Car accident severity

Coursera capstone project

Submitted by Ali Bou-Imajdil – October 2020

1. Introduction :

"Should I stay or should I go?" This famous song could also be the title of our problem. Whenever I need to ride my car, to go from point A to point B, I wish I knew how the traffic is and if I am running a risk of being involved in a car accident. The audience is actually anyone driving their car willing to stay on the safe side. If our model tells us that we have a high probability to meet (including being involved) in a very severe accident, maybe it is better to stay home that day. Choices in that case could be postponing one's duties for example. On the other hand, if the model tells us that we could be involved in a less severe accident, we can willingly chose to run that risk and drive our car that very day.

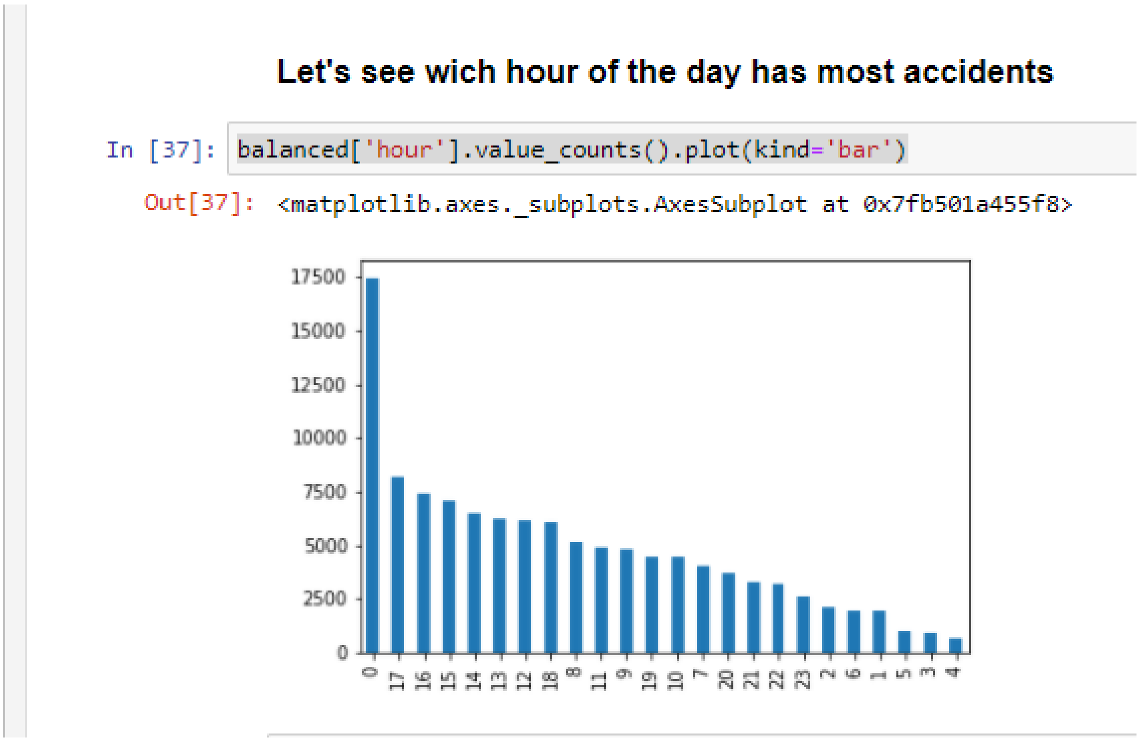
2. Our Data :

The Data we will be using is the example dataset provided in the course. The file name is data-collision.csv.

We will use the columns "severitycode" as our label (2 being very severe with injuries and 1 being less severe with property damage only) and our predictor variables are "weather", "roadcond" and "lightcond". We will also use the column date to extract the hour of the day and the day of the week to feed our algorithm. The categorical variables will be dummified to turn them into Boolean variables.

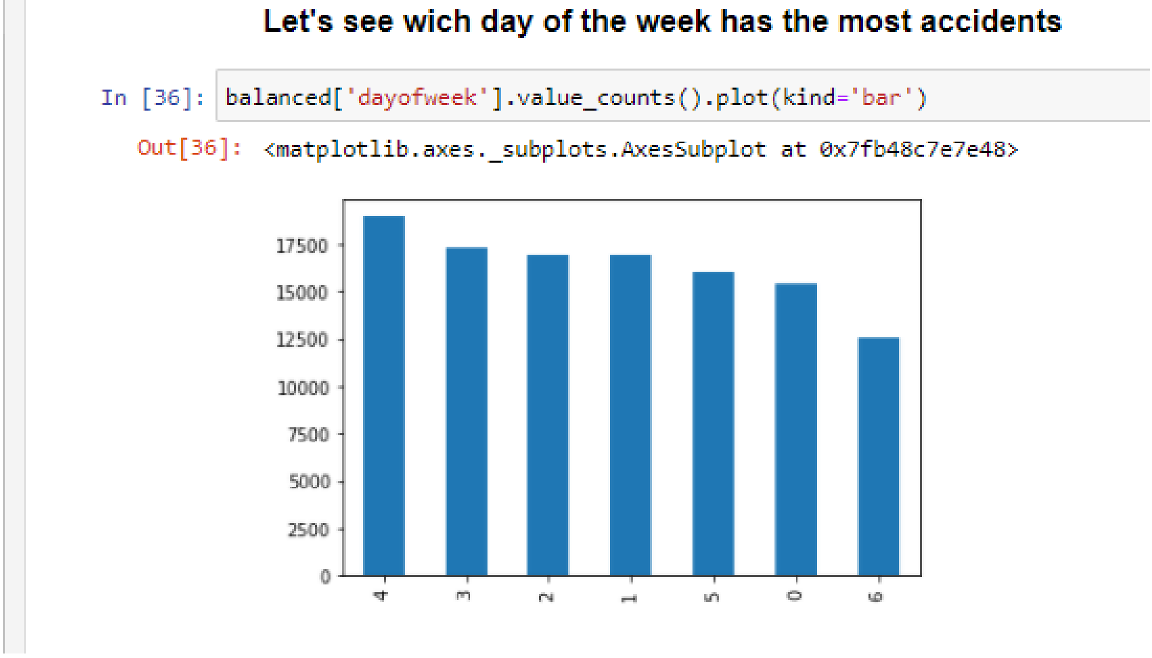
We will as well balance our data to have the same proportion of very severe and less severe accidents and we will remove the rows with missing information.

We therefore can check our data :



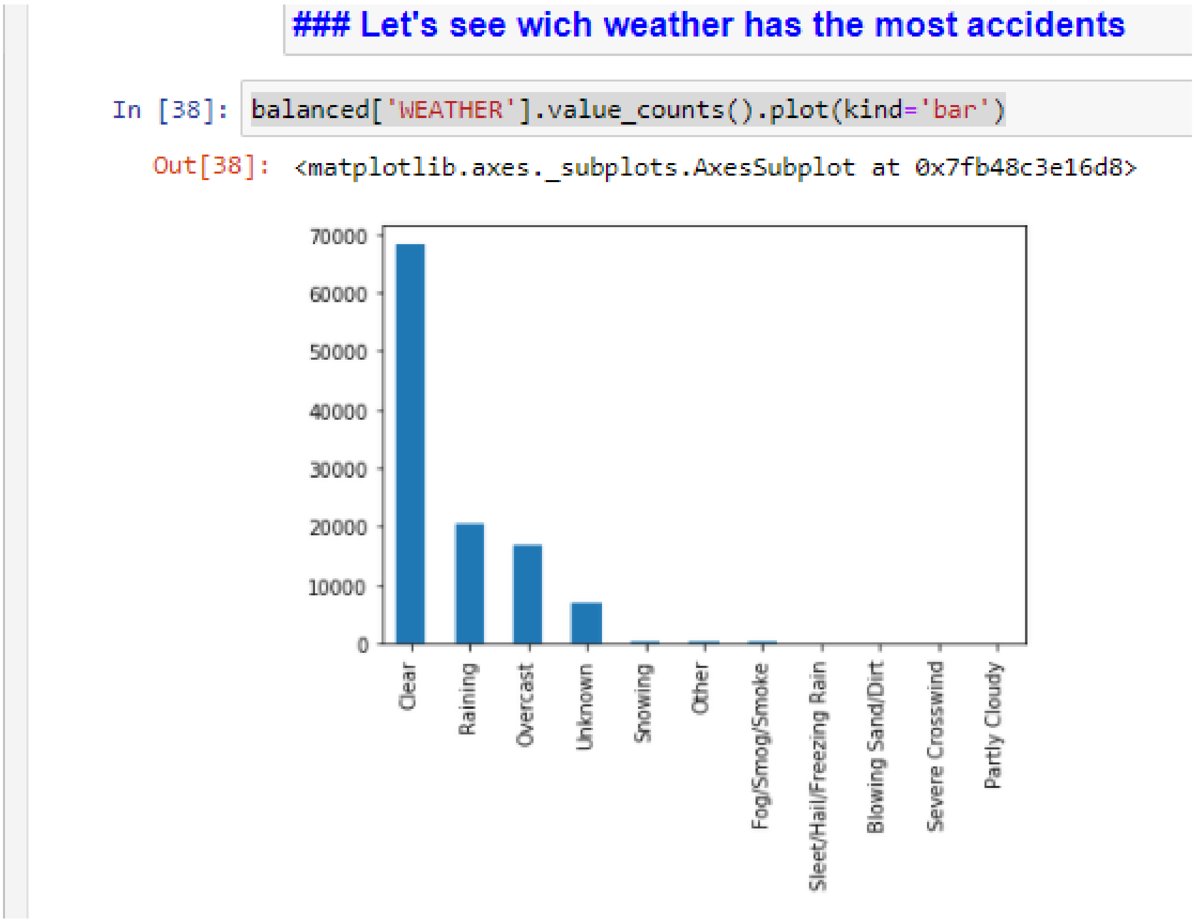
Let's see wich hour of the day has most accidents

```
In [37]: balanced['hour'].value_counts().plot(kind='bar')

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb501a455f8>
```



Let's see wich day of the week has the most accidents

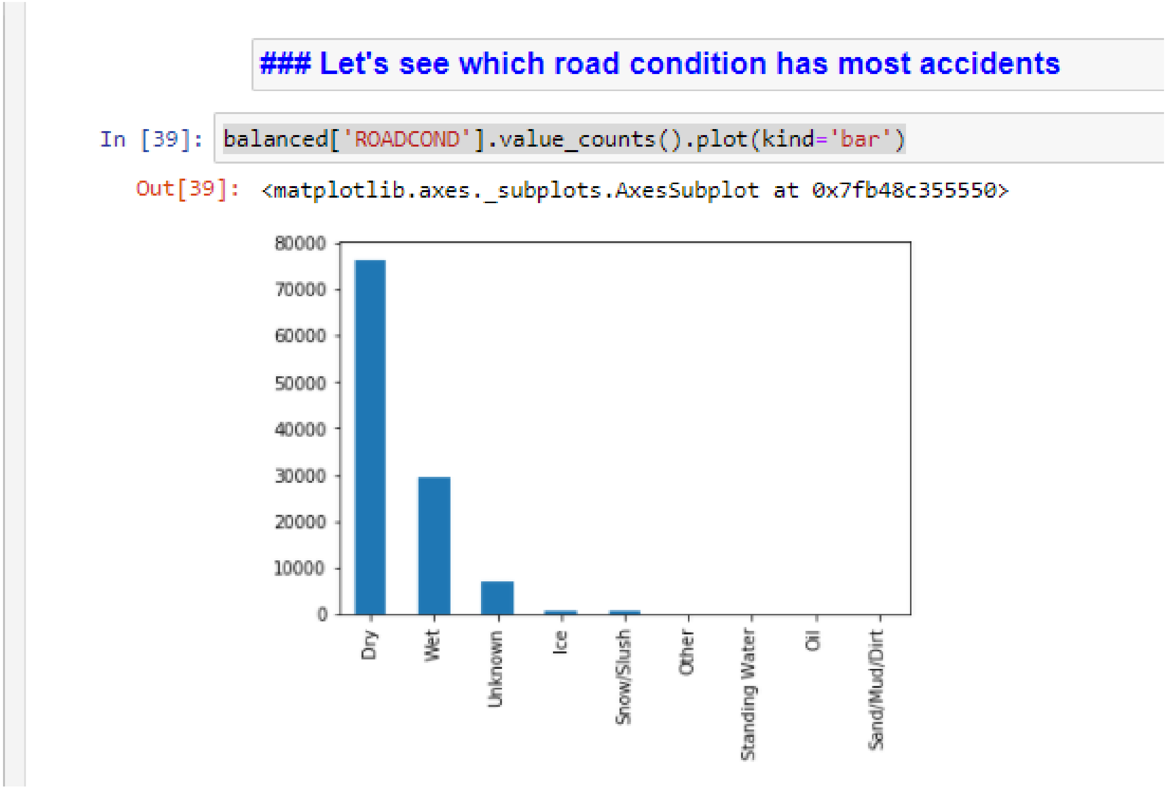
```
In [36]: balanced['dayofweek'].value_counts().plot(kind='bar')

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb48c7e7e48>
```



```
### Let's see wich weather has the most accidents
```

```
In [38]: balanced['WEATHER'].value_counts().plot(kind='bar')
```

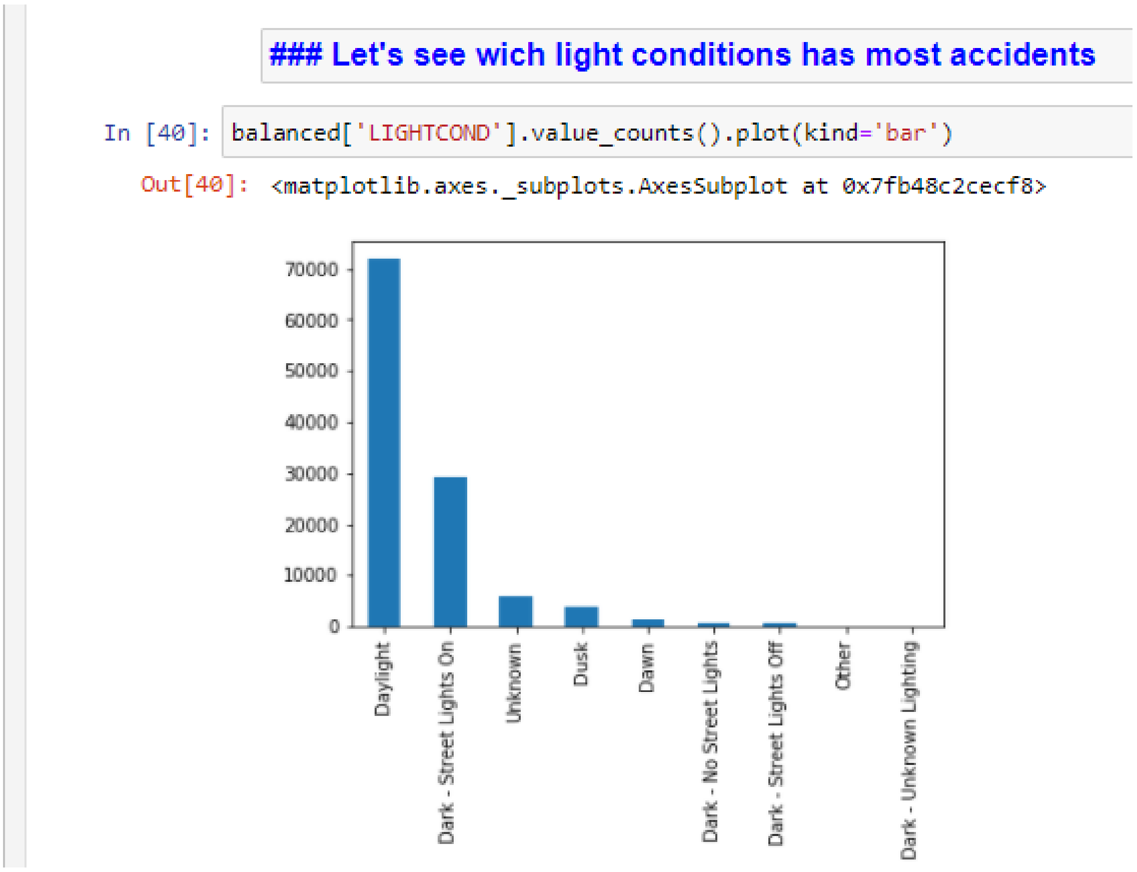
```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb48c3e16d8>
```



```
### Let's see which road condition has most accidents
```

```
In [39]: balanced['ROADCOND'].value_counts().plot(kind='bar')
```

```
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb48c355550>
```



```
### Let's see wich light conditions has most accidents

In [40]: balanced['LIGHTCOND'].value_counts().plot(kind='bar')

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb48c2cecf8>
```

3. Methodology section :

Once our data is properly prepared, we can feed it to our machine learning algorithms, namely K nearest neighbors, Decision Tree, SVM and logistic regression. We will of course chose the model with the best accuracy on the test set. However, since we have a large data set, computational time is important as well. For instance, finding the best K for KNN takes 45 minutes and running the KNN with the best K takes 7 minutes. I tried SVM as well and it took more than two hours without converging, so it will be disqualified in our report.

4. Results

Test set KNN Accuracy: 0.5368076653423697

Test set KNN jaccard : 0.37081861830879725

Test set KNN f1_score : 0.5367493260892071

Test set Decission Tree Accuracy: 0.558424865622809

Test set Decission Tree jaccard : 0.4827539009033671

Test set Decission Tree f1_score : 0.5242714956667659

Test set Logistic regression Accuracy: 0.5622809067539145

Test set Logistic regression jaccard : 0.5011153654070252

Test set Logistic regression f1_score : 0.512653273275834

A part from SVM that did not converge, KNN is the slowest of the three remaining models. It is not as good as a “simple” decision tree with a depth of 4 or a logistic regression.

Regarding the accuracy results of our models, and taking into account convergence time, it seems that logistic regression is doing pretty well.

We should always chose the most simple model whenever we can

5. Conclusion

Since our data is balanced fifty-fifty and logistic regression accuracy is 56%, we can claim that our model gives us an edge in predicting accident severity. I f we look at the logistic regression coefficient, we will figure out that most severe accidents happen in daylight and when road conditions are unknown. Our conclusion is that drivers should be careful at anytime