Capstone Project Report

Seattle Car Accident Severity

Submited by,
Akshay Kumar
Oct/2020

Introduction | Business Understanding

The Seattle government is going to prevent avoidable car accidents by employing methods that alert drivers, health system, and police to remind them to be more careful in critical situations.

In most cases, not paying enough attention during driving, abusing drugs and alcohol or driving at very high speed are the main causes of occurring accidents that can be prevented by enacting harsher regulations. Besides the aforementioned reasons, weather, visibility, or road conditions are the major uncontrollable factors that can be prevented by revealing hidden patterns in the data and announcing warning to the local government, police and drivers on the targeted roads.

The target audience of the project is local Seattle government, police, rescue groups, and last but not least, car insurance institutes. The model and its results are going to provide some advice for the target audience to make insightful decisions for reducing the number of accidents and injuries for the city.

Data Understanding

The data was collected by the Seattle Police Department and Accident Traffic Records Department from 2004 to present.

The data consists of 37 independent variables and 194,673 rows. The dependent variable, “SEVERITYCODE”, contains numbers that correspond to different levels of severity caused by an accident from 0 to 4.
Severity codes are as follows:
0: Little to no Probability (Clear Conditions)
1: Very Low Probability — Chance or Property Damage
2: Low Probability — Chance of Injury
3: Mild Probability — Chance of Serious Injury
4: High Probability — Chance of Fatality

Attributes used to weigh the severity of an accident are 'WEATHER', 'ROADCOND' and 'LIGHTCOND'.

Data Preprocessing

In its original form, this data is not fit for analysis. For one, there are many columns that we will not use for this mode so we need to drop the non-relevant columns. Also, most of the features are of type object, when they should be numerical type, so I used label encoding to covert the features to our desired data type.
After that our Data will look like this.

	SEVERITYCODE	WEATHER	ROADCOND	LIGHTCOND	WEATHER_ENC	ROADCOND_ENC	LIGHTCOND_ENC
0	2	Overcast	Wet	Daylight	4	8	5
1	1	Raining	Wet	Dark - Street Lights On	6	8	2
2	1	Overcast	Dry	Daylight	4	0	5
3	1	Clear	Dry	Daylight	1	0	5
4	2	Raining	Wet	Daylight	6	8	5

With the new columns, we can now use this data in our analysis and ML models!

To get a good understanding of the dataset, I have checked different values in the features. The results show, the target feature is imbalance, so we use a simple statistical technique to balance it.

```
df2['SEVERITYCODE'].value_counts()
```

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

As you can see, the number of rows in class 1 is almost three times bigger than the number of rows in class 2. It is possible to solve the issue by down sampling the class 1.

```
from sklearn.utils import resample
```

```
df2_major=df2[df2.SEVERITYCODE==1]
df2_minor=df2[df2.SEVERITYCODE==2]
df2_major_dsample=resample(df2_major,
                           replace= False,
                           n_samples=58188,
                           random_state=123)
balanced_df2=pd.concat([df2_major_dsample,df2_minor])
balanced_df2['SEVERITYCODE'].value_counts()
```

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

Now our data is perfectly balanced.