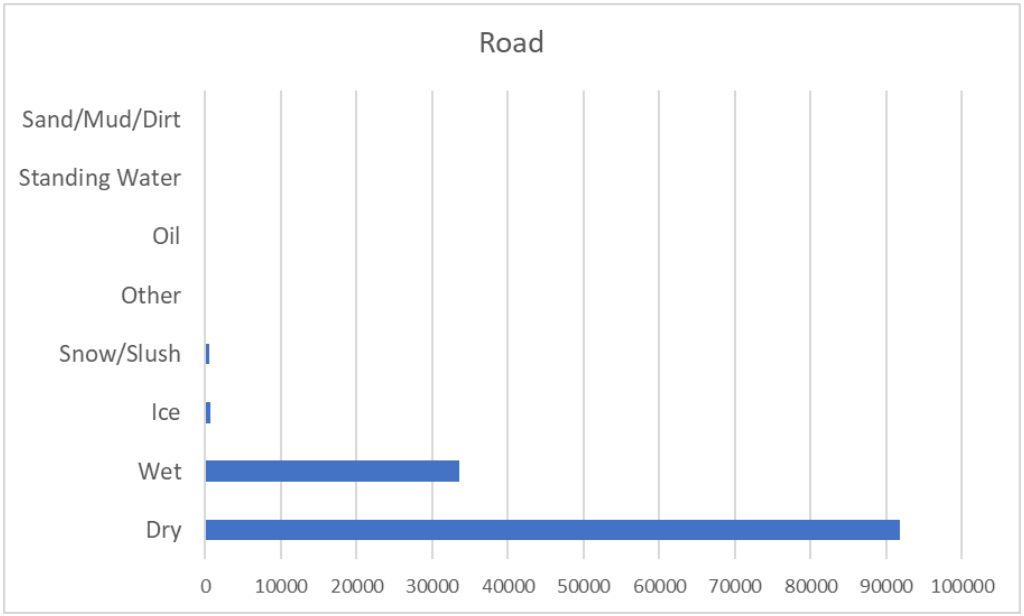

Figure 4. Distribution of data by ROADCOND

Most accidents occurred on a dry road. Many accidents also happened on a wet road. Other conditions are very unlikely. Least of all accidents happened on standing water and sand/mud/dirt.

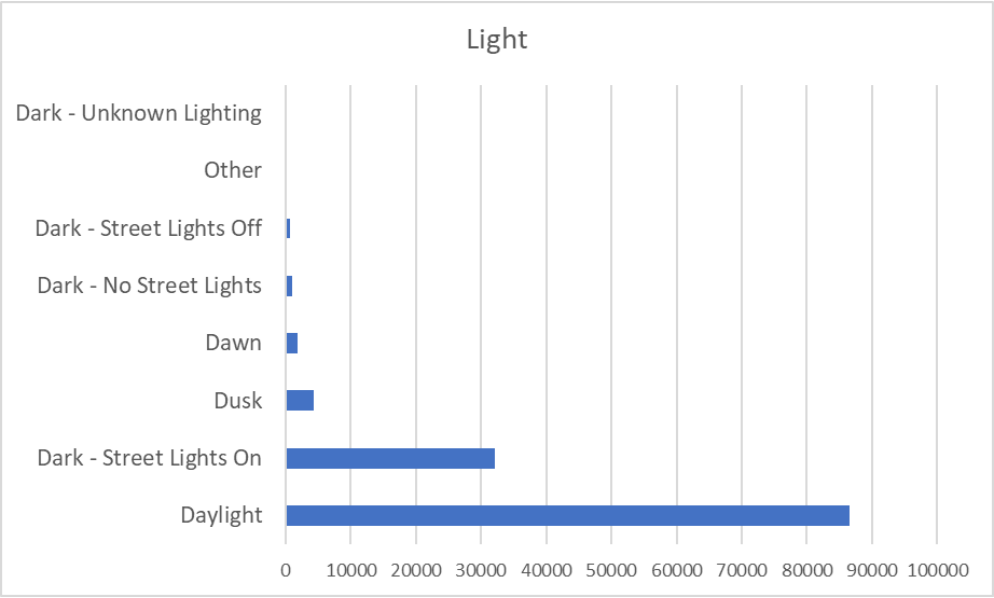

Figure 5. Distribution of data by LIGHTCOND

Most accidents occurred in daylight. Many accidents also happened in the dark with the lights on.

This information only gives us an understanding of how the data is distributed in our dataset. From this distribution we can conclude which conditions are the most frequent.

Let's take a look at the percentage of property damage and injury collisions for each type of condition:

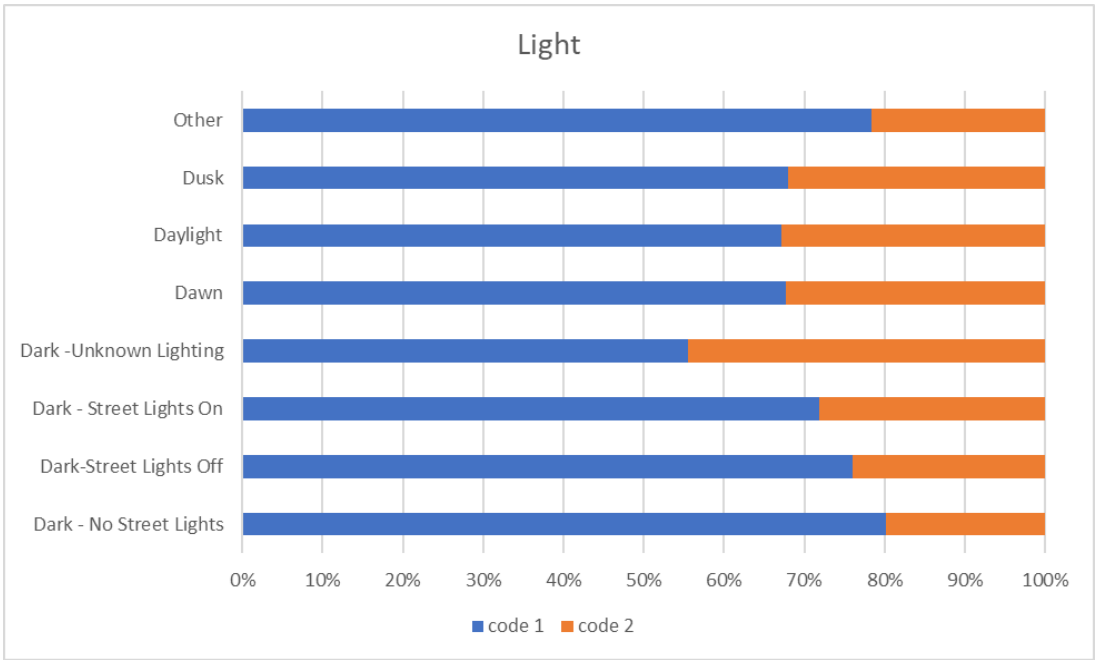

Figure 6. Distribution of data by each light condition type

Consider the distribution of data under the most common lighting conditions: Daylight and Dark-Street Lights On. The distribution roughly corresponds to the distribution of data across classes.

The greatest deviation from the general distribution is observed under the rarest lighting conditions.

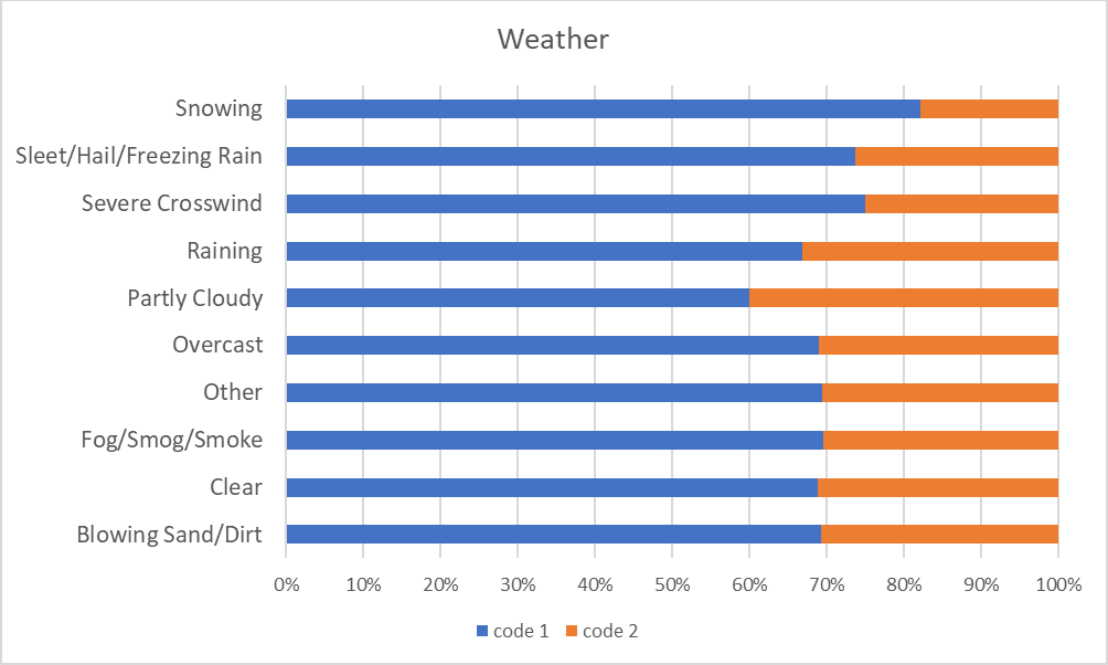

Figure 7. Distribution of data by each weather condition type

When looking at the distribution of the data for the most frequent weather conditions (Clear, Raining and Overcast), it is noticeable that they almost perfectly correspond to the general distribution. The greatest deviations from the general distribution by classes are observed in snowy and partly cloudy weather. This can be explained by the fact that under these conditions there were few accidents.

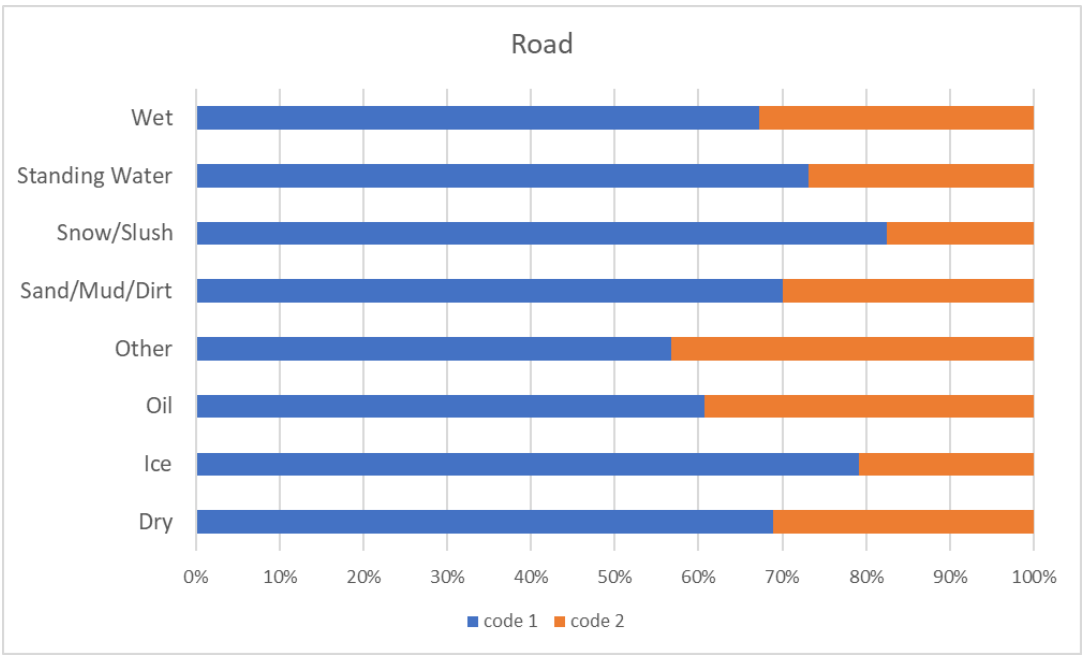

Figure 8. Distribution of data by each road condition type

Under the most common conditions (Dry and Wet), the distribution of the data is very close to the general distribution. In most other road conditions, the distribution deviates from the overall distribution. However, these conditions are very rare.

From this analysis, it was concluded that the distribution for individual road, weather and lighting conditions does not differ from the overall distribution of the data. This is a bad signal as it may mean that there is no clear correlation between the conditions chosen and the severity of the accident.

After that, the average value of the severity of the accident was calculated for each combination of conditions. There were 220 unique combinations in total. Under 21 combinations of conditions, the average accident severity was 2.0. This means that the probability of injury under such conditions was 100%. However, such combinations were rare. Most of them only take one time.

2.4.3 Turning categorical variables into quantitative variables

Most statistical models cannot take in objects or strings as input and for model training only take the numbers as inputs. In our dataset all input values are categorical values. For further analysis, these variables were converted into some form of numeric format. After one-hot-encoding was dataset ready to be used for machine learning algorithms.

2.4.4 Feature selection

Feature selection was carried out within the previous steps. In this step, the corresponding variables were created and assigned values. Since the size and appearance of our dataset after encoding has been changed, the number of parameters in the label set has increased. Now every single condition is a parameter. As a result of this step, a feature set (X) and a label set (Y) were determined.

2.4.5 Feature Scaling

Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled. Distance algorithms like KNN, K-means, and SVM are most affected by the range of features. This is because behind the scenes they are using distances between data points to determine their similarity.

For this reason, a feature scaling was made before training machine learning algorithms. Standardization was chosen as a feature scaling technique

Standardization is scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

3. Predictive Modeling

Our data is ready to be used for machine learning algorithms. As it was found out in the previous steps, our target is composed of discrete values. This is a classification problem. That is, given the dataset with predefined labels, I needed to build a model to be used to predict the class of a new or unknown case. This means that classification algorithms must be used to build models. For this I used the following algorithms:

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

After training the algorithms, it is necessary to evaluate the obtained models in real conditions. For this, I used the cross-validation technique.

Cross validation is one of the techniques used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data. To perform cross validation we need to keep aside a sample/portion of the data on which is not used to train the model, later use this sample for testing/validating.

I chose Train Test Split approach for cross validation. In this approach I randomly split the complete data into training and test sets. Then Perform the model training on the training set and use the test set for validation purpose. I split the data into 75:25.

3.1 K Nearest Neighbor (KNN)

The K-Nearest Neighbors algorithm is a classification algorithm that takes a bunch of labeled points and uses them to learn how to label other points. This algorithm classifies cases based on their similarity to other cases.

First, I determined which value of K must be chosen for our algorithm. To do this, I tried several options and saw at what value of K the algorithm shows the best accuracy on the test set. As a metric, I used the Jaccard similarity score. Once I have chosen the K value, I trained the model.

3.2 Decision Tree

Decision trees are built by splitting the training set into distinct nodes, where one node contains all of one category or most of one category of the data. A decision tree can be constructed by considering the attributes one by one.

3.3 Support Vector Machine

A Support Vector Machine is a supervised algorithm that can classify cases by finding a separator. SVM works by first mapping data to a high dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. Then, a separator is estimated for the data. The data should be transformed in such a way that a separator could be drawn as a hyperplane. I trained the SVM algorithm with 'rbf' kernel.

3.4 Logistic Regression

A characteristic property of logistic regression is that it can predict the probability of sample and we map the cases to a discrete class based on that probability. I trained with C = 0.1 and 'liblinear' solver.

3.5 Model Evaluation Results

After the models of the four classification algorithms have been obtained, I compared them with each other using metrics such as the F1-Score and the Jaccard similarity score. Logistic regression was additionally evaluated using log loss. The results are presented in Table 1. Best performance labeled in green cells.

Table 1. Performance of classification models

Algorithm	Jaccard	F1-score	Log Loss
KNN	0.689777	0.565658	NaN
Decision Tree	0.690991	0.564857	NaN
SVM	0.690869	0.565135	NaN
Logistic Regression	0.691021	0.564816	0.615403

Among the individual models, the Logistic Regression model performed the best (~69.1% accuracy), though the differences between models were very small.

The accuracy of the models does not actually differ from the general distribution of data across classes. That is, if the models always predicted only the more common class, then the accuracy would actually be the same.

This confirmed fears that the severity of road accidents does not actually correlate with the parameters chosen.

4. Conclusions

In this study, I analyzed the conditions that can affect the severity of road traffic accidents in order to create a navigation application that can alert drivers to potential danger. This app can be very useful for drivers to change their route if possible or to drive more carefully and accurately. This will reduce the number of serious road accidents. The data source will be government agencies that are potential sponsors of this application.

An important factor was that the input parameters of the models could be collected in real time and could be applied to different regions. Of all the possible parameters, only three were selected that meet the task at hand: road conditions, lighting conditions and weather conditions. I have developed classification models to predict how severe (property damage only or injury collision) an accident is more likely to occur under the conditions that currently exist along a given route.

I was able to achieve an accuracy of about 69% of the developed classification models. This is no different from class distribution of data. Unfortunately, as a result of preliminary analysis and analysis of final models, it was determined that the severity of road accidents does not actually depend on the parameters chosen.

5. Future directions

It was not possible to create a reliable model from this dataset that could warn drivers about the severity of an accident based on real-time data. Therefore, it is necessary to try to look for other factors that could be obtained in real time and that would have an impact on the severity of accidents.

There are no fatal cases left in this dataset after preprocessing. Such cases must be considered without fail.

In addition, it may be worth trying to take into account the type of car body, because this can significantly affect the consequences of an accident. The user can set this parameter himself in the application, so there will be no problem to collect this information. It can be difficult to find statistics that take this factor into account. But it can improve the model and use it for the final product.