COURSERA FINAL CAPSTONE

PREDICTING CAR ACCIDENT SEVERITY

September 6, 2020

Kerman Sanjuan Malaxechevarria

Index

Chapter 1

Introduction

1.1 Background

Everyone knows the danger of the roads. Just in USA there are 6 million car accidents per year, involving more than one million deaths. Car accidents are considered one of the biggest causes of mortality in the world.

Moreover, the accidents have more consequences, due to the fact that road traffic accidents lead to a large number of fatal incapacitating injuries, the consequences of these accidents are fundamentally reflected in the social sphere. This concerns job losses and the related financial hardships, loss of amenity and a fatal impact on the functioning of the whole family. We should not forget that the psychological impact of the consequences of road traffic accidents do not only to affect the direct participants, but also their families. The costs associated with road traffic accidents are shouldered by the whole society.

Accident prediction is important for optimizing public transportation, enabling safer routes, and cost-effectively improving the transportation infrastructure, all in order to make the roads safer. Given its significance, accident analysis and prediction has been a topic of research in the past few decades.

In short it would be interesting to be able to predict the dangerousness of a road/place in case of accident, simply by knowing the characteristics of it, right?

1.2 Problem

With the obtained data, project aims to predict the severity of an accident, depending on the features of it. For example, the numbers of cars which are involved, speed, road condition, weather...etc

1.3 INTEREST

The main audience of this report could be any organization or government, obviously the ability to predict the consequence of an accident is something that generates a lot of interest. The applications of this predictions could be used for example, for real-time accident prediction, studying accident hot-spot locations, casualty analysis and extracting cause and effect rules to predict accidents, or studying the impact of precipitation or other environmental stimuli on accident occurrence.

Chapter 2

Data acquisition and cleaning

2.1 Acquisition of data

All these data is obtained from Cornell University, specifically from Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath from "A Countrywide Traffic Accident Dataset.", and "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights."

2.2 About the data

First, I decided to use the data which Coursera give it to me, but the lack of variety when talking about the severity of an accident force myself to choose another source. The obtained data contains information about traffic accidents, covering the 49 states of United States of America. This source has been continuously obtained since 2016, so we're working with actual data. Currently, there 3.5 million rows (we will use just 175k).Each row contains information about a single accident,with all these data, we should be able to create a prediction model and discover some interesting information about this kind of events.

For explaining the data, I will use the resource that this dataset provides. There are 45 different data features, which are distributed as follows

Type	Description
Amenity	Refers to particular places such as restaurant, library, college, bar, etc.
Bump	Refers to speed bump or hump to reduce the speed.
Crossing	Refers to any crossing across roads for pedestrians, cyclists, etc.
Give-way	A sign on road which shows priority of passing.
Junction	Refers to any highway ramp, exit, or entrance.
No-exit	Indicates there is no possibility to travel further by any transport mode along a formal path or route.
Railway	Indicates the presence of railways.
Roundabout	Refers to a circular road junction.
Station	Refers to public transportation station (bus, metro, etc.).
Stop	Refers to stop sign.
Traffic Calming	Refers to any means for slowing down traffic speed.
Traffic Signal	Refers to traffic signal on intersections.
Turning Loop	Indicates a widened area of a highway with a non-traversable island for turning around.

As we see, there's a POI table, where all Points Of Interest are defined, these are the main descriptions:

Total Attributes	45
Traffic Attributes (10)	id, source, TMC [23], severity, start_time, end_time, start_point, end_point, distance, and description
Address Attributes (8)	number, street, side (left/right), city, county, state, zip-code, country
Weather Attributes (10)	time, temperature, wind_chill, humidity, pressure, visibility, wind_direction, wind_speed, precipitation, and condition (e.g., rain, snow, etc.)
POI Attributes (13)	All cases in Table 1
Period-of-Day (4)	Sunrise/Sunset, Civil Twilight, Nautical Twilight, and Astronomical Twilight
Total Accidents	2,243,939
# MapQuest Accidents	1,702,565 (75.9%)
# Bing Accidents	516,762 (23%)
# Reported by Both	24,612 (1.1%)
Top States	California (485K), Texas (238K), Florida (177K), North Carolina (109K), New York (106K)

Most of these data is categorical, so it will be dropt out of the table for a correct model developing. By the way, there some irrelevant features which will be erased too.

2.3 DATA CLEANING

Now we have talked about the data, is time to start working with it.First of all, we have to clean the data, but, what it means? We will see it

2.3.1 Removing unnecessary information

So, as we see before, there's data that we don't need, so we will get rid of it. Which are examples of irrelevant data?

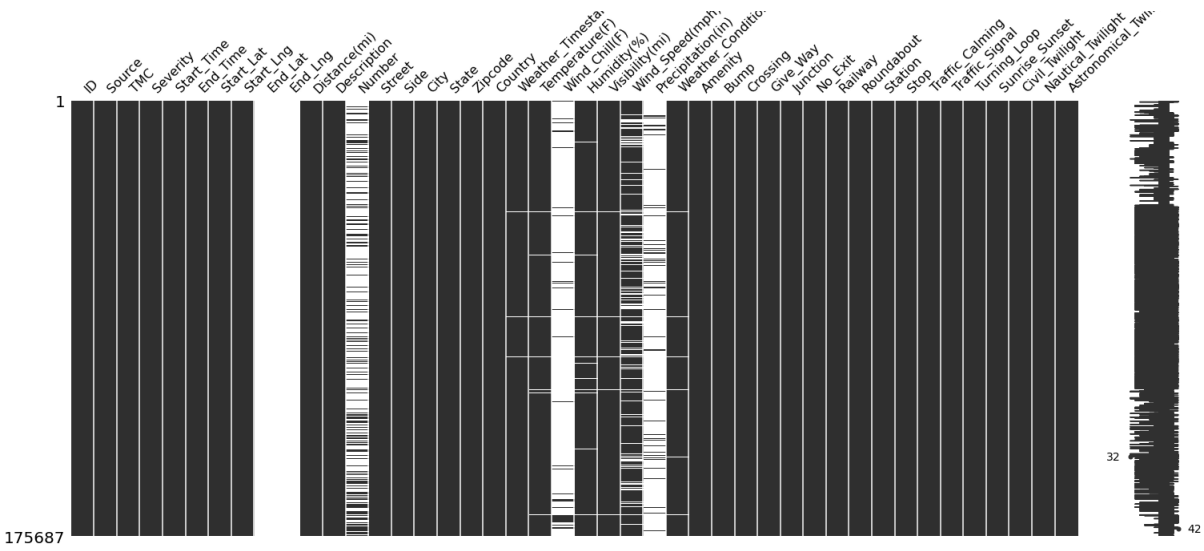
- ID
- Pressure:
- All address attributes: number, street, side (left/right), city, county, state, zip-code...
- Wind Direction
- Timezone
- Airport Code
- Nautical Twilight
- Astronomical Twilight

The Twilight we will use will be the Civil one, which is the period when enough natural light remains that artificial light is not needed. As you see, all of this information has no importance when talking about car accidents, so we erased it. There's still more data, but know its time to drop the data with a bunch of missing values. This dataset has it's own index, but we will remove it to let Pandas library use it's own.

2.3.2 Deleting the lack of data

In this type of data-set, is normal to have some missing values, but in some cases, this can bring problems. In the cases where the missing values are above the 15% of all the samples, we will drop it. So, lets see how much missing values we have on a visual way.

Figure 2.1: Missing values before cleaning.



As we see, there are some features with more than 15% of missing values, these are the features we will drop:

- Start Time
- End Time
- Latitude
- Longitude
- Description
- Weather Timestamp
- Wind chill
- Precipitation
- Number
- Source

Now, We've gone from 47 to 25 features, but there are still features to drop.

2.3.3 Deleting some categorical values

A categorical variable (sometimes called a nominal variable) is one that has two or more categories, but there is no intrinsic ordering to the categories. For example, gender is a categorical variable having two categories (male and female) and there is no intrinsic ordering to the categories. This features are hard to compute, so we're going to delete the ones with more than 5 different categories.

2.4 DATA WRANGLING

After all this cleaning, we ended up having just 25 features out of 47, which is a pretty god selection. Now it's time to replace all the missing values and transforming categorical values to discrete ones.

2.4.1 Replace missing values

First, we will replace the missing values with the mean of the feature, this is the best way to avoid having a negative impact on the data. The missing values appear with the word NaN, as Pandas default. In the case of categorical, we will use the most common

2.4.2 Transformation of categorical values

All the boolean values are changed with 1 or 0 values, depending on True or False. The remaining categorical features are the following ones: Weather Condition and Sunrise Sunset. For solving this problem, we will use some one-hot encoding, this is a method to quantify categorical data. In short, this method produces a vector with length equal to the number of categories in the data set. If a data point belongs to the -ith category then components of this vector are assigned the value 0 except for the ith component, which is assigned a value of 1. In this way one can keep track of the categories in a numerically meaningful way.

Figure 2.2: Hot encoding example.

SAFETY-LEVEL (TEXT)	SAFETY-LEVEL (None)	SAFETY-LEVEL (Low)	SAFETY-LEVEL (Medium)	SAFETY-LEVEL (High)	SAFETY-LEVEL (Very High)
None	1	0	0	0	0
Low	0	1	0	0	0
Medium	0	0	1	0	0
High	0	0	0	1	0
Very-High	0	0	0	0	1

2.5 FEATURE SELECTION

Now, we will see the relation between this features and severity. After printing it, the remaining features have some important relation with severity. The reason is that there are a lot of factors behind a car accident. So we will not delete any more features.

2.6 FINAL DATAFRAME

After all this cleaning, we get this dataframe, with 25 features, this will our working material, now its time to start crushing the data and taking some conclusions about it.

Figure 2.3: Final Dataframe

	TMC	Severity	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Visibility(mi)	Amenity	Bump	...	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop	Sunrise_Sunset_10.0	Sunrise_Sunset_Day	Sunrise_Sunset_Night	Sunrise_Sunset_nan
0	201.0	3	39.865147	-84.058723	0.01	36.9	91.0	10.0	0	0	...	0	0	0	0	0	0	0	0	1	0
1	201.0	2	39.928059	-82.831184	0.01	37.9	100.0	10.0	0	0	...	0	0	0	0	0	0	0	0	1	0
2	201.0	2	39.063148	-84.032608	0.01	36.0	100.0	10.0	0	0	...	0	0	0	0	1	0	0	0	1	0
3	201.0	3	39.747753	-84.205582	0.01	35.1	96.0	9.0	0	0	...	0	0	0	0	0	0	0	0	1	0
4	201.0	2	39.627781	-84.188354	0.01	36.0	89.0	6.0	0	0	...	0	0	0	0	1	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
175682	201.0	2	38.631863	-90.383659	0.00	82.0	65.0	10.0	0	0	...	0	0	0	0	0	0	0	1	0	0
175683	201.0	2	41.835400	-88.010834	0.00	73.0	90.0	10.0	0	0	...	0	0	0	0	1	0	0	1	0	0
175684	229.0	2	41.995152	-88.165894	0.00	75.0	82.0	10.0	0	0	...	0	0	0	0	1	0	0	1	0	0
175685	229.0	2	41.844688	-87.957802	0.00	73.9	85.0	10.0	0	0	...	0	0	0	0	0	0	0	1	0	0
175686	229.0	3	41.760910	-87.892639	0.00	73.9	85.0	10.0	0	0	...	0	0	0	0	0	0	0	1	0	0

175687 rows × 25 columns

Chapter 3

Exploratory Data Analysis (EDA)

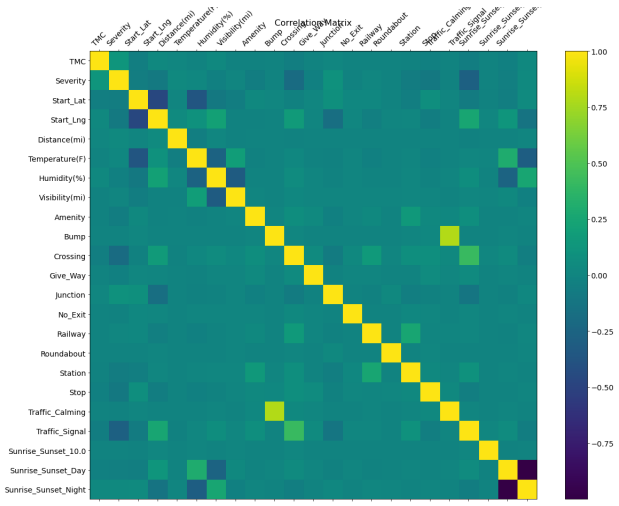
3.1 Descriptive Statistical Analysis

After cleaning all the data and selecting the features we will use, it's time to make some visualizations of data and take some information before making the formal prediction model. Describing the data can give us some interesting statistical information about the data.

3.2 Correlation and Causation

The conclusions taken by describing the data are not enough, so we need to display the correlation of features to be more learn more about the most valuable data.

Figure 3.1: Correlation matrix



The correlation is a value in range of -1 and 1, closer values to the limits involves stronger correlation. But we have to be carefull, correlation doesn't mean causation, in other words, causation explicitly applies to cases where action causes outcome B. On the other hand, correlation is simply a relationship. Action A relates to Action B- but one event doesn't necessarily cause the other event to happen.

3.3 VALUE COUNTS

Value-counts is a good way of understanding how many units of each characteristic/-variable we have. We can apply the "value-counts" method every column . Don't forget the method "value-counts" only works on Pandas series, not Pandas Dataframes. As a result, we only include one bracket not two brackets.

3.4 GROUPING

The "groupby" method groups data by different categories. The data is grouped based on one or several variables and analysis is performed on the individual groups. The grouped data is much easier to visualize when it is made into a pivot table. A pivot table is like an Excel spreadsheet, with one variable along the column and another along the row. We can convert the dataframe to a pivot table using the method "pivot " to create a pivot table from the groups.

3.5 ANOVA

The Analysis of Variance (ANOVA) is a statistical method used to test whether there are significant differences between the means of two or more groups. ANOVA returns two parameters:

- F-test score: ANOVA assumes the means of all groups are the same, calculates how much the actual means deviate from the assumption, and reports it as the F-test score. A larger score means there is a larger difference between the means.
- P-value: P-value tells how statistically significant is our calculated score value.

If our severity variable is strongly correlated with some variables we are analyzing, expect ANOVA to return a sizeable F-test score and a small p-value.

Chapter 4

Model Development

We will try three different classification models to perform our prediction model, and then we will choose the one which best average score as the definitive one.

4.1 K-Nearest Neighbors

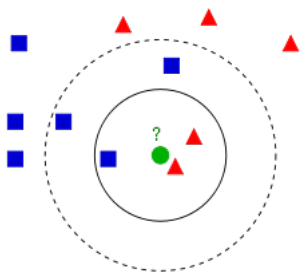
4.1.1 Definition

A k-nearest-neighbor is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it.

An algorithm, looking at one point on a grid, trying to determine if a point is in group A or B (in our particular case, A, B, C and, looks at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa.

The k-nearest-neighbor is an example of a "lazy learner" algorithm because it does not generate a model of the data set beforehand. The only calculations it makes are when it is asked to poll the data point's neighbors. This makes k-nn very easy to implement for data mining.

Figure 4.1: Graphical Example

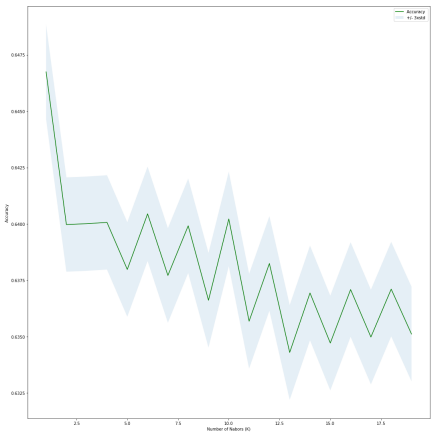


4.1.2 Implementation

Using the Scikit-Library, we will implement the k-nn algorithm, just with one parameter, the number of neighbors we want to take in account when making the label assignation. In other words, how much points we will check before determining the output of the prediction.

To make the best decission, we will plot the scores depending on the value of the number of neighbors, and we will use the one with the best score on the test set.

Figure 4.2: Selection of the number of neighbors

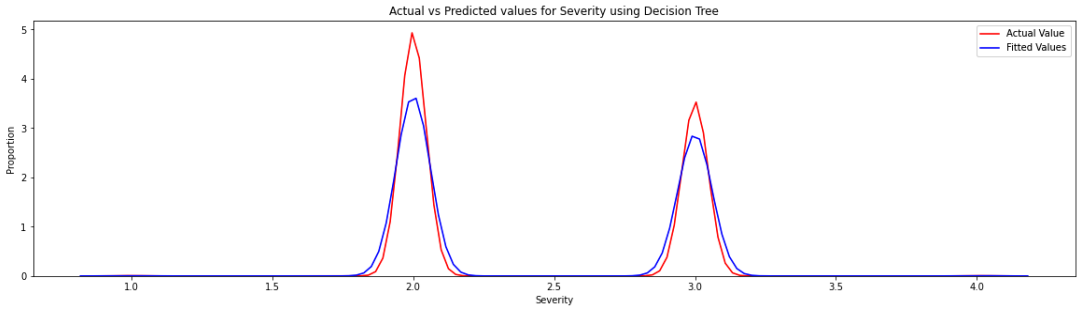


As we see, the best score was performed with just one neighbor, so we will train and fit the model with this value

4.1.3 Model Score

With all parametrization done, now its time to train the model. Before spliting the data into a train and test set, we fit the model, and finally, we print the score.

Figure 4.3: Relation between predicted values and the actual ones



Now, using the R2, F, precission and recall score, we will evaluate the model.

Accuracy Score	R2 Score	F1 Score
0.6467	-0.4416	0.6477

Table 4.1: k-Nearest Neighbors Score

As we see, this isn't the best score, that's why kNN isn't the best model to fit this particular case , so we will try another classification models and see their performance.

4.2 DECISION TREE

4.2.1 Definition

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Figure 4.4: Graphical Example

4.2.2 Implementation

Using the Scikit-Library, we will implement a decission tree, just with two parameters, criterion, which will be Entropy and the length of the tree. Entropy,In the most layman terms, Entropy is nothing but the measure of disorder, in other words,

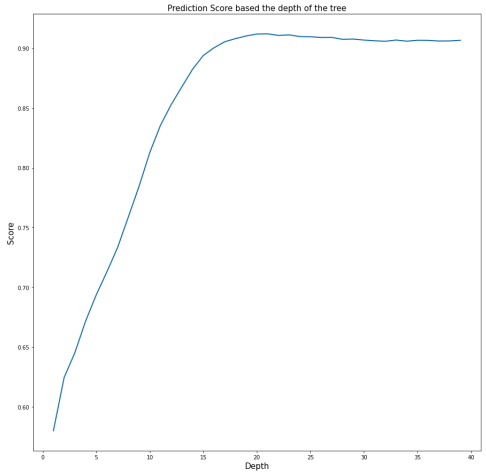
Figure 4.5: Entropy Formula

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

There is an other important parameter, the lenght, this length will be the maximum number of layers of the tree. Depending on this number, the score and the precision of the model will change, so we will have to decide the number with the best performance on the test set.

As we see, the depth of the tree with the best score is 21. So this will be one we will use to develop the prediction model.

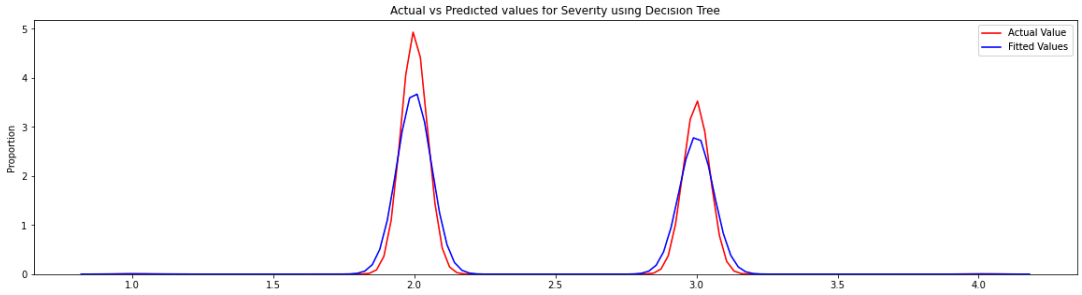
Figure 4.6: Depth selection



4.2.3 Model Score

With all parametrization done, now its time to train the model. Before spliting the data into a train and test set, we fit the model, and finally, we print the score.

Figure 4.7: Relation between predicted values and the actual ones



Now, using the R2, F, precission and recall score, we will evaluate the model.

Accuracy Score	R2 Score	F1 Score
0.9116	0.6342	0.9117

Table 4.2: Decision tree score

4.3 SUPPORT VECTOR MACHINE (SVM)

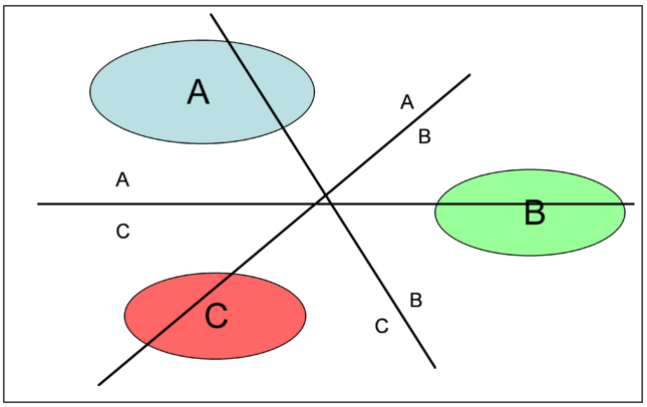
4.3.1 Definition

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. The algorithms draw lines (hyperplanes) to separate the groups according to patterns.

An SVM builds a learning model that assigns new examples to one group or another. By these functions, SVMs are called a non-probabilistic, binary linear classifier. In probabilistic classification settings, SVMs can use methods such as Platt Scaling.

Like other supervised learning machines, an SVM requires labeled data to be trained. Groups of materials are labeled for classification. Training materials for SVMs are classified separately in different points in space and organized into clearly separated groups. After processing numerous training examples, SVMs can perform unsupervised learning. The algorithms will try to achieve the best separation of data with the boundary around the hyperplane being maximized and even between both sides.

Figure 4.8: Graphical Example



4.3.2 Implementation

The SVM algorithm offers a choice of kernel functions for performing its processing. Basically, mapping data into a higher dimensional space is called kernelling. The mathematical function used for the transformation is known as the kernel function, and can be of different types, such as:

1. Linear
2. Polynomial
3. Radial basis function (RBF)
4. Sigmoid

Each of these functions has its characteristics, its pros and cons, and its equation, but as there's no easy way of knowing which function performs best with any given dataset, we usually choose different functions in turn and compare the results. Let's just use the default as RBF (Radial Basis Function) as starting point.

4.3.3 Model Score

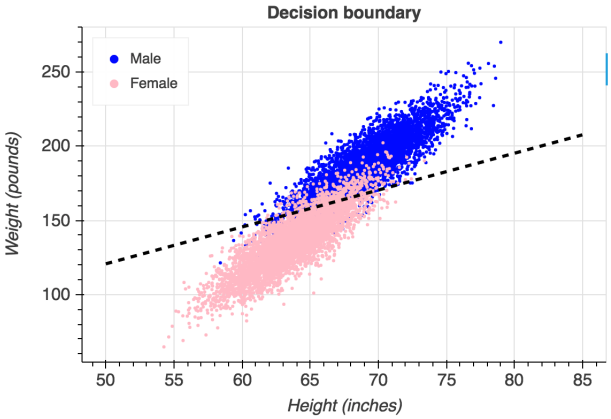
Due to the size of the dataset, the model is to complex and hard to compute, so SVM is not a proper model for this dataset.

4.4 LOGISTIC REGRESSION

4.4.1 Definition

Multinomial logistic regression is a form of logistic regression used to predict a target variable have more than 2 classes. It is a modification of logistic regression using the softmax function instead of the sigmoid function the cross entropy loss function. The softmax function squashes all values to the range [0,1] and the sum of the elements is 1.

Figure 4.9: Graphical Example



4.4.2 Implementation

Lets build our model using LogisticRegression from Scikit-learn package. This function implements logistic regression and can use different numerical optimizers to find parameters, including 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga' solvers.

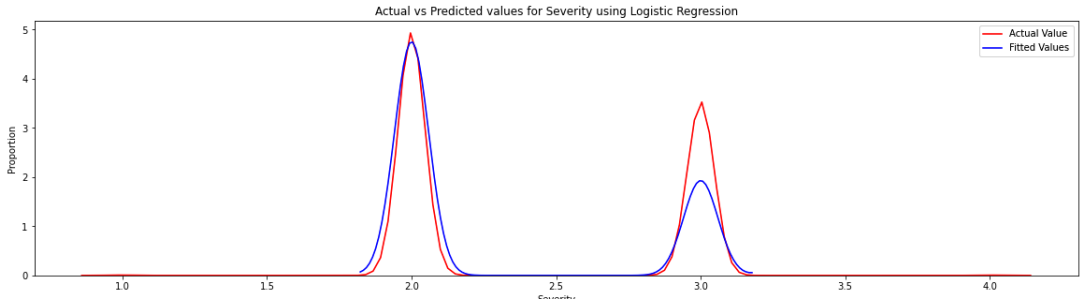
The version of Logistic Regression in Scikit-learn, support regularization. Regularization is a technique used to solve the overfitting problem in machine learning models. C parameter indicates inverse of regularization strength which must be a positive float. Smaller values specify stronger regularization. Now lets fit our model with train set:

4.4.3 Model Score

With all parametrization done, now its time to train the model. Before spliting the datainto a train and test set, we fit the model, and finally, we print the score.Now, using the R2, F, precission and recall score, we will evaluate the model.As we see, this isn't the best

score, that's why kNN isn't the best model to fit this particularcase , so we will try another classification models and see their performance.

Figure 4.10: Relation between predicted values and the actual ones



As we see, this model performs well predicting case 2 incidents, but poorly with case 3. This is the score of this model.

Accuracy Score	R2 Score	F1 Score
0.6322	-0.4980	0.6166

Table 4.3: Logistic Regression score

Chapter 5

Conclusion

5.1 Resume

In this study, I analyzed the relationship between Accident severity and their main characteristics. I identified weather, position, temperature, among the most important features that affect on this kind of events. I built four different prediction models to try to predict the level of severity of an accident, for example, Logistic Regression or k-Nearest Neighbors. These models can be very useful in helping goverments and organizations, ffor instance, it could help identify and help to determine the safety of a road in case of accident.

5.2 Acknowledgments

Thanks to my family and my girlfriend, who support me during this long summer while I was working on this professional certificate. Thanks to some of my teacher's for encouraging me to work hard towards my dreams, and finally, thank you for reading this report.

Chapter 6

Future Directions

Tanking in account that all this report is made by a 20 year old student, I'm sure that this work can be improved. I was able to predict with a 91 score the severity of an accident, so a better feature management and more data about 1 and 4 severity type data could improve much all these conclusions. By the way, more data, with more relevant information could give us more accurate models.