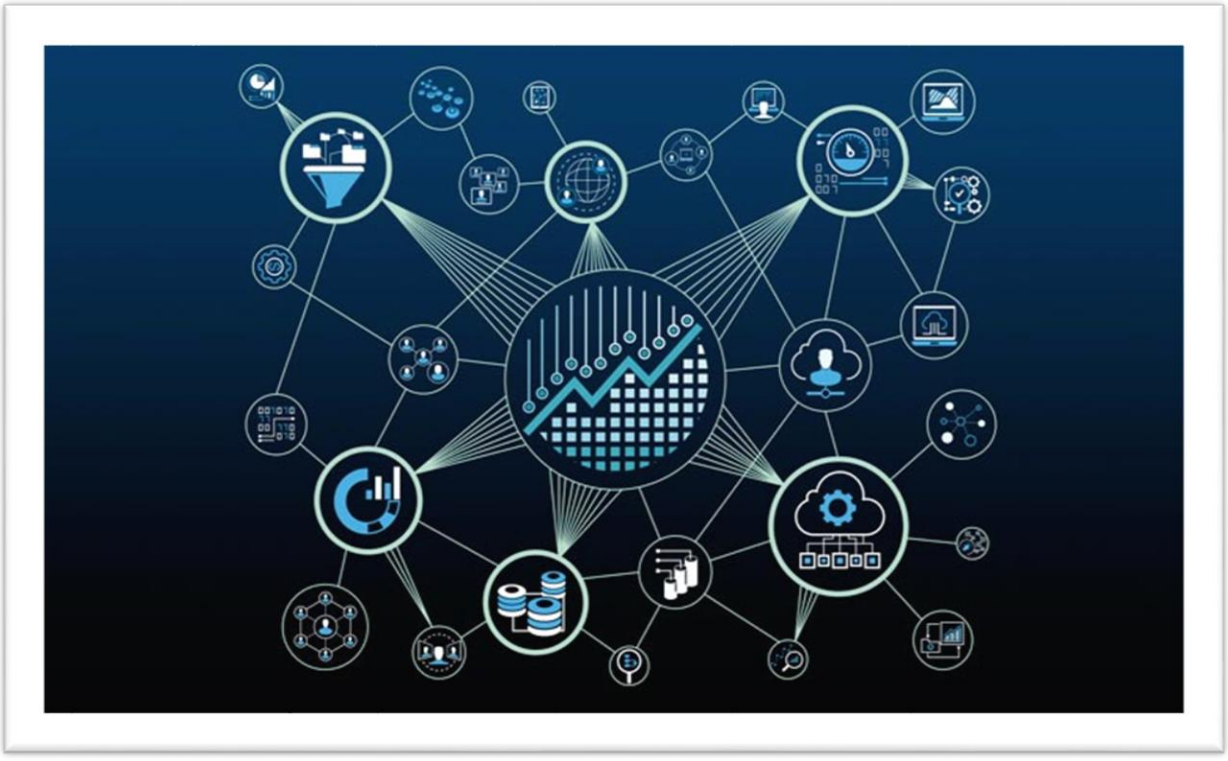
IBM DATA SCIENCE CAPSTONE FINAL PROJECT

AUTHOR: JOVANNY ULLOA

TABLE OF CONTENTS



AUTHOR: JOVANNY ULLOA

INTRODUCTION

Nowadays, data science represents an opportunity to analyze and give us some areas of information that we did´nt see in the past. With this area we can predict and understand a phenomenon according to the variables that are being related to a target property of a situation or event.

Car accidents are being a worldwide problem in our days, not only because they represent an economic problem and also, lifes are in game within these kind of events. However, now it is possible to detect all those conditions or factors that can decrease or increase the probability of an accident with the behaviour understanding of all the variables that participate.

BUSINESS PROBLEM

To help people to confront this problem, within an effort to contribute on the management of these fatal events, an algorythm is going to be developed to predict car accidents and advertise a driver when conditions are wrong enough to represent a risk. This can bring some benefits as: saving lives of drivers (our target audience) and decrease costs for goverment (our stakeholders)

DATA (COLLECT, UNDERSTANDING, VISUALIZATION)

COLLECT

To start this project we need data, in this case our source, which belongs to reports from 2004 to 2020, is organized with 38 attributes and 194673 rows.

```
Registers: 194673
Cathegories: 38
```

```
Source address: https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
```

After some enough data analysis of this CSV file, our target or dependent variable is going to be "SEVERITYCODE", this field or attribute is used to measure the severity of a car accident with a range from 0 to 5. So, all the algorytm development is going to be focused on this variable, with the next values:

- ✓ 0 : Little to no Probability
- ✓ 1 : Very Low Probablility - Chance or Property Damage
- ✓ 2 : Low Probability - Chance of Injury

- ✓ 3 : Mild Probability - Chance of Serious Injury
- ✓ 4 : High Probability - Chance of Fatality

By the other side, these are our independent variables or most important attributes:

1. WEATHER
2. ROADCOND
3. LIGHTCOND
4. COLLISIONYTPE
5. UNDERINFL

UNDERSTANDING

This process consisted in the next steps:

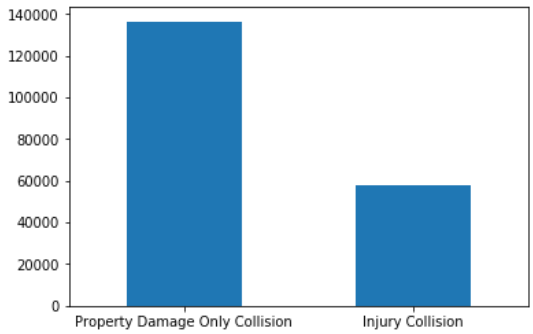
1. Setting data from source.
2. Understanding data: fields, registers, types.
3. Check for missing values.
4. Balance data. The next pictures show this stage.

```
1    136485
2     58188
Name: SEVERITYCODE, dtype: int64
```

```
Out[52]: 2    58188
         1    58188
         Name: SEVERITYCODE, dtype: int64
```

VISUALIZATION

To have a better understanding of the information, we can view it with the help of some plots, for example, in the next one we can appreciate the SEVERTY DESC values to see that most of the accidents had an important weight over Properly Damage Only Collision in contrast of the collisions that had injuries.

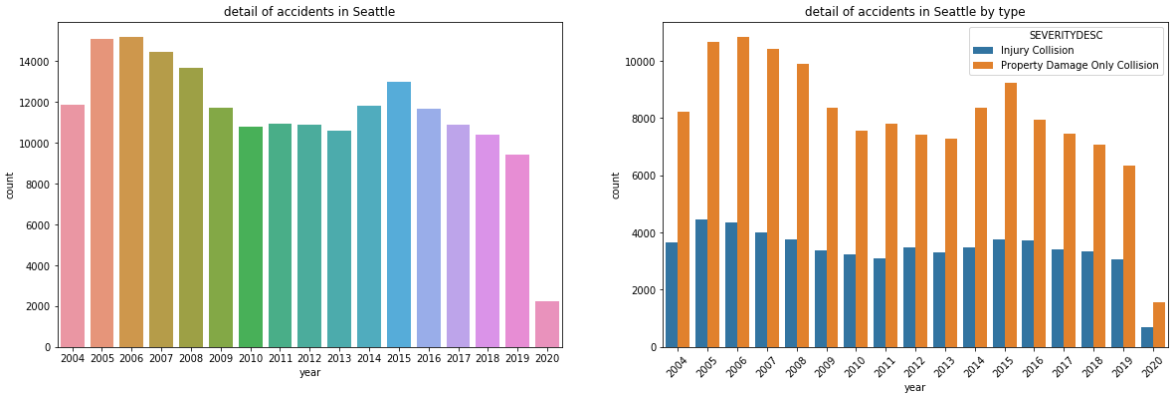


Graphic 1. Severity Description plot

Next, we can view how frequent were the accidents by the years, in the Graphic 2 there is general information and in Graphic 2.1, this data is divided by the SEVERITYDESC.

What information can we get of these plots?

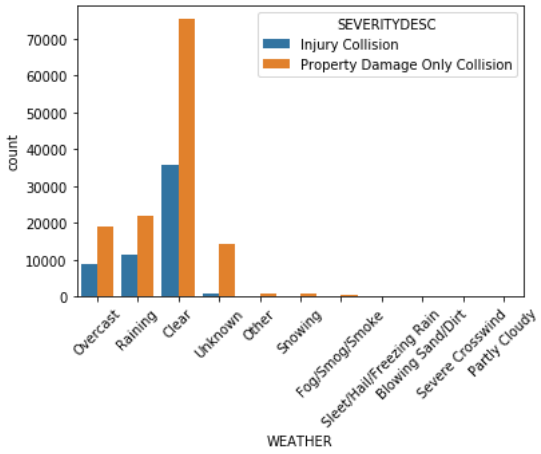
By all time, properly damaged only collisions are more recurrent, however in general terms, there is a decrease since the last 4 years, waiting to have the 2020 total information.



Graphic 2 and 2.1

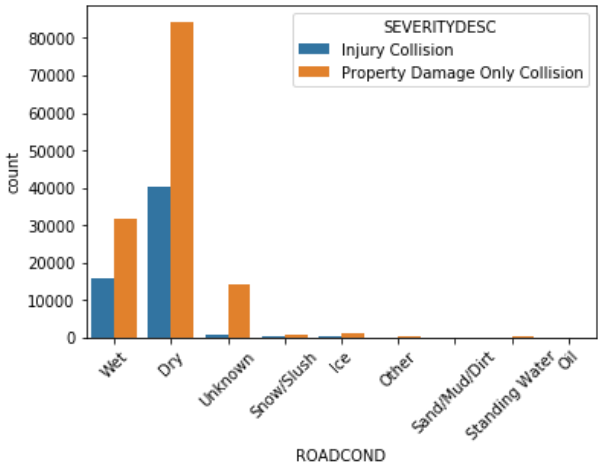
Now, we are going to analyze our most important independent variables to get some interesting facts:

Weather: the most recurrent accidents occur on clear days, when in theory drivers have enough visibility and conditions to go on the roads.



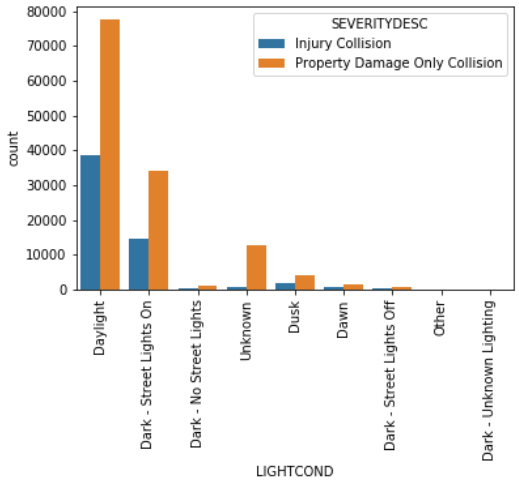
Graphic 3

Road conditions: the most recurrent accidents occur on dry roads, where there are no factors as water or snow that in theory cause many problems to drivers.



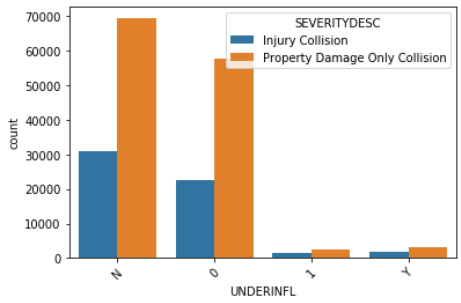
Graphic 4

Light conditions: most of the accidents occur on daylighted environments, and it is razonable to be followed by dark days with lights on.



Graphic 5

Under influency of drugs or alcohol: in most of the cases, drivers had no registered a substance that affected their state.



Graphic 6

METHODOLOGY

DATA PREPARATION

STEP 1:

It is important to do a data cleaning of our registers to make data readable and useful for the models and balance it.

STEP 2:

There were more than 30 cathegories or variables to analyze these future models, however the ones that are going to be used are: WEATHER, ROADCOND, LIGHTCOND, UNDERINFL AND COLLISION TYPE, they will be related to our target that is SEVERITYCODE.

```
COLLISIONTYPE     object
WEATHER           object
ROADCOND          object
LIGHTCOND         object
UNDERINFL         object
SEVERITYCODE       int64
dtype: object
```

STEP 3:

For the cathegoric variables, it will be necessary to apply a label encoding.

	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND	UNDERINFL
0	0	4	8	5	0
1	9	6	8	2	0
2	5	4	0	5	0
3	4	1	0	5	0
4	0	6	8	5	0

STEP 4:

When data is ready, a dataset with independent variables (X) is going to be created, by (Y) for the dependent ones. We will use a split function to train data by these rules: 70% for trainning 30% for testing but normalized

	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND	UNDERINFL
109717	0	1	0	5	0
9615	7	1	0	5	0
133991	3	1	0	5	0
76012	5	1	0	5	0
97913	9	10	7	8	0

MODELING

After the data preparation, we are going to use next models. When each model is developed, it will be evaluated with the next parameters:

- Accuracy
- Precision
- F1-SCORE

Description of the models:

- Logistic Regression: to classify data by calculating the probability of its classes
- Decision Tree: to classify by creating roots of subsets from data to take decisions
- KNN: to classify by catching its neighbours. Note: Default K=2 (Number of classes/SEVERITYCODES).

In [81]:
```
# MODELING

# Logistic Regression

from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(C=0.0001, solver='liblinear')
lr.fit(X_train, y_train)
lr
```

Out[81]:
```
LogisticRegression(C=0.0001, 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)
```

In [82]:
```
# Decision tree

from sklearn.tree import DecisionTreeClassifier

model_tree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
model_tree.fit(X_train, y_train)
model_tree
```

Out[82]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

In [83]:
```
# KNN
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 2).fit(X_train, y_train)
knn
```

RESULTS

In this last dataframe we resume the final results of the parameters that we measured for each model. As we can see the Jaccard Score equals more than 0.7 in the 3 models, however the best accuracy is for the Decision Tree. It also has a great F1_SCORE.

```
Results:

LR - Train Accuracy =   0.6989156435583794
LR - TesT Accuracy =   0.7363676379963024
DECISION TREE - Train Accuracy =  0.6989156435583794
DECISION TREEE - Test Accuracy =  0.7363676379963024
```

In the next table, we have another visualization and comparision of these results.

	Algorithm	Jaccard	F1-score	Precision
0	Logistic Regression	0.7	0.58	0.68
1	Decision Tree	0.7	0.58	0.68
2	KNN	0.74	0.67	0.74

DISCUSSIONS

After developing 3 models to compare them within their results, we can see that they have a behavour with almost the same level of performance, but in this case the Decision Tree had an small difference in contrast to Logistic Regression and KNN. This context would give to the Decision Tree the role to be implemented as the core of a future application.

CONCLUSION

This project had the purpose of understanding the data over an interesting perspective, because with the described process we could see how variables interact each other to generate a kind of relation that for research purposes can give us answsers about our environment. It was interesting to say that less than 30% of the files or attributes of this dataset could be useful to determine possible situations through a model, however we cannot loose the opportunity to continue improving these kind of projects in order to get more useful results and in the case of our world, reduce those costs that are invested in many accidents and mainly, the lives that we need to preserve.