Accident Severity Probability Prediction in Seattle City

Abednego Kristanto

8 September 2020

1. Introduction

1.1. Background

Safety is always everyone priority, whenever and wherever they are, especially on the road. However as stated in a well-known Murphy's Law: "Anything that can go wrong will go wrong", traveling on the road is not without danger, since it could go wrong very quickly. No matter how fast the driver's reflex or how experienced the driver is, without proper warning, fatal accident still could occur. It would be nice if there are some system that inform the driver about the danger present along the way during particular road condition. It will be very useful when the driver is unfamiliar with the local road condition. Such system will definitely reduce the traffic accident because with the risk information available, the driver will drive more carefully when the road condition become dangerous according to this information system. This system will definitely be useful for government to improve the safety of the road. Besides that, most vehicle's GPS system manufacturers will be interested in a system that could analyse and predict accident risk on the road because such system could be their useful and attractive selling point of their products.

1.2. Problem

Data to represent road condition that lead to accident is needed for accident severity analysis and prediction. These data could be the the road condition during the accident, the weather, the light condition, the driver condition, etc. From these data, a model could be built to predict the severity of accident if it occurs during that particular road, and driver conditions.

2. Data Acquisition, Selection, and Cleaning

2.1. Data Source

Dataset for this model are accidents report recorded in Seattle City between January 1st, 2004, and May 20th, 2020. This dataset is available as example dataset in Coursera Applied Data Science Capstone Course, and can be downloaded in https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv. This dataset contain many features that are described in its metadata. The metadata for this dataset can be downloaded in https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf. Not all features in the dataset will be utilized in the model. The next sections will describe the features selection, and data cleaning.

2.2. Features Selection

The machine learning model will be used to predict the severity of the accident. Therefore, the dataset attribute "SEVERITYCODE", or a code that corresponds to the severity of the collision will be maintained as training data. After checking the features in the dataset, features are selected for accident severity prediction model building. Features selected are features that can be related with road condition, and driver condition during the accident. These selected features are summarized in the table below:

Table 1. Summary of selected features with their description

Feature Name	Description
"X"	Location's latitude
"Y"	Location's longitude
"ADDRTYPE"	Collision address type.
"PERSONCOUNT"	The total number of people involved in the collision
"VEHCOUNT"	The number of vehicles involved in the collision.
"INATTENTIONIND"	Whether or not collision was due to inattention.
"UNDERINFL"	Whether or not a driver involved was under the influence of drugs or alcohol.
"WEATHER"	A description of the weather conditions during the time of the collision.
"ROADCOND"	The condition of the road during the collision.
"LIGHTCOND"	The light conditions during the collision.

“SPEEDING”	Whether or not speeding was a factor in the collision.

There are several features that have been drop because of several reasons which are: unknown features, features with very few data, redundant features, unused identification codes, description that cannot be quantified with number, date and location features that could not be used for prediction, and features that not related to road condition, and driver condition during the accident. Dropped features, their descriptions, and the reason why those features are dropped is summarize in the table below:

Table 2. Summary of dropped features with their description and reason for dropping

Feature Name	Description	Reason for dropping
“OBJECTID”	Unique identifier	Unused identification codes
“INCKEY”	A unique key for the incident	Unused identification codes
“COLDETKEY”	Secondary key for the incident	Unused identification codes
“REPORTNO”	Report number	Unused identification codes
“STATUS”	Data status	Unknown feature
“INTKEY”	Key that corresponds to the intersection associated with a collision	Unused identification codes
“LOCATION”	Description of the general location of the collision	Description that cannot be quantified
“EXCEPTRSNCODE”	Unknown	Unknown feature
“EXCEPTRSNDESC”	Unknown	Unknown feature
“SEVERITYCODE.1”	A code that corresponds to the severity of the collision	Redundant feature with “SEVERITYCODE” feature
“SEVERITYDESC”	A detailed description of the severity of the collision	Description that cannot be quantified
“COLLISIONTYPE”	Collision type	Not related to road condition
“PEDCOUNT”	The number of pedestrians involved in the collision.	Redundant feature with “PERSONCOUNT” feature
“PEDCYLCOUNT”	The number of bicycles involved in the collision	Redundant feature with “VEHCOUNT” feature
“INCDATE”	The date of the incident.	Date and time feature
“INCDTTM”	The date and time of the incident.	Date and time feature

“JUNCTIONTYPE”	Category of junction at which collision took place	Redundant feature with “ADDRTYPE” feature
“SDOT_COLCODE”	A code given to the collision by SDOT.	Unused identification codes
“SDOT_COLDESC”	A description of the collision corresponding to the collision code.	Unused identification codes
“PEDROWNOTGRNT”	Whether or not the pedestrian right of way was not granted.	Features with very few data < 5% occurrence
“SDOT_COLNUM”	A number given to the collision by SDOT.	Unused identification codes
“ST_COLCODE”	A code provided by the state that describes the collision.	Unused identification codes
“ST_COLDESC”	A description that corresponds to the state’s coding designation.	Unused identification codes
“SEGLANEKEY”	A key for the lane segment in which the collision occurred.	Unused identification codes
“CROSSWALKKEY”	A key for the crosswalk at which the collision occurred.	Unused identification codes
“HITPARKEDCAR”	Whether or not the collision involved hitting a parked car.	Redundant feature with “COLLISIONTYPE” feature that has type “Parked Car”

2.3. Data Cleaning

It is clear that after just checking few rows in the dataset, the dataset needs some cleaning to be done. The first step is to change or remove features with NaN or empty value. All features that have NaN or empty values have to be identified, and will be dealt accordingly. In the feature “ADDRTYPE”, there are 1,926 rows with empty values. These empty rows will be dropped because there is no way to check this data. There are no missing values in feature “PERSONCOUNT”, and “VEHCOUNT”. The features “INATTENTIONIND”, “UNDERINFL”, and “SPEEDING” are features with yes or no value. The missing data can be interpreted as “no” value, and string “Y” can be interpreted as “yes” value. Yes or no value will be converted to integer 1 for “yes”, and 0 for “no”, so that it could be used in the predictive model. There are a few exceptions in the “UNDERINFL” feature since multiple formats are used to represent yes and no values. In the “UNDERINFL” feature, the value ‘Y’, and ‘1’ will be converted to integer 1, then,

the missing values, ‘N’, and ‘1’ will be converted to integer 0. The features 'WEATHER', ‘ROADCOND’, and ‘LIGHTCOND’ have missing values of 5,081; 5,012; and 5,170, consecutively. The missing values in these features will be change to “Unknown” since values “Unknown” are already in the data. After this data cleaning process, the data will be ready to explore, and transform as needed by the machine learning algorithm. After cleaning data frame is containing 189,339 rows of data in 12 columns of features.

3. Methodology

3.1. Exploratory Data Analysis

The problem in this modelling is a classification problem with discrete target class, accident severity. In the dataset, there are only two unique values in feature “SEVERITYCODE”, which are 1 for property damage with no injury, and 2 for accident that lead to human injury. There are no number of people injured mentioned in the dataset, therefore, new class could not be created. However, there are still many data to be explored from this dataset. In this exploratory data analysis, the combination of road condition that could lead to accident is explored. In the dataset, the road condition can be determined by features “WEATHER”, “ROADCOND”, and “LIGHTCOND”. After the unique values of each feature has been checked, there are two categories that cannot be interpreted, which are categories: “Unknown”, and “Other”. These uninterpretable categories will be dropped from exploratory data analysis data frame, but will be kept for predictive modelling because of their significant amount.

3.1.1. Conditions That Lead to Property Damage

This accident severity class is the least severe. The road condition that leads to property damage will be grouped and counted with pandas group by, and value counts methods. The data is grouped by three features: “WEATHER”, “ROADCOND”, and “LIGHTCOND”. Then, accident counts that lead to property damage or “SEVERITYCODE” equal to 1 is selected and plotted using bar chart. The resulting bar chart is shown in the figure below:

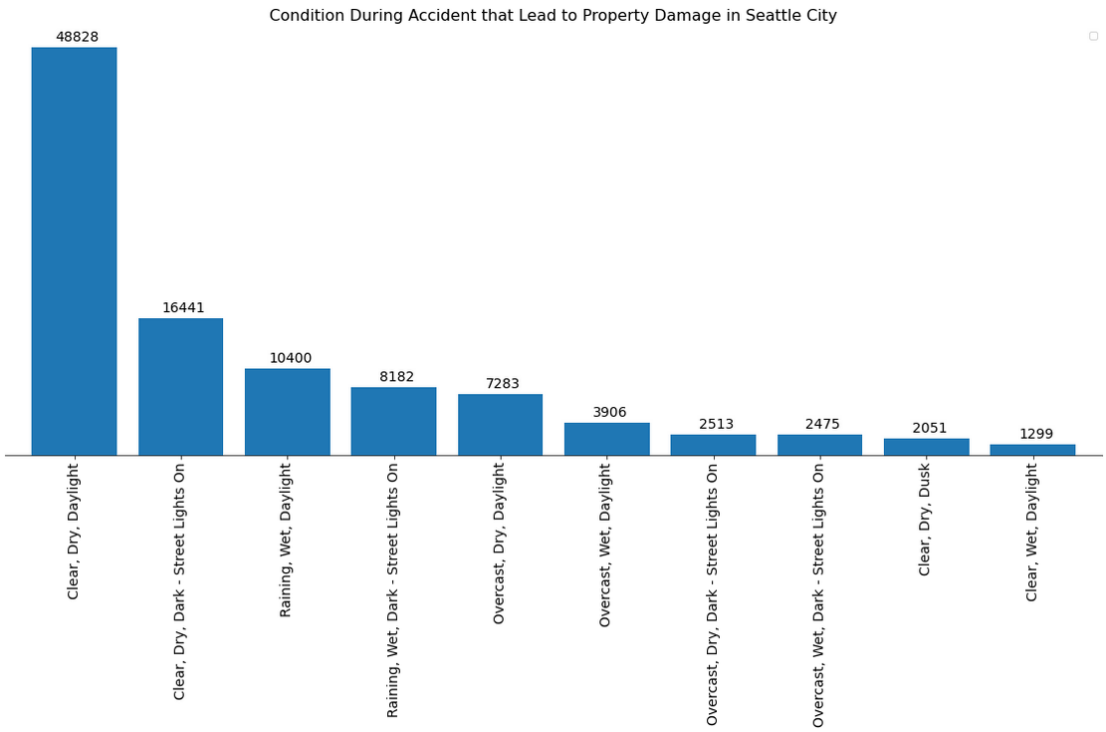

Figure 1. Condition during accidents that lead to property damage in Seattle City

From bar chart above, interestingly, most accidents that lead to property damage were occurred during clear, and dry road condition in day light and evening with street lights on, being in the first and second place. The third and fourth place is all during rainy condition.

3.1.2. Conditions That Lead to Injury

Similar to the data exploration above, accident counts that lead to person or people injury is selected by taking data from exploratory data frame with “SEVERITYCODE” equal to 2 is selected and then plotted using matplotlib bar chart. The resulting bar chart is shown in the figure below:

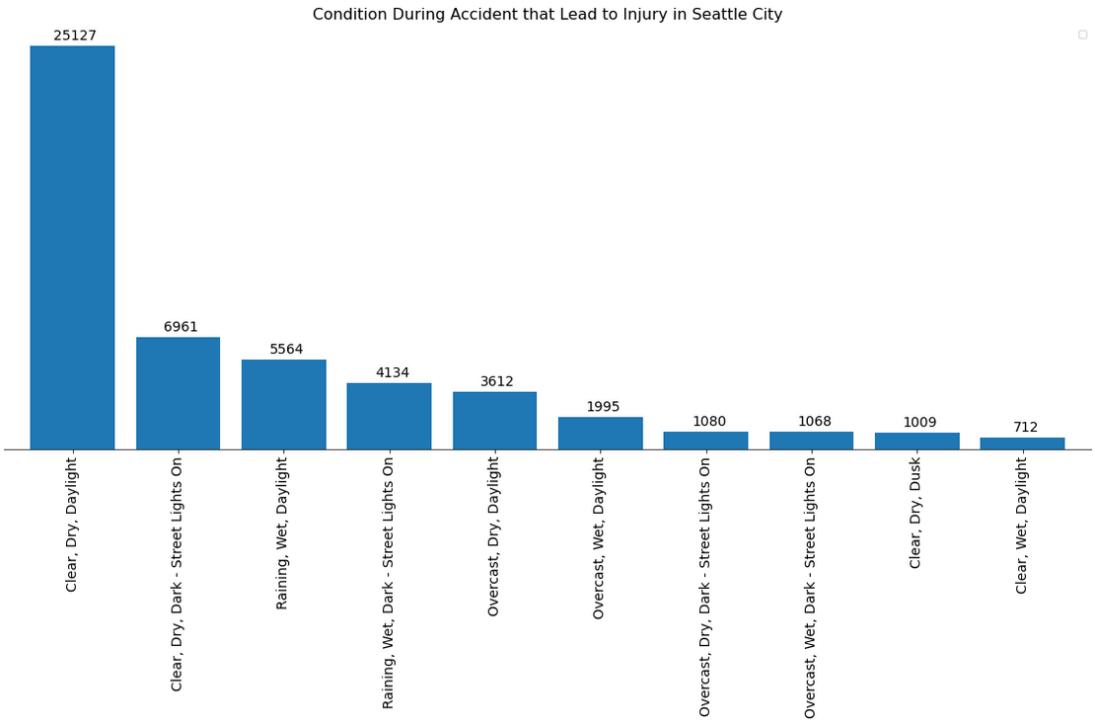

Figure 2. Condition during accidents that lead to injury in Seattle City

The result is quite similar with previous chart, clear, and dry road condition still the condition during most of accident that lead to injury in Seattle City. The high number could be caused by the weather of Seattle City itself that might be sunny most of the time, so that most accidents occurred during that condition. Another explanation is that most of the accidents might be occurred during crowded time on the road which explain most of the accidents occurred during daylight, when people went to their work, and during evening, when people went back to their home.

3.2. Predictive Classification Modelling

3.2.1. Converting Categorical Data into Numerical Values

There are several categorical data in the dataset that need to be converted into numerical values before being used to build a machine learning model. The first step is checking the unique values in every categorical data in the data frame. There are values Unknown and Other in the “WEATHER”, “ROADCOND”, and “LIGHTCOND” features, to simplify the data, the data with value Other is merged with Unknown. Then, the data need to be prepared by separating features to predict with target feature, in this problem “SEVERITYCODE”. Latitude and longitude data is also included

in the target for map plotting, but it will also be separated from “SEVERITYCODE” after training and testing data split.

The conversion from categorical data into numerical values is using Ordinal Encoder. This encoder module is available in sklearn.preprocessing, and will convert each values in categorical data into a number. Ordinal Encoder transformation process is done by fitting features that will be used to train and test the model into Ordinal Encoder function. After this transformation, the data will be ready for data splitting and normalization.

3.2.2. Data Spliting, and Normalization

Data splitting process for this model will use train_test_split function from sklearn.model_selection. Data test size for this model is 30%, and the rest will be data for model training. Data splitting with this function is set randomly, however to ease the debugging effort, random_state seed number 9 is used. If random_state parameter is set to an integer number, the training, and testing data will not change during each run. After data splitting, latitude and longitude in the target will be separated, but the data index will be maintained for map plotting.

The next step is data normalization. This step is necessary to ensure that the different scale in the feature’s values will not affecting the machine learning model. StandardScaler function from sklearn preprocessing is used for data normalization. This function will transform the data in each feature and try to make the feature’s mean equal to zero.

3.2.3. Logistic Regression Classification

Logistic Regression is used for this classification problem because this algorithm produces classification result that based in the most probable class for a particular data point. This means this algorithm also produces probability of a given data point belong to a class. Therefore, it is a suitable algorithm to reach the objectives of the model which is predicting the probability of accident severity.

Other machine learning algorithm called Support Vector Machine (SVM) is also capable of producing the probability of class prediction. However, this algorithm is very computationally intensive, especially if working with large dataset as used in this

modelling. Thus, Logistic Regression model is used because of it need much less computational resource than SVM, and it also means that a model using Logistic Regression algorithm will be easier to expand if there are more data available.

The model then will be evaluated using Jaccard similarity score, f1-score, and logarithmic loss, to measure its accuracy. Beside those evaluation metrics, confusion matrix will also be used to evaluate how many data predicted correctly or incorrectly by the model. Classification report will also be displayed to check the precision, recall, and f1-score of the model.

4. Result and Discussion

4.1. Logistic Regression Classification Modelling

Logistic Regression model has been created with regularization value C equal to 0.01, and using solver lbfgs. The model is trained using training dataset, and tested using testing dataset, which had been separated in previous step. The predicted value is saved in yhat_LR, and the probability of data classification is saved in yhat_LR_proba. Three evaluation metrics are used to evaluate the accuracy of the model. The result of the model evaluation using Jaccard similarity index is 0.7151, using f1-score with weighted average is 0.6588, and using logarithmic loss is 0.5692. The accuracy of the model is not very good but it is capable to predict correctly most of the test dataset.

4.2. Confusion Matrix and Classification Report

Confusion matrix evaluation is also done for this model to see deeper into the result of the model prediction. The confusion matrix of this model that tested with testing dataset is shown in figure below:

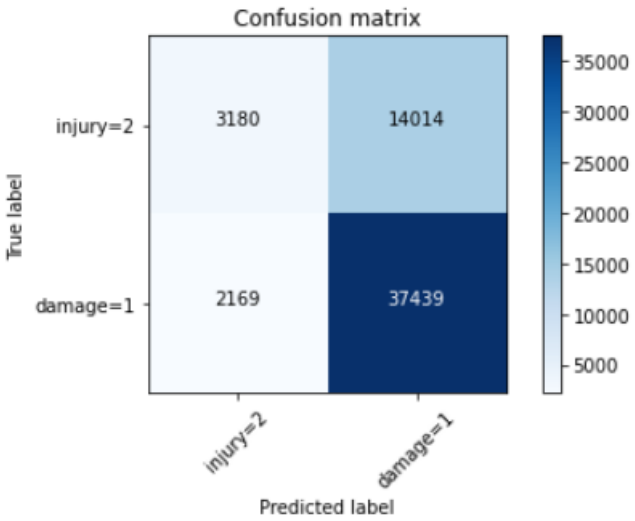

Figure 3. Confusion matrix of the logistic regression model to predict accident severity

As seen in the figure above, most of data with label 1 or property damage is predicted correctly. However, most of data with label 2 or injury is predicted incorrectly. Classification report can be used to see this clearly by showing precision, recall, and f1-score for each severity categories. The printed classification report from this model is shown in the figure below:

	precision	recall	f1-score	support
1	0.73	0.95	0.82	39608
2	0.59	0.18	0.28	17194
micro avg	0.72	0.72	0.72	56802
macro avg	0.66	0.57	0.55	56802
weighted avg	0.69	0.72	0.66	56802

Figure 4. Classification Report of the logistic regression model to predict accident severity

The classification report above showed that even though the accuracy is not very good, it is still acceptable since majority of the data is predicted correctly. However, model improvement by using more data or more feature could be done to improve the accuracy. Because of this model using logistic regression classification, the model could be easily expanded without taking too much computation resource.

4.3. Application Example Map Plotting

Usage example of this model is to predict the severity of an accident, and its probability, that could occur in a particular road type. This prediction value could be

sent to the driver in real-time, or it could be a part of a digital city map. This map plotting will try to use 100 y_test data as an example to plot the accident severity probability prediction in a folium map. The index from testing data after random split will be used to select data from clean dataset that contain accident locations' latitudes, and longitudes. The predicted classification probability then added to the dataset that contain location coordinate. After the dataset ready, folium module is imported, and map of Seattle City with location latitude = 47.608013, and longitude = -122.335167 is generated. For the labels, a series is created from the information of road type, weather, road condition, property damage probability, and injury probability in each rows in the dataset. The result of this example map plotting is shown in the figure below:

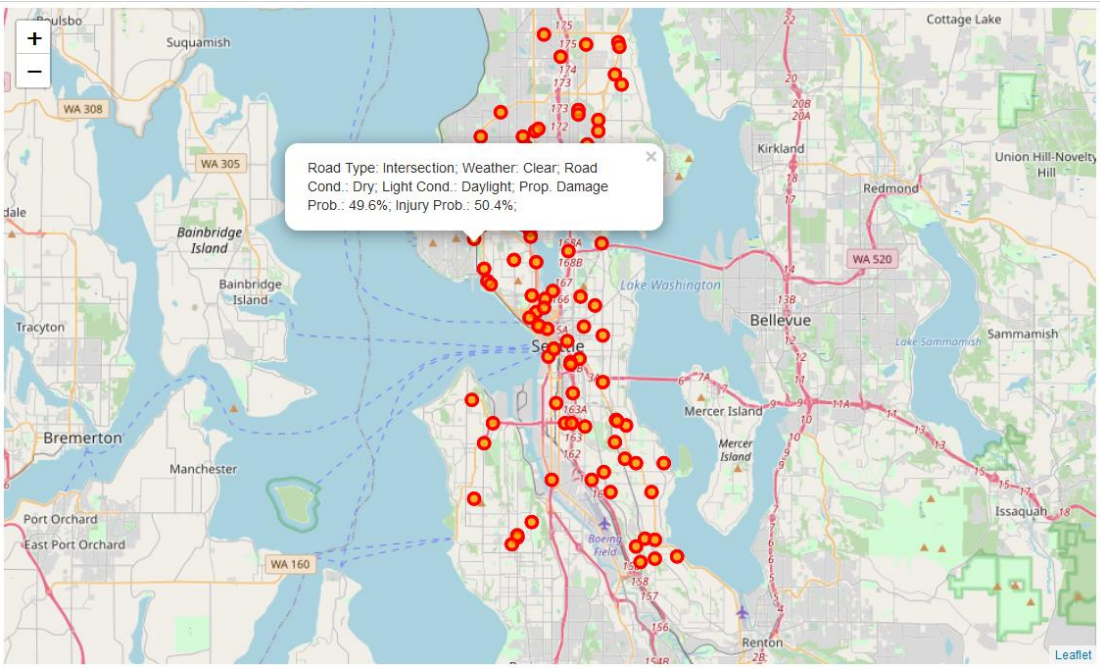

Figure 5. Interactive folium map plot that showing the severity probability if accident occurs in a location during stated specific condition.

4.4. Possible Future Works

There are many possible future works that can be done with this model. The first and the foremost is improving the model accuracy. This can be done by adding more data, features, and accident severity class. Because the features are very diverse, the model is having difficulties to classify it into only 2 class. I think adding more class, for

example separating injury and serious injury. Adding feature on how many people injured and use it to further classify injured class is also a good option to improve the model accuracy.

Further development on the application of this model as in map plotting example above is also a good option for future work. A system can be built where the weather and road condition is inputted in real-time during user driving a vehicle, then using this model to predict the severity probability if accident occurs in the road type where the user is on. This system could warn the driver of the road with high-risk of severe accident in a real time, and it is very useful to improve the travel safety even in an unfamiliar location and condition.

5. Conclusion

A model to predict accident severity probability prediction in Seattle City has been created using Logistic Regression Classification model. The model has the following accuracy: Jaccard similarity index = 0.7151, f1-score with weighted average = 0.6588, and logarithmic loss = 0.5692. An example application of the model has also been created by plotting testing data features and accident severity probability prediction in an interactive folium map. The model is very potential to be developed further in the future, for example by improving its accuracy, and expand its application using real-time data.