Report

IBM Capstone Project: Car Accident Severity Prediction

ANUBHAV ROY

Contents

1. Introduction

Seattle Department of Transportation (SDOT) is on a mission to deliver a transportation system that provides safe and affordable access to places and opportunities. The council's goal is to create safe transportation environments and eliminate serious and fatal crashes in Seattle.

1.1 Background

Say you are driving to another city for work or to visit some friends. It is rainy and windy, and on the way, you come across a strenuous traffic jam on your side of the highway. Long lines of cars barely moving. Imagine the highway is shut down. It's an accident and rescue workers are busy transporting the ones involved in the crash to the nearest hospital. They must be in critical condition for all of this to be happening.

1.2 The need

Making sure people can get around the growing city safely is the council's top priority. SDOT collects data of every accident happening in the city and it has preserved data since 2002 in structured manner. It looks for analysing this huge data and draw out meaningful forecasts about the causes and impacts of fatal accidents.

1.3 The Problem

Now, wouldn't it be great if there is something in place that could warn you, given the weather and the road conditions about the possibility of you getting into a car accident and how severe it could be, so that you would drive more carefully or even change your travel if you are able to.
Well, this is exactly what we want to accomplish under this case study titled 'Capstone Project – Car accident severity', which would help predict the severity of an accident.

1.4 Audience and stakeholders

The severity impact prediction model (which is scope of this project) could be published as a REST API or web service (future scope of work) for the Seattle Department of Transportation (SDOT). The SDOT may have options to own or to subscribe to this service. By inputting necessary data to the service it could receive predictions regarding severity of accidents. This would help SDOT formulate traffic routing decisions or alerts in the geography under its monitoring.

Daily commuters and road travellers would find it much convenient to know about live traffic information, traffic diversion alerts and notifications when they tune with the SDOT broadcast channels. It would help save everyone's precious time, hectic travels and help avert mishaps or accidents due to such forewarnings.

We aim to design the model for a reliable accuracy of its prediction.

In fact the project foresees very high potential and ambitious goals to offer such services of human safety to most of the city councils across United States and across continents globally.

2. Methodology

We introduce here the research methods and data source used for the analysis. We would discuss in detail in below sections about the data, choice of variables, modelling methods and how they would help answer the problem statement. The methodology steps essentially are as follows;

- ✓ Data collection and understanding
  - Data source
  - Data understanding
- ✓ Data preparation
  - Basic insight of dataset
  - Feature selection
  - Data cleansing
  - Data transforming
  - Test of correlation and significance
  - Conclusion of important variables
- ✓ Model development
  - Algorithms and empirical findings
  - Results summary
- ✓ Discussion
- ✓ Conclusion
- ✓ References
- ✓ Acknowledgement

2.1 Data collection and understanding

Data source

We have used shared data of Seattle city as basis to deal with the accidents data (source: http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0.csv).
At first glance at the CSV file, we could see what type of data we have with us. The label for the data set is Severity, which describes the fatality of an accident. The remaining columns have different types of attributes. Also noticed that the data had some unbalanced attributes which need to be normalised during next steps.

We also used the collisions meta data available for about 16 years to understand the nature of all attributes. Having about 2.21L data observations, we could notice that a split of these could be used to train and test the prospective model.

Data understanding

The dataset basics were provided as follows;

Title: Collisions—All Years
Abstract: All collisions provided by Traffic Records.
Description: This includes all types of collisions. Collisions are displayed at the intersection or mid-block of a segment in the Annexure.
Timeframe: 2004 to Present.
Keyword(s): SDOT, Seattle, Transportation, Accidents, Bicycle, Car, Collisions, Pedestrian, Traffic, Vehicle
Types: The data is a mix of numerical and categorical types.
The data set provides labelled data for severity of accident. It shows a dual class categorical type of variable. The attributes (38 columns) convey information mainly about;

- the incident: such as identification no., location coordinates, date, time etc.
- the collision: such as code, type, description, injuries, fatalities etc.
- the impact: such as count of pedestrians, cyclists, vehicles involved etc.
- preconditions: such as inattention, influence of drugs, road condition, weather, speeding etc.

Attributes are almost complete with the information such as name, data type, length and description as shown in next section. State Collision Code Dictionary comprising about 85 codes with descriptions is also provided in supplement.

In the given dataset, SeverityCode is identified as the target variable (labelled or dependent) while rest of the fields are construed as independent variables or the attributes. The case objective with the given data, does qualify it as a classification problem of the supervised machine learning. All columns that could influence the cause and impact of an accident need to be selected for training and testing the model.

2.2 Data preparation

Basic insight of dataset

After reading data into Pandas data frame, it becomes a good start to explore the dataset. Following ways are followed to obtain essential insights of the data to help better understand the dataset;

Columns:
It provides list of columns that exist in the dataset.

Data types:
This step is to know the variety of types viz. object, float, int, bool and datetime64. In order to better learn about each attribute, it is necessary to know the data type of each column, which was identified using Python Info() method as in screen shot below;

```
[6]: 1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221389 entries, 0 to 221388
Data columns (total 40 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   X                213918 non-null  float64
 1   Y                213918 non-null  float64
 2   OBJECTID         221389 non-null  int64
 3   INCKEY           221389 non-null  int64
 4   COLDETKEY        221389 non-null  int64
 5   REPORTNO         221389 non-null  object
 6   STATUS           221389 non-null  object
 7   ADDRTYPE         217677 non-null  object
 8   INTKEY           71884 non-null   float64
 9   LOCATION         216801 non-null  object
 10  EXCEPTRSNCODE    100986 non-null  object
 11  EXCEPTRSNDESC    11779 non-null   object
 12  SEVERITYCODE     221388 non-null  object
 13  SEVERITYDESC     221389 non-null  object
 14  COLLISIONTYPE    195159 non-null  object
 15  PERSONCOUNT      221389 non-null  int64
 16  PEDCOUNT         221389 non-null  int64
 17  PEDCYLCOUNT      221389 non-null  int64
 18  VEHCOUNT         221389 non-null  int64
 19  INJURIES         221389 non-null  int64
 20  SERIOUSINJURIES  221389 non-null  int64
 21  FATALITIES       221389 non-null  int64
 22  INCDATE          221389 non-null  object
 23  INCDTTM          221389 non-null  object
 24  JUNCTIONTYPE     209417 non-null  object
 25  SDOT_COLCODE     221388 non-null  float64
 26  SDOT_COLDESC     221388 non-null  object
 27  INATTENTIONIND   30188 non-null   object
 28  UNDERINFL        195179 non-null  object
 29  WEATHER          194969 non-null  object
 30  ROADCOND         195050 non-null  object
 31  LIGHTCOND        194880 non-null  object
 32  PEDROWNOTGRNT    5192 non-null    object
 33  SDOTCOLNUM       127205 non-null  float64
 34  SPEEDING         9928 non-null    object
```

```
 33  SDOTCOLNUM       127205 non-null  float64
 34  SPEEDING         9928 non-null    object
 35  ST_COLCODE       211976 non-null  object
 36  ST_COLDESC       195159 non-null  object
 37  SEGLANEKEY       221389 non-null  int64
 38  CROSSWALKKEY     221389 non-null  int64
 39  HITPARKEDCAR     221389 non-null  object
dtypes: float64(5), int64(12), object(23)
memory usage: 67.6+ MB
```

1   Python | Idle   Saving completed   Mode: Command   Ln 1, Col 1   Capstone Project_Prasad Jaywant_Submission.ipynb

Data Description:

We could get statistical summary, such as count, unique value, column mean value, column standard deviation, etc of each column. It provides various summary statistics, excluding NaN (Not a Number) values.

```
[7]: 1 df.describe(include='all')
```

[7]:

	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	LOCATION	...	ROADCOND	LIGHTCOND	PEDROWNOTGRN
count	213918.000000	213918.000000	221389.00000	221389.000000	221389.000000	221389	221389	217677	71884.000000	216801	...	195050	194880	519
unique	NaN	NaN	NaN	NaN	NaN	221386	2	3	NaN	25198	...	9	9	
top	NaN	NaN	NaN	NaN	NaN	1782439	Matched	Block	NaN	BATTERY ST TUNNEL NB BETWEEN ALASKAN WY VI NB ...	...	Dry	Daylight	
freq	NaN	NaN	NaN	NaN	NaN	2	195232	144917	NaN	298	...	128535	119448	519
mean	-122.330756	47.620199	110695.00000	144708.701914	144936.934541	NaN	NaN	NaN	37612.330964	NaN	...	NaN	NaN	Nal
std	0.030055	0.056043	63909.64371	89126.729589	89501.312920	NaN	NaN	NaN	51886.084219	NaN	...	NaN	NaN	Nal
min	-122.419091	47.495573	1.00000	1001.000000	1001.000000	NaN	NaN	NaN	23807.000000	NaN	...	NaN	NaN	Nal
25%	-122.349280	47.577151	55348.00000	71634.000000	71634.000000	NaN	NaN	NaN	28652.750000	NaN	...	NaN	NaN	Nal
50%	-122.330363	47.616053	110695.00000	127184.000000	127184.000000	NaN	NaN	NaN	29973.000000	NaN	...	NaN	NaN	Nal
75%	-122.311998	47.664290	166042.00000	209783.000000	210003.000000	NaN	NaN	NaN	33984.000000	NaN	...	NaN	NaN	Nal
max	-122.238949	47.734142	221389.00000	333843.000000	335343.000000	NaN	NaN	NaN	757580.000000	NaN	...	NaN	NaN	Nal

11 rows × 40 columns

Feature selection

In the first screening, it was noticed that some of the attributes are not significant to the cause of or to assess the impact of the severity. So these could be dropped for removing bias while designing the model. Rest of the columns were retained for further analysis.

Data cleansing

In the second step, rest of the columns were analysed for missing values. Columns were treated for substituting missing values as shown in table below;

Sr. No.	Attribute	Data type, length	Description	Wrangling Method	Rationale
1	OBJECTID	OBJECTID	ESRI unique identifier	Dropped	Insignificance
2	X	Longitude	ESRI geometry field	Dropped	Insignificance
3	Y	Latitude	ESRI geometry field	Dropped	Insignificance
4	ADDRTYPE	Text, 12	Collision address type: • Alley • Block • Intersection	Retained	2% missing data, replaced by max. frequency
5	INTKEY	Double	Key that corresponds to the intersection associated with a collision	Dropped	Insignificance
6	LOCATION	Text, 255	Description of the general location of the collision	Dropped	Insignificance
7	EXCEPTRSNCODE	Text, 10		Dropped	Insignificance
8	EXCEPTRSNDESC	Text, 300		Dropped	Insignificance
9	SEVERITYCODE	Text, 100	A code that corresponds to the severity of the collision: • 3—fatality • 2b—serious injury • 2—injury • 1—prop damage • 0—unknown	Retained	Target variable
10	SEVERITYDESC	Text	A detailed description of the severity of the collision	Retained	Target variable
11	COLLISIONTYPE	Text, 300	Collision type	Retained	12% missing data, replaced by max. frequency
12	PERSONCOUNT	Double	The total number of people involved in the collision	Retained	
13	PEDCOUNT	Double	The number of pedestrians involved in the collision. This is entered by the state.	Retained	
14	PEDCYLCOUNT	Double	The number of bicycles involved in the collision. This is entered by the state.	Retained	

Sr. No.	Attribute	Data type, length	Description	Wrangling Method	Rationale
15	VEHCOUNT	Double	The number of vehicles involved in the collision. This is entered by the state.	Retained	
16	INJURIES	Double	The number of total injuries involved in the collision. This is entered by the state.	Retained	
17	SERIOUSINJURIES	Double	The number of serious injuries involved in the collision. This is entered by the state.	Retained	
18	FATALITIES	Double	The number of fatalities involved in the collision. This is entered by the state.	Retained	
19	INCDATE	Date	The date of the incident.	Dropped	Insignificance
20	INCDTTM	Text, 30	The date and time of the incident.	Dropped	Insignificance
21	JUNCTIONTYPE	Text, 300	Category of junction at which collision took place	Retained	5.5% missing data, replaced by max. frequency
22	SDOT_COLCODE	Text, 10	A code given to the collision by SDOT.	Dropped	Insignificance
23	SDOT_COLDESC	Text, 300	A description of the collision corresponding to the collision code.		
24	INATTENTIONIND	Text, 1	Whether or not collision was due to inattention (Y/N).	Dropped	86% data is missing
25	UNDERINFL	Text, 10	Whether or not a driver involved was under the influence of drugs or alcohol.	Dropped	only 4.5% observations are influencing
26	WEATHER	Text, 300	A description of the weather conditions during the time of the collision.	Retained	12% missing data, replaced by max. frequency
27	ROADCOND	Text, 300	The condition of the road during the collision.	Retained	12% missing data, replaced by max. frequency
28	LIGHTCOND	Text, 300	The light conditions during the collision.	Retained	12% missing data, replaced by max. frequency
29	PEDROWNOTGRNT	Text, 1	Whether or not the pedestrian right of way was not granted. (Y/N)	Dropped	97.7% data missing

Sr. No.	Attribute	Data type, length	Description	Wrangling Method	Rationale
30	SDOTCOLNUM	Text, 10	A number given to the collision by SDOT.	Dropped	Insignificance
31	SPEEDING	Text, 1	Whether or not speeding was a factor in the collision. (Y/N)	Dropped	only 4.5% observations are influencing, rest data unavailable
32	ST_COLCODE	Text, 10	A code provided by the state that describes the collision. For more information about these codes, please see the State Collision Code Dictionary.	Dropped	Insignificance
33	ST_COLDESC	Text, 300	A description that corresponds to the state's coding designation.	Retained	12% missing data, replaced by max. frequency
34	SEGLANEKEY	Long	A key for the lane segment in which the collision occurred.	Dropped	Insignificance
35	CROSSWALKKEY	Long	A key for the crosswalk at which the collision occurred.	Dropped	Insignificance
36	HITPARKEDCAR	Text, 1	Whether or not the collision involved hitting a parked car. (Y/N)	Retained	
37	STATUS	Text, 10	Matched, Unmatched	Dropped	Insignificance
38	REPORTNO	Long	Sr. No. of report for internal purposes	Dropped	Insignificance
39	COLDETKEY	Long	Secondary key for the incident	Dropped	Insignificance
40	INCKEY	Long	A unique key for the incident	Dropped	Insignificance

The cleansed data set is structured as follows;

```
[11]:  1  df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 199794 entries, 0 to 221388
Data columns (total 20 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   ADDRTYPE         199794 non-null  object
 1   SEVERITYCODE     199794 non-null  int64
 2   SEVERITYDESC     199794 non-null  object
 3   COLLISIONTYPE    199794 non-null  object
 4   PERSONCOUNT      199794 non-null  int64
 5   PEDCOUNT         199794 non-null  int64
 6   PEDCYLCOUNT      199794 non-null  int64
 7   VEHCOUNT         199794 non-null  int64
 8   INJURIES         199794 non-null  int64
 9   SERIOUSINJURIES  199794 non-null  int64
 10  FATALITIES       199794 non-null  int64
 11  JUNCTIONTYPE     199794 non-null  object
 12  SDOT_COLCODE     199794 non-null  float64
 13  SDOT_COLDESC     199794 non-null  object
 14  WEATHER          199794 non-null  object
 15  ROADCOND         199794 non-null  object
 16  LIGHTCOND        199794 non-null  object
 17  ST_COLCODE       199794 non-null  object
 18  ST_COLDESC       195157 non-null  object
 19  HITPARKEDCAR     199794 non-null  object
dtypes: float64(1), int64(8), object(11)
memory usage: 32.0+ MB
```

Data transforming

The last step in data cleansing would be to check and make sure that all data is in the correct format (int, float, text or other). To use categorical variables for regression analysis, indicator variables (or dummy variable) were used for transforming categorical variables into binary (0s and 1s) or numeric values. Also we changed the target data type to be integer, as it is a requirement by the Skitlearn algorithms used. This would make the data ready for next tests of correlation and determining significance. The results are as shown in screen shots below;

```
[14]:  1 df_clean.info(max_cols=130)

<class 'pandas.core.frame.DataFrame'>
Int64Index: 199794 entries, 0 to 221388
Data columns (total 126 columns):
 #   Column                                                    Non-Null Count   Dtype
---  ------                                                    --------------   -----
 0   SEVERITYCODE                                              199794 non-null  int64
 1   SEVERITYDESC                                              199794 non-null  object
 2   PERSONCOUNT                                               199794 non-null  int64
 3   PEDCOUNT                                                  199794 non-null  int64
 4   PEDCYLCOUNT                                               199794 non-null  int64
 5   VEHCOUNT                                                  199794 non-null  int64
 6   INJURIES                                                  199794 non-null  int64
 7   SERIOUSINJURIES                                           199794 non-null  int64
 8   FATALITIES                                                199794 non-null  int64
 9   SDOT_COLDESC                                              199794 non-null  object
 10  ST_COLDESC                                                195157 non-null  object
 11  ADDRTYPE_Alley                                            199794 non-null  uint8
 12  ADDRTYPE_Block                                            199794 non-null  uint8
 13  ADDRTYPE_Intersection                                     199794 non-null  uint8
 14  COLLISIONTYPE_Angles                                      199794 non-null  uint8
 15  COLLISIONTYPE_Cycles                                      199794 non-null  uint8
 16  COLLISIONTYPE_Head On                                     199794 non-null  uint8
 17  COLLISIONTYPE_Left Turn                                   199794 non-null  uint8
 18  COLLISIONTYPE_Other                                       199794 non-null  uint8
 19  COLLISIONTYPE_Parked Car                                  199794 non-null  uint8
 20  COLLISIONTYPE_Pedestrian                                  199794 non-null  uint8
 21  COLLISIONTYPE_Rear Ended                                  199794 non-null  uint8
 22  COLLISIONTYPE_Right Turn                                  199794 non-null  uint8
 23  COLLISIONTYPE_Sideswipe                                   199794 non-null  uint8
 24  JUNCTIONTYPE_At Intersection (but not related to intersection)  199794 non-null  uint8
 25  JUNCTIONTYPE_At Intersection (intersection related)       199794 non-null  uint8
 26  JUNCTIONTYPE_Driveway Junction                            199794 non-null  uint8
```

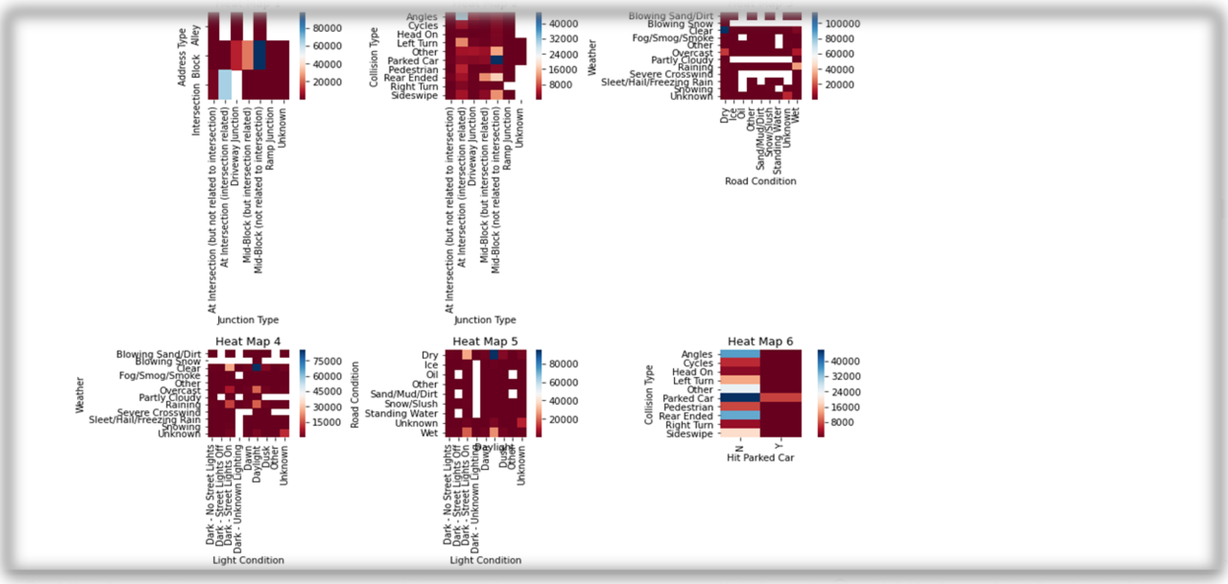
```
 24  JUNCTIONTYPE_At Intersection (but not related to intersection)  199794 non-null  uint8
 25  JUNCTIONTYPE_At Intersection (intersection related)       199794 non-null  uint8
 26  JUNCTIONTYPE_Driveway Junction                            199794 non-null  uint8
 27  JUNCTIONTYPE_Mid-Block (but intersection related)         199794 non-null  uint8
 28  JUNCTIONTYPE_Mid-Block (not related to intersection)      199794 non-null  uint8
 29  JUNCTIONTYPE_Ramp Junction                                199794 non-null  uint8
 30  JUNCTIONTYPE_Unknown                                      199794 non-null  uint8
 31  WEATHER_Blowing Sand/Dirt                                 199794 non-null  uint8
 32  WEATHER_Blowing Snow                                      199794 non-null  uint8
 33  WEATHER_Clear                                             199794 non-null  uint8
 34  WEATHER_Fog/Smog/Smoke                                    199794 non-null  uint8
 35  WEATHER_Other                                             199794 non-null  uint8
 36  WEATHER_Overcast                                          199794 non-null  uint8
 37  WEATHER_Partly Cloudy                                     199794 non-null  uint8
 38  WEATHER_Raining                                           199794 non-null  uint8
 39  WEATHER_Severe Crosswind                                  199794 non-null  uint8
 40  WEATHER_Sleet/Hail/Freezing Rain                          199794 non-null  uint8
 41  WEATHER_Snowing                                           199794 non-null  uint8
 42  WEATHER_Unknown                                           199794 non-null  uint8
 43  ROADCOND_Dry                                              199794 non-null  uint8
 44  ROADCOND_Ice                                              199794 non-null  uint8
 45  ROADCOND_Oil                                              199794 non-null  uint8
 46  ROADCOND_Other                                            199794 non-null  uint8
 47  ROADCOND_Sand/Mud/Dirt                                    199794 non-null  uint8
 48  ROADCOND_Snow/Slush                                       199794 non-null  uint8
 49  ROADCOND_Standing Water                                   199794 non-null  uint8
 50  ROADCOND_Unknown                                          199794 non-null  uint8
 51  ROADCOND_Wet                                              199794 non-null  uint8
 52  LIGHTCOND_Dark - No Street Lights                         199794 non-null  uint8
 53  LIGHTCOND_Dark - Street Lights Off                        199794 non-null  uint8
 54  LIGHTCOND_Dark - Street Lights On                         199794 non-null  uint8
 55  LIGHTCOND_Dark - Unknown Lighting                         199794 non-null  uint8
 56  LIGHTCOND_Dawn                                            199794 non-null  uint8
 57  LIGHTCOND_Daylight                                        199794 non-null  uint8
 58  LIGHTCOND_Dusk                                            199794 non-null  uint8
 59  LIGHTCOND_Other                                           199794 non-null  uint8
```

```
 59  LIGHTCOND_Other                              199794 non-null  uint8
 60  LIGHTCOND_Unknown                            199794 non-null  uint8
 61  ST_COLCODE_                                  199794 non-null  uint8
 62  ST_COLCODE_0                                 199794 non-null  uint8
 63  ST_COLCODE_1                                 199794 non-null  uint8
 64  ST_COLCODE_10                                199794 non-null  uint8
 65  ST_COLCODE_11                                199794 non-null  uint8
 66  ST_COLCODE_12                                199794 non-null  uint8
 67  ST_COLCODE_13                                199794 non-null  uint8
 68  ST_COLCODE_14                                199794 non-null  uint8
 69  ST_COLCODE_15                                199794 non-null  uint8
 70  ST_COLCODE_16                                199794 non-null  uint8
 71  ST_COLCODE_17                                199794 non-null  uint8
 72  ST_COLCODE_18                                199794 non-null  uint8
 73  ST_COLCODE_19                                199794 non-null  uint8
 74  ST_COLCODE_2                                 199794 non-null  uint8
 75  ST_COLCODE_20                                199794 non-null  uint8
 76  ST_COLCODE_21                                199794 non-null  uint8
 77  ST_COLCODE_22                                199794 non-null  uint8
 78  ST_COLCODE_23                                199794 non-null  uint8
 79  ST_COLCODE_24                                199794 non-null  uint8
 80  ST_COLCODE_25                                199794 non-null  uint8
 81  ST_COLCODE_26                                199794 non-null  uint8
 82  ST_COLCODE_27                                199794 non-null  uint8
 83  ST_COLCODE_28                                199794 non-null  uint8
 84  ST_COLCODE_29                                199794 non-null  uint8
 85  ST_COLCODE_3                                 199794 non-null  uint8
 86  ST_COLCODE_30                                199794 non-null  uint8
 87  ST_COLCODE_31                                199794 non-null  uint8
 88  ST_COLCODE_32                                199794 non-null  uint8
 89  ST_COLCODE_4                                 199794 non-null  uint8
 90  ST_COLCODE_40                                199794 non-null  uint8
 91  ST_COLCODE_41                                199794 non-null  uint8
 92  ST_COLCODE_42                                199794 non-null  uint8
 93  ST_COLCODE_43                                199794 non-null  uint8
 94  ST_COLCODE_45                                199794 non-null  uint8
 95  ST_COLCODE_48                                199794 non-null  uint8
 96  ST_COLCODE_49                                199794 non-null  uint8
 97  ST_COLCODE_5                                 199794 non-null  uint8
 98  ST_COLCODE_50                                199794 non-null  uint8
 99  ST_COLCODE_51                                199794 non-null  uint8
 100 ST_COLCODE_52                                199794 non-null  uint8
 101 ST_COLCODE_53                                199794 non-null  uint8
 102 ST_COLCODE_54                                199794 non-null  uint8
```

```
 101 ST_COLCODE_53                                199794 non-null  uint8
 102 ST_COLCODE_54                                199794 non-null  uint8
 103 ST_COLCODE_56                                199794 non-null  uint8
 104 ST_COLCODE_57                                199794 non-null  uint8
 105 ST_COLCODE_6                                 199794 non-null  uint8
 106 ST_COLCODE_60                                199794 non-null  uint8
 107 ST_COLCODE_64                                199794 non-null  uint8
 108 ST_COLCODE_65                                199794 non-null  uint8
 109 ST_COLCODE_66                                199794 non-null  uint8
 110 ST_COLCODE_67                                199794 non-null  uint8
 111 ST_COLCODE_7                                 199794 non-null  uint8
 112 ST_COLCODE_71                                199794 non-null  uint8
 113 ST_COLCODE_72                                199794 non-null  uint8
 114 ST_COLCODE_73                                199794 non-null  uint8
 115 ST_COLCODE_74                                199794 non-null  uint8
 116 ST_COLCODE_8                                 199794 non-null  uint8
 117 ST_COLCODE_81                                199794 non-null  uint8
 118 ST_COLCODE_82                                199794 non-null  uint8
 119 ST_COLCODE_83                                199794 non-null  uint8
 120 ST_COLCODE_84                                199794 non-null  uint8
 121 ST_COLCODE_85                                199794 non-null  uint8
 122 ST_COLCODE_87                                199794 non-null  uint8
 123 ST_COLCODE_88                                199794 non-null  uint8
 124 HITPARKEDCAR_N                               199794 non-null  uint8
 125 HITPARKEDCAR_Y                               199794 non-null  uint8
dtypes: int64(8), object(3), uint8(115)
memory usage: 40.2+ MB
```

Test of correlation and significance

To get a better measure of the important characteristics, we looked at the correlation of attributes vis-a-vis target variable i.e. Accident Severity. The correlations are depicted by constructing heat maps between pairs of the variables.



Pearson Correlation:

The Pearson Correlation measures the linear dependence between two variables X and Y of 'int64' or 'float64' types.

The resulting coefficient is a value between -1 and 1 inclusive, where:

1: Total positive linear correlation.
0: No linear correlation, the two variables most likely do not affect each other.
-1: Total negative linear correlation.
The closeness to terminal values (-1 and 1) would decide strength of the correlation.

P-value:

The P-value is the probability value that the correlation between these two variables is statistically significant. Normally, we choose a significance level of 0.05, which means that we are 95% confident that the correlation between the variables is significant. We would use "stats" module in the "Scipy" library to get the P-value.

By convention, when the

p-value is < 0.001: we say there is strong evidence that the correlation is significant.
the p-value is < 0.05: there is moderate evidence that the correlation is significant.
the p-value is < 0.1: there is weak evidence that the correlation is significant.
the p-value is > 0.1: there is no evidence that the correlation is significant.

Conclusion: Important Variables

By now we would have a better idea of what our data looks like and which variables are important for consideration while predicting the ‘Severity’ class.
As we move into building machine learning models to automate our analysis, feeding the model with variables that meaningfully affect our target variable would help improve the model's prediction performance.

[19]:

	index	Pearson Correlation Coefficient	P-value
6	INJURIES	0.735846	0
7	SERIOUSINJURIES	0.468547	0

[20]:

```
df_corr.info(max_cols=130)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 199794 entries, 0 to 221388
Data columns (total 123 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   SEVERITYCODE                    199794 non-null  int64
 1   PERSONCOUNT                     199794 non-null  int64
 2   PEDCOUNT                        199794 non-null  int64
 3   PEDCYLCOUNT                     199794 non-null  int64
 4   VEHCOUNT                        199794 non-null  int64
 5   INJURIES                        199794 non-null  int64
 6   SERIOUSINJURIES                 199794 non-null  int64
 7   FATALITIES                      199794 non-null  int64
 8   ADDRTYPE_Alley                  199794 non-null  uint8
 9   ADDRTYPE_Block                  199794 non-null  uint8
 10  ADDRTYPE_Intersection           199794 non-null  uint8
 11  COLLISIONTYPE_Angles            199794 non-null  uint8
 12  COLLISIONTYPE_Cycles            199794 non-null  uint8
 13  COLLISIONTYPE_Head On           199794 non-null  uint8
 14  COLLISIONTYPE_Left Turn         199794 non-null  uint8
 15  COLLISIONTYPE_Other             199794 non-null  uint8
 16  COLLISIONTYPE_Parked Car        199794 non-null  uint8
 17  COLLISIONTYPE_Pedestrian        199794 non-null  uint8
 18  COLLISIONTYPE_Rear Ended        199794 non-null  uint8
 19  COLLISIONTYPE_Right Turn        199794 non-null  uint8
 20  COLLISIONTYPE_Sideswipe         199794 non-null  uint8
```

Python | Disconnected   Saving failed   Mode: Command   Ln 1, Col 2   Capstone Project_Prasad Jaywant_Submission.ipynb

```
 111 ST_COLCODE_73                   199794 non-null  uint8
 112 ST_COLCODE_74                   199794 non-null  uint8
 113 ST_COLCODE_8                    199794 non-null  uint8
 114 ST_COLCODE_81                   199794 non-null  uint8
 115 ST_COLCODE_82                   199794 non-null  uint8
 116 ST_COLCODE_83                   199794 non-null  uint8
 117 ST_COLCODE_84                   199794 non-null  uint8
 118 ST_COLCODE_85                   199794 non-null  uint8
 119 ST_COLCODE_87                   199794 non-null  uint8
 120 ST_COLCODE_88                   199794 non-null  uint8
 121 HITPARKEDCAR_N                  199794 non-null  uint8
 122 HITPARKEDCAR_Y                  199794 non-null  uint8
dtypes: int64(8), uint8(115)
memory usage: 35.6 MB
```

2.3 Model development

In this section, we developed several models that will predict the severity of the accident using the variables or features. A Model would help us understand the exact relationship between different variables and how these variables are used to predict the result.

Algorithms used

We developed Classification model based on following algorithms that would predict the severity of an accident using the variables or features.

Logistic regression

While Linear Regression is suited for estimating continuous values (e.g. estimating house price), it is not the best tool for predicting the class of an observed data point. To estimate the class of a data point, as is our current case, we use Logistic Regression. It produces a formula that predicts the probability of a class label as function of the independent variables. Logistic regression fits a special s-shaped curve by transforming the numeric estimate into a probability with the sigmoid function σ.

Data pre-processing and selection

We selected some likely causal features for modelling in the first step. We further defined X as the Feature Matrix values (Numpy array) and y as the response vector (target) for our dataset. and then normalized the dataset. Data Standardization makes data zero mean and unit variance.

Train/Test dataset

Here we split our dataset into train and test set in the ratio of 70:30 as illustration. The sets are mutually exclusive.

Modelling

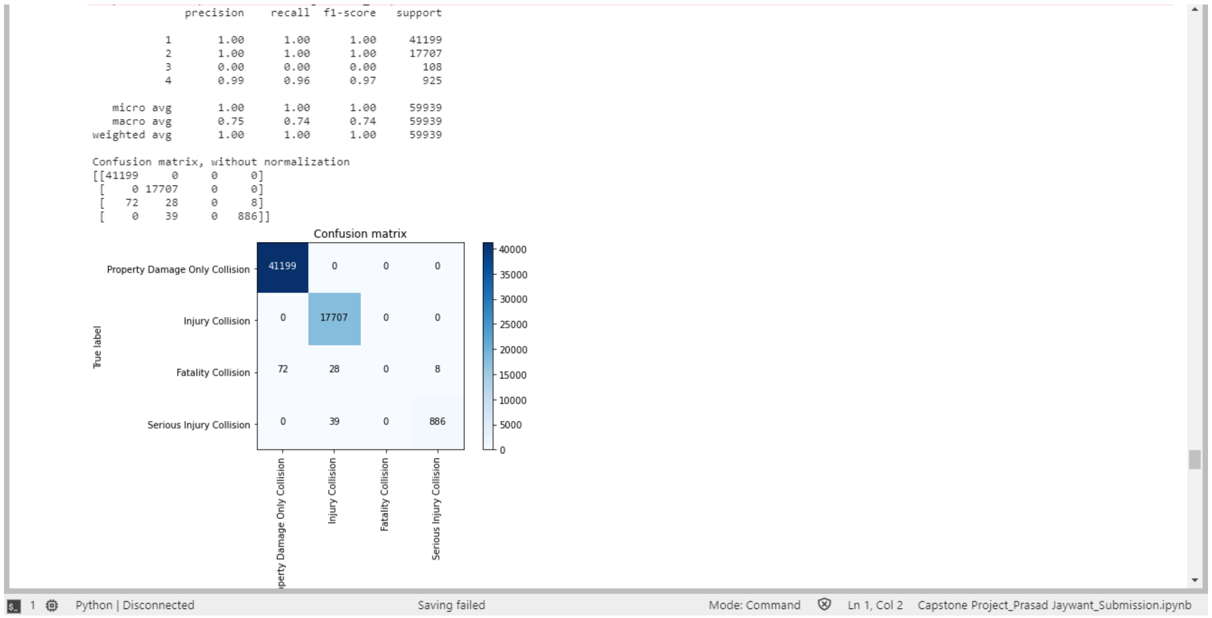
This function implements logistic regression and can use different numerical optimizers to find parameters, including ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’ solvers. We used ‘saga’ for multi-class regression solver.
The version of LogisticRegression in Scikit-learn, support regularization. Regularization is a technique used to solve the overfitting problem in machine learning models. C parameter indicates inverse of regularization strength which must be a positive float. Smaller values specify stronger regularization. Now lets fit our model with train set;

We predicted model performance using our test set. predict_proba returns estimates for all classes, ordered by the label of classes. So, the first column is the probability of class 1, P(Y=1|X), and second column is probability of class 0, P(Y=0|X):

Evaluation

Jaccard index
We define jaccard as the size of the intersection divided by the size of the union of two label sets. If the entire set of predicted labels for a sample strictly match with the true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0.

Confusion matrix
Another way of looking at accuracy of classifier is to look at confusion matrix.
In specific case of multi-class classifier, such as our case, we can interpret these numbers as the count of true positives, false positives, true negatives, and false negatives.



Based on the count of each section, we calculate precision and recall of each label:

Precision is a measure of the accuracy provided that a class label has been predicted. It is defined by: precision = TP / (TP + FP)

Recall is true positive rate. It is defined as: Recall = TP / (TP + FN)
So, we can calculate precision and recall of each class.

F1 score: Now we are in the position to calculate the F1 scores for each label based on the precision and recall of that label.
The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. It is a good way to show that a classifier has a good value for both recall and precision.
And finally, we can tell the average accuracy for this classifier is the average of the F1-score for both labels, which is 0.72 in our case.

Log loss

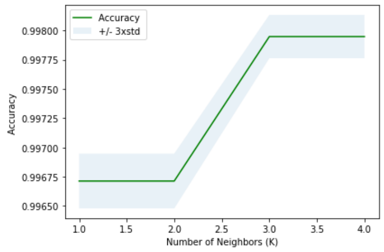
In logistic regression, the output can be the probability of customer churn is yes (or equals to 1). This probability is a value between 0 and 1. Log loss(Logarithmic loss) measures the performance of a classifier where the predicted output is a probability value between 0 and 1.

K-Nearest Neighbors (KNN)

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 considers the 'K' nearest points to it to determine its classification.

```
KNN Testset Accuracy:  0.9979479137122741
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

[30]:
```python
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 Neighbors (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 0.9979479137122741 with k = 3
```

Python | Disconnected   Saving failed   Mode: Command   Ln 1, Col 2   Capstone Project_Prasad Jaywant_Submission.ipynb

It follows the same steps as used in logistic Regression measure. It was noted that KNN takes highest time (15 minutes or more) among all modelling methods followed in our exercise.

Decision Trees

Followed same preparatory steps as in other models. We first created an instance of the DecisionTreeClassifier called SeverityTree. It's based on the 'minimizing entropy (degree of randomness)' and 'maximising information gain (level of certainty)' criteria of each node.

[34]:
```python
treeclf = DecisionTreeClassifier(max_depth=5, random_state=1)
treeclf.fit(X_train,y_train)
# compute the feature importances
df_imp_feature=pd.DataFrame({'feature':x, 'importance':treeclf.feature_importances_}).sort_values(by='importance', ascending=False)
df_imp_feature.head(10)
```

[34]:

	feature	importance
0	(I, N, J, U, R, I, E, S)	0.932811
1	(S, E, R, I, O, U, S, I, N, J, U, R, I, E, S)	0.067189

[35]:
```python
df_imp_feature[df_imp_feature['importance'] >= 0.3]
```

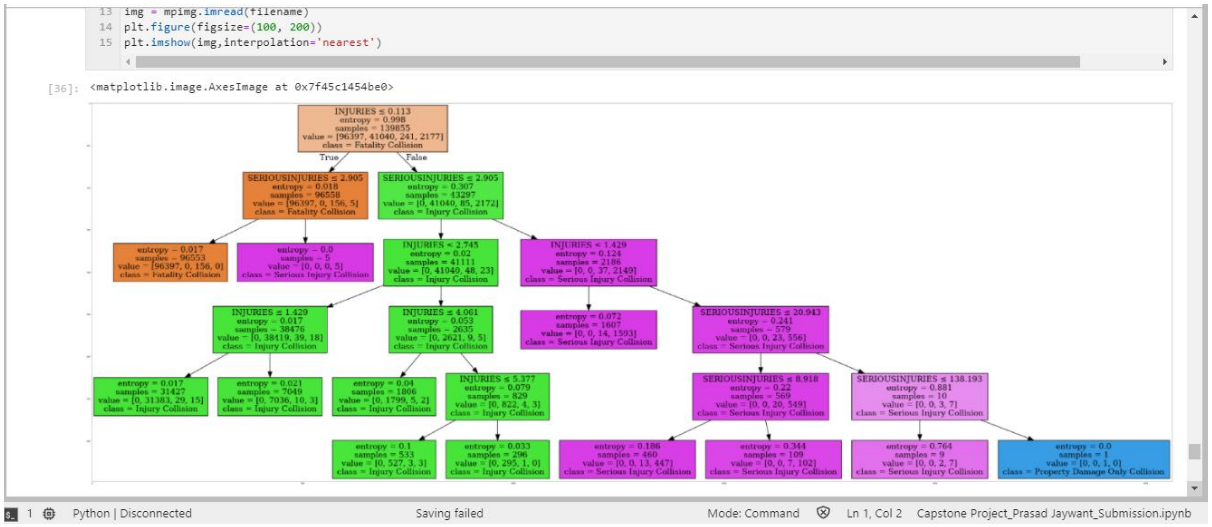
[35]:

	feature	importance
0	(I, N, J, U, R, I, E, S)	0.932811

[36]:
```python
from sklearn.externals.six import StringIO
import pydotplus
import matplotlib.image as mpimg

dot_data = StringIO()
filename = "severitydesc.png"
y=df_clean['SEVERITYDESC']
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=4)
featureNames = x.columns[0:123]
out=tree.export_graphviz(SeverityTree,feature_names=featureNames, out_file=dot_data, class_names= np.unique(y_train), filled=True,  special_characters=True,rot
```

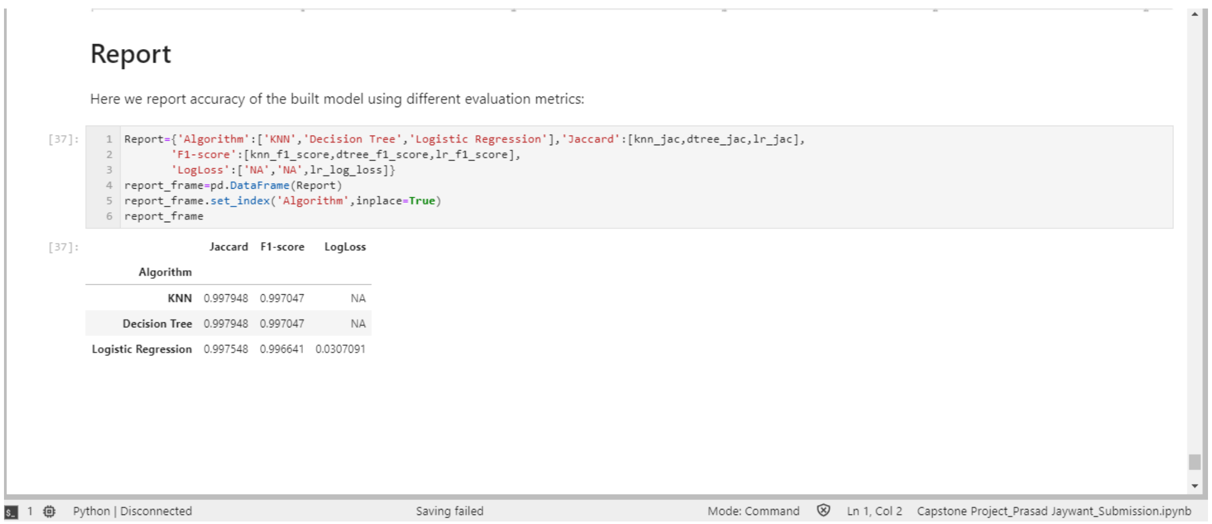
Python | Disconnected   Saving failed   Mode: Command   Ln 1, Col 2   Capstone Project_Prasad Jaywant_Submission.ipynb



Accuracy classification score computes subset accuracy i.e. the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true.

Results summary

The accuracy of the models built using different evaluation metrics can be summarized as follows;



Report

Here we report accuracy of the built model using different evaluation metrics:

```
[37]: Report={'Algorithm':['KNN','Decision Tree','Logistic Regression'],'Jaccard':[knn_jac,dtree_jac,lr_jac],
        'F1-score':[knn_f1_score,dtree_f1_score,lr_f1_score],
        'LogLoss':['NA','NA',lr_log_loss]}
report_frame=pd.DataFrame(Report)
report_frame.set_index('Algorithm',inplace=True)
report_frame
```

[37]:

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.997948	0.997047	NA
Decision Tree	0.997948	0.997047	NA
Logistic Regression	0.997548	0.996641	0.0307091

Python | Disconnected Saving failed Mode: Command Ln 1, Col 2 Capstone Project_Prasad Jaywant_Submission.ipynb

3. Discussion

The data set is well structured and offers good number of useful observations (about 2L). The data wrangling was mostly accomplished by substituting the values with maximum frequency of the available data.

The correlation method shortlisted some variables such as injuries, which are related to the impact of accident, contributed moderately to the severity. Though the influence of causal factors such as weather, road/light conditions on accident severity was expected, they seemed not significant in contribution as was suggested by the low values (<0.4) of

Pearson coefficients. Correlation of address type and junction type to severity was also not significantly evident.

We had split given data set into 70:30 ratio for training/testing the model. Model prediction accuracy seems acceptable due to high Jaccard and F1-score and near-zero Log loss values.

4. Conclusion

The model has fairly taken care of the missing values which are of common occurrence in the real data gathering scenarios. The selected algorithms are in sync with the prediction accuracy, thereby poses high confidence in predicting the real cases. As envisioned in section 1.4, the model seems capable of implementing it at the client site. Also poses high potential for extending it to more city councils having similar data sources.

In the roadmap ahead, the model could be enriched with deeper analysis of causation factors, although the focus at present was more on the correlations within given data. The model deployment and integration with client systems could be the next steps of project implementation. With study of advanced Python capabilities, statistical/probabilistic algorithms and graphical visualizations, it could provide opportunity for iterative improvements in the model.

5. References

Preparation of this report must cite help of valuable references as follows;

- ✓ IBM Data Science Professional Certificate Course – All 9 modules, labs, tutorials and links therein: https://www.coursera.org/professional-certificates/ibm-data-science
- ✓ IBM Watson Studio Resources: https://cloud.ibm.com/resources
- ✓ Github Repository: https://github.com/
- ✓ Pandas open source literature: http://pandas.pydata.org
- ✓ Scikit Learn open source content: https://scikit-learn.org/
- ✓ Technology community sites: https://stackoverflow.com/ and many more from Google Search

6. Acknowledgement

I must thank all the mentor team members; Alex Aklson, Polong Lin, Romeo Kienzler, Svetlana Levitan, Joseph Santarcangelo, Hima Vasudevan, Rav Ahuja, SAEED AGHABOZORGI for their valued guidance, exciting videos and structuring comprehensive labs and the tutorials. The illustrations, case studies showcased in these sessions helped me understand the concepts and practical applications of Data Science tools and techniques.

I also take this opportunity to thank my peers who took out their time to review my graded submissions and provided the valued feedbacks. I owe my special thanks to the contributors who are active in posting replies to various queries or issues being raised in the discussion forums. Their efforts in helping people to come out of the stuck up or no-clue situations are much appreciated.

Equally important, please convey my best regards to the Coursera organizing team for designing this unique course for the benefit of hundreds of thousands of such Data Science enthusiasts globally.