Accidents with cyclists — A guide to avoid them

Author: Patrick Queiroz dos Anjos

Business Understanding

- The use of bicycles has increased in recent years, due to the encouragement of sports practices and less use of polluting vehicles;

- But, unfortunately, there are many accidents involving cyclists, due to several variables, such as road and weather conditions, for example;

- This project aims to describe, and predict, the accidents that occurred in Seattle (USA) as a basis to inform the conditions where more accidents occur, so that the cyclist can provide information to prevent future accidents;

Data Understanding

- The data used in this project will use the .csv (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv);

- 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;

Data Understanding

- The column ADDRTYPE indicates the collision address type in 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;

- PEDCYLCOUNT is the applicant table for the accidents involveds with cyclists and SEVERITYCODE is the column that establishes the relationship between the accident and a code that returns a value;

Data Preparation

```
<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
```

- The dataset was processed to obtain only accidents involving cyclists and subsequent balancing;
- SEVERITYCODE data has 50% value 1 and 50% value 2;
- Value 1 is "Property Damage" and value 2 is "Injury";

Data Preparation

- The chosen columns underwent the One Hot Encoding process, to establish numerical values for each categorical value and then there was the process of partitioning the data between training data and test data;

- The target values were changed from 1 to "Property Damage" and 2 to "Injury", in order to provide a better visualization and interpretation of the model made;

- The pandas.get_dummies and sklearn.model_selection.train_test_split were used for One Hot Encoding and the partitioning of training and test data, respectively;

Modeling

- Decision tree was chosen because it is easy to implement and easy to interpret. The optimal values for each hyperparameter were found using the GridSearchCV algorithm;

```
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.
```

Modeling

- The values of the hyperparameters were implemented in the decision tree and therefore. the decision tree was drawn up;

```
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')
```

Evaluation

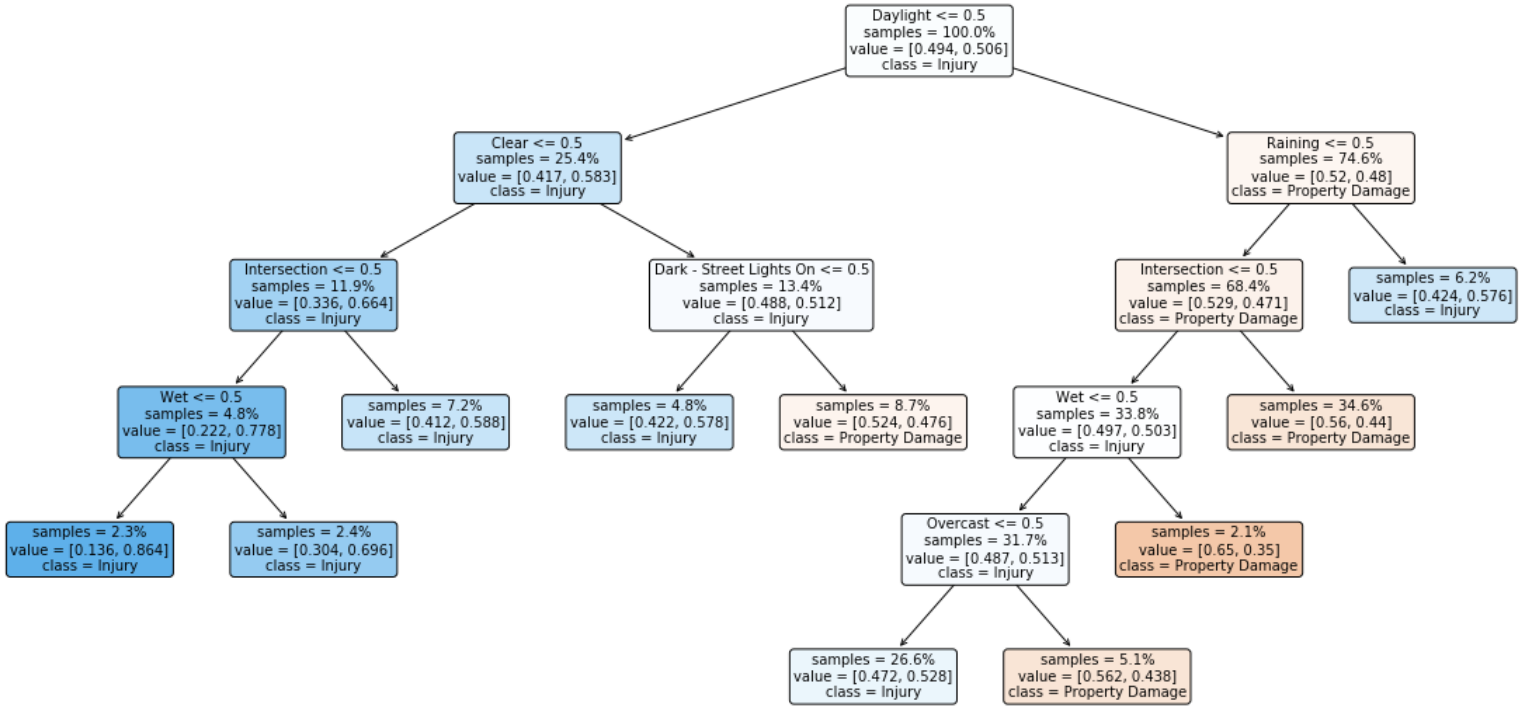
- The values of accuracy, recall score, precision score and f1 score were 51.23%, 51.23%, 51.2% and 51.21%, respectively. The values proved to be well below the desired values for predictive algorithms;

- The reason is that the occurrences of accidents, in this case involving cyclists, are complex and are dependent on many variables, which are outside the scope of this project;

- However, the decision tree provides highly useful information to assist cyclists and to exercise greater caution in conditions where there is a greater accident rate;

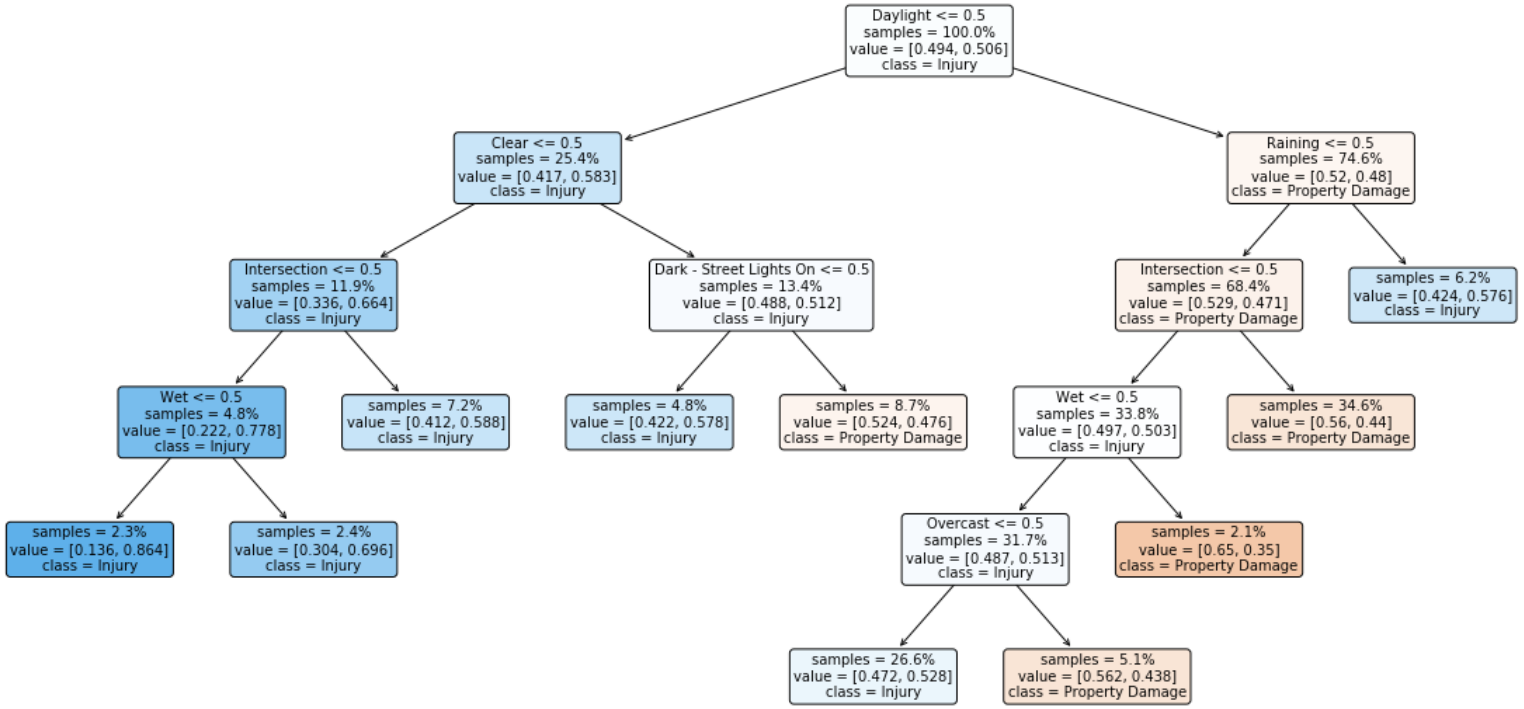
Deployment

- The "deploy" of the decision tree is the image below, which shows the variables with the greatest weight in the decision and achievable probabilities of "Property Damage" or "Injury":



Deployment

- Values on the left are when the main sentence of each node is False, the opposite, when True the decision is on the right;



Deployment

- As there are only binary values (0 and 1) for each sentence, the values are then interpreted as True os False for each explicit variable;

- The main variable in the decision tree is "Daylight", where, if True, there is 52% Property Damage and 48% Injury, by probability. The opposite is 41.7% chances for Property Damage and 58.3% chances for Injury;

- The following analyzes are applicable in the same way, thus providing information such as on days with "Daylight" and "Raining" there are greater chances of Injury and on days with "Clear" and "Dark - Street Lights On" there is less chance of Injury;

Conclusion

- The model for purposes of description and prediction, for accidents involving cyclists, was implemented and built, through a decision tree;

- The data were pre-processed to provide primary information on weather and road conditions, for example, for cyclists in order to have a better understanding of the accidents that occurred;

- The model provided low evaluation values. Even so, the implemented model has a good description for accidents and how to avoid them. The deployment is easy to view and interpret, so it helps the global understanding of the project.

Appendix

- The notebook, with each step commented and specified can be viewed on link https://github.com/patrick21081995/Coursera_Capstone/blob/master/capstone_project_PatrickQueirozdosAnjos.ipynb.