Accidents with cyclists — A guide to avoid them

Author: Patrick Queiroz dos Anjos

Business Understanding

The use of bicycles as a means of transport, as sports or just for recreation, has increased dramatically during the recent years. The bicycle is a vehicle that has no pollution (direct, in its use) and, therefore, is considered one of the cleanest means of transport, in terms of the environment, to use.

But, unfortunately like any other means of transport, the bicycle and the cyclist are liable to suffer some accidents, from simple impacts to severe actions. To ride a bicycle a cyclist must pay attention to several issues on the road, such as weather conditions on the day.

These conditions can help in the fact that in a rain condition, for example, the tendency that the cyclist has to lose control of his bicycle is greater, due to the aspect that, at a given speed, the control of his vehicle is impaired. Other climates are more severe, such as snow and milder ones, such as its use on sunny days.

Lighting in front of the road where the bicycle will be used is also important. Road conditions must be suitable for use, such as a well-lit environment. Environments in low light or with different colors, such as at dusk, can be prone to possible accidents.

The environment where the cyclist will be used, usually on roads, is also relevant. Conditions where there is a road with sand, or with a road after rain indicate a greater inclination for accidents. The opposite is also true, as conditions where the road is dry denotes greater initial safety for cyclists.

The environment where the bicycle will be used can also influence accidents, such as at intersections where, by some different means, such as rain and a wet road, they can increase the likelihood of accidents.

So, this project aims to identify certain parameters, such as weather, road conditions, light conditions and the collision address type to demonstrate and indicate patterns where accidents occur with cyclists and thus avoid, or be attentive to, the use of bicycles in the indicated parameters, through of a decision tree.

Data understanding

The data used in this project will use the .csv file (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv) made available in the first week of the Applied Data Science Capstone course, in the "Downloading Example Dataset". It shows a file where can get various information, such as weather conditions, number of pedestrians involved in the specified accident and the severity of the accident, among others.

The dataset has 194673 different rows with a total of 38 different columns, each detailing some variable inherent to the accident. A summary of the file can be seen at the link https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf.

The data that will be collected, processed to suit the later machine learning and primary data analysis processes will be the columns ADDRTYPE, ROADCOND, LIGHTCOND, WEATHER, PEDCYLCOUNT and SEVERITYCODE.

The column ADDRTYPE indicates the collision address type, where it shows the environment where the accident occurred and the ROADCOND column shows the conditions of the road where the accident occurred.

The LIGHTCOND column denotes the lighting conditions in the accident and the WEATHER column indicates the weather conditions that the accident occurred.

The PEDCYLCOUNT column and the SEVERITYCODE are the high impact columns in the final table, a table suitable for the machine learning algorithm, in this case a decision tree.

PEDCYLCOUNT is the applicant table for the accidents involveds with cyclists. There are, clearly, accidents that involved pedestrians only, only the driver of the vehicle involved in the accident, or both. But the focus of this project is on accidents involving cyclists, so this column is extremely important.

SEVERITYCODE is the column that establishes the relationship between the accident and a code that returns a value. This value denotes how severe the accident was. So, therefore, this will be the target column for later machine learning methods.

Then, by means of a dataframe equipped with these specified columns, we will have the process of modeling through machine learning, through the decision tree algorithm.

Data Preparation

The original dataframe was converted to have only 6 columns, the columns ADDRTYPE, ROADCOND, LIGHTCOND, WEATHER, PEDCYLCOUNT and SEVERITYCODE, described earlier. The 'df' dataframe has a total of 187525 rows and the 6 columns chosen. From that choice, some methods of choosing variables, transforming and analyzing were specified:

— The df was specified to have data only with accidents involving cyclists; The target values in the decision tree algorithm, SEVERITYCODE, now only have a value of 1 or 2;

— The accident severity code values are 1 for "Property Damage" and 2 for "Injury". But the amount of data with value 1 (“Property Damage”) and value 2 (“Injury”) is unbalanced, with 4793 data for value 2 and 676 data for value 1.

Then the dataframe was balanced so that there was 50% of each dataframe that will be submitted to the machine learning process. The Under-Sampling technique was used and, subsequently, the amount of data with value 2 was equal to that of value 1.

The dataframe now has 676 data with values 1 and 2, totaling 1352 rows. Image 1, just below, shows some information from the pre-processed dataframe:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1352 entries, 192552 to 194585
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   ADDRTYPE      1352 non-null   object
 1   ROADCOND      1352 non-null   object
 2   LIGHTCOND     1352 non-null   object
 3   WEATHER       1352 non-null   object
 4   PEDCYLCOUNT   1352 non-null   int64 
 5   SEVERITYCODE  1352 non-null   int64 
dtypes: int64(2), object(4)
memory usage: 73.9+ KB
```

Figure 1 – Dataset partially pre-processed

The image shows the number of columns, non-null count, dtype and other parameters for information methods.

After these processes, information was collected about the chosen columns:

— The WEATHER column has 8 values specified in a table and displayed in a bar graph, with the count of each categorical value. They can be seen in image 2 below:

```
(<matplotlib.axes._subplots.AxesSubplot at 0x1c24802e448>,
 Clear                       944
 Overcast                    207
 Raining                     150
 Unknown                      46
 Other                         2
 Snowing                       1
 Sleet/Hail/Freezing Rain      1
 Fog/Smog/Smoke                1
 Name: WEATHER, dtype: int64)
```

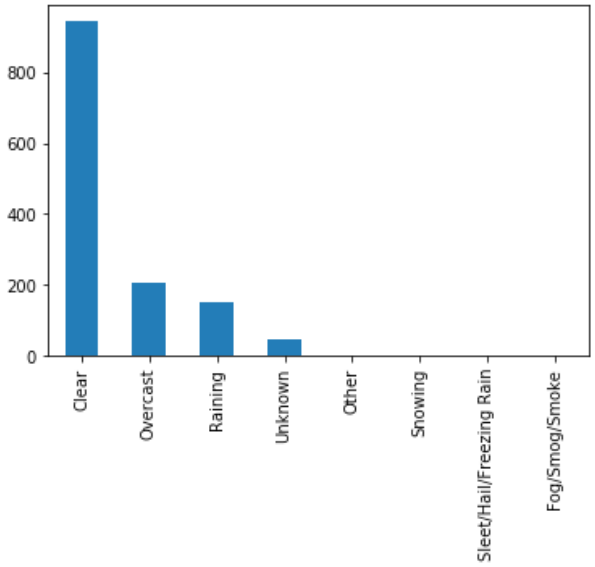

Figure 2 - WEATHER column description

The categorical values "Unknown" and "Other" were removed, as they do not provide useful information in the face of an initial situation for the cyclist.

— The ADDRTYPE column has 3 values with the demonstrated count and a descriptive bar graph. Figure 3 below shows these values:

```
(<matplotlib.axes._subplots.AxesSubplot at 0x1c248095d88>,
 Intersection    733
 Block           617
 Alley             2
 Name: ADDRTYPE, dtype: int64)
```

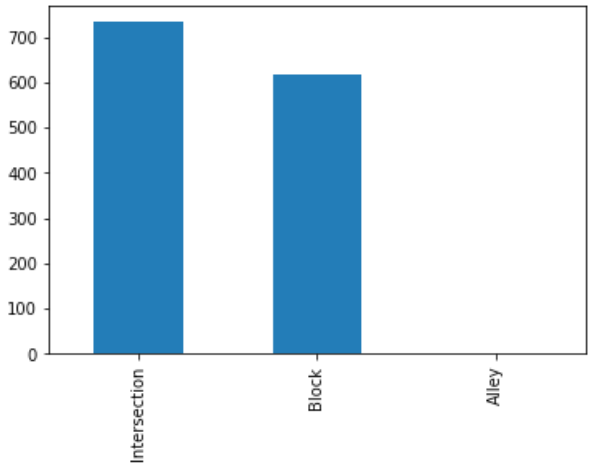

Figure 3 - ADDRTYPE column description

—- The ROADCOND column has 8 categorical values. The count of each category and therefore the bar graph are shown in image 4 below:

```
(<matplotlib.axes._subplots.AxesSubplot at 0x1c255b8eac8>,
 Dry                1061
 Wet                 227
 Unknown              53
 Standing Water        3
 Ice                   3
 Snow/Slush            2
 Sand/Mud/Dirt         2
 Other                 1
 Name: ROADCOND, dtype: int64)
```

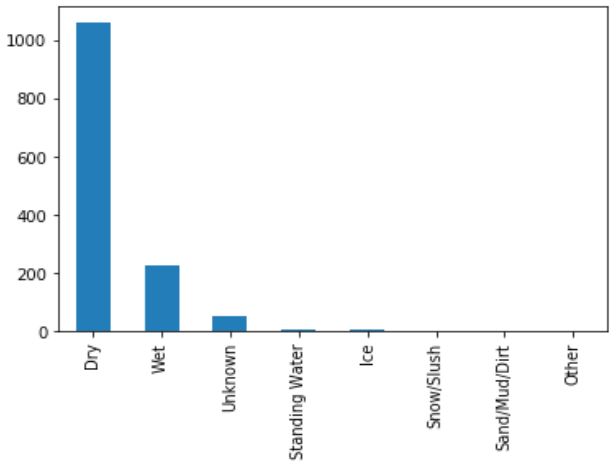

Figure 4 - ROADCOND column description

The “Unknown” and “Other” categories have been removed as well.

— The LIGHTCOND column has 7 categories. The count for each category and also the intrinsic bar graph are shown in image 5 below:

```
(<matplotlib.axes._subplots.AxesSubplot at 0x1c248185988>,
 Daylight                   994
 Dark - Street Lights On    225
 Dusk                        53
 Unknown                     46
 Dawn                        19
 Dark - No Street Lights      8
 Dark - Street Lights Off     7
 Name: LIGHTCOND, dtype: int64)
```

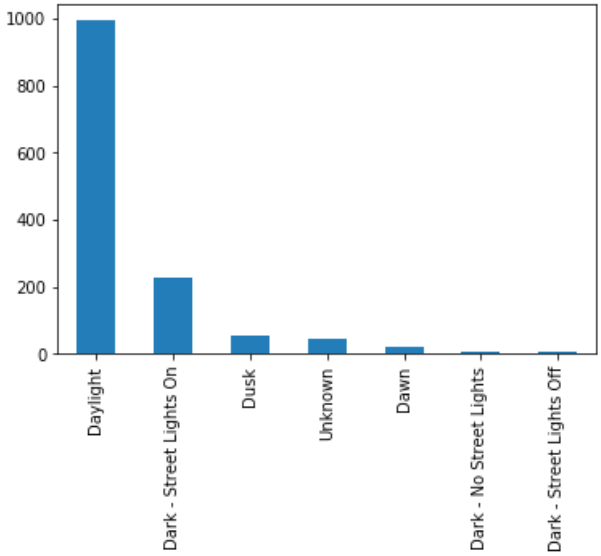

Figure 5 - LIGHTCOND column description

The “Unknown” column has been removed.

All 4 columns underwent the One Hot Endoding process and were concatenated and converted to values that will be submitted to the machine learning algorithm, the decision tree.

The target values in the SEVERITYCODE column have been modified. The value "1" was reset to "Property Damage" and the value "2" to "Injury".

The values of each predictor column and the target column were submitted to Train-Test-Split, to partition the training and test values, for training the decision tree and for its test, respectively. The result is seen below, in image 6:

```
from sklearn.model_selection import train_test_split

# normalized, value 0 or 1 after One Hot Encoding
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)

print ('Train set:\n'
       '- The x_train shape is', x_train.shape, 'and y_train shape is',  y_train.shape, '\n',
       'Test set:\n'
       '- The x_test shape is', x_test.shape, 'and y_test shape is', y_test.shape)
```

```
Train set:
- The x_train shape is (946, 21) and y_train shape is (946,)
 Test set:
- The x_test shape is (406, 21) and y_test shape is (406,)
```

Figure 6 - Train-Test-Split applied

Modeling

The model chosen was the decision tree for some reasons, such as easy implementation, good description and because it is highly interpretable. You can use the decision tree to make a choice for an event, for example.

The decision tree for the classification of parameter values such as weather conditions, lighting, the road the cyclist uses, as well as the type of address where the accidents occurred, were selected and submitted to predict the class of the accident, if it occurred a Property Damage or an Injury.

The hyperparameters of the decision tree, such as “criterion”, “max_depth”, “max_leaf_nodes” and “min_samples_leaf” were chosen as a result of the “GridSearchCV” algorithm (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html). These values provide the best training results for a given algorithm, in this case a decision tree.

The results are shown below, in image 7:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV

dt_model = DecisionTreeClassifier()

parameters = {'criterion':['gini', 'entropy'],
              'max_depth':range(5,11),
              'max_leaf_nodes':range(5,11),
              'min_samples_leaf':range(5,11)}
grid = GridSearchCV(estimator = dt_model, param_grid = parameters).fit(x_train, y_train)

print(f'The best parameters are {grid.best_params_};\n'
      f'And the best score is {grid.best_score_.round(4)}.')
```

```
The best parameters are {'criterion': 'gini', 'max_depth': 5, 'max_leaf_nodes': 10, 'min_samples_leaf': 10};
And the best score is 0.5264.
```

Figure 7 - Best hyperparameters found in GridSearchCV algorithm

The implementation of the decision tree is also shown below in image 8, with each hyperparameter optimized.

```
dt_model = DecisionTreeClassifier(criterion=grid.best_params_.get('criterion'),
                                  max_depth=grid.best_params_.get('max_depth'),
                                  max_leaf_nodes=grid.best_params_.get('max_leaf_nodes'),
                                  min_samples_leaf=grid.best_params_.get('min_samples_leaf'))
dt_model.fit(x_train,y_train)
dt_model
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=5, max_features=None, max_leaf_nodes=10,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=10, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

Figure 8 - Implemented decision tree

Evaluation and Deployment

The algorithm has been trained and tested. The best precision value for the training algorithm was approximately 52.64%. After this step, the algorithm was trained and demonstrated the following results for “Accuracy”, “Recall Score”, “Precision Score” and “F1 Score”, as described below:

Table 1 - Valuation values of decision tree

Method	Results (%)
Accuracy	51.23
Recall Score	51.23
Precision Score	51.20
F1 Score	51.21

The values demonstrate that the values, considered here as predictive variables, do not have as much predictability for the machine learning model.

The questioning of the logic of the project process itself is highly complex, the processes involving acts of accidents are changeable and with a wide range of inherent variables. So the global process itself is difficult, so also, the low performance of the chosen machine learning algorithm, the decision tree, has low accuracy.

However, the decision tree implemented can correlate variables of greater weight in the event of accidents involving cyclists, the main scope of this project.

The decision tree has been implemented and can be viewed below:

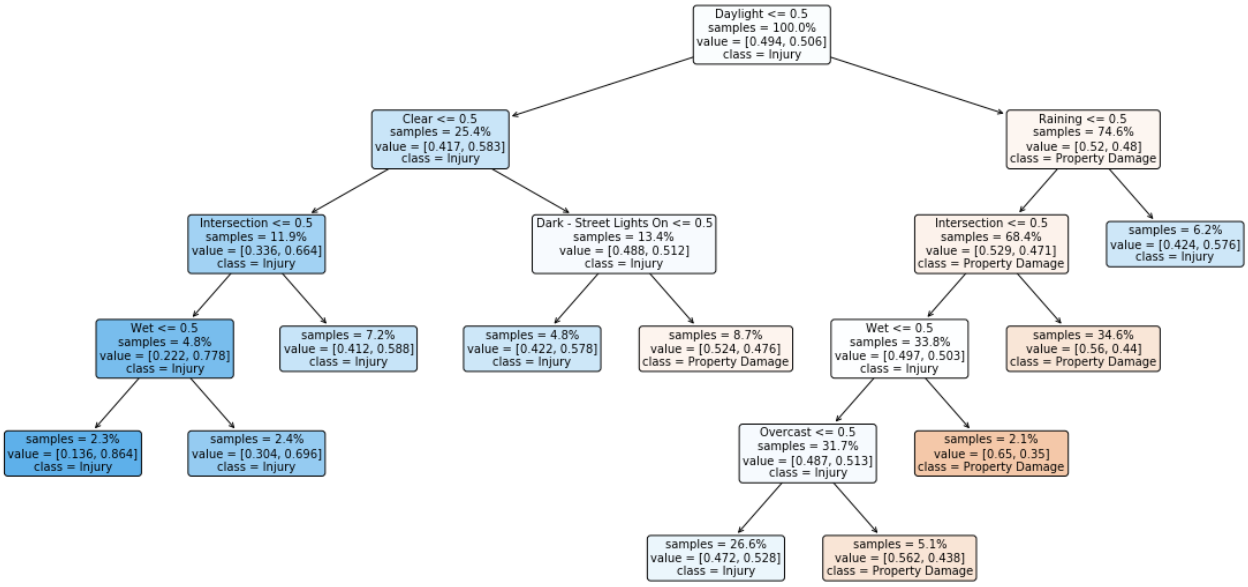

Figure 7 – Image of implemented decision tree

Here the step by step description of the implemented decision tree is demonstrated. The analysis takes place by hierarchy, citing the first node (the one with the highest level in the image above) to the lowest levels, where all the samples present in this work come from.

There is a sentence in each node, and if the sentence is true the interpretation is on the left. The opposite is on the right. All variables are binary, that is, if equal to 1 the variable is present and if equal to 0 the variable is absent. The complete analysis of the first node is described below:

— The most important variable in the decision tree is “Daylight”. When it is daylight and bright, the forecast is 52% for Property Damage and 48% for Injury. The opposite has a 58.3% chance of being Injury and 41.7% of Property Damage. This sentence is logical, because in the daylight there is less chance of more severe accidents occurring.

Now all other nodes will be described in a simplified way, but with a highly interpretable method that is easy to observe and read. By logical sentences, we have:

IF **DAYLIGHT = FALSE** (or <= 0.5), THEN LEFT (Decision tree image)

— When the weather condition is “Clear” and when the lighting condition is “Dark - Street Lights On” there is a forecast of 52.4% Property Damage and 47.6% Injury. There is the “Clear” condition, but without the “Dark - Street Lights On” condition there is 42.2% Property Damage and 57.8% prediction Injury;

— When the condition of adrress type is “Intersection”, without the condition “Clear”, there is 41.2% of Property Damage and 58.8% of Injury, in probability;

— When the road condition is “Wet”, without the condition of “Clear” or “Intersection”, there is 30.4% Property Damage and 69.6% Prediction Injury. When there is no “Wet” condition, there is 13.6% Property Damage and 86.4% Injury, in probability.

IF **DAYLIGHT = TRUE** (or> 0.5), THEN RIGHT (Image of the decision tree)

— And the weather condition is “Raining” there is a probability of 42.4% for Property Damage and 57.6% for Injury;

— And condition of adrress type for “Intersection” there are 56% chances for Property Damage and 44% chances for Injury;

— And road condition is “Wet”, there is a 65% probability for Property Damage and 35% for Injury;

— And the weather condition is “Overcast” there is 56.2% for Property Damage and 43.8% for Injury, by probability;

— **ONLY** when the lighting condition is “Daylight” there is a 47.2% chance for Property Damage and 52.8% for Injury.

CONCLUSIONS

The conclusions can be summarized below:

— The accident data were pre-processed to obtain the correlation between some variables, such as weather conditions and lighting conditions with the severity of accidents involving cyclists;

— The machine learning algorithm chosen was the decision tree, as it is an easy-to-implement algorithm and with a resulting model that is easy to view;

— The decision tree was then implemented using optimal hyperparameters defined by the GridSearchCV algorithm, and was subsequently evaluated;

— The evaluation of the implemented, trained and tested decision tree was carried out. The values did not demonstrate a good evaluation of the predictive method of the implemented decision tree;

— Even with the low prediction yield, the decision tree can provide interesting parameters to demonstrate certain patterns to inform cyclists about variables where one should be more cautious, for example;

— The model was implemented by means of an image indicating the values of each important variable in the process. Through logical sentences, variables such as “Daylight”, “Raining” and “Wet” can be analyzed and proved to be relevant to the process of describing accidents involved by cyclists.

APPENDIX

The notebook, with the programming steps, explanatory notes and the results of the data analysis, pre-processing, the implementation of the decision tree algorithm and its "deploy" is available in:
https://github.com/patrick21081995/Coursera_Capstone/blob/master/capstone_project_PatrickQueirozdosAnjos.ipynb.

Day — 5
Month — October
Year — 2020