Data Science Capstone Project

Predicting Accident Severity in the Greater Seattle Area

Suman Bhardwaj

Coursera

Author Note

This is a report containing the results from completing the capstone project for the IBM Data Science Professional Certificate. The data to create charts and tables are available in a Jupyter Notebook.

Abstract

This project aims to answer a question can the severity of accidents be predicted? Using a data set from 2004 to early 2020 in the Greater Seattle Area this problem will be analyzed. Through the use of a Jupyter Notebook running Python 3.6 the data set will be cleaned and preprocessed for evaluation. To understand the data, it will be visualized, and relationships made with the attributes and the target values. This is classification problem; machine learning algorithms will be used to train and predict the data set. The results will be a seventy percent accuracy in the prediction on the severity of an accident.

Keywords: severity accidents, Seattle, machine learning

Contents

Data Science Capstone Project

Predicting Accident Severity in the Greater Seattle Area

Predictions are a way of life in our busy schedules. We check the weather and it always seems to be a 50% chance of rain, especially in the Greater Seattle Washington area. There is even a section of the news that covers traffic. But it is limited to letting you know if a traffic accident has occurred. What if we can predict accidents, in addiction the severity of these accidents. Better yet why not deploy measures to avoid or even reduce the number and severity of accidents.

Problem Statement

This project will look at data provided by the Seattle Police Department and made available by the Seattle Department of Transportation. We will look at this data and see if we can use it to predict the severity of accidents. Using the attributes in the data set we will examine different forms of data science tools to check if machine learning can aid in this question.

Can the data provided in this file provided by the Seattle Department of Transportation (WSDOT, n.d.) and supplied by Coursera contain the labels needed to predict the severity of an accident. If the data set can then resources can be put to better use and deployed in the areas that require it the most.

Data Set

The data was imported from a repository managed by the Coursera course as a comma separated (.csv) file. The file was inserted into a data frame to be examined. A sample of the first five rows is in *Table one (1)*. In looking at the data set using several useful Python tools, the set has 194,673 rows and 38 columns. The rows represent an accident, and the columns represent possible attributes that coincide with the event. The first column is the target label,

SEVERITYCODE, that has two (2) values 1 for property damage, and 2 for injury accident. The 37 attributes are a mix of objects and integers *see Table two (2),* that needs to be preprocessed for evaluation.

Cleaning

The first step is to remove the attributes with missing values. Using Python to count all on the null values and creating a bar graph. The bar graph is labeled *Figure one (1)* in the Figures section. The bar graph shows that there are seven of the labels that are missing more than twenty percent of the values. These attributes will not be useful in the project and are dropped from the evaluation. All of the other fields in the data set have less than four percent of missing values, but there are thirty-one values remaining. Values that have meaning and lack redundancy are the next to be dropped.

Examining the data set we receive location information from the X and Y columns, which represent longitude and latitude respectfully. Any location information can be dropped. The bureaucracy of public infrastructure gives many codes, these codes are not very useful for the project. Any additional time and date information is dropped other than the one that is used in the evaluation. What is left is a cleaned-up set of twelve attributes, from the original thirty-seven.

Preprocessing

With a cleaned set of data there is still some missing values to deal with, see *Table three (3)* for remaining missing value count. The first ones will be the X and Y columns, as these are very specific and only about three percent of the values are missing, these values will be dropped. The WEATHER, ROADCOND (Road Conditions), LIGHTCOND (Light Conditions) have a large amount of missing values but are valuable attributes. *Table Four (4*) through *Table six (6)* has the counts of the various values. The null values were replaced by the dominant

values Clear, Dry, Daylight. The last two attributes COLLISIONTYPE, and UNDERINFL (under the influence) did not have a clear dominate values so these nulls will be dropped.

The data set started as a large group of thirty-seven attributes and one target label. Now the data set had been cleaned of unnecessary attributes and redundancies for a more manageable evaluation. The data has also been processed to where there are no null values. The data is now available for examination.

Methodology

It is important in data science to understand the data and the relationships. Visualization of the data is a great way to look at the data instead of in numerical or text. This can become overwhelming and cumbersome. Bar charts, scatter plots and geospatial maps are great communicators for data. Exploratory Analysis of the data is necessary to make sure we are working with the right information for the project.

Data Visualization

Since this section is data visualization the figures of the bar graphs and geospatial will be in this section and not in the figures section.

Bar Graph comparisons

This section will look at a comparison of the severity code as it relates to an accident attribute. The values of the SEVERITYCODE is “1” for Property Damage and “2” for Injury. The graphs will be bar graphs where the x-axis is the attribute value name and the y-axis is the count of the unique attribute value. The first graph is accidents with property damage and the second graph will be accidents with injuries.

Figure A – Collision Type Values

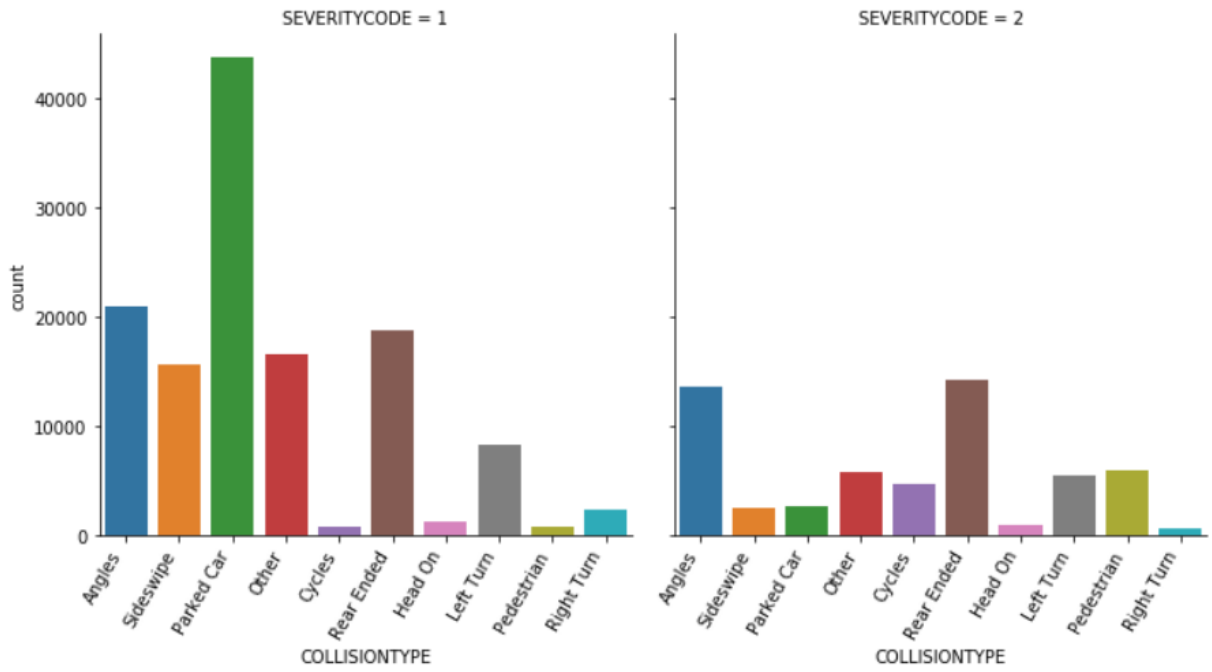


Figure B – Weather Values

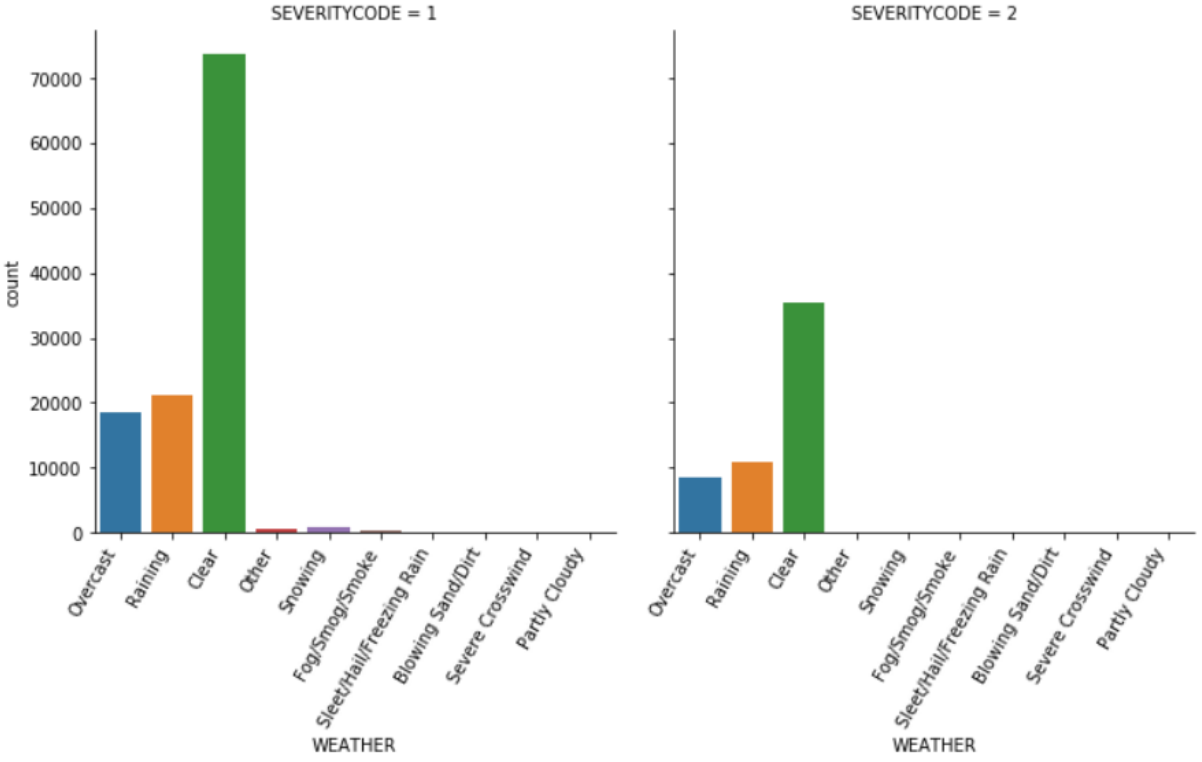


Figure C – Road Condition Values

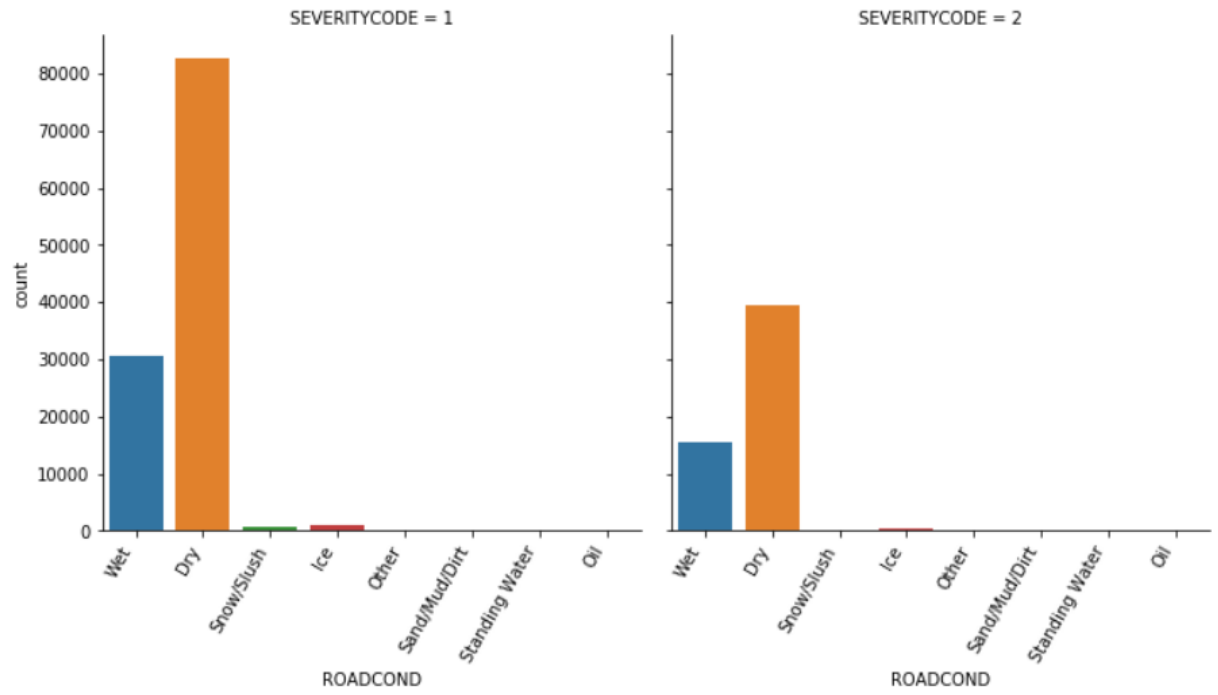
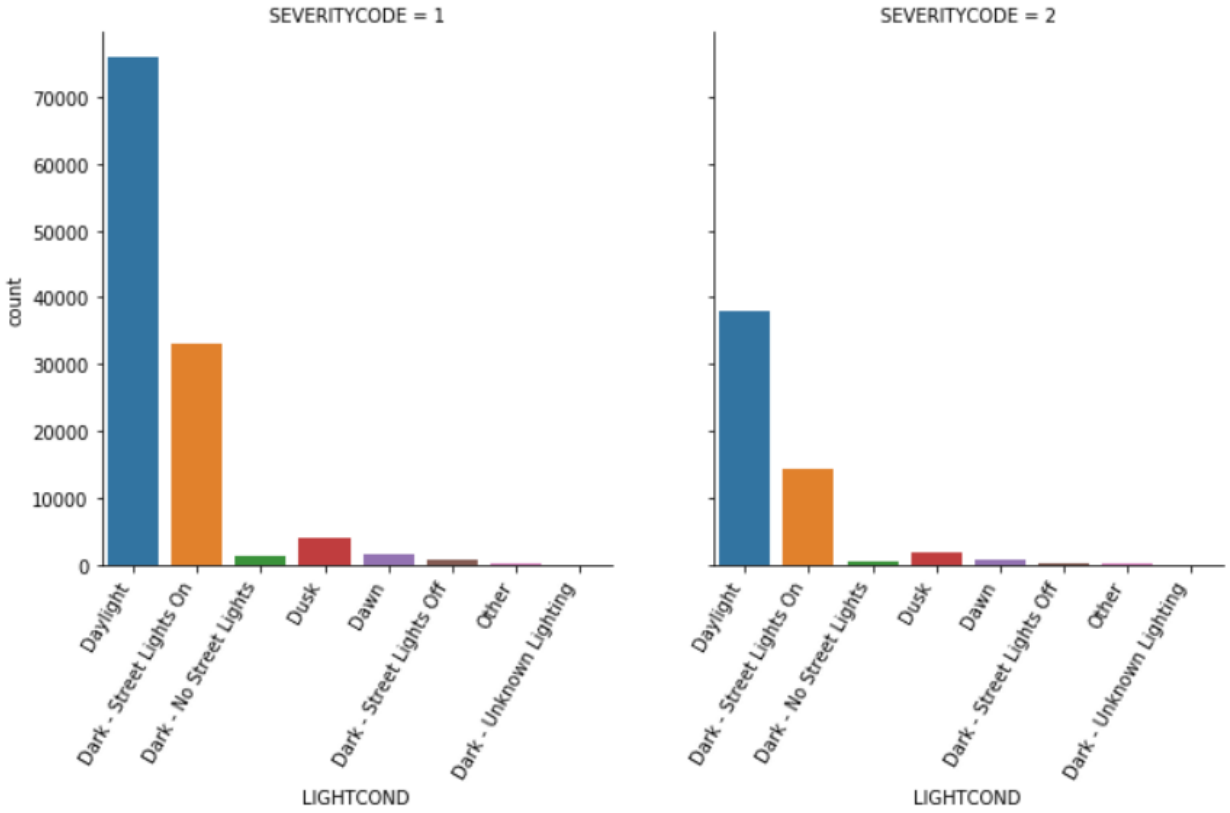


Figure D – Lighting Condition Values

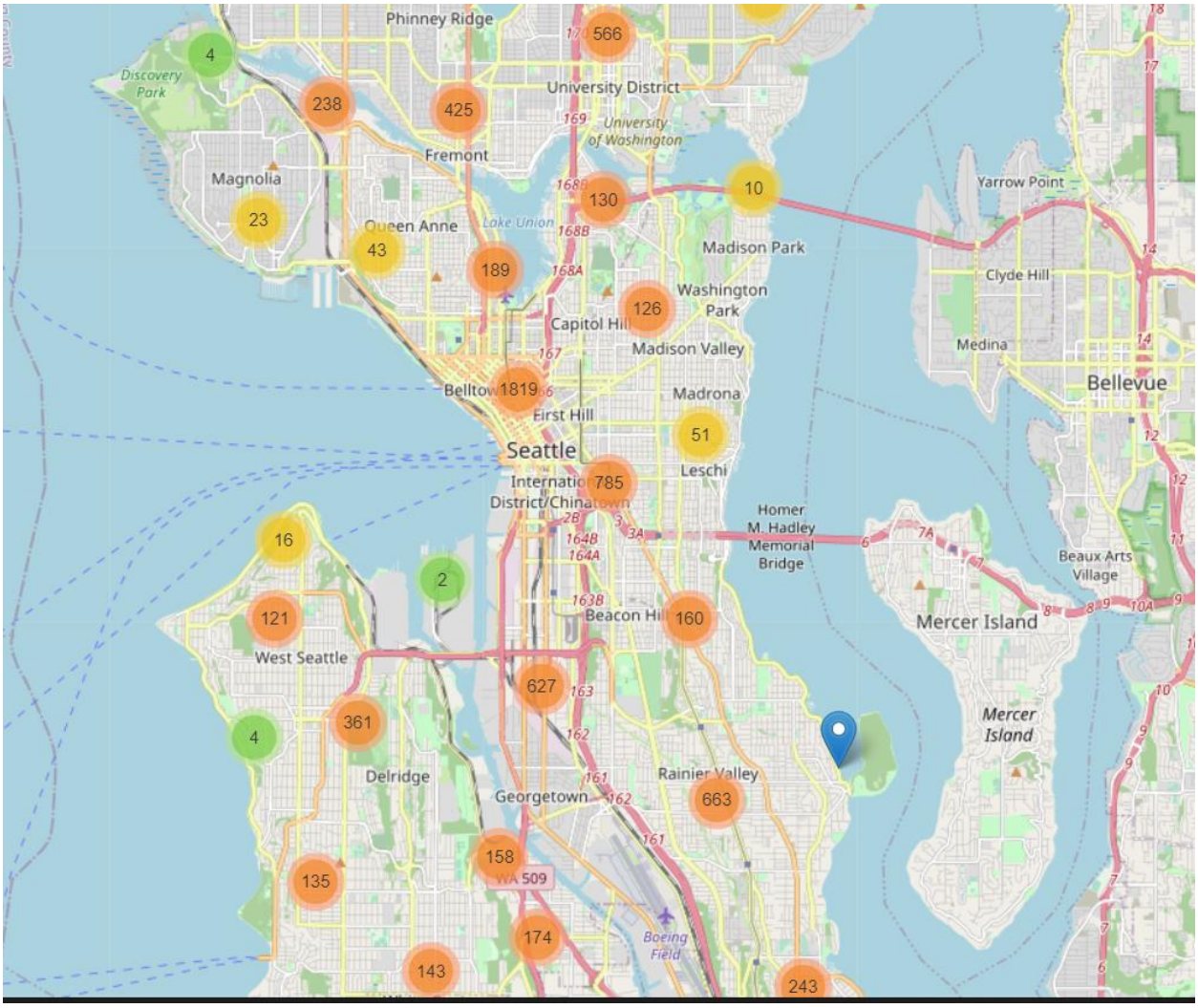


In the data comparison there are clear values that dominate the attribute. Most accidents happen on Dry, Clear, and during Day Light. The type of collision has variation, most property damage is with a parked car. Injury accidents have two types of struck incidents. Bar graphs have a great ability to show a quick comparison of data values.

Geospatial

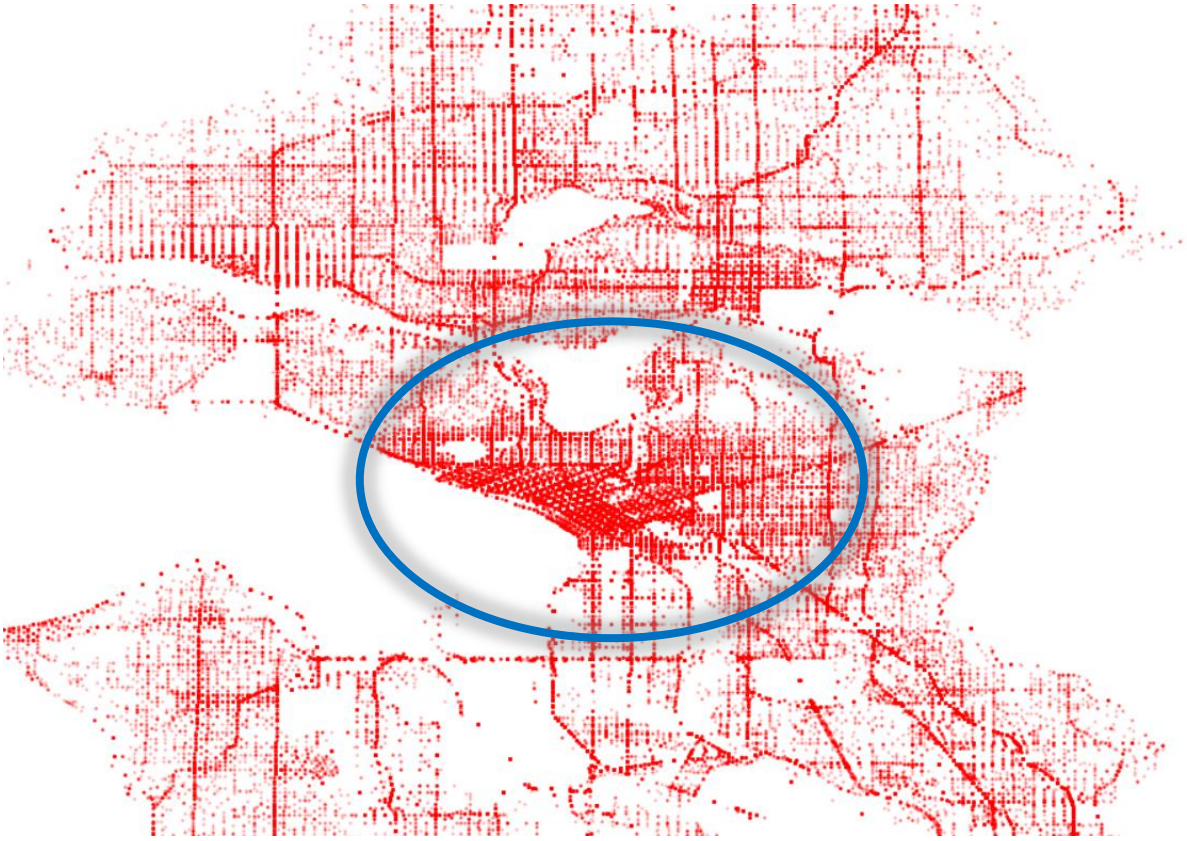
The next visualization is geospatial, where data is shown over a geographical representation of the data. In this project the folium library was used to show accidents within zip codes.

Figure E – 2019 Greater Seattle Area Count of Accidents by Zip Code



The data set is from 2004 to the beginning of 2020. To save on resources the date time stamp was broken down to month, day, and year to only use one year of data. 2019 was chosen for it being the most recent data with one full year of data values. Clearly the downtown area had the largest number of incidents. Additionally, heat maps were created to show the location of the accidents.

Figure F – Heat map of Greater Seattle Area Accident Locations



As should be expected the greatest concentration was in the downtown area of Seattle. The geospatial visualization can easily give a "bird's eye view" of the data values and show the concentration of the values by location.

Exploratory Data Analysis

The relationships of the data values have been analyzed, now it needs to be modeled. In the visualization the values were categorical, this makes it difficult to model numerically. *Table seven (7)* illustrates the attributes that are integers, float, datetime and object. The attributes that are object need to be converted. Using sklearn library and LabelEncoder the object data was converted from string to an integer value. The datetime stamps are no longer needed so this attribute is dropped. *Table eight (8) and nine (9)* show the results of the data values after EDA.

For the modeling sklearn is continued to be used to transform the attributes into a Standard Scaler and creates an array of the attribute values. This normalizes and standardizes the data. The data then is split into a training set and a testing set. In this project a thirty percent test set is used, leaving seventy percent for the training set of the data, see *Figure two (2)*.

Modeling

In this project a Classification Model will need to be used because this model is designed to predict categorical values. The target variable is the SEVERITYCODE and this is a simple 1 for property damage and a 2 for injury. There are two supervised machine learning algorithms' available in sklearn. Decision Tree Model and Logistic Regression will be used to model and each model will output the accuracy of the test. In the project X is the attributes (input) and Y is the target value (output)

Decision Tree Model

This model is flow chart structure that uses nodes to complete a test on an attribute. Each branch is the outcome of the test. Each leaf created is a label after all attributes have been considered. The path from the root to the leaf is the classification model. In this project a depth

of ten was applied to the algorithm. Trained a model with the train data, and then ran the prediction model to predict the results. The output is a flow chart of tests and decisions.

Logistic Regression

Since this is not a continuous value Linear Regression is not available as a usable algorithm. A Logistical Regression is needed in a classification target value. Similar to the Decision Tree the training data is introduced into an object. In the prediction all estimates are returned as an array of two columns. The probability of each class is ordered by the label of the class, in this case when Y is 0 with respect to X and when Y is 1 with respect to X.

Results

The results are output in the form of functions. The accuracy score is the most basic of the scores if all the predicted values match the score is 1.0 if not it is 0.0 (1.4. Support Vector Machines — scikit-learn 0.20.2 documentation, n.d.) . A confusion matrix is used to give a visual way to see the accuracy of the classifier values. *Figure three (3)* illustrates the confusion matrix object created for the project. The classification report for the project is listed below.

Table A – Classification Report

	precision	recall	f1-score	support
1	0.73	0.95	0.82	38547
2	0.61	0.18	0.27	16828
micro avg	0.72	0.72	0.72	55375
macro avg	0.67	0.56	0.55	55375
weighted avg	0.69	0.72	0.66	55375

Decision Tree Model

```
Decision Tree accuracy is 0.7524334085778781
F1 Score accuracy is 0.7190817586710851
Jaccard Similarity Score is 0.7524334085778781
```

The above results are very good, scoring in the seventy percentile shows good modeling (1.4. Support Vector Machines — scikit-learn 0.20.2 documentation, n.d.).

Logistical Regression

```
Logistic Regression accuracy is 0.7158645598194131
F1 Score accuracy is 0.6563652386708057
Jaccard Similarity Score is 0.7158645598194131
LogLoss is 0.5696148997474206
```

The Logistical Regression faired well, but not as well as the Decision Tree. The F1 score is almost six points less again this is still high for these types of models.

Discussion.

The question raised was can the severity of an accident be predicted? With the data set that was used the answer is yes with about a seventy percent accuracy. Sounds almost like a weather forecast? Interesting trends did debunk some preconceived notions, most accidents happen in dry, clear, daylight times. Then again that is probably when most people are driving. Most accidents happen in high population densities, like in this project that showed downtown to have the highest concentration. Most accidents are not under the influence of something. Collision type was the most interesting as the values very with no clear dominating value. This would be a great question to investigate further.

Conclusion.

It looks like accident severity can be predicted accurately. It is a matter of time as more data is collected and less no values enter a database, the better predictions can be made. Garbage in is garbage out. The tools used in this project is only a fraction of the powerful tools in data science. Many city and state transportation departments can use this data and concepts to predict what resources are needed and where to deploy these assets.

References

1.4. Support Vector Machines — scikit-learn 0.20.2 documentation. (n.d.). Retrieved 10 13, 2020, from http://scikit-learn.org/stable/modules/svm.html

WSDOT. (n.d.). *WSDOT Regions-North Central*. Retrieved 10 9, 2020, from http://www.wsdot.wa.gov/regions/NorthCentral/

Tables

Table one (1)

First Five (5) rows of data set:

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEDROWNOTGRNT	SDOTCOLNUM	SPEEDING
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight	NaN	NaN	NaN
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On	NaN	6354039.0	NaN
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight	NaN	4323031.0	NaN
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight	NaN	NaN	NaN
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight	NaN	4028032.0	NaN

5 rows × 38 columns

Table two (2)

Sample list of attributes and data types:

```
df.dtypes
```

```
]: 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
```

Table three (3)

Final attributes to use in project. Missing value count that must be preprocessed.

```
df.isnull().sum()
```

```
4]: SEVERITYCODE        0
    X                5334
    Y                5334
    ADDRTYPE         1926
    COLLISIONTYPE    4904
    PERSONCOUNT         0
    VEHCOUNT            0
    INCDTTM             0
    UNDERINFL        4884
    WEATHER          5081
    ROADCOND         5012
    LIGHTCOND        5170
    HITPARKEDCAR        0
    dtype: int64
```

Table four (4)

Count of values for Lighting Conditions attribute

```
df['LIGHTCOND'].value_counts()
```

```
5]: Daylight                   113582
    Dark - Street Lights On     47314
    Unknown                    12432
    Dusk                        5775
    Dawn                        2422
    Dark - No Street Lights     1451
    Dark - Street Lights Off    1152
    Other                        188
    Dark - Unknown Lighting       11
    Name: LIGHTCOND, dtype: int64
```

Table five (5)

Count of values for Weather attribute

```
df['WEATHER'].value_counts()
```

```
]: Clear                       108959
   Raining                      32015
   Overcast                     27136
   Unknown                      13893
   Snowing                        894
   Other                          773
   Fog/Smog/Smoke                 553
   Sleet/Hail/Freezing Rain       112
   Blowing Sand/Dirt               50
   Severe Crosswind                24
   Partly Cloudy                    5
   Name: WEATHER, dtype: int64
```

Table six (6)

Count of unique values for Road Conditions attribute

```
f['ROADCOND'].value_counts()
```

```
: Dry                122076
  Wet                 46064
  Unknown             13839
  Ice                  1177
  Snow/Slush            989
  Other                 117
  Standing Water        102
  Sand/Mud/Dirt          64
  Oil                    53
  Name: ROADCOND, dtype: int64
```

Table seven (7)

List of Data Types before Exploratory Data Analysis

```
df.dtypes
```

```
SEVERITYCODE              int64
X                       float64
Y                       float64
ADDRTYPE                 object
COLLISIONTYPE            object
PERSONCOUNT               int64
VEHCOUNT                  int64
INCDTTM          datetime64[ns]
UNDERINFL                object
WEATHER                  object
ROADCOND                 object
LIGHTCOND                object
HITPARKEDCAR             object
year                      int64
month                     int64
weekday                   int64
dtype: object
```

Table eight (8)

List of Attributes after Exploratory Data Analysis

```
df.dtypes
```

```
SEVERITYCODE       int64
X                float64
Y                float64
ADDRTYPE           int64
COLLISIONTYPE      int64
PERSONCOUNT        int64
VEHCOUNT           int64
UNDERINFL          int64
WEATHER            int64
ROADCOND           int64
LIGHTCOND          int64
HITPARKEDCAR       int64
year               int64
month              int64
weekday            int64
dtype: object
```

Table nine (9)

First Five rows of Values after Exploratory Data Analysis

X	Y	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	VEHCOUNT	UNDERINFL	WEATHER	ROADCOND	LIGHTCOND	HITPARKEDCAR
-122.323148	47.703140	1	0	2	2	2	4	7	5	0
-122.347294	47.647172	0	9	2	2	0	6	7	2	0
-122.334540	47.607871	0	5	4	3	0	4	0	5	0
-122.334803	47.604803	0	4	3	3	2	1	0	5	0
-122.306426	47.545739	1	0	2	2	0	6	7	5	0

Note: All Tables and Figures are available in the Jupyter Notebook for this project

Figures

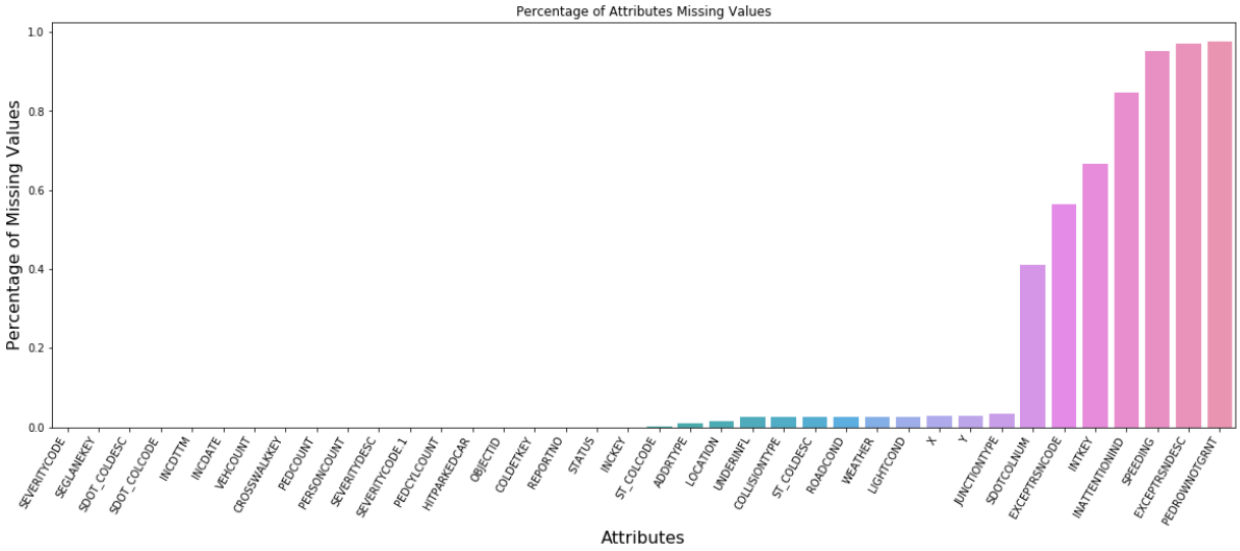


Figure one (1). This bar graph illustrates the count of missing values from the attributes in the data set. The attributes to the right side, above 20 percent is unusable and needs to be removed.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size = 0.3, random_state = 3)
print('Training X =', X_train.shape, ' y =', y_train.shape)
print('Testing X =', X_test.shape, ' y =', y_test.shape)
```

```
Training X = (129206, 10)  y = (129206,)
Testing X = (55375, 10)  y = (55375,)
```

Figure two (2). The above code is the splitting of the data thirty percent for testing. The output is the number a rows and columns in the array.

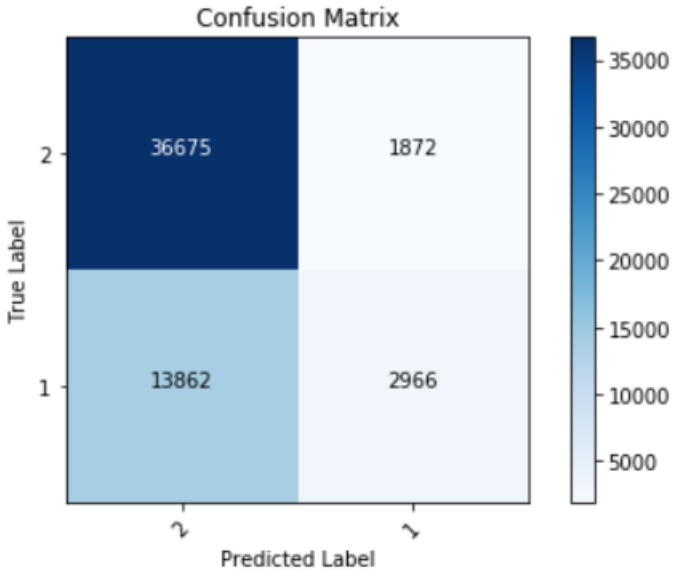


Figure three (3) – Confusion Matrix of predicted and true values for target classifier