

Capstone Project - Car accident severity (Week 3)

By: Leandro Recova

1 – Introduction and Business problem.

The goal of this capstone project is to provide a machine learning model that can predict the severity of car accident based on a dataset provided by the course.

Based on the information of previous accidents, weather condition, traffic jam, the model could provide the likelihood of an accident for a driver that might be driving along the road. This kind of warning would be very important so the driver can make decisions while driving in such conditions. This model could be integrated with a GPS software application that runs in the user application based on the trajectory followed by the driver.

2 – Dataset Description

The dataset that will be used during this capstone project was provided in the week 1 of the course. It has a list of 38 fields described as listed in the table below. By analyzing the data, the dataset will have to go through a pre-processing problem since some columns still have blanks, inconsistency of data description (E.g: Y, N, 0, 1 values in the same column).

Based on the fields presented in this dataset, we will have to go through a pre-processing of the fields, make sure we select the features that will have a impact in the model, and remove those columns that will not be necessarily for this project.

We will use a logistic regression model to predict the severity of car accident based on this dataset.

Field	Field Description	Comments
1	SEVERITYCODE	Code 1 or 2 depending on the accident.
2	X	Longitude of the location of the accident
3	Y	Latitude of the location of the accident
4	OBJECTID	Primary key of the table
5	INCKEY	Table Keys
6	COLDETKEY	Table Keys
7	REPORTNO	Police Report Number
8	STATUS	Matched or Unmatched status
9	ADDRTYPE	Alley, Roadblock, Intersection, and blanks
10	INTKEY	Key parameter - Some fields are blank
11	LOCATION	Address of the accident
12	EXCEPTSNCODE	Key for except: NEI (Not enough information), blanks

13	EXCEPTRSNDESC	Description of the code if any
14	SEVERITYCODE	Severity code 1 or 2.
15	SEVERITYDESC	Two possible fields: Injury Collision and Property Damage Collision.
16	COLLISIONTYPE	Collision type.
17	PERSONCOUNT	Number of people inside the car.
18	PEDCOUNT	Pedestrians count
19	PEDCYLCOUNT	Pedestrians cyclists count
20	VEHCOUNT	Number of vehicles.
21	INCDATE	Incident date
22	INCDTTM	Incident date with time
23	JUNCTIONTYPE	Junction type
24	SDOT_COLCODE	69 different sdot codes
25	SDOT_COLDESC	Sdot codes description
26	INATTENTIONIND	In attention in mind: Just a Yes comment.
27	UNDERINFL	Mixed of N, Y, O, and 1 parameters.
28	WEATHER	Weather conditions
29	ROADCOND	Road conditions
30	LIGHTCOND	Light conditions
31	PEDROWNOTGRNT	Pedestrian not granted ROW indicator
32	SDOTCOLNUM	Sdot column number
33	SPEEDING	Mix of blanks and Y.
34	ST_COLCODE	Street code
35	ST_COLDESC	Street code description
36	SEGLANEKEY	Code
37	CROSSWALKKEY	Cross walk key
38	HITPARKEDCAR	Hit parked car (Yes or No).

3 – Methodology

As mentioned in the previous section, the data was analyzed and we selected few features for this model. They are listed below:

- **Location:** Fields X and Y representing the longitude and latitude, respectively. We also renamed these fields during the data processing.
- **Road Condition:** Field Road Cond. We created an additional column called ROADCONDID to have a number associated with one of each type of conditions listed in the dataset.
- **Weather:** This is an important factor and we also categorized this field by associating a number with each of the conditions listed.
- **Light conditions:** Like the previous item, we also created a column called WeatherID to represent the particular category.

- **Speeding:** This field had multiple blanks and also a “Y” when speeding was flagged. We added fill it out the blanks with 0 and the “Y” field with 1 in a new column called SpeedingID.
- **Junction Type:** We categorized each of the junction types by creating a new column called JunctionTypeID.

Below is a summary of each of the categories:

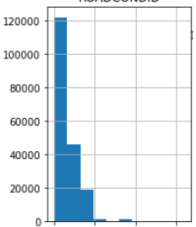
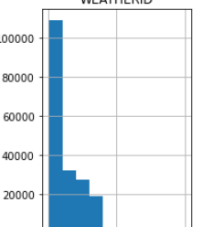
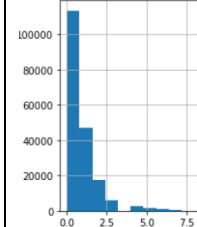
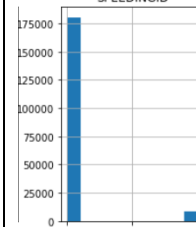
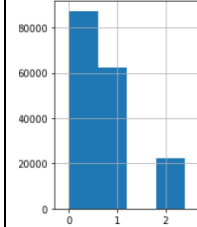
Road Conditions	Weather	Light Conditions	Speeding	Junction Type
Dry 0 Wet 1 Unknown 2 Ice 3 Snow/Slush 4 Other 5 Standing Water 6 Sand/Mud/Dirt 7 Oil 8	Clear 0 Raining 1 Overcast 2 Unknown 3 Snowing 4 Other 5 Fog/Smog/Smoke 6 Sleet/Hail/Freezing 6 Rain 7 Blowing Sand/Dirt 8 Severe Crosswind 9 Partly Cloudy 10	Daylight 0 Dark - Street Lights On 1 Unknown 2 Dusk 3 Dawn 4 Dark - No Street Lights 5 Dark - Street Lights Off 6 Other 7 Dark - Unknown Lighting 8	Unknown 0 Y 1	Mid-Block (not related to intersection) 0 At Intersection (intersection related) 1 Mid-Block (but intersection related) 2 Driveway Junction 3 Unknown 4 At Intersection (but not related to intersection) 5 Ramp Junction 6
				

Table 2: Categories used for each of the selected features and its histograms.

The histograms presented on table 2 provides some interesting information. In most of the accidents, it is clear the road conditions dry, weather clear, light condition day light, not speeding, and junction types mid-block are the ones with most of the accidents.

Once the categories have been defined, we created a panda dataframe called **finalIDF** with the fields required for the logistic regression algorithm. Below is a snapshot of the first rows of the input data.

SEVERITYCODE	LAT	LONG	PERSONCOUNT	VEHCOUNT	ROADCONDID	WEATHERID	LIGHTCONDID	SPEEDINGID	JUNCTIONTYPEID
0	1	47.703140	-122.323148	2	2	1.0	2.0	0.0	1.0
1	0	47.647172	-122.347294	2	2	1.0	1.0	1.0	0.0
2	0	47.607871	-122.334540	4	3	0.0	2.0	0.0	0.0
3	0	47.604803	-122.334803	3	3	0.0	0.0	0.0	0.0
4	1	47.545739	-122.306426	2	2	1.0	1.0	0.0	1.0

Figure 1: Snapshot of the **finalIDF** PANDA dataframe as an input to the Logistic regression algorithm.

4 – Results and Discussion of the results.

The Logistic Regression algorithm was used with the following configuration:

```
#SPLIT THE TRAIN AND TEST SET
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=4)
print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)

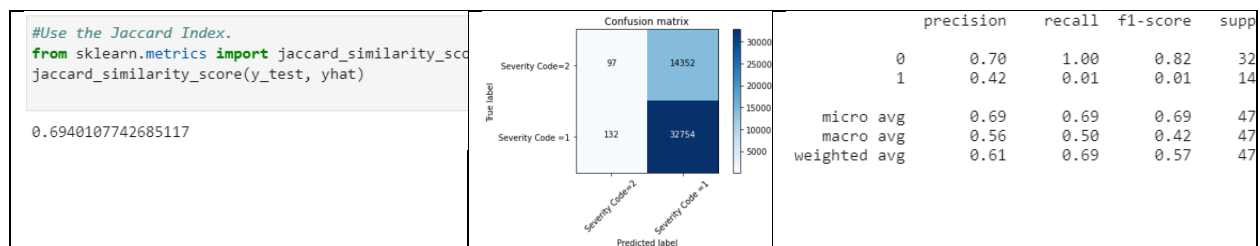
Train set: (142004, 8) (142004,)
Test set: (47335, 8) (47335,)

#LET'S USE THE LOGISTIC REGRESSION WITH REGULARIZATION
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
LR

LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='warn',
n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
tol=0.0001, verbose=0, warm_start=False)
```

Figure 2: Snapshot of python notebook of the LR Model parameters utilized.

We used the solver “liblinear” with the regularization constant $C=0.01$. We split the train and test set in a ratio of 75% and 25%, respectively. The jaccard index, confusion matrix, and precision and F-1 scores snapshots are presented below.



We obtained a 70% precision of accuracy in estimating the severity code 1 and 42% for severity code 2. The model did not estimate well the severity code 2. One of the reasons could be the fact that there might be some human decisions in classifying the codes based on all factors used in this model. We also tried to play with different values of the regularization factor, but the results seem to be within the same values obtained above.

5 – Conclusion

In this capstone project, our goal was to estimate the severity of the accident based on the dataset provided in the week 1 of the course.

There was a significant time spent during the data preparation, choice of the features, categorizing the features that could have a more impact in the outcome prediction, and the choice of the method to evaluate the results.

The logistic regression method was utilized since we had to estimate between two severity codes. We used the liblinear solver and obtained a jaccard index of 0.69 and a precision of 70% of severity code 1 and 42% for severity code 2. The estimation for severity code 2 was not the result expected, but, after analyzing the dataset, it seems there might be a human decision factor to decide when to choose these codes.

The final recommendation left for the dataset owner would be to reevaluate how the codes 1 and are being assigned and if any human factor is used to decide between the severity of these codes. A evaluation of this method against future decisions would help to better classify these codes and provide a better prediction of accidents for insurance companies and drivers that might be driving in a high risk road.

