Capstone Project - Car accident severity

Seemant Arora

26th September 2020

Table of Contents

Introduction: Business Problem

This Caston Project, which come under my IBM certification course, I would like to perform the Data Analysis of Road accident data with purpose of checking correlation of Road Accident at different conditions of Road, Weather and Light.

Generally road accidents are creating Sevier injuries to people, vehicles or both, To avoid and reduce the frequency of these type of accidents, I would like to build a model to predict the severity of an accident given the Weather and the Road conditions. This way we would be able to bring awareness about the possibility and severity of an accident. This way people will drive with full of attention or will change the drive plan. The main purpose of algorithm will be to know the severity of accident at given Weather and Road condition.

Data Understanding

The data is collected by the Seattle Police Department, recorded by Traffic Records and provided by Coursera via a download link. The time for this data starts from 2004 and consist 194,673 observations and **38 variables**.

```
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       object
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
dtype: object
```

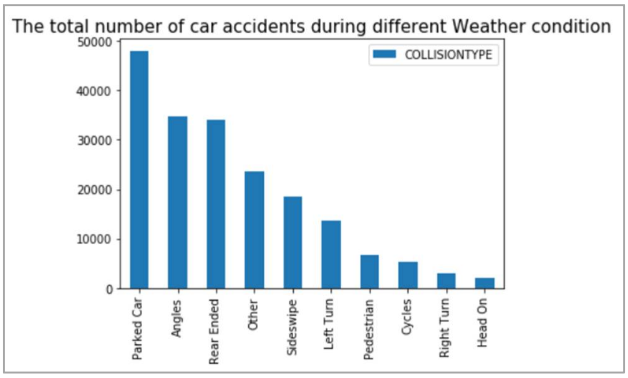
As mentioned in Introduction part, we will use **SEVERITYCODE** as our dependent variable Y and try different combinations of independent variables X to see the impact of Independent Variable on dependent one. Moreover, below variables will behave as Independent ones

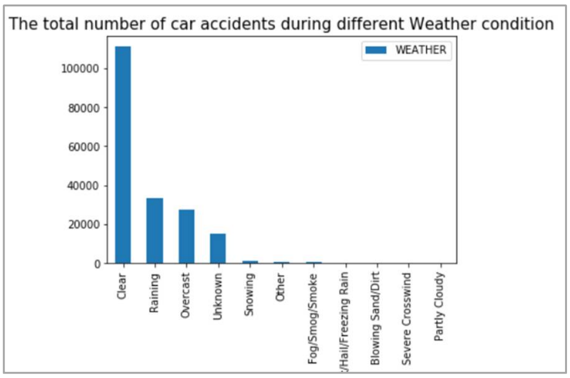
- COLLISIONTYPE: Collision type
- WEATHER: Weather conditions during the time of the collision.
- ROADCOND: The condition of the road during the collision.

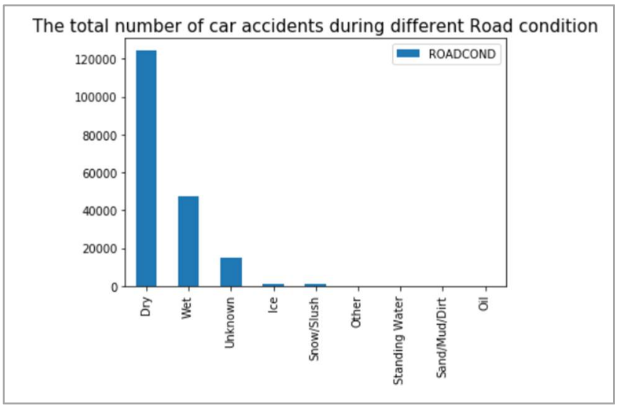
- LIGHTCOND: The light conditions during the collision.
- UNDERINFL: Whether or not a driver involved was under the influence of drugs or alcohol

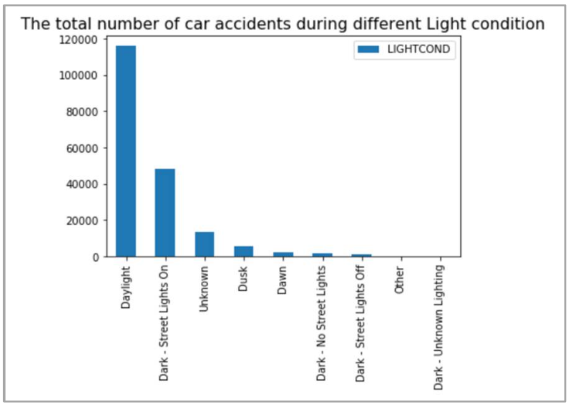
Data Visualisation

Impact of different condition on no of Accidents









Data Analysis

- Step1: Data preparation and cleaning
- Step2: Converting the Categorical variables in Numeric Value
- Step3: Normalize Data
- Step4: Split the Data set in to Train and Test set
- Step5: Classification Modeling and Evaluation

Step1: Data preparation and cleaning

In this step we will select only the relevant fields by dropping the irrelevant data which are having lots of missing value.

	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND	UNDERINFL	SEVERITYCODE
0	Angles	Overcast	Wet	Daylight	N	2
1	Sideswipe	Raining	Wet	Dark - Street Lights On	0	1
2	Parked Car	Overcast	Dry	Daylight	0	1
3	Other	Clear	Dry	Daylight	N	1
4	Angles	Raining	Wet	Daylight	0	2

Step2: Converting the Categorical variables in Numeric Value

In this step, we will convert all categorical variables in to Numeric one, and converting them to feature so they will act as independent variable. Severity code will behave like dependent variable.

	COLLISIONTYPE	WEATHER	ROADCOND	LIGHTCOND	UNDERINFL
0	0	4	8	5	0
1	9	6	8	2	0
2	5	4	0	5	0
3	4	1	0	5	0
4	0	6	8	5	0

Step3: Normalize Data

```
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
X[0:5]
```

```
array([[-1.61715866,  0.32150987,  1.47904464,  0.3500893 , -0.22467193],
       [ 1.61435927,  1.02230214,  1.47904464, -1.40093682, -0.22467193],
       [ 0.17812908,  0.32150987, -0.71198344,  0.3500893 , -0.22467193],
       [-0.18092847, -0.72967854, -0.71198344,  0.3500893 , -0.22467193],
       [-1.61715866,  1.02230214,  1.47904464,  0.3500893 , -0.22467193]])
```

Step4: Split the Data set in to Train and Test set

Train/Test Split involves splitting the dataset into training and testing sets respectively, which are mutually exclusive. After which, you train with the training set and test with the testing set.

This will provide a more accurate evaluation on out-of-sample accuracy because the testing dataset is not part of the dataset that have been used to train the data. It is more realistic for real world problems.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

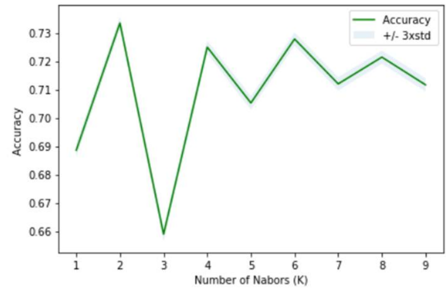
```
Train set: (151452, 5) (151452,)
Test set: (37864, 5) (37864,)
```

Step5: Classification Modeling and Evaluation

In this step, we would like to use 3 different algorithms to check the accuracy on test data. Final selection will be based upon accuracy result.

- ## K nearest neighbor (KNN)

KNN will help us predict the severity code of an outcome by finding the most similar to data point within k distance. KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses '**feature similarity**' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.




```
print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
The best accuracy was with 0.7334407352630467 with k= 2
```

- ## Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

- ## Decision tree

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Discussion and Conclusion

When we started analysisng the data, we had some categorical data with data type 'object'. This categorical data we can not feed to algorithm so we converted this data in to variable and later we tried 3 different algorithm to check which gives us better result. During KNN classification we also checked whcih K value gives us best result to improve the accuracy of modal. Evaluation metrics used to test the accuracy of our models were Jaccard index, f-1 score and precision score

	Algorithm	Jaccard	F1-score	Precision
0	KNN	0.72	0.7	0.7
1	Logistic Regression	0.7	0.58	0.63
2	Decision Tree	0.75	0.69	0.78

In this excercise we evaluated 3 machine learning algorithms to predict the severity of an accident knowing the weather and road conditions. The three models performed very similary, but Decision Tree stood out after comparision of model's accuracy.