US Accident

Distributed Data Analysis and Mining

Stefano Carrera
Elvis Lleshi
Mattia Gallucci

Contents

1 Introduction

1.1 Dataset Description

The US-Accidents dataset is a comprehensive collection of motor vehicle crash data encompassing 49 states across the United States, with data collection commencing in February 2016. As of March 2023, it comprises approximately 7.7 million accident records.

The dataset aggregates information from multiple data providers, including APIs that stream traffic event data. These sources capture traffic incidents reported by entities such as state and federal Departments of Transportation, law enforcement agencies, traffic cameras, and road-network sensors.

Each accident record includes 46 attributes:

Variable Name	**Description**	**Data Type**
ID	Unique identifier for each accident record.	String
Severity	Numerical value (1 to 4) indicating the severity of the accident.	Integer
Start_Time	Local time when the accident impact started.	Datetime
End_Time	Local time when the accident impact ended.	Datetime
Start_Lat	Latitude of the accident's starting location.	Float
Start_Lng	Longitude of the accident's starting location.	Float
End_Lat	Latitude of the accident's ending location.	Float
End_Lng	Longitude of the accident's ending location.	Float
Distance(mi)	The length of road affected by the accident.	Float
Description	Natural language description of the accident.	String
Number	Street number where the accident occurred.	String
Street	Street name where the accident occurred.	String
City	City where the accident occurred.	String
County	County where the accident occurred.	String
State	State where the accident occurred.	String

Variable Name	Description	Data Type
Zipcode	ZIP code of the accident location.	String
Country	Country where the accident occurred.	String
Timezone	Timezone of the accident location.	String
Airport_Code	Code of the nearest airport-based weather station.	String
Weather_Timestamp	Time when weather observations were recorded.	Datetime
Temperature(F)	Temperature at the time of the accident.	Float
Wind_Chill(F)	Wind chill at the time of the accident.	Float
Humidity(%)	Humidity percentage at the time of the accident.	Float
Pressure(in)	Atmospheric pressure at the time of the accident.	Float
Visibility(mi)	Visibility in miles at the time of the accident.	Float
Wind_Direction	Direction of the wind.	String
Wind_Speed(mph)	Wind speed in miles per hour.	Float
Precipitation(in)	Precipitation level in inches.	Float
Weather_Condition	General weather condition (e.g., rain, snow).	String
Amenity	Whether an amenity was nearby.	Boolean
Bump	Whether a speed bump was nearby.	Boolean
Crossing	Whether a crossing was nearby.	Boolean
Give_Way	Whether a give-way sign was nearby.	Boolean
Junction	Whether a junction was nearby.	Boolean
No_Exit	Whether a no-exit sign was nearby.	Boolean
Railway	Whether a railway was nearby.	Boolean
Roundabout	Whether a roundabout was nearby.	Boolean
Station	Whether a station was nearby.	Boolean
Stop	Whether a stop sign was nearby.	Boolean
Traffic_Calming	Whether a traffic calming measure was nearby.	Boolean
Traffic_Signal	Whether a traffic signal was nearby.	Boolean

Variable Name	Description	Data Type
Turning_Loop	Whether a turning loop was nearby.	Boolean
Sunrise_Sunset	Time of the day (day/night).	String
Civil_Twilight	Civil twilight status.	String
Nautical_Twilight	Nautical twilight status.	String
Astronomical_Twilight	Astronomical twilight status.	String

For more detailed information and access to the dataset, please visit the US-Accidents Dataset Page

1.2 Project Objectives

The main goal of this project is trying to apply some machine learning algorithms as well as some data understanding and processing using python module PySpark for Big Data.

For this reason the tasks, which correspond to the report sections, are the following:

1. **Data Understanding and Preprocessing**: Understand the variables and their values, some basic insight of the data, feature engineering and variable selection/elimination for further sections.
2. **Clustering**: Kmeans implementation on preprocessed data.
3. **Classification**: Decision Tree and Logistic Regression algorithms compared.
4. **Regression**: Multiple Linear Regression and discussion about its effectiveness and interpretaion power.
5. **Conclusion**

2 Data Understanding and Exploration

This section exploits all the variables of the dataset as well as a brief description and some issues solving about NaNs or other inconsistent values

2.1 Position

Variables: `End_Lat`, `End_Lng`
Many NAs values were present in `End_Lat` and `End_Lng` (roughly half of the data: 3,402,762). Our assumption is that `End_` positions were either not recordable or the same as the start positions. In both cases assuming that the End coordinates coincide with the `Start_` could be considered the right way.

NAs Action: Replaced NaN values in `End_Lat` and `End_Lng` with the corresponding values from `Start_Lat` and `Start_Lng`.

2.2 Weather Timestamp

This variable indicates when the weather, reported in `Weather_Condition` variable was recorded. Observations with missing `Weather_Timestamp` also had other important missing features, suggesting a possible problem in the recording of weather data.

NAs Action: Dropped all observations with missing `Weather_Timestamp`.

2.3 City

NAs Action:

- Grouped data by `County` and `City`.
- Computed the most frequent city for each county.
- Replaced missing values in `City` with the most frequent city for the corresponding `County`.

2.4 Timezone

NAs Action:

- Grouped data by `State` and `Timezone`.
- Computed the most frequent timezone for each state.
- Replaced missing values in `Timezone` with the most frequent timezone for the corresponding `State`.

2.5 Twilights

Variables: `Civil_Twilight`, `Astronomical_Twilight`, `Nautical_Twilight`, `Sunrise_Sunset`

These variables are all boolean ones and they reported 'True' if the accident is in daytime, 'False' if it has been reported in nightime. The difference between the variables lays in the way they consider daytime and nightime. For example `Sunrise_Sunset` consider the day as the hours in which the sun is strictly above the horizon, while `Civil_Twilight` consider as 'daytime' all the moments in which the sun is at most 6° under the horizon, for both sunrise and sunset. Other informations about Twilights at SeaChest.

The most interesting variable, considering the purpose and the nature of this dataset is the `Civil_Twilight` because it suddivision of Day/Night is a perfect partition of the hours where it is possible to see distinctly shapes and objects without any artificial light help, and hours where this is not possible. For this reason we decided to keep only `Civil_Twilight` and to discard the other twilight variables.

NAs Action:

- Grouped data by hour and twilight type.
- Computed the mode for each hour.
- Replaced missing values in `Civil_Twilight` with the mode for the corresponding hour.

2.6 Numerical Attributes

2.6.1 Wind Chill (F)

The variable is quite correlated with `Temperature(F)`, as it's possible to see in figure 1.

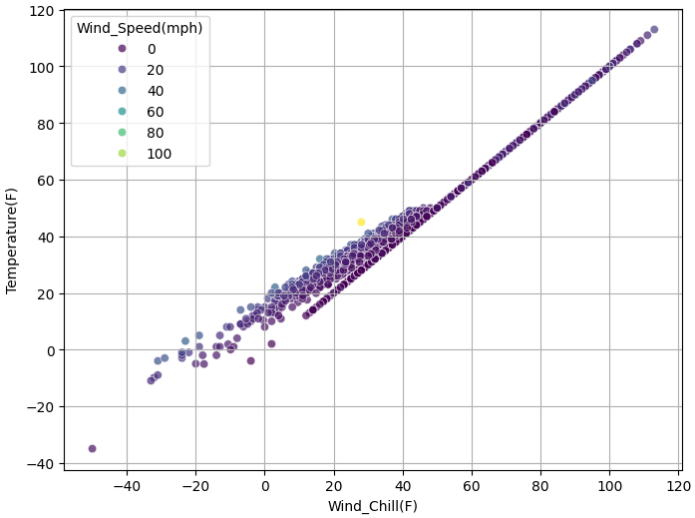

Figure 1: Scatter plot for `Wind_Chill(F)` vs `Temperature(F)`

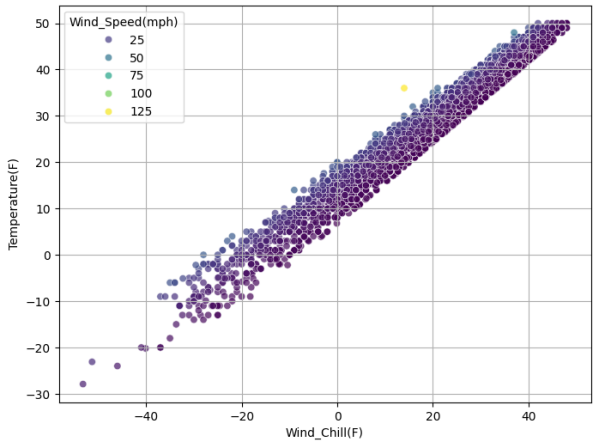

Figure 2: Scatter plot for `Wind_Chill(F)` vs `Temperature(F)` for accidents with `Wind_Chill(F)` smaller than `Temperature(F)`

It's seems like there's an hidden partition in data, maybe in how they're recorded (some recording station could not have the possibility to record the chill of the wind and so automatically assign

it to the temperature of the air). We didn't managed to retrive the origin of this anomaly but few considerations are possible.

- It's clear how there's a mass of points which follow the exact same values as `Temperature(F)`, while there's another mass, especially when the temperature is under 50 F, that follow a more irregular relation. This points represents accidents where the air temperature was above wind temperature, as figure 2 shows.
- Furthermore we managed to identify a sort of relation with `Wind_Speed(mph)`. In fact it seems like at same levels of air temperature, wind temperature is lower when the speed is higher compared to situations where the speed is lower.
- We also figured out how when the `Wind_Speed(mph)` was under 3, `Wind_Chill` and `Temperature(F)` have a correlation near 1.

NAs Action: Replaced missing values in `Wind_Chill(F)` with values from `Temperature(F)` when `Wind_Speed(mph)` is smaller than 3 or when `Temperature(F)` is above 50 °F.

At this point, with some few remaining Null values, we computed the correlation again and its value was near 0.99. For this reason we decided not to find out new ways to fill remaining Nulls but to directly discard `Wind_Chill(F)` variable.

2.6.2 Precipitation (in)

Observed that missing values occurred mostly during non-precipitation weather conditions. Since they are a considerable proportion of data (roughly 2,500,000) we assume they represent 'No-precipitation'

NAs Action: Replaced NAs values with 0 and checked and corrected negative precipitation values by setting them to 0.

2.6.3 Humidity (%)

NAs Action:

- Grouped data by month and weather condition.
- Computed the average of each group.
- Replaced missing values in `Humidity(%)` with the average for the corresponding month and weather.

2.6.4 Wind Speed (mph)

NAs Action:

- Grouped data by weather condition.
- Computed the average `Wind_Speed(mph)` of each group.
- Replaced missing values in `Wind_Speed(mph)` with the average for the corresponding weather.

2.6.5 Visibility (mi)

NAs Action:

- Grouped data by weather condition.
- Computed the average `Visibility(mi)` of each group.
- Replaced missing values in `Visibility(mi)` with the average for the corresponding weather.

2.6.6 Pressure (in)

NAs Action:

- Grouped data by weather condition.
- Computed the average `Pressure(in)` of each group.
- Replaced missing values in `Pressure(in)` with the average for the corresponding weather.

2.7 Weather Condition

This categorical variable come with 140 different types of value. We decided to shrink as much as possible this number to facilitate further models and analysis. First we noticed that some values consist in an aggregation of two other values separated by a '/' (ex. `Snow / Storm`). From all these values only the first one before the '/' has been extracted and incorporated with the existing one, if any.

With this oparation we managed to reduce the number of values from 140 to 92. Then we managed to reduce even more the categories by aggregating similar weather conditions (e.g., `Rain`, `Light_Rain`, `Showers`,..., all under the category `Rain`)

NAs Action: Observations with missing `Weather_Condition` were discarded.

2.8 New Variables

- `Traffic_Duration`**:** Calculated as the difference between `End_Time` and `Start_Time`, in minutes.
- **Time Features:** Extracted `Hour`, `Day_of_Week`, and `Day_of_Year` from `Start_Time`.
- `Weekend_day`**:** Created a binary variable indicating whether the day was a weekend.

2.9 Transformations

Some Numeric Variables has been Log-transformed due to their skewness in distribution (3, 4), also some clipping has been made for anomalous values. **Note:** log-transformed and original variables have been both kept in the dataset for further analysis.

- **Traffic Duration:** Clipped at 720 minutes and log-transformed.
- **Wind Speed:** Clipped at 200 mph and log-transformed.

- **Precipitation:** Clipped at 5 inches and log-transformed.
- **Visibility:** Log-transformed.
- **Distance:** Log-transformed.

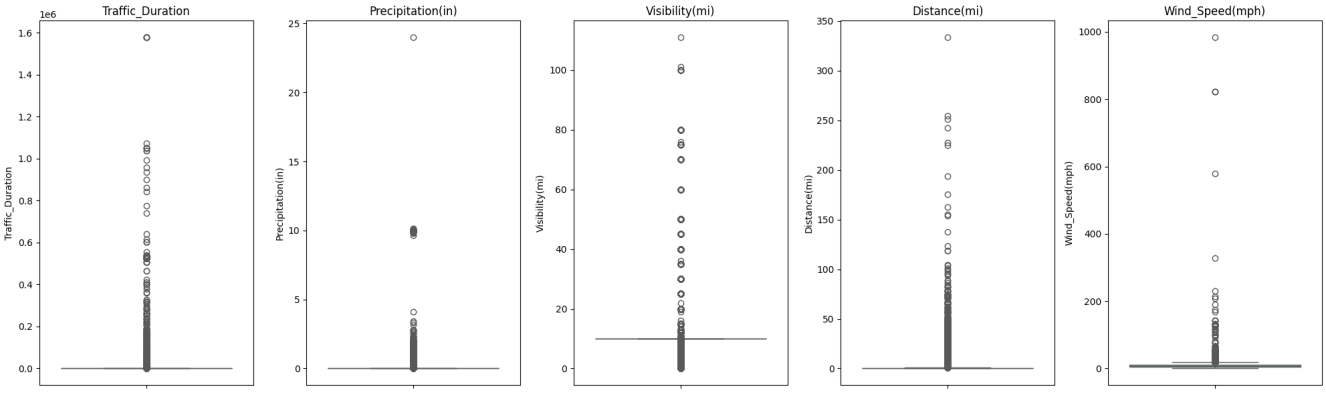

Figure 3: Original numeric variables

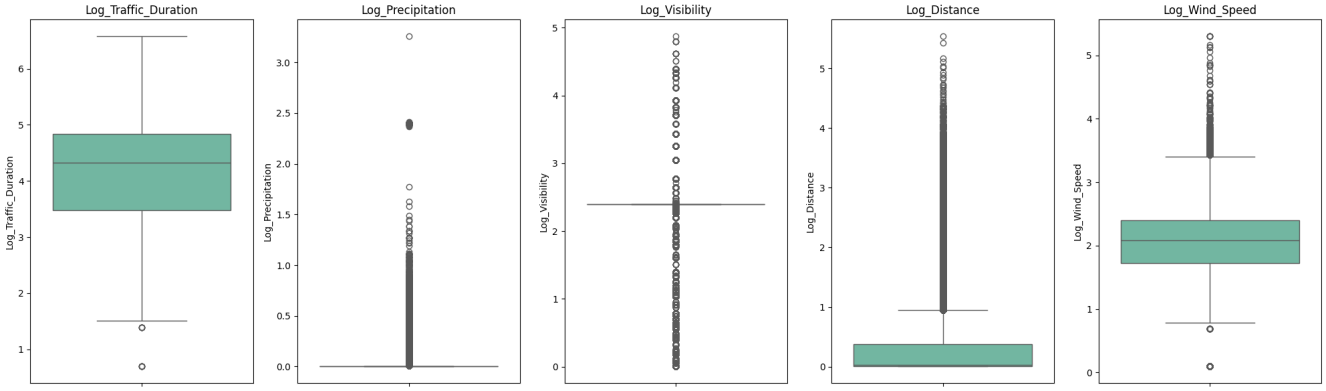

Figure 4: Log-transformed numeric variables

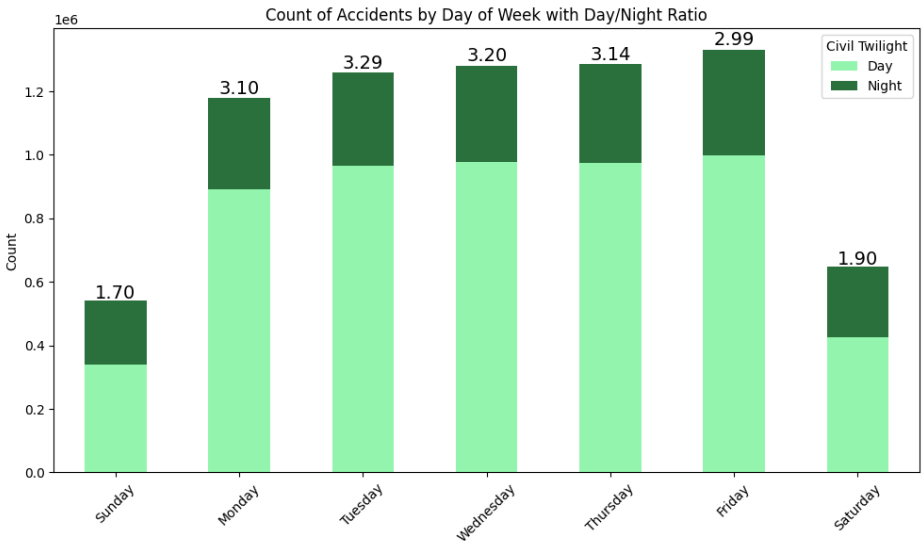

Figure 5: Accidents per day of week with Day/Night accidents ratio

3 Clustering

The primary goal was to determine an optimal number of clusters using the KMeans algorithm, with a focus on numeric features derived from the dataset. The elbow method was employed to evaluate the clustering performance across various values of k.

3.1 Data Preprocessing

The data preparation involved some preliminary steps:

- Identification of numeric columns, excluding specific features such as the original ones where the logarithmic transformation was present and some time variables such as `Hour`, `Day_of_Week`, `Day_of_Year`, `Year` and `Month`.
- Conversion of some categorical features in numerical ones, such as `Civil_Twilight`, which assume value 1 for 'Day' and 0 for 'Night'
- Standardization of numerical features (not the binary ones) to ensure all features were on a similar scale.

3.2 Methodology

The KMeans algorithm was trained on the standardized data for different values of k ranging from 4 to 15. The clustering cost, defined as the within-cluster sum of squared errors (WSSE), was calculated for each value of k. The goal was to identify the point at which the decrease in cost became negligible, indicating the optimal number of clusters.

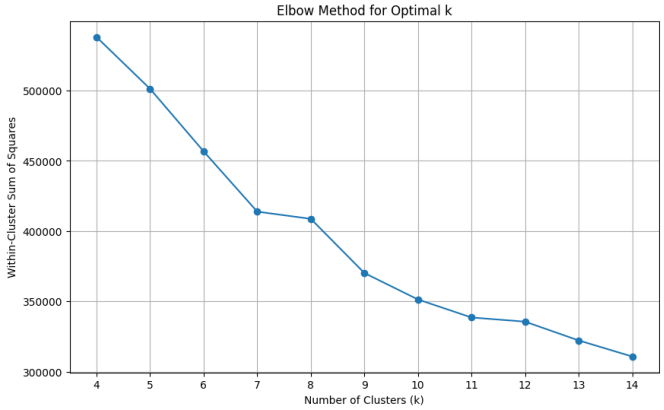

Figure 6: Elbow method for optimal number of clusters

Figure 6 indicates a significant decrease in cost from $k = 4$ to $k = 7$, followed by a stabilization. This suggests that $k = 7$ could be a suitable choice for the number of clusters. Even 11 could be a good candidate for the same stabilization achieved after and because 11 clusters could capture more details.

3.3 Results

Figure 7 shows how many records have been assigned to each cluster while figure 8 summarizes the cluster centroids using a heatmap for centroid feature values.

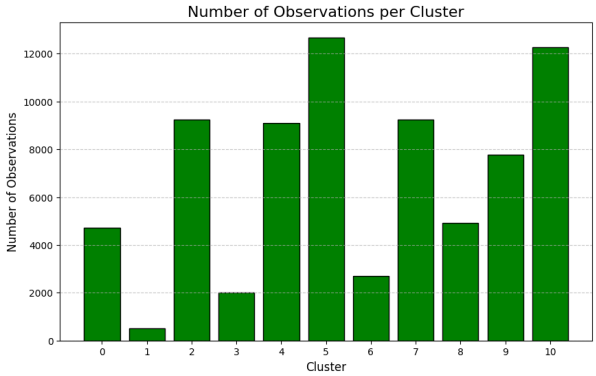

Figure 7: Number of observations per cluster

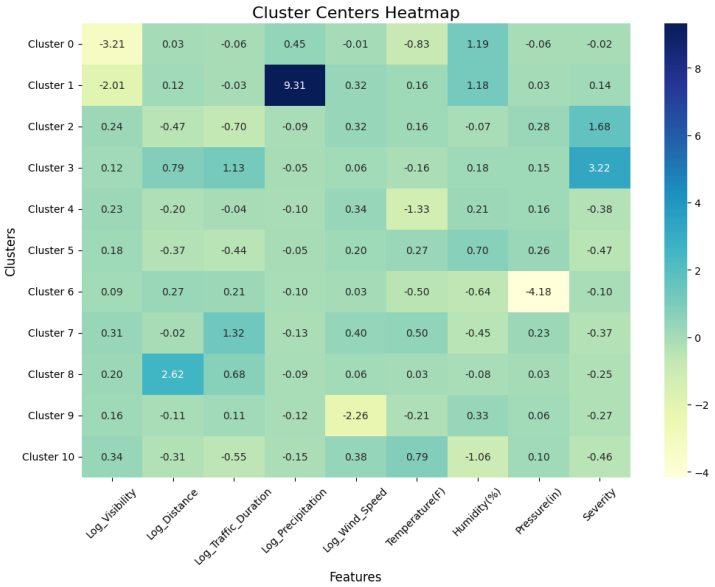

Figure 8: Heatmap for centroids features

The results show some peculiar clusters such as:

- **Cluster 1**: composed by accidents where the precipitations were extreme and the visibility was terrible. It is also the cluster with less observations.
- **Cluster 3**: composed by most severe accidents. This is probably the most interesting one because shows that severe accidents are clustered together with observations with high traffic duration and relatively high distance, underlining a possible relations between these variables.
- **Cluster 8**: composed by accidents which affected on average the most ditance on the road, not surprisingly this cluster contains on average also high level of traffic duration but normal levels for severity.

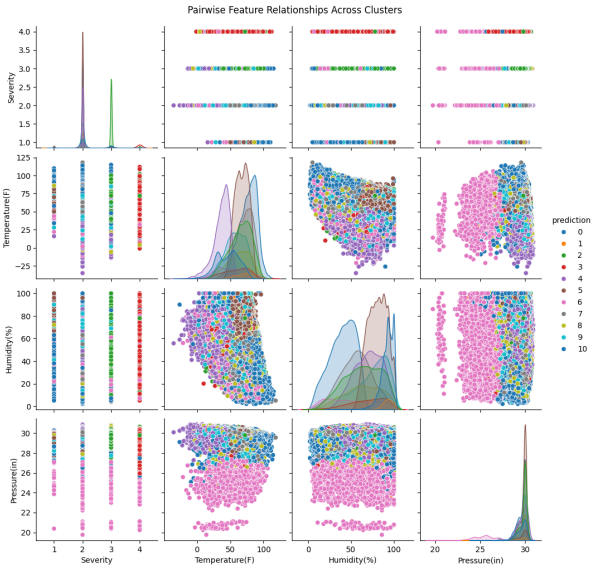

Figure 9: Clusters summary for some features

4 Classification

The primary goal was to predict accident severity (`Severity`) using a decision tree classifier and the logistic regression evaluating performance across different hyperparameter combinations to optimize model accuracy. Two tasks are defined for each model:

1. Predict only the '4' class of `Severity` against all the others (Unbalanced problem).
2. Predict class '3' or '4' (Severe) against '1' and '2' (Not Severe)

Both cases represent a binary classification problem.

4.1 Data Preparation

The data preparation involved the following steps:

- Identification of relevant features for prediction.
- Conversion of categorical variables into numerical format using One Hot Encoding.
- Splitting the dataset into training, validation and testing subsets.

For task 2 a random oversampling combined with random undersampling have been made trying to improve the accuracy of the models but in both cases the accuracy get worse than the model run on unbalanced data.

4.2 Decision Tree

A Decision Tree Classifier was employed for classification tasks. The models were run based on the following key parameters:

- `maxDepth` - Maximum depth of the decision tree.
- `maxBins` - Maximum number of bins for splitting features.
- `minInstancesPerNode` - Minimum number of instances required in a node for it to split.
- `impurity` - Criterion for splitting (`gini` or `entropy`).

An iterative Random Search approach was used to optimize these hyperparameters by evaluating the accuracy of the model on the validation dataset. Final accuracy was then calculated from test set, running best hyperparameters combination model.

Figure 11 shows the features importance extracted from the Decsion Tree model trained for task 2 with balanced data. It's possible to notice how distance and traffic duration play a key role in determining if an accident will be severe or not. Also Pressure play a good role: the reason could be that pressure is a sort of summary indicator of weather condition, since high-pressure zones are characterized by good weather while low-pressure one by bad weather.

4.3 Logistic Regression

Standard logistic regression model has been construct for task 1 and 2. Figure 10 shows the coefficient magnitude of the best obtained (in terms of Recall) logistic regression model. **Note**: the accuracy of this model is worse than the trivial classifier, even if this model could better detect 'Severe' accidents.

It's possible to see how `Bump`, `Weather_Condition_Ice_Pellets` and `Weather_Condition_Dust` are the most influent variables for predicting the fact an accident will be severe or not. For example, the presence of a bump decrease the probability to observe a severe accident *ceteris paribus* by $1 - e^{-148.252} = 99.999\%$, meaning that almost surely in this condition the accident will not be severe. Another example: if the weather condition is `Overcast`, then the probability of observing a severe accident *ceteris paribus* increase by $e^8 = 2980$ times.

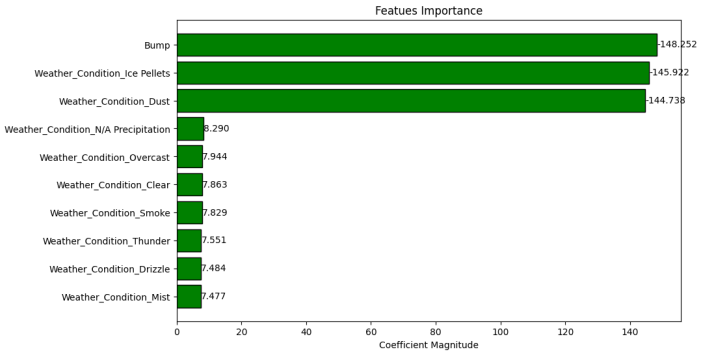

Figure 10: Coefficients magnitude for logistic regression model obtained from balanced data for task 2

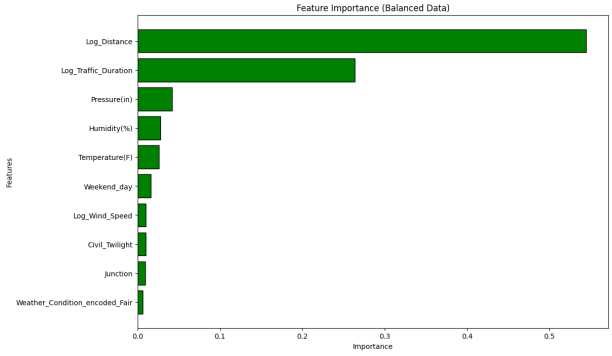

Figure 11: Features Importance for Decision Tree model obtained from balanced data for task 2

4.3.1 Results

Model	Task	Balancing	Recall	Accuracy	Trivial Accuracy
Decision Tree	1	Normal	0.27	0.82	0.81
Decision Tree	2	Unbalanced	0.07	0.97	0.97
Decision Tree	2	Balanced	0.61	0.90	0.97
Logistic Regression	1	Normal	0.05	0.73	0.81
Logistic Regression	2	Unbalanced	0.05	0.76	0.97
Logistic Regression	2	Balanced	0.30	0.77	0.97

Table 2: Classification Metrics

From this table we can retrive several considerations:

1. Best model in terms of accuracy and recall for Task 1 is surely Decision Tree one.
2. Best model in terms of accuracy for Task 2 is the Decision Tree trained with unbalanced data, though the accuracy is not better than the trivial model. Although if we're searching for a model with higher Recall (more false allarm but better catching of Severe accident) we can use the Decision Tree model trained on balanced data, sacrifying some levels of accuracy.

5 Regression

Lastly we dive into Regression with Linear Regression. The target variable identified in this section is `Traffic_Duration`, even if `Log_Traffic_Duration` has been implemented as target variable for its better distribution and also because a logarithm-transformed variable could help in some interpretation of the coefficients. The actual model described in this section is so a Log-Linear Regression model. We've tried several combination of regressor but eventually in this section has been reported the model with the highest R^2 metric.

5.1 Data Preparation

The data preparation involved the following steps:

1. Conversion of categorical variables into numerical format using One Hot Encoding.
2. Splitting the dataset into training, validation and testing subsets
3. Identification of relevant features for regression (trial and errors).

5.2 Final Model and Results

The final model is a Log-Linear regression one with `Log_Traffic_Duration` as dependent variable and the following as regressors: `'Temperature(F)', 'Humidity(%)', 'Pressure(in)', 'Precipitation(in)', 'Amenity', 'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway', 'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal', 'Civil_Twilight', 'Log_Wind_Speed', 'Log_Precipitation', 'Log_Distance', 'Log_Visibility', 'Weekend_day', 'Hour' 'Month', 'Weather_Condition', 'Severity', 'Day_of_Week'`

This model achieved an R^2 metric of 0.15. The maximum value this metric could take is 1, describing a perfect model, while the minimum is 0, describing an useless one.

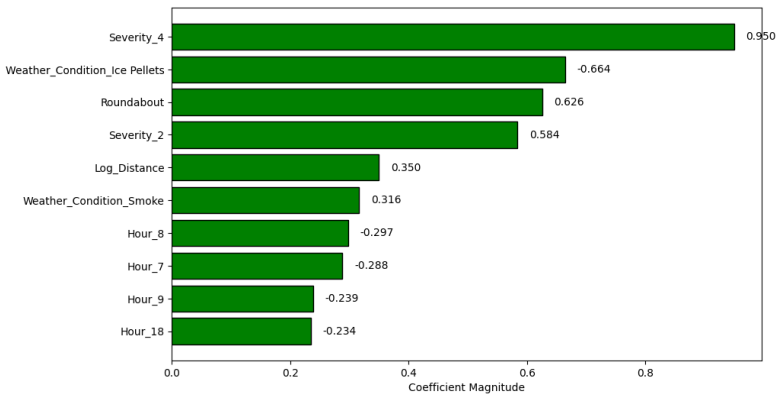

Figure 12: Linear Regression Coefficients values

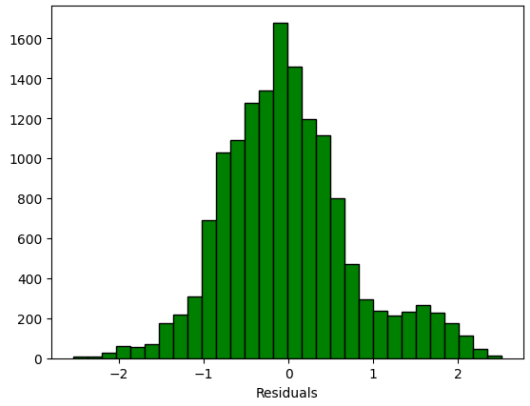

Figure 13: Residuals distribution

Figure 12 shows the most important regressor based on their coefficient values. Let's try to give an interpretation of some of these:

- **Severity_4** = 0.950: the fact that the accident is of severity 4 lead to an increase of 95% of traffic duration *ceteris paribus*

- **Roundabout** = 0.626: the fact that there's a roundabout in the proximity of the accident lead to an increase of 63% of traffic duration *ceteris paribus*
- **Log_Distance** = 0.35: a 1% increase in Distance will lead to an increase of 0.35% in traffic duration *ceteris paribus*