A Simple Logistic Regression Model to Predict Accident Severity Based on Seattle GIS Data

Roel Christian Yambao

August 2020

Contents

List of Tables

List of Figures

1 Introduction

According to data published by the National Highway Traffic Safety Administration, more than 30,000 Americans die each year of vehicle-related accidents. The per capita figure, despite trending downwards over the years, had seen a significant uptick in 2015 and has been on the rise ever since.

Transportation safety issues affect not just passengers and drivers but perhaps more importantly, pedestrians, as well. Any project addressing this must therefore acknowledge the fact that vehicular accidents are community issues and affect not just the individual drivers. This study looks at both community (e.g., neighborhood and location data) and individual (e.g., state of the driver/passenger) factors in addition to other external circumstances outside of the control of the two to develop a simple regression model aimed at predicting (and hopefully avoiding) severe accidents.

This study combs through 16 years of accident data from Seattle to design a machine-learning algorithm that considers a multitude of factors and which, once implemented, will help both local governments and private individuals in minimizing variables that have been proven to correlated with severe accidents. This information, though derived from data from Seattle, is not geographically limited to the city but could, when adapted properly, be used as a prediction model for anywhere else.

2 Exploratory Data Analysis

2.1 The Dataset

The dataset contains accident severity information on incidents recorded in Seattle, Washington, from 2004 to the present. The metadata is available here:

```
https://s3.us.cloud-object-storage.appdomain.cloud/
cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf
```

For the purposes of this project,the individual columns are not discussed in detail unless the column or columns have been manipulated for analysis. The description available in the metadata file to get acquainted with the dataset.

2.2 Data Cleanup

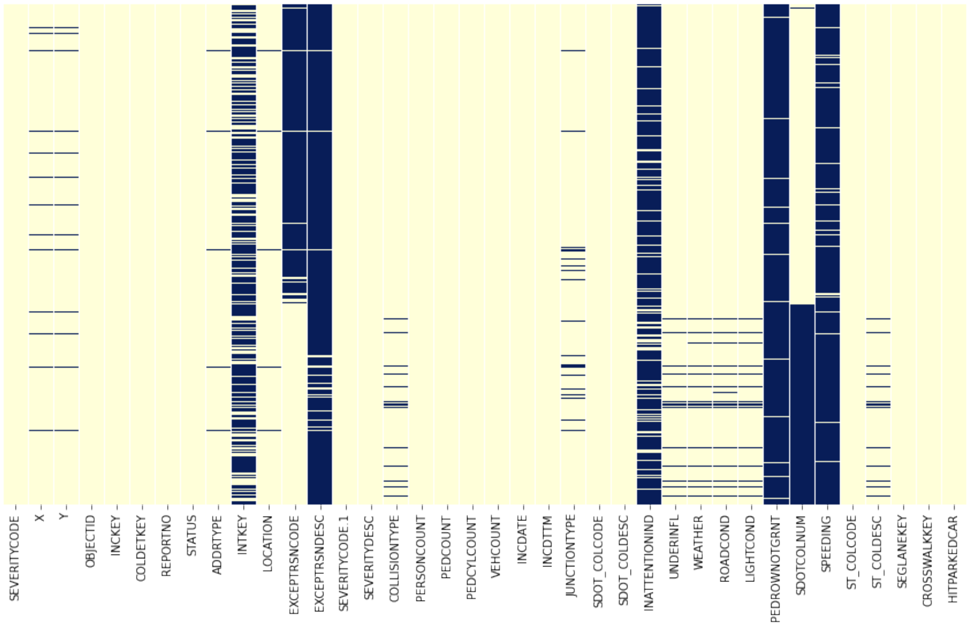


Figure 1: Heatmap showing NaN values in the data set.

As can be seen in 1 the dataset does not contain a lot of missing values with the exception of the following columns INTKEY, EXCEPTRSNCODE, EXCEPTRSNDESC, INATTENTIONIND, PEDROWNOTGRNT, SDOTCOLNUM, SPEEDING. Fortunately most of these data points are merely identifying keys, as explained in § 2.4 and can be easily dropped without any effect on our model.

Both the SPEEDING and the INATTENTIONIND column was created as a boolean (i.e., a Y/N data point) but we see that only the Y values have been filled up and the N values left blank. To be able to use this data point in our model, we replace cells containing the string 'Y' with 1 and empty cells with 0.

2.3 Treating accident severity as binary data

GIS classifies accident severity into five categories: Severity Code 3 for accident causing deaths, Code 2b for those causing serious injuries, Code 2 for those causing minor injuries, Code 1 for those causing only property damage, and Code 0 for those where the extent of the damaged caused is unknown.

A quick inspection of the data from Seattle reveals that the recorded accidents in the city are all classified under severity code 1 or 2. Since there might be accidents with severity codes 2b or 3 that has not been captured in the available data, it would be more prudent to consider accident severity as a binary feature, with accident with a severity code of 2 classified as severe and those with a code of 1 as non-severe. Using this approach, the prediction will yield not the severity of the accident itself but the likelihood that a severe accident, given the conditions, will occur.

2.4 Feature selection

We first eliminate features that were originally used as database keys or identifiers from the original data source, since they do not represent actual variables. This means dropping the following columns: OBJECTID, INCKEY, COLDETKEY, REPORTNO, STATUS, INTKEY, EXCEPTRSNCODE, SEGLANEKEY, and CROSSWALKKEY.

Since our designing a logistic regression model, we will not be using geographical data other than the ADDRTYPE feature which classifies the place where the accident occured into one of three types. Although it is important to know where an accident occured, we are not concerned on the geographical coordinates as much as we are withidentifying whether severe accidents are more common in locations of a certain type than in others.

We can now review the three binary variables in the feature set. We can draw a couple of interesting conclusion from the odds ratio and log odds ratio for the variables in Table 1. First we see that driving under the influence or driving above the prescribed speed limit increases the likelihood of a severe accident (with odds ratios of 1.536 and 1.455, respectively).[1] Second, we see that hitting a parked car makes the resulting accident about ten times less likely to be severe (i.e., an odds ratio of 0.149). These relationships are statistically significant and is supported by

[1] Note however that this does not reflect the odds of having an accident itself, which is outside the scope of this report.

Table 1: Odds ratio table for binary features in the original feature set.

	Estimate	Std. Err.	LCB	UCB	*p*-value
is_dui					
Odds ratio	1.536		1.471	1.604	0.000
Log odds ratio	0.429	0.022	0.386	0.472	0.000
is_speeding					
Odds ratio	1.455		1.394	1.519	0.000
Log odds ratio	0.375	0.022	0.332	0.418	0.000
hit_parked_car					
Odds ratio	0.149		0.135	0.164	0.000
Log odds ratio	-1.906	0.049	-2.002	-1.810	0.000

a p-value significantly lower than our $p = 0.05$ threshold. These initial regression models however treat the data points in isolation and their effect on accident severity when occuring in conjunction will only be dealt with when we introduce our model.

After an initial assessment of the available features, we are then left with the following feature set:

Table 2: Initial feature set

Feature	Data Type	Cleaned up?	Description
is_speeding	bool	Yes	positive correlation
is_dui	bool	Yes	positive correlation
addrtype	categorical	No	relationship unknown
weathercond	categorical	No	relationship unknown
roadcond	categorical	No	relationship unknown
lightcond	categorical	No	relationship unknown
weather	categorical	No	relationship unknown
hit_parked_car	bool	Yes	negative correlation
personcount	integer	Yes	relationship unknown
vehcount	integer	Yes	relationship unknown

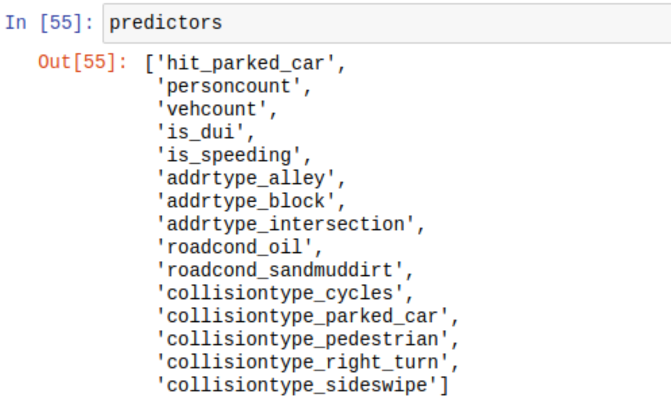
From this we see that of the initial feature set, the categorical variables are yet to be cleaned up. To be successfully used in our model, we need to create these categories into numerical values. We can replace each category using an index (similar to a record id in SQL) but this risks our model interpreting these numerical values as something inherently meaningful.

Since none of the remaining variables represent a scale we can objectively replicate[2] we will instead use dummy variables based on the feature and the categories under said feature. To visualize this, we can use the recursion below:

```
# introduce dummy variables to transform categorical data to binary data
cat_vars = ['ADDRTYPE','LIGHTCOND','ROADCOND','WEATHER','COLLISIONTYPE']
# replace column name with column name using the format columnname_categoryname
for var in cat_vars:
    cat_list='var'+'_'+var
    cat_list = pd.get_dummies(acc[var], prefix=var)
    acc1=acc.join(cat_list)
    acc = acc1
    acc.drop(var, axis=1, inplace=True)
# cleanup the generated names by replacing illegal characters etc.
acc.columns = acc.columns.str.replace('-', '')
acc.columns = acc.columns.str.replace(' ', '_')
acc.columns = acc.columns.str.replace('/', '')
acc.columns = acc.columns.str.replace('__', '_')
acc.columns = map(str.lower, acc.colu
```

The summary statistics are shown in Table 3.

Based on this summary, we will only be keeping features with a p-value less than or equal to 0.05. This leaves us the following final feature set for our model.



```
In [55]: predictors

Out[55]: ['hit_parked_car',
          'personcount',
          'vehcount',
          'is_dui',
          'is_speeding',
          'addrtype_alley',
          'addrtype_block',
          'addrtype_intersection',
          'roadcond_oil',
          'roadcond_sandmuddirt',
          'collisiontype_cycles',
          'collisiontype_parked_car',
          'collisiontype_pedestrian',
          'collisiontype_right_turn',
          'collisiontype_sideswipe']
```

Figure 2: Final feature set after removing variables with p-value $>$ 0.05.

[2] We can argue for example that the weather condition, which includes 11 categories in our data set can be ranked from 0 to 10 with the ideal weather at 10 and the worst condition at 0, but this requires designing an objective hierarchical system but the complexity of such model is not justifies our need for it.

Table 3: Summary statistics for the initial feature set.

	Coef.	Std.Err.	z	$P > \|z\|$	[0.025	0.975]
hit_parked_car[T.1]	-0.160	0.054	-2.966	0.003	-0.266	-0.054
personcount	0.150	0.005	31.170	0.000	0.141	0.160
vehcount	0.295	0.014	21.862	0.000	0.268	0.321
is_dui	0.597	0.027	22.582	0.000	0.546	0.649
is_speeding	0.437	0.025	17.586	0.000	0.389	0.486
addrtype_alley	-0.770	0.165	-4.681	0.000	-1.093	-0.448
addrtype_block	0.212	0.085	2.482	0.013	0.045	0.379
addrtype_intersection	0.394	0.086	4.584	0.000	0.226	0.563
lightcond_dark_no_street_lights	0.085	0.197	0.433	0.665	-0.301	0.472
lightcond_dark_street_lights_off	0.173	0.199	0.873	0.383	-0.216	0.562
lightcond_dark_street_lights_on	0.188	0.184	1.022	0.307	-0.172	0.548
lightcond_dark_unknown_lighting	0.105	0.882	0.119	0.905	-1.623	1.834
lightcond_dawn	0.260	0.190	1.367	0.172	-0.113	0.632
lightcond_daylight	0.276	0.183	1.504	0.133	-0.084	0.635
lightcond_dusk	0.264	0.186	1.421	0.155	-0.100	0.628
lightcond_other	0.273	0.258	1.058	0.290	-0.233	0.779
lightcond_unknown	-0.332	0.192	-1.734	0.083	-0.708	0.043
roadcond_dry	0.375	0.288	1.302	0.193	-0.190	0.940
roadcond_ice	0.209	0.298	0.701	0.484	-0.375	0.792
roadcond_oil	0.892	0.399	2.234	0.026	0.109	1.675
roadcond_other	0.696	0.356	1.958	0.050	-0.001	1.393
roadcond_sandmuddirt	0.797	0.402	1.982	0.048	0.009	1.585
roadcond_snowslush	-0.032	0.310	-0.104	0.917	-0.640	0.575
roadcond_standing_water	0.066	0.371	0.179	0.858	-0.660	0.793
roadcond_unknown	-0.381	0.296	-1.290	0.197	-0.961	0.198
roadcond_wet	0.338	0.289	1.170	0.242	-0.228	0.903
weather_blowing_sanddirt	0.145	0.413	0.351	0.726	-0.665	0.955
weather_clear	-0.210	0.217	-0.968	0.333	-0.635	0.215
weather_fogsmogsmoke	-0.055	0.238	-0.231	0.818	-0.522	0.412
weather_other	-0.182	0.248	-0.732	0.464	-0.668	0.305
weather_overcast	-0.235	0.217	-1.082	0.279	-0.660	0.191
weather_partly_cloudy	1.602	1.054	1.519	0.129	-0.465	3.668
weather_raining	-0.246	0.218	-1.131	0.258	-0.673	0.181
weather_severe_crosswind	-0.125	0.517	-0.241	0.810	-1.138	0.889
weather_sleethailfreezing_rain	-0.586	0.331	-1.770	0.077	-1.234	0.063
weather_snowing	-0.369	0.245	-1.508	0.132	-0.849	0.111
weather_unknown	-0.480	0.226	-2.122	0.034	-0.923	-0.037
collisiontype_angles	-0.372	0.252	-1.477	0.140	-0.866	0.122
collisiontype_cycles	2.524	0.254	9.931	0.000	2.026	3.022
collisiontype_head_on	-0.177	0.256	-0.693	0.488	-0.679	0.324
collisiontype_left_turn	-0.346	0.252	-1.372	0.170	-0.841	0.148
collisiontype_other	-0.638	0.252	-2.536	0.011	-1.131	-0.145
collisiontype_parked_car	-2.416	0.253	-9.549	0.000	-2.911	-1.920
collisiontype_pedestrian	2.730	0.254	10.747	0.000	2.232	3.228
collisiontype_rear_ended	-0.153	0.252	-0.605	0.545	-0.647	0.342
collisiontype_right_turn	-1.216	0.256	-4.750	0.000	-1.717	-0.714
collisiontype_sideswipe	-1.678	0.253	-6.640	0.000	-2.173	-1.183

3 Implementing the model

Since the value of the variable severe_bool which we are trying to predict is binary, we will be fitting a logistic regression model to our data. For this we used the scikitlearn library in Python.

The original data set consisting of 194,673 entries has been randomly split into a training set and a testing set with a ratio 75:25. A portion of the code for the model is reproduced below.

```python
# model.py
# The code below assumes the dataframe is named acc.

# define variables
X = acc[predictors]
y = acc['severe_bool']

# split training sets
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,
test_size=0.25,random_state=0)

# import the class
from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression(max_iter=10000)


# fit the model with data
logreg.fit(X_train,y_train)
y_pred=logreg.predict(X_test)
```

4 Evaluating the model

4.1 Model metrics

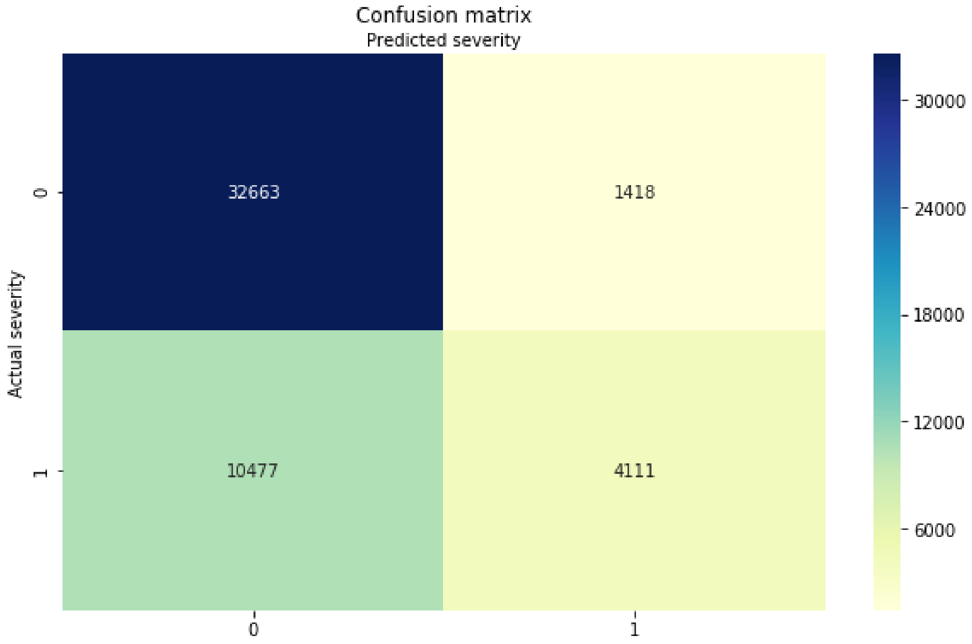


Figure 3: Confusion matrix for a testing sample of size $n = 48669$ (25% of the original data set).

As seen in Figure 3, the model was able to produce 32,663 true negative results and 4,111 true positive results from the testing sample of size $n = 48669$. This translates to a model accuracy of 76%, which is quite good. The generated classification report below shows the rest of the model metrics.

Table 4: Generated classification report showing model metrics.

	precision	recall	f1-score	support
0	0.76	0.96	0.85	34081
1	0.74	0.28	0.41	14588
accuracy			0.76	48669
macro avg	0.75	0.62	0.63	48669
weighted avg	0.75	0.76	0.71	48669

4.2 Goodness-of-fit

Earlier we have calculated the odds of an accident being severe given the other binary variables are true. We can extend this to our current predictors (changing transforming first the value of the log-odds of each feature to a simple odds ratio by using the natural exponential function) with the following results:

Table 5: Generated odds report and confidence intervals for the feature set.

	odds ratio	z-value	2.50%	97.50%
Intercept	0.162683	1.81E-108	0.138509	0.191076
hit_parked_car[T.1]	0.795618	1.89E-05	0.716482	0.883494
personcount	1.279422	0.00E+00	1.268536	1.290402
is_dui	1.684449	3.24E-94	1.602888	1.770159
is_speeding	1.375667	5.29E-41	1.313012	1.441312
addrtype_alley	0.506095	1.90E-05	0.370413	0.691477
addrtype_block	1.703737	7.10E-11	1.451531	1.999765
addrtype_intersection	2.006791	1.99E-17	1.708855	2.356671
roadcond_oil	1.42606	1.88E-01	0.841187	2.41759
roadcond_sandmuddirt	1.294969	3.57E-01	0.746829	2.245421
collisiontype_cycles	13.7757	0.00E+00	12.688599	14.955938
collisiontype_parked_car	0.117833	0.00E+00	0.112776	0.123117
collisiontype_pedestrian	16.580313	0.00E+00	15.284286	17.986236
collisiontype_right_turn	0.420828	1.55E-76	0.383985	0.461207
collisiontype_sideswipe	0.268319	0.00E+00	0.256545	0.280633

The odds ratio is multiplicative, and interestingly enough we see several variables with significantly higher odds than the rest of the feature set. Focussing on the variables with odds ratio greater than one, we can interpret the results in the table above as follows: incidents involving pedestrian collisions are 16.58 times more likely to cause a severe injury (or worse) than a typical accident, those involving cycle collisions 13.7 times, those occuring in intersections twice as likely to be fatal, those where one or more of the drivers involved was under the influence 1.68 times, and those where one or more of the drivers involved was speeding, among others. Understandably, too, we see that accidents happening in alleys are half as likely to be severe than a normal accident (i.e., about 0.50 times) while side swipe collisions are about a fourth as likely to cause any injury or fatality (about 0.26 times).

Finally to illustrate the adequacy of the model we plot both the deviance residuals and the Studentized Pearson residuals against the fitted values. The Lowess smooth of both residuals approximate a horizontal line with an intercept of zero, as expected.

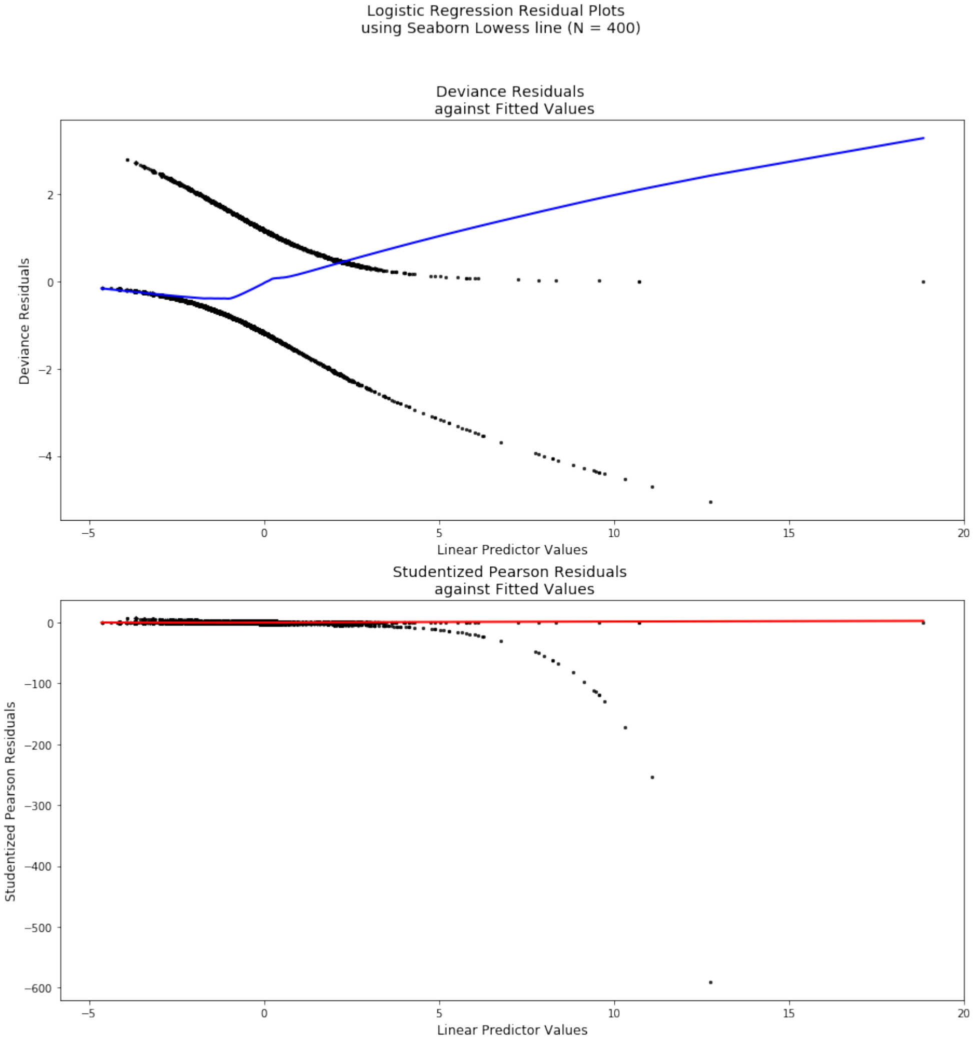


Figure 4: Logistic regression residuals.

5 Conclusion

We were able to design a multivariate logistic regression model for predicting whether or not an accident will be severe (i.e., cause an injury or fatality) based on a set of extenuating circumstances. The model's accuracy is around 75%. We were also able to demonstrate the relationship of various factors both inherently within the control of people (such as driving under the influence or above the speed limit) and the otherwise external factors (such as accident location or the weather or road conditions) with the accident severity itself as well as with the other variables. The results of this project can be used in predicting and thus avoiding the occurence of severe accidents.

6 Further Research

The current report can be further improved by introducing two main variable groups that have not been dealt with here: seasonality and geographical location.

The data set spans 16 years and is therefore ideal in modelling whether accident severity exhibits any seasonal pattern. We can also use the geographical data to identify high-risk clusters, and therefore focus the city's resources on zones that are accident-prone in certain sets of circumstances.

A major assumption made when designing the model is that of the accident severity data point being binary. While it is justified in the context of this study, we can further expand this model by including data from other cities or other sources which include accidents with severity code higher than 2. Since the severity can be thought of as a gradient, we can easily adapt the logistic regression model to a non-logistic regression model (most likely a linear one) that predicts the severity from a pre-established scale.

References

[1] National Highway Traffic Safety Administration. *2018 Fatal Motor Vehicle Crashes: Overview.* URL: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812826.

[2] David W. Hosmer and Stanley Lemesbow. “Goodness of fit tests for the multiple logistic regression model”. In: *Communications in Statistics - Theory and Methods* 9.10 (1980), pp. 1043–1069. DOI: 10.1080/03610928008827941. eprint: https://www.tandfonline.com/doi/pdf/10.1080/03610928008827941. URL: https://www.tandfonline.com/doi/abs/10.1080/03610928008827941.

[3] *Logistic Regression.* URL: https://www.pythonfordatascience.org/logistic-regression-python/#test_with_python. (accessed: 11.07.2020).