Predict Car Accidents Severity

Liuqi Qian

August 2020

Introduction

As the most commonly used transports, automobiles play an important role in daily life. People drive cars to work, study, travel, and even move to a new house. With the widespread use of automobiles, the possibility of traffic accidents increases inevitably. Besides, the environmental factors including weather, location light, and more will also affect the probability of accidents.

However, sometimes emergency departments are hard to evaluate severity when they received the call from on-site which may cause irreparable loss and even people's death. Based on the situation, predict the severity is one of the crucial points.

To help the people who are involved in the accidents, this project will utilize certain know conditions to predict the severity of the accidents and reduce the risks of accidents happen by taking actions.

Data Source

To address the problem, the weekly collision from 2004 to present in Seattle is going to be used in this case. The original dataset is a csv file and can be found here. Also, there is a metedata form to give the basic information about the dataset.

Data Description and Cleaning

There are total 194,673 rows and 38 columns in the original dataset.

Right is the list of all the columns (features):

FIELD	TYPE	FIELD	TYPE	FIELD	TYPE	FIELD	TYPE
SEVERITYCODE	int64	X	float64	Y	float64	OBJECTID	int64
INCKEY	int64	COLDETKEY	int64	REPORTNO	object	STATUS	object
ADDRTYPE	object	INTKEY	float64	LOCATION	object	EXCEPTRSNCODE	object
EXCEPTRSNDESC	object	SEVERITYCODE.1	int64	SEVERITYDESC	object	COLLISIONTYPE	object
PERSONCOUNT	int64	PEDCOUNT	int64	PEDCYLCOUNT	int64	VEHCOUNT	int64
INCDATE	object	INCDTTM	object	JUNCTIONTYPE	object	SDOT_COLCODE	int64
SDOT_COLDESC	int64	INATTENTIONIND	object	UNDERINFL	object	WEATHER	object
ROADCOND	object	LIGHTCOND	object	PEDROWNOTGRNT	object	SDOTCOLNUM	float64
SPEEDING	object	ST_COLCODE	object	ST_COLDESC	object	SEGLANEKEY	int64
CROSSWALKKEY	int64	HITPARKEDCAR	object				

In this project, the first column "SEVERITYCODE" is the target variable that will be predicted by other fields. According to the metadata file, there are a total of 5 codes that correspond to the severity. However, there are only two types recorded in the dataset: code 1 and code 2. There is 136,485 number of code 1 and it stands for "prop damage". 58,188 observations represent "injury" as code 2.

After checking the NA number in each column, there are seven fields that have more than 40% of the missing values including "INTKEY", "EXCEPTRSNCODE", "EXCEPTRSNDESC", "INATTENTIONIND", "PEDROWNOTGRNT", "SDOTCOLNUM", SPEEDING". These columns will be dropped in this project.

```
# check the na value
nulls = df.isnull().sum() /df.shape[0]*100
nulls = nulls[nulls > 40]
print(nulls)
```

```
INTKEY            66.574718
EXCEPTRSNCODE     56.434123
EXCEPTRSNDESC     97.103861
INATTENTIONIND    84.689710
PEDROWNOTGRNT     97.602646
SDOTCOLNUM        40.959455
SPEEDING          95.205807
dtype: float64
```

Also, following columns will be kept or dropped based on their reasons:

- The "WEATHER" column is related to the "ROADCOND" so only "ROADCOND" will be kept
- "X", "Y" columns are describing the geographic point of the "LOCATION" so only "LOCATION" will be kept
- "INCDTTM" includes the information in the "INCDATE" and only "INCDTTM" will be kept
- "PEDCOUNT" and "PEDCYLCOUNT" are related to "COLLISIONTYPE" and only "COLLISIONTYPE" will be kept
- "ST_COLCODE" includes the information in the "HITPARKEDCAR" so "HITPARKEDCAR" will be dropped
- "COLLISTIONTYPE" includes the information about "STOD_COLCODE", "STOD_COLDESC", "ST_COLCODE" as well as "ST_CODESC", and only "COLLISTIONTYPE" will be kept
- "JUNCTIONTYPE" is similar to "ADDTYPE" and only "ADDTYPE" will be kept
- "STATUS" is meaningless in this project and will be dropped
- "LOCATION" is too massive in this project and will be dropped

After dropping the columns, there are only 9 columns left. Following is the screenshot of top 5 rows:

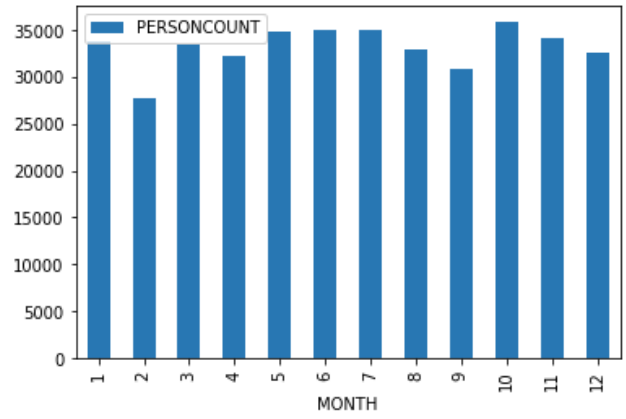
	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	VEHCOUNT	INCDTTM	UNDERINFL	ROADCOND	LIGHTCOND
0	2	Intersection	Angles	2	2	3/27/13 14:54	N	Wet	Daylight
1	1	Block	Sideswipe	2	2	12/20/06 18:55	0	Wet	Dark - Street Lights On
2	1	Block	Parked Car	4	3	11/18/04 10:20	0	Dry	Daylight
3	1	Block	Other	3	3	3/29/13 09:26	N	Dry	Daylight
4	2	Intersection	Angles	2	2	1/28/04 08:04	0	Wet	Daylight

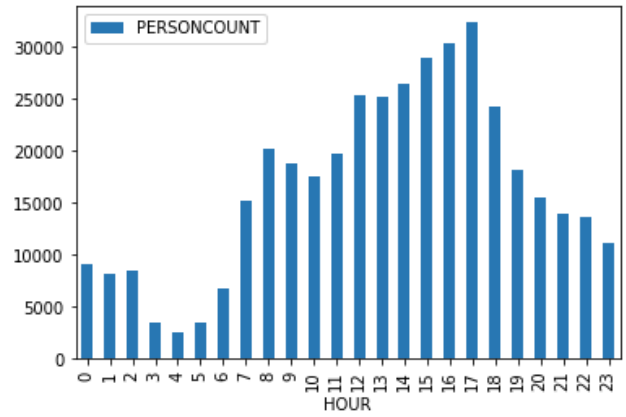
Since there are still many NA values and there are total 194673 records, the rows which contains NA values will be dropped. And finally there are 187,609 rows with 9 columns.

Convert Data Type

Since the "INCDTTM" is meaningless but the month and the hour time makes senses, we will extract the month and hour value from the "INCDTTM" column. And create two new columns, “MONTH” and “HOUR”. Followings are the bar chart shows the person count bar plot that group by month and hour.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb3671d38d0>
```

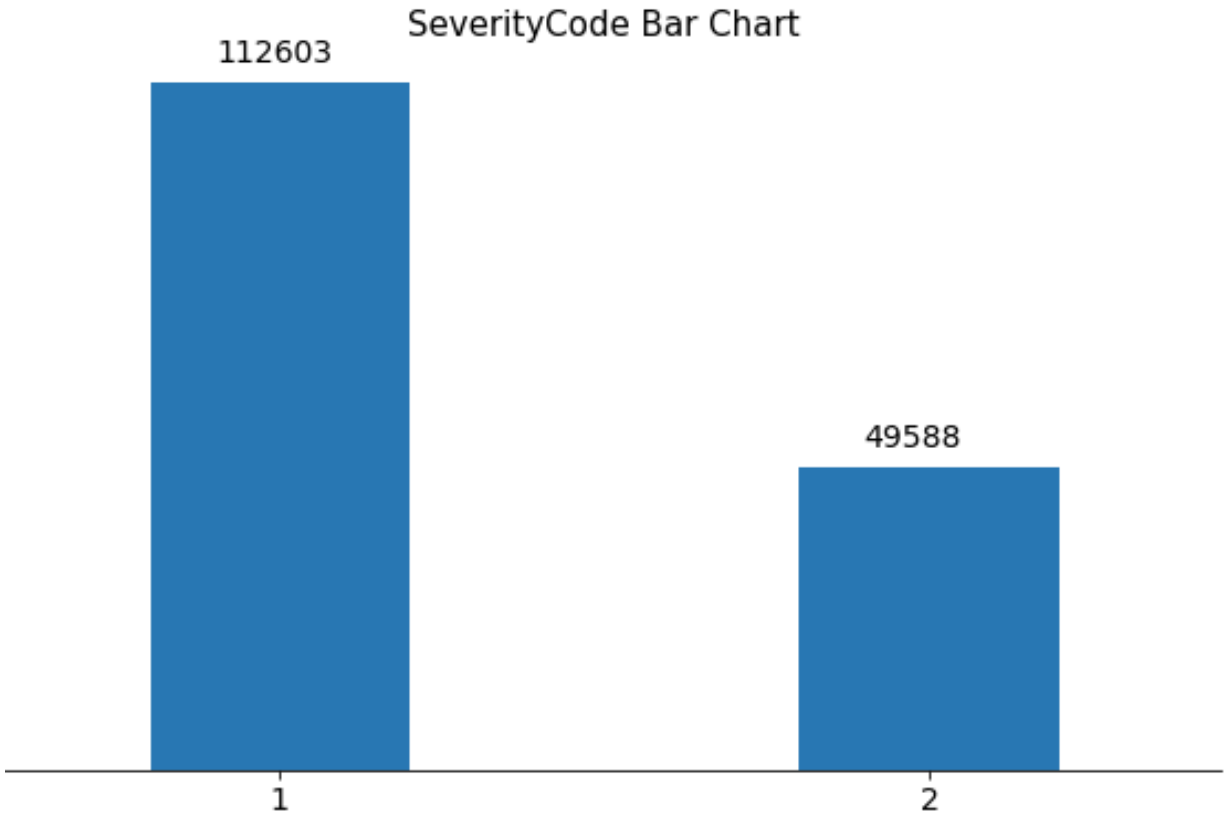




Also, all the string type columns are be converted into integer type including “ADDRTYPE”, “COLLISIONTYPE”, “UNDERINFL”, “ROADCOND”, “LIGHTCOND”.

```
df["ADDRTYPE"]=df["ADDRTYPE"].replace(["Block","Intersection","Alley"], [1,2,3])
df["COLLISIONTYPE"]=df["COLLISIONTYPE"].replace(["Parked Car","Angles","Rear Ended","Other","Sideswipe","Left Tu
df["UNDERINFL"]=df["UNDERINFL"].replace(["Y","N"], ["1","0"])
df=df.astype({"UNDERINFL":int})
df["ROADCOND"]=df["ROADCOND"].replace(["Dry","Wet","Unknown","Ice","Snow/Slush","Other","Standing Water","Sand/M
df["LIGHTCOND"]=df["LIGHTCOND"].replace(["Daylight","Dark - Street Lights On","Unknown","Dusk","Dawn","Dark - No
```

Following is the severity code bar plot in the dataset.



Predict Model

First try the Linear Regression

```
reg = linear_model.LinearRegression()
reg.fit(X_train,y_train)
y_pred = reg.predict(X_test)
reg.score(X_test,y_test)
```

```
/Users/liuqi/anaconda3/lib/python3.7/site-packages/sklearn/base.py:420: FutureWarning: The default value of multiou
tput (not exposed in score method) will change from 'variance_weighted' to 'uniform_average' in 0.23 to keep consis
tent with 'metrics.r2_score'. To specify the default value manually and avoid the warning, please either call 'metr
ics.r2_score' directly or make a custom scorer with 'metrics.make_scorer' (the built-in scorer 'r2' uses multioutpu
t='uniform_average').
  "multioutput='uniform_average').", FutureWarning)
```

```
0.006945335590865957
```

```
print('Coefficients: \n', reg.coef_)
```

```
Coefficients:
 [[ 0.19920801]
 [ 1.1295742 ]
 [ 0.3686585 ]
 [-0.09957209]
 [ 0.01848707]
 [-0.16525701]
 [-0.16830183]
 [ 0.02008139]
 [ 0.31911644]]
```

After testing, it seems like a non-linear regression model.
Hence, decision tree is used for the future predicting.

```
tree = DecisionTreeClassifier(criterion="entropy",max_depth=10)
tree.fit(X_train,y_train)
tree.predict(X_test)
```

```
array([[ 1,  1,  2, ...,  1, 10, 17],
       [ 1,  1,  2, ...,  1, 10, 17],
       [ 1,  1,  2, ...,  1, 10, 17],
       ...,
       [ 1,  3,  2, ...,  1, 10, 17],
       [ 1,  1,  2, ...,  1, 10, 17],
       [ 1,  1,  2, ...,  1, 10, 17]])
```

```
plot_tree(tree)
```

```
[Text(167.4, 163.07999999999998, 'X[0] <= 1.5\nentropy = 1.992\nsamples = 97314\nvalue =
```

Conclusion

It is useful to use the machine learning to predict the severity of car accidents. Most of the algorithms are biased towards to most frequent class. Proper preprocessing of the data will give optimal results.