Predicting Severity in Car Crashes

By: DUDAM CHANDANA

1. Introduction

1.1 Background

Currently, driving through the streets in vehicles, whether public or owned, is one of the most daily activities that people carry out. Even so, traffic accidents cause about 40,000 deaths a year only in the United States.

1.2 Solution

To reduce the frequency and the severity of car collisions in a community, an algorithm must be developed to predict the severity of an accident given the current weather, road and visibility conditions.

When conditions are prone to fatal accidents, the model developed will allow drivers to be alerted to the likelihood of an accident and to be reminded to drive carefully.

2. Data acquisition and cleaning

the data set used contains data about car accidents provided by SPD and recorded by Traffic Records. The label for the data set is 'SEVERITYCODE' (Target variable), which describes the fatality of an accident, it can take 4 values:

- 0: Little to no Probability
- 1: Very Low Probability - Chance or Property Damage
- 2: Low Probability - Chance of Injury
- 2b: Mild Probability - Chance of Serious Injury
- 3: High Probability - Chance of Fatality

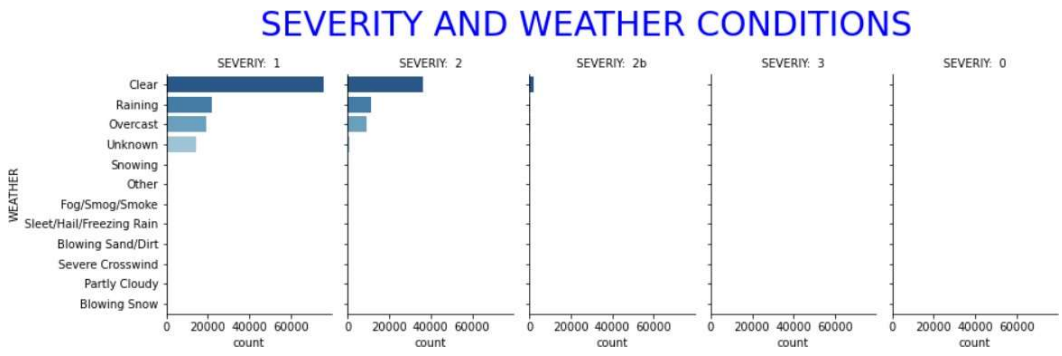
Dataset: https://opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0.csv

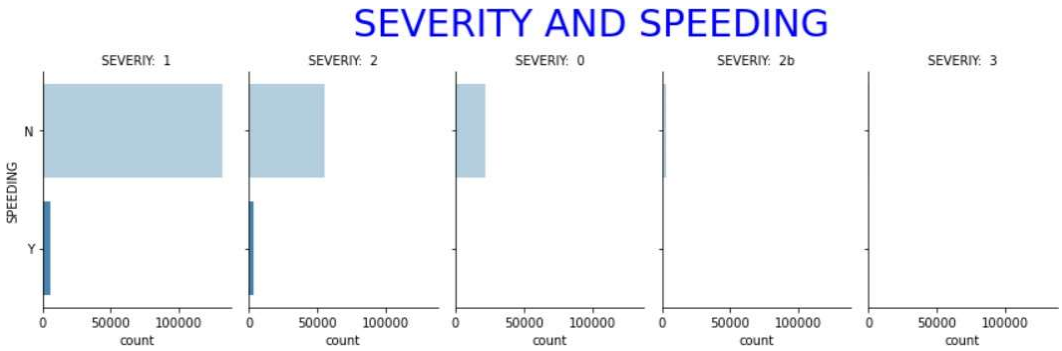
Metadata: https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf

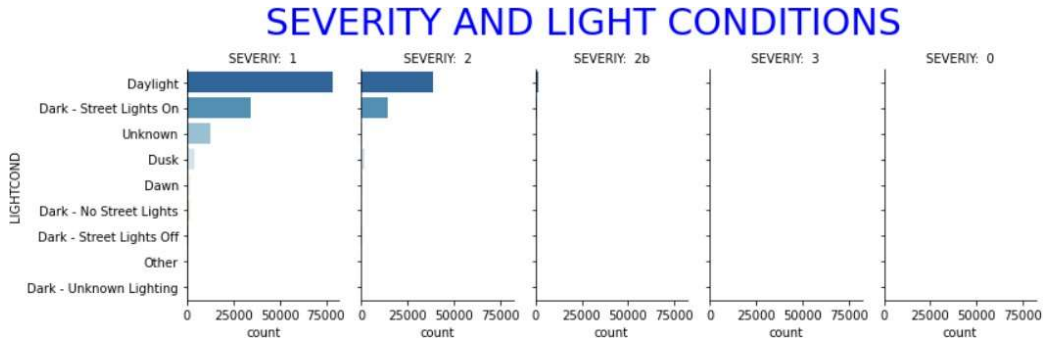
after some data exploration and discarding variables corresponding to id's, it is decided to work with the following variables:

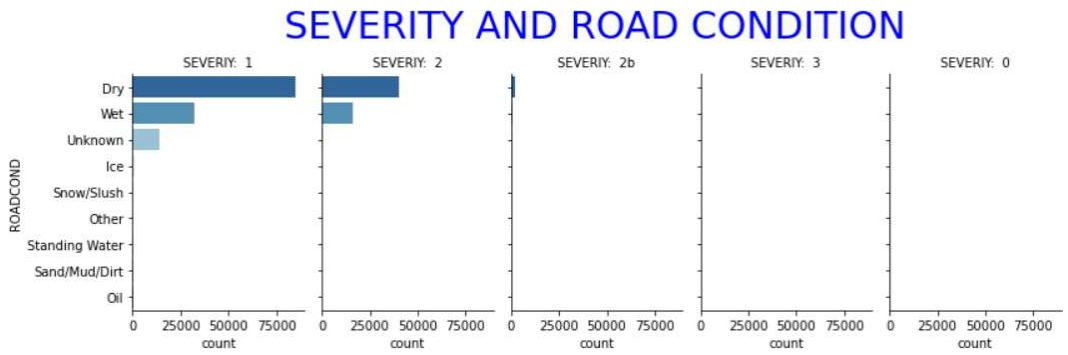
Variables	Description
WEATHER	A description of the weather conditions during the time of the collision
SPEEDING	Whether or not speeding was a factor in the collision. (Y/N)
LIGHTCOND	The light conditions during the collision
ROADCOND	The condition of the road during the collision
JUNCTION TYPE	Category of junction at which collision took place
PERSONCOUNT	The total number of people involved in the collision
VEHCOUNT	The number of vehicles involved in the collision

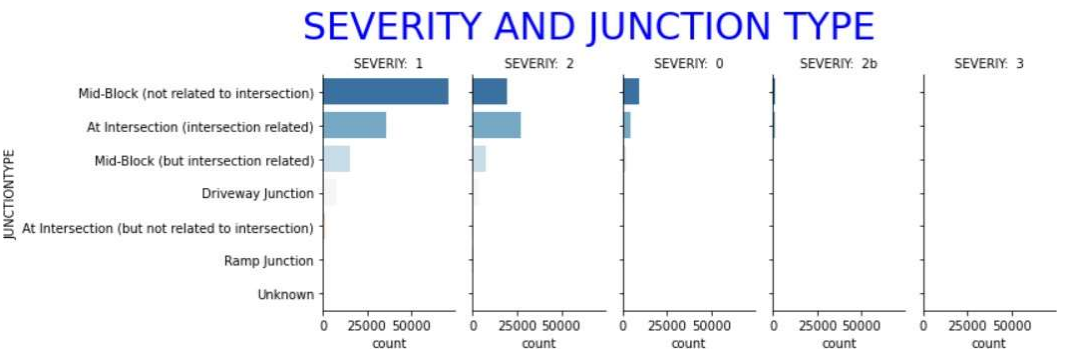
Each selected variable is analyzed:











the values of the categorical variables are replaced by numerical values, and the continuous variables are combined into categories according to the quartiles of each variable to later convert the dataframe into a numpy array and apply the models from the sklearn library, finally the data is balanced.

Code Mapping

WEATHER	CODE	----	SPEEDING	CODE	----	LIGHTCOND	CODE
Blowing Sand/Dirt	0	----	N	0	----	Dark - No Street Lights / Street Lights Off	0
Blowing Snow	1	----	Y	1	----	Dark - Street Lights On	1
Clear	2	----			----	Dark - Unknown Lighting	2
Fog/Smog/Smoke	3	----			----	Dawn	3
Overcast	4	----			----	Daylight	4
Partly Cloudy	5	----			----	Dusk	5
Raining	6	----			----		
Severe Crosswind	7	----			----		
Sleet/Hail/Freezing Rain	8	----			----		
Snowing	9	----			----		

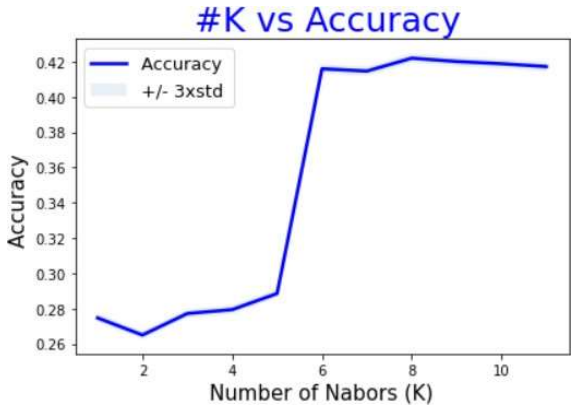
ROADCOND	CODE	---	JUNCTIONTYPE	CODE	---	PERSONCOUNT	CODE	---	VEHCOUNT	CODE
Dry	0	---	At Intersection (but not related to intersection)	0	---	0 <= x < 2	0	---	0 <= x < 2	0
Ice	1	---	At Intersection (intersection related)	1	---	2 <= x < 3	1	---	2 <= x < 3	1
Oil	2	---	Driveway Junction	2	---	3 <= x < 93	2	---	3 <= x < 93	2
Other	3	---	Mid-Block (but intersection related)	3	---					
Sand/Mud/Dirt	4	---	Mid-Block (not related to intersection)	4	---					
Snow/Slush	5	---	Ramp Junction	5	---					
Standing Water	6	---			---					
Wet	7	---			---					

3. Predictive Modeling

There are two types of models, regression and classification Since the target variable is categorical, the following categorical models are applied:

3.1 K nearest neighbor (KNN)

first the model is run with different values of K to determine which K is more appropriate:

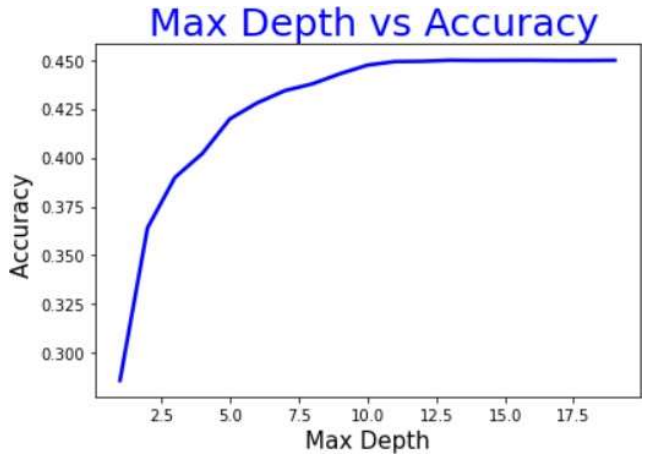


K = 8 is chosen to perform the model. Then the model in fit with the train data

```
k = 8
neigh = KNeighborsClassifier(n_neighbors = k).fit(x_train,y_train)
yhat_knn = neigh.predict(x_test)
```

3.2 Decision Tree

The model in run with different maximum depth values to see when the accuracy stabilizes



Max depth = 12 is chosen and the model is fit with train values

```
dectree = DecisionTreeClassifier(criterion="entropy", max_depth = 12)
dectree.fit(x_train,y_train)
predtree = dectree.predict(x_test)
```

3.3 Logistic Regression

The model is fit with train values:

```
lr = LogisticRegression(C=0.01, solver='liblinear').fit(x_train,y_train)
yhat_lr = lr.predict(x_test)
yhat_prob_lr = lr.predict_proba(x_test)
```

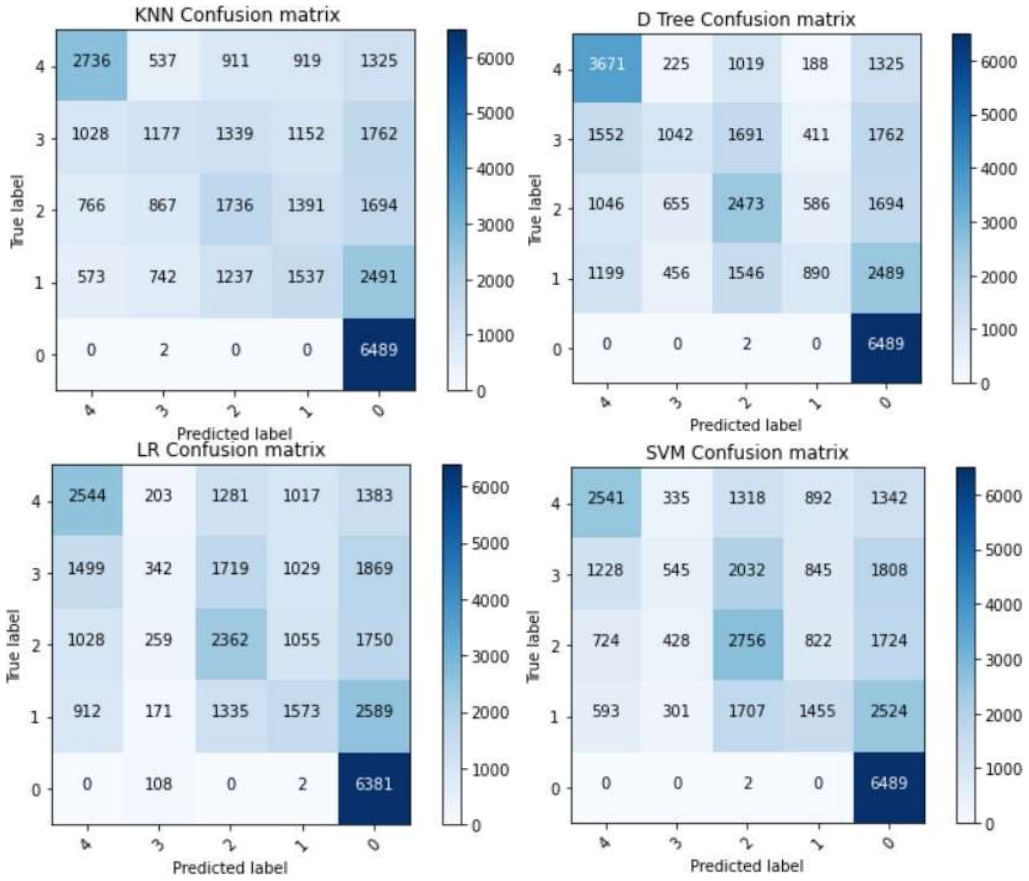
3.4 Support Vector Machine (SVM)

The model is fit with train values:

```
clf = svm.SVC(kernel='rbf')
clf.fit(x_train, y_train)
```

4. Model Evaluation

for all models a confusion matrix is made:



all models are evaluated according to 2 parameters, Score and F1 Score

	Score	F1 Score
KNN	0.422	0.384
Decision Tree	0.449	0.397
Logistic Regression	0.407	0.353
SVM	0.425	0.375

5. Conclusion

Purpose of this project was to predict the severity of a car collission based on the current weather, road and visibility conditions through the development of an algorithm in which all the principles of data science were developed.
Using different prediction models it was obtained that the most suitable model is the decision tree with an accuracy of 45%, this is not a desired value as it is too low to consider that the prediction model is correct.
work should continue on the model to improve its accuracy by prior manipulation of the data, rethinking the variables that are considered important.