Predicting car accident severity

IBM capstone project 2020

Introduction

The number of annual road traffic deaths has reached 1.35 million worldwide[1]. Road traffic injuries are the first cause of death of people aged 5-29 years. Every 24 seconds someone dies on the road. Those numbers are astonishingly high but there is another cost of these accidents : the medical costs for the survivors. Injured people often need medical care whereas for short or long term injuries. The economic consequences of motor vehicle crashes have been estimated between 1% and 3% of the respective GNP of the world countries, reaching a total over $500 billion[2]

This project aims to determine if we can predict the severity of a car accident using features as date, time, location, weather conditions and road conditions.

[1] World Health Organization (WHO) - Global Status Report on Road Safety 2018 :
https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/external

[2] World Health Organization (WHO) - Global Plan for the Decade of Action for Road Safety 2011-2020 :
https://www.who.int/roadsafety/decade_of_action/plan/plan_english.pdf?ua=1

Interests

If such a prediction is possible, we may be able to find a correlation between the severity and a specific feature which could lead to improve the roads safety. Addressing this problem the right way, whereas it is prevention messages, adding traffic signage or redesigning roads would mean less road traffic injuries and less money spent on medical care. For a government or local authority taking care of the medical bill, it means they could spend this budget for other purposes.

Data 1/2

The dataset used is the US Accidents (3.5 million records)[3] dataset from Kaggle.

Description

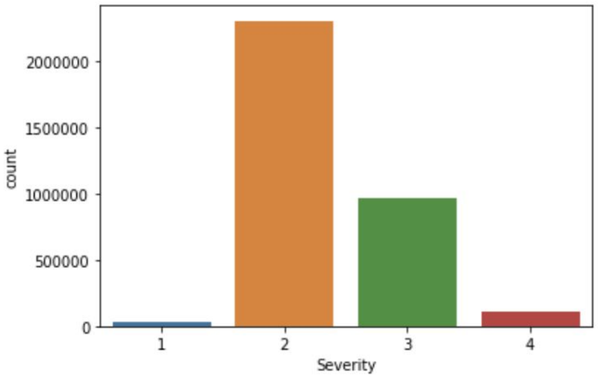
This dataset contains accident data covering 49 states of the USA. It has been collected from February 2016 and June 2020 using multiple Traffic APIs. It contains features as start and end time, location, weather conditions and road conditions

[3] Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, 2019. Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights." In proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.

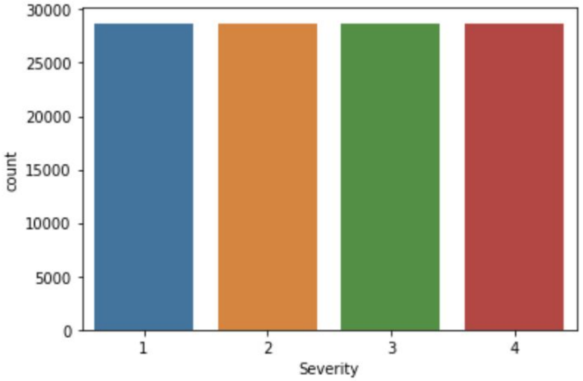
Data 2/2

Getting a balanced dataset

We can see the dataset is unbalanced for the severity feature. We will use an undersampling method to get a balanced dataset



Severity repartition before undersampling



Severity repartition after undersampling

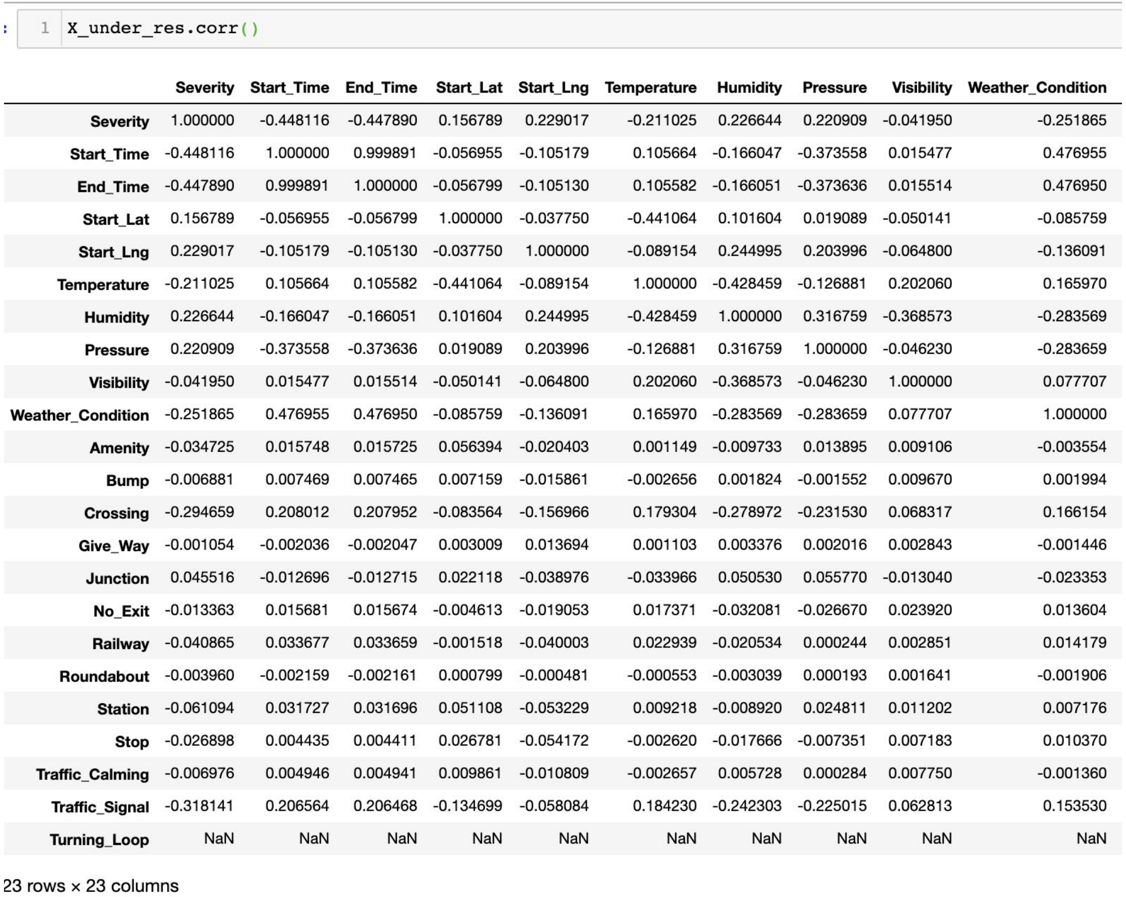
Methodology 1/3

Finding correlation

We will use the Pearson Correlation to find out if some specific variables have an interdependence with the Severity of an accident

We can see that there is no variable that have a significant interdependence with the Severity feature.

As 2 variables: Humidity and pressure have the best results, we will use the meteorological variables and see if we can have good results predicting the severity of a car accident with them.



```
X_under_res.corr()
```

	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	Temperature	Humidity	Pressure	Visibility	Weather_Condition
Severity	1.000000	-0.448116	-0.447890	0.156789	0.229017	-0.211025	0.226644	0.220909	-0.041950	-0.251865
Start_Time	-0.448116	1.000000	0.999891	-0.056955	-0.105179	0.105664	-0.166047	-0.373558	0.015477	0.476955
End_Time	-0.447890	0.999891	1.000000	-0.056799	-0.105130	0.105582	-0.166051	-0.373636	0.015514	0.476950
Start_Lat	0.156789	-0.056955	-0.056799	1.000000	-0.037750	-0.441064	0.101604	0.019089	-0.050141	-0.085759
Start_Lng	0.229017	-0.105179	-0.105130	-0.037750	1.000000	-0.089154	0.244995	0.203996	-0.064800	-0.136091
Temperature	-0.211025	0.105664	0.105582	-0.441064	-0.089154	1.000000	-0.428459	-0.126881	0.202060	0.165970
Humidity	0.226644	-0.166047	-0.166051	0.101604	0.244995	-0.428459	1.000000	0.316759	-0.368573	-0.283569
Pressure	0.220909	-0.373558	-0.373636	0.019089	0.203996	-0.126881	0.316759	1.000000	-0.046230	-0.283659
Visibility	-0.041950	0.015477	0.015514	-0.050141	-0.064800	0.202060	-0.368573	-0.046230	1.000000	0.077707
Weather_Condition	-0.251865	0.476955	0.476950	-0.085759	-0.136091	0.165970	-0.283569	-0.283659	0.077707	1.000000
Amenity	-0.034725	0.015748	0.015725	0.056394	-0.020403	0.001149	-0.009733	0.013895	0.009106	-0.003554
Bump	-0.006881	0.007469	0.007465	0.007159	-0.015861	-0.002656	0.001824	-0.001552	0.009670	0.001994
Crossing	-0.294659	0.208012	0.207952	-0.083564	-0.156966	0.179304	-0.278972	-0.231530	0.068317	0.166154
Give_Way	-0.001054	-0.002036	-0.002047	0.003009	0.013694	0.001103	0.003376	0.002016	0.002843	-0.001446
Junction	0.045516	-0.012696	-0.012715	0.022118	-0.038976	-0.033966	0.050530	0.055770	-0.013040	-0.023353
No_Exit	-0.013363	0.015681	0.015674	-0.004613	-0.019053	0.017371	-0.032081	-0.026670	0.023920	0.013604
Railway	-0.040865	0.033677	0.033659	-0.001518	-0.040003	0.022939	-0.020534	0.000244	0.002851	0.014179
Roundabout	-0.003960	-0.002159	-0.002161	0.000799	-0.000481	-0.000553	-0.003039	0.000193	0.001641	-0.001906
Station	-0.061094	0.031727	0.031696	0.051108	-0.053229	0.009218	-0.008920	0.024811	0.011202	0.007176
Stop	-0.026898	0.004435	0.004411	0.026781	-0.054172	-0.002620	-0.017666	-0.007351	0.007183	0.010370
Traffic_Calming	-0.006976	0.004946	0.004941	0.009861	-0.010809	-0.002657	0.005728	0.000284	0.007750	-0.001360
Traffic_Signal	-0.318141	0.206564	0.206468	-0.134699	-0.058084	0.184230	-0.242303	-0.225015	0.062813	0.153530
Turning_Loop	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

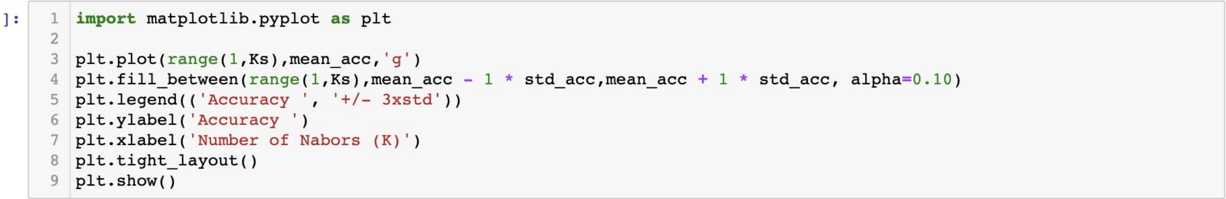
23 rows × 23 columns

Methodology 2/3

K-Nearest Neighbors

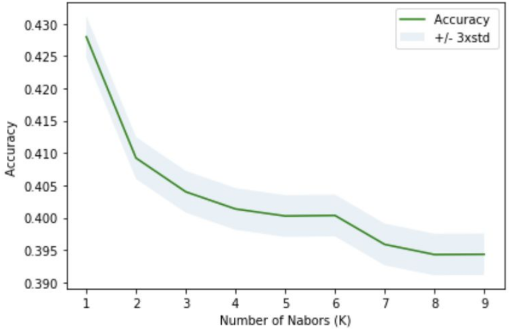
K-Nearest Neighbors is an algorithm for supervised learning. Where the data is 'trained' with data points corresponding to their classification. Once a point is to be predicted, it takes into account the 'K' nearest points to it to determine its classification.

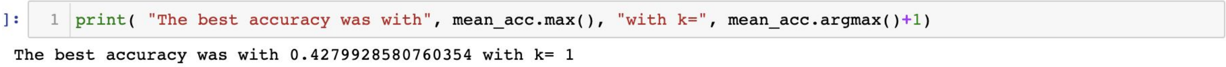
Plot model accuracy for Different number of Neighbors



```
import matplotlib.pyplot as plt

plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Nabors (K)')
plt.tight_layout()
plt.show()
```





```
print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

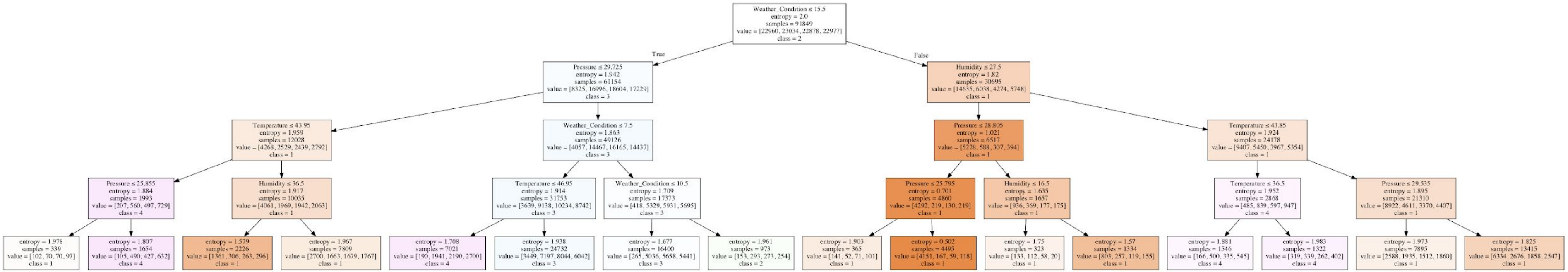
```
The best accuracy was with 0.4279928580760354 with k= 1
```

Methodology 3/3

Decision Tree

The basic intuition behind a decision tree is to map out all possible decision paths in the form of a tree

Each internal node corresponds to a test Each branch corresponds to a result of the test Each leaf node assigns a classification We will first create an instance of the DecisionTreeClassifier called severityTree. Inside of the classifier, specify criterion="entropy" so we can see the information gain of each node.



Weather_Condition ≤ 15.5 entropy = 2.0 samples = 91849 value = [22960, 23034, 22878, 22977] class = 2
True
False
Pressure ≤ 29.725 entropy = 1.942 samples = 61154 value = [8325, 16996, 18604, 17229] class = 3
Humidity ≤ 27.5 entropy = 1.82 samples = 30695 value = [14635, 6038, 4274, 5748] class = 1
Temperature ≤ 43.95 entropy = 1.959 samples = 12028 value = [4268, 2529, 2439, 2792] class = 1
Weather_Condition ≤ 7.5 entropy = 1.863 samples = 49126 value = [4057, 14467, 16165, 14437] class = 3
Pressure ≤ 28.805 entropy = 1.021 samples = 6517 value = [5228, 588, 307, 394] class = 1
Temperature ≤ 43.85 entropy = 1.924 samples = 24178 value = [9407, 5450, 3967, 5354] class = 1
Pressure ≤ 25.855 entropy = 1.884 samples = 1993 value = [207, 560, 497, 729] class = 4
Humidity ≤ 36.5 entropy = 1.917 samples = 10035 value = [4061, 1969, 1942, 2063] class = 1
Temperature ≤ 46.95 entropy = 1.914 samples = 31753 value = [3639, 9138, 10234, 8742] class = 3
Weather_Condition ≤ 10.5 entropy = 1.709 samples = 17373 value = [418, 5329, 5931, 5695] class = 3
Pressure ≤ 25.795 entropy = 0.701 samples = 4860 value = [4292, 219, 130, 219] class = 1
Humidity ≤ 16.5 entropy = 1.635 samples = 1657 value = [936, 369, 177, 175] class = 1
Temperature ≤ 36.5 entropy = 1.952 samples = 2868 value = [485, 839, 597, 947] class = 4
Pressure ≤ 29.535 entropy = 1.895 samples = 21310 value = [8922, 4611, 3370, 4407] class = 1
entropy = 1.978 samples = 339 value = [102, 70, 70, 97] class = 1
entropy = 1.807 samples = 1654 value = [105, 490, 427, 632] class = 4
entropy = 1.579 samples = 2226 value = [1361, 306, 263, 296] class = 1
entropy = 1.967 samples = 7809 value = [2700, 1663, 1679, 1767] class = 1
entropy = 1.708 samples = 7021 value = [190, 1941, 2190, 2700] class = 4
entropy = 1.938 samples = 24732 value = [3449, 7197, 8044, 6042] class = 3
entropy = 1.677 samples = 16400 value = [265, 5036, 5658, 5441] class = 3
entropy = 1.961 samples = 973 value = [153, 293, 273, 254] class = 2
entropy = 1.903 samples = 365 value = [141, 52, 71, 101] class = 1
entropy = 0.502 samples = 4495 value = [4151, 167, 59, 118] class = 1
entropy = 1.75 samples = 323 value = [133, 112, 58, 20] class = 1
entropy = 1.57 samples = 1334 value = [803, 257, 119, 155] class = 1
entropy = 1.881 samples = 1546 value = [166, 500, 335, 545] class = 4
entropy = 1.983 samples = 1322 value = [319, 339, 262, 402] class = 4
entropy = 1.973 samples = 7895 value = [2588, 1935, 1512, 1860] class = 1
entropy = 1.825 samples = 13415 value = [6334, 2676, 1858, 2547] class = 1

Conclusion

In this study, we analyzed the relationship between the severity of an accident and the meteorological conditions. We built and compared two different models to predict the severity of a car accident.

As we can see, both KNN and decision tree did not give us good results. The results for these models are less efficient than a random which would give us 0,5.

Unfortunately, the results were not conclusive and we were not able to make a good model predicting the car accident severity from meteorological data.