PREDICTING CAR SEVERITY ACCIDENT

IBM Applied Data Science Capstone

Final Assignment

October 2020

INTRODUCTION

This study provides a comprehensive analysis to car accident severity problem in Seattle city. Car accidents might not only affect those who are involved in a car crash physically, emotionally and financially, but also affect others by causing traffic delay. The National Highway Traffic Safety Administration [1] reported the total number of fatalities in car accident crashes increased from 26 to 36,560 starting from year 1899 to 2018. This is an enormous increase and we aim to address this issue in this study.

The objective of this study is to develop a model that could predict the severity of car accidents given by the factors affecting the collision in Seattle city. These factors are not restricted to road and visibility and weather condition. However, we will identify the number of significant effective factors and develop a model which is able to predict the severity of accident in the Seattle city.

The developed model could assist drivers who tend to travel by any motor vehicle with the required information on road traffic and also the possibility of getting into a car accident. Furthermore, the users would know how severe the accident would be. Therefore they are

able to make decision in advance prior to the travel. It potentially will result in reduced number of motor vehicle crashes, injury and fatality rate. Other beneficiaries in this study are the emergency unit departments that could potentially present more advanced help carrier to the community.

METHODOLOGY: PROCESS MODEL

In this study, we follow CRISP-Data Mining (CRoss-Industry Standard Process for Data Mining) methodology [2]. It provides a guideline of a data mining\data science project life cycle (refer to Figure 1). We have implemented these phases and the detailed process is explained in the following sections.

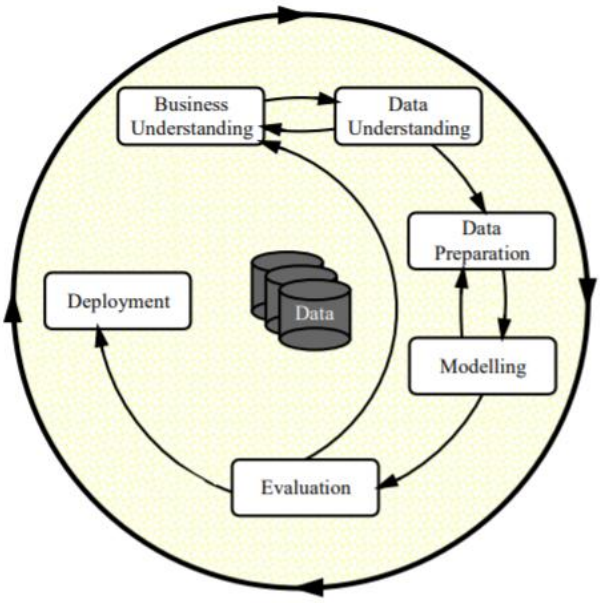


Figure 1: Phases of the CRISP-DM Process Model

DATA UNDERSTANDING

1. Introduction

This section initiates with data collection process and it is comprised of the steps involved in getting familiar with dataset and to identify the quality of collected data. Basically, the data describe all types of collisions displayed at the intersection or mid-block of a segment from January 2004 to May 2020. The collisions are provided by Seattle Police Department (SPD) and recorded by Traffic Records group. They are accessed and collected from the following link (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv).

The original dataset is comprised of 194,673 rows and 38 columns (Figure 2). The objective is to identify the impact of traffic using the severity of accident. Therefore, the "SEVERITYCODE" attribute, which describes the fatality of an accident, will be used as the dependant (target) variable. This code corresponds to the severity of the collision through 5 values (3: fatality, 2b: serious injury, 2: injury, 1: property damage, 0: unknown).

```
In [7]: df.size
Out[7]: 7397574

In [8]: df.shape
Out[8]: (194673, 38)

In [9]: print("The number of null values for severity code is:", df['SEVERITYCODE'].isnull().sum())
        The number of null values for severity code is: 0
```

Figure 2: Dataset size

In the current dataset, the total number of cases relevant to code 1 and 2 (injury and property damage respectively) is available. We identified there are no missing values in this attribute (Figure 2). Moreover, the remaining 37 columns are described as independent variables and their corresponding values (Figure 3 and Figure 4).

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGH
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Dayli
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark Light
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Dayli
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Dayli
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Dayli

Figure 3: Sample of original dataset (dependant and independent variables)

The data types of these attributes are presented in Figure 4. There are total numbers of 22 attributes that are presented as object data type and the remaining 16 are presented as integer or float data types. It provides a clear perspective about which attributes needs to be normalized before applying into the models.

```
In [3]: df.dtypes

Out[3]: 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
```

Figure 4: Attribute data types

2. Explanation of Attribute Values

This section presents the attributes codes and their description collected for the study. This phase is essential since it assists to identify and use the attributes in the analysis precisely. It is also critical to the success of pre-processing phase. Table 1 presents these attributes and their description.

Table 1: Attributes, vales and descriptions

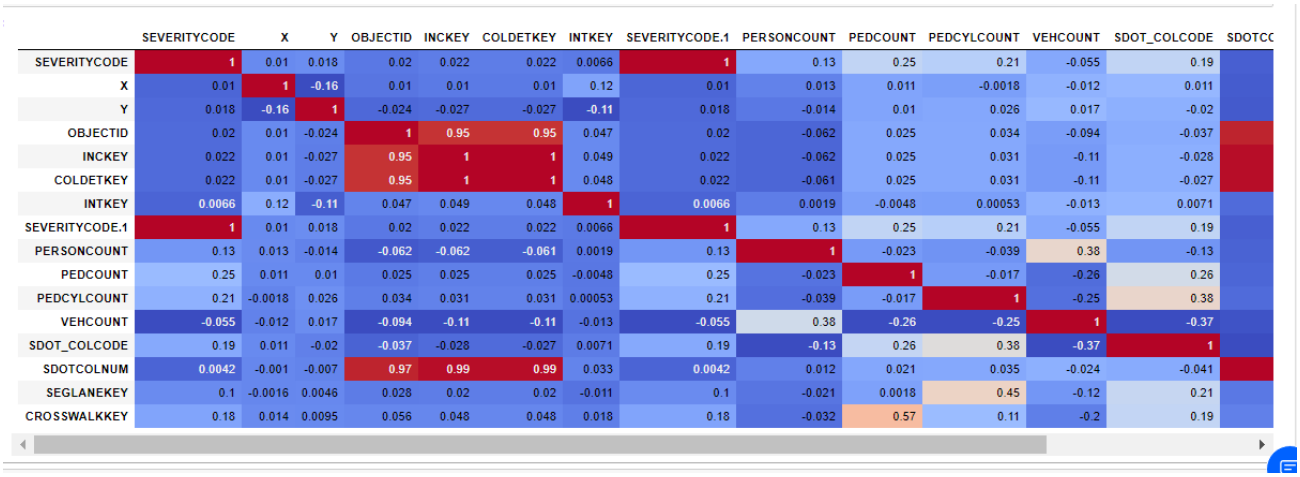
Attribute Code	*Description*
OBJECTID	ESRI unique identifier
SHAPE(X, Y)	ESRI geometry field
INCKEY	A unique key for the incident
COLDETKEY	Secondary key for the incident
ADDRTYPE	Collision address type including; alley, block, intersection
INTKEY	Key that corresponds to the intersection associated with a collision
LOCATION	Description of the general location of the collision
EXCEPTRSNCODE	A code—not known
EXCEPTRSNDESC	A code description —not known
SEVERITYCODE	A code that corresponds to the severity of the collision: 3—fatality, 2b—serious injury, 2—injury, 1—prop damage, 0—unknown
SEVERITYDESC	A detailed description of the severity of the collision
COLLISIONTYPE	Collision type
PERSONCOUNT	The total number of people involved in the collision
PEDCOUNT	The number of pedestrians involved in the collision
PEDCYLCOUNT	The number of bicycles involved in the collision.
VEHCOUNT	The number of vehicles involved in the collision
INJURIES	The number of total injuries in the collision
SERIOUSINJURIES	The number of serious injuries in the collision
FATALITIES	The number of fatalities in the collision
INCDATE	The date of the incident
INCDTTM	The date and time of the incident
JUNCTIONTYPE	Category of junction at which collision took place
SDOT_COLCODE	A code given to the collision by SDOT
SDOT_COLDESC	A description of the collision corresponding to the collision code
INATTENTIONIND	Whether or not collision was due to inattention. (Y/N)
UNDERINFL	Whether or not a driver involved was under the influence of drugs or alcohol

Attribute Code	*Description*
WEATHER	A description of the weather conditions during the time of the collision
ROADCOND	The condition of the road during the collision
LIGHTCOND	300 The light conditions during the collision
PEDROWNOTGRNT	Whether or not the pedestrian right of way was not granted. (Y/N)
SDOTCOLNUM	A number given to the collision by SDOT
SPEEDING	Whether or not speeding was a factor in the collision. (Y/N)
ST_COLCODE	A code provided by the state that describes the collision.
ST_COLDESC	A description that corresponds to the state's coding designation
SEGLANEKEY	A key for the lane segment in which the collision occurred
CROSSWALKKEY	A key for the crosswalk at which the collision occurred
HITPARKEDCAR	Whether or not the collision involved hitting a parked car. (Y/N)

3. Data Redundancy analysis

This section presents the data redundancy/ data incompleteness analysis based on the result of correlation analysis. Basically, correlation analysis provides a brief overview of the original data. Figure 5 visualizes a portion of this analysis. The red sections present the existence of highly correlated attributes. These attributes are "OBJECT ID", "INCKEY" and "COLDETKEY". According to the description of these attributes presented in Table 1, these are the unique keys which eventually has no effect in prediction of car severity accident. Therefore, they will be excluded from the analysis.

Moreover, “SEVERITYCODE.1” which is highly correlated to “SEVERITYCODE” found to be a duplicate attribute. We believe “SEVERITYCODE.1” is redundant and it needs to be excluded from the analysis.



	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	INTKEY	SEVERITYCODE.1	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	SDOT_COLCODE	SDOTCC
SEVERITYCODE	1	0.01	0.018	0.02	0.022	0.022	0.0066	1	0.13	0.25	0.21	-0.055	0.19	
X	0.01	1	-0.16	0.01	0.01	0.01	0.12	0.01	0.013	0.011	-0.0018	-0.012	0.011	
Y	0.018	-0.16	1	-0.024	-0.027	-0.027	-0.11	0.018	-0.014	0.01	0.026	0.017	-0.02	
OBJECTID	0.02	0.01	-0.024	1	0.95	0.95	0.047	0.02	-0.062	0.025	0.034	-0.094	-0.037	
INCKEY	0.022	0.01	-0.027	0.95	1	1	0.049	0.022	-0.062	0.025	0.031	-0.11	-0.028	
COLDETKEY	0.022	0.01	-0.027	0.95	1	1	0.048	0.022	-0.061	0.025	0.031	-0.11	-0.027	
INTKEY	0.0066	0.12	-0.11	0.047	0.049	0.048	1	0.0066	0.0019	-0.0048	0.00053	-0.013	0.0071	
SEVERITYCODE.1	1	0.01	0.018	0.02	0.022	0.022	0.0066	1	0.13	0.25	0.21	-0.055	0.19	
PERSONCOUNT	0.13	0.013	-0.014	-0.062	-0.062	-0.061	0.0019	0.13	1	-0.023	-0.039	0.38	-0.13	
PEDCOUNT	0.25	0.011	0.01	0.025	0.025	0.025	-0.0048	0.25	-0.023	1	-0.017	-0.26	0.26	
PEDCYLCOUNT	0.21	-0.0018	0.026	0.034	0.031	0.031	0.00053	0.21	-0.039	-0.017	1	-0.25	0.38	
VEHCOUNT	-0.055	-0.012	0.017	-0.094	-0.11	-0.11	-0.013	-0.055	0.38	-0.26	-0.25	1	-0.37	
SDOT_COLCODE	0.19	0.011	-0.02	-0.037	-0.028	-0.027	0.0071	0.19	-0.13	0.26	0.38	-0.37	1	
SDOTCOLNUM	0.0042	-0.001	-0.007	0.97	0.99	0.99	0.033	0.0042	0.012	0.021	0.035	-0.024	-0.041	
SEGLANEKEY	0.1	-0.0016	0.0046	0.028	0.02	0.02	-0.011	0.1	-0.021	0.0018	0.45	-0.12	0.21	
CROSSWALKKEY	0.18	0.014	0.0095	0.056	0.048	0.048	0.018	0.18	-0.032	0.57	0.11	-0.2	0.19	

Figure 5: Visualizing Data Correlation Analysis

SDOTCOLNUM (number given to the collision by SDOT) is also highly correlated to the mentioned key numbers. However, the coding of this attribute is not provided by experts. Moreover, the number by itself does not provide any meaningful information. Therefore, it is also excluded from the analysis (Figure 6).

Out[55]:

	SEGLANEKEY
0	110885
6532	18
6078	16
10336	13
10342	13
12162	13
8985	12
10420	10
12179	10
10354	10
8816	9

Figure 6: Portion of “SDOTCOLNUM” Attribute values and its counts

Since the data is not pre-processed yet, the correlation analysis does not present very strong correlations. This indicates that data need to be cleaned first. We will pre-process the data and present the results in the subsequent sections.

DATA PREPARATION

The data preparation is a critical task of data pre-processing because it mainly includes all the required activities to construct the final dataset which will be fed into the modelling tools. The data preparation section aims to develop a clean dataset. In this section, the process of balancing the labelled data, handling missing data, data standardization and all the required feature engineering tasks for some attributes are explained.

1. Balancing the Labelled Data

To identify whether the data is in a good state used for the modelling algorithm, we have analyzed the data, attributes and their values. Figure 7 illustrates the distribution of dependant variable in respects to its values; property damage and injury (code 1 and 2 respectively). It presents that the data has unbalanced labels. The unbalanced labels potentially create a biased machine learning model. Therefore, the first task is to balance the data according to the labels.

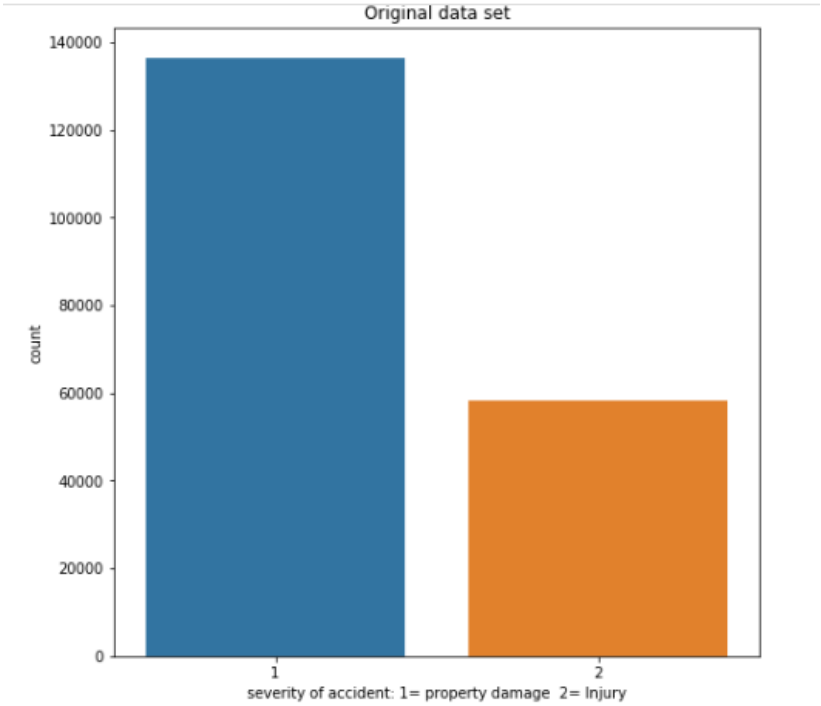


Figure 7: Dependant variable values before balancing

The following figure presents how the data is balanced corresponds to the two code values 1 and 2. As presented next, there are exactly 58188 numbers of labels for each category in the dataset.

```
In [6]: df_balanced['SEVERITYCODE'].value_counts().to_frame()
```

Out[6]:

	SEVERITYCODE
2	58188
1	58188

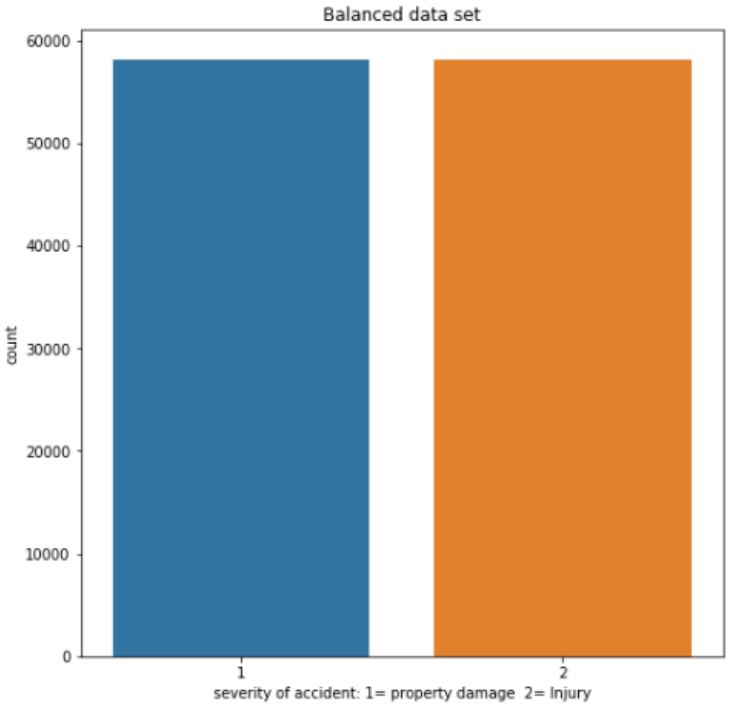


Figure 8: Balanced dataset

2. Handling Missing Data

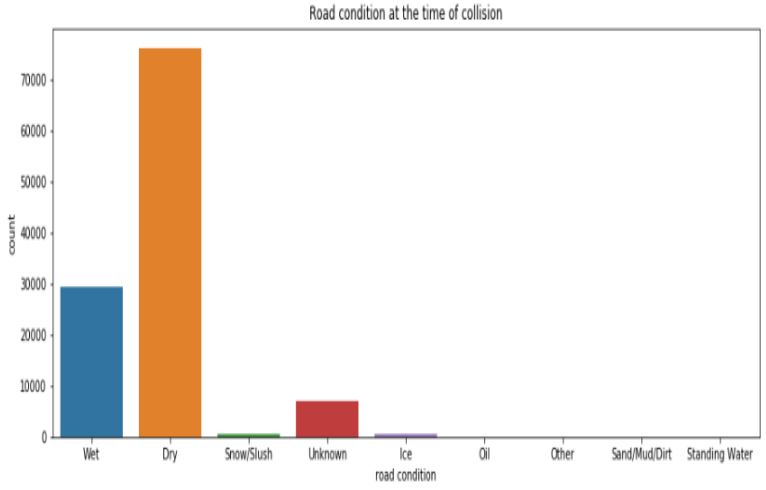
In this section, the attributes and their values are analysed to find the availability of missing values. Figure 9 presents a sample of codes illustrating the number of missing values for the attributes (True presents the number of missing values).

```
ADDRTYPE
False    115464
True        912
Name: ADDRTYPE, dtype: int64

INTKEY
True     72684
False    43692
Name: INTKEY, dtype: int64

LOCATION
False    115101
True       1275
Name: LOCATION, dtype: int64

EXCEPTRSNCODE
True     64850
False    51526
Name: EXCEPTRSNCODE, dtype: int64
```

```
COLLISIONTYPE
False    113699
True       2677
Name: COLLISIONTYPE, dtype: int64

PERSONCOUNT
False    116376
Name: PERSONCOUNT, dtype: int64

PEDCOUNT
False    116376
Name: PEDCOUNT, dtype: int64

PEDCYLCOUNT
False    116376
Name: PEDCYLCOUNT, dtype: int64

VEHCOUNT
False    116376
Name: VEHCOUNT, dtype: int64
```

Figure 9: Number of Missing Values (Corresponds to True value)

Next, all the attributes are evaluated individually to identify the quality of the attribute values. The result identifies that the missing values for the possible predictable attributes are removed. For example, the attribute “ROADCOND” which presents the condition of road at the time collision has less than 0.1% missing values, so the rows related to the missing values are excluded from dataset (Figure 10). Similarly the attribute “LIGHTCOND” which includes 0.1% missing values, the related missing rows are excluded.



Out[59]:

	ROADCOND
False	113568
True	28

Figure 10: Left diagram presents attribute values “ROADCOND” and right diagram presents the number of missing values (28 missing value)

Attribute “SPEEDING” which presents whether or not speeding was a factor in the collision presented as one values “Y” (Figure 11). The corresponding missing values don’t confirm whether they are missed or speed is not a factor of collision. Therefore due to lack of expert knowledge, we have to exclude it from analysis.

Out[54]:

	SPEEDING
Y	5967

Figure 11: Attribute “SPEEDING” value

3. Data Standardization and Encoding

Based on the existing date that the incidents occurred, we generated new attributes which are meaningful to our analysis. The first attribute refers to the days of the week that the accident happened. We encoded it as “dayofweek”. The other one is encoded as "weekend". It presents whether the incident occurs during the weekend or not. We applied *.dt.dayofweek* function that returns the days of the week (Monday as 0 to Sunday as 6). Then, we created and added attribute “weekend” to dataset based on the corresponding number of days.

Figure 12 presents the location of incidents in regards to the day of week and severity of accident. As presented below, the possibility of accidents is higher on Fridays for all locations. This is because most people are commonly in rush to finalize the week duties. In addition, the possibility of accidents to be injury (presented as blue) is likely to happen at the intersections especially on Fridays. This is probably due to the fact that drivers want to pass the yellow traffic light and they consequently increase the speed.

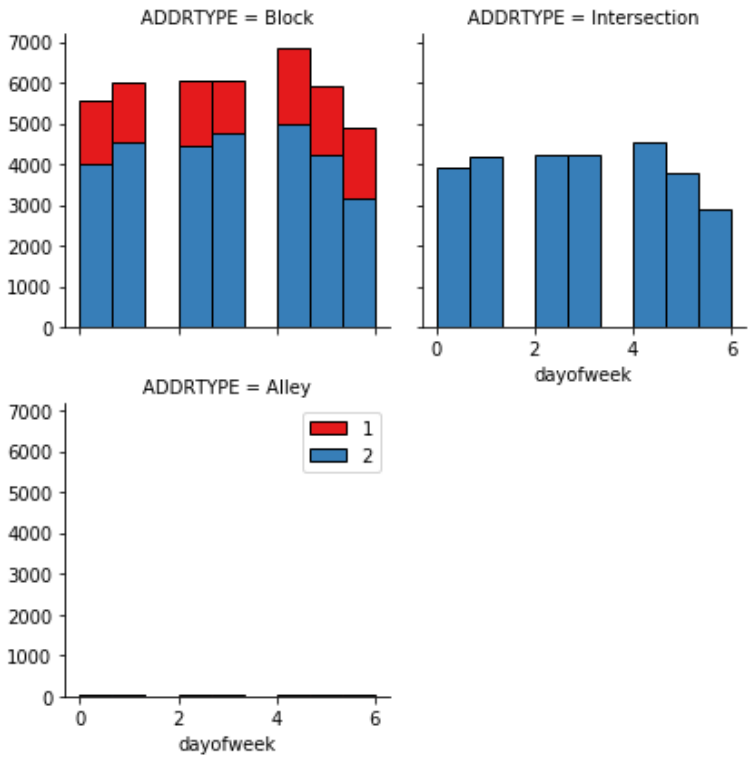


Figure 12: Analysis of the location of accidents in respect to severity and the time of accident

.

The attribute "UNDERINFL" which presents whether or not a driver involved was under influence of drugs or alcohol is presented as "N", "Y", "0" and "1" (Figure 13). We assume "0" means "N" and "1" means "Y". Therefore the values are standardized to form a consistent data attribute.

```
Attribute vale before standardizing data:    UNDERINFL
N      60018
0      47554
Y       3245
1       2635


Attribute value ater standardizing data:    UNDERINFL
N     107572
Y       5880
```

Figure 13: Attribute "UNDERINFL" - Before and After Data Standardization

Similarly, attribute "INATTENTIONIND" which presents whether or not collision was due to inattention is presented as "Y". We assume the missing values are not related to being inattention. Therefore, we encoded the missing values to "N".

MODELING

The fourth phases in CRISM-DM methodology is modeling. In modeling phase, various or a single algorithms are selected and applied to build the models. The studyfocuse on supervised machine learning techniques. They are used to infer a solution based on the labeled training set from the given training set. In this study, the supervised learning algrithms analyse the data about car accident severity in Seattle city and infer a function that can be used for predicting new data samples and accurately determine the class labels for unseen instances.

Among the supervised learning algorithm, we first select decision tree algorithm. Then we apply K-Nearest Neighbourhood and logistic regression to compare the performance of classification results with decision tree. The reason that we applied these three algorithms is that these algorithms are the most commonly classification algorithms and has been widely used in many application domains. In addition, we applid decision tree as the first classification algorithm due to the fact that it is fast and more efficient compared to K-Nearest Neighborhood and other classification algorithms. K-Nearest Neighborhood is slow due to expensive real time execution. It generally has to keep track of all training set and find the neighboorhood nodes.

Decision Tree Algorithm

The classification decision tree algorithm is applied to the car accident data set to build a model from historical accident data. Consequently, the developed trained decision tree is used to predict the class of unknown car accident severity and to identify the most efective factors affecting car accident. The model is developed using the class *sklearn.tree.DecisionTreeClassifier*. The function to measure the quality of a split is specified as “entropy” for the information gain. The maximum depth of the tree and the respective accuracy is calculated and summarized in Table 2. Due to the limited space, the decision tree with maximum depth of 4 is presented in Figure 14 top diagram and a portion of tree with maximum depth of 6 is presented in Figure 14 bottom diagram.

Table 2: Decision tree maximum depth and accuracy

Maximum depth	3	4	5	6
Accuracy	68.4%	69.92%	70.2%	70.3%

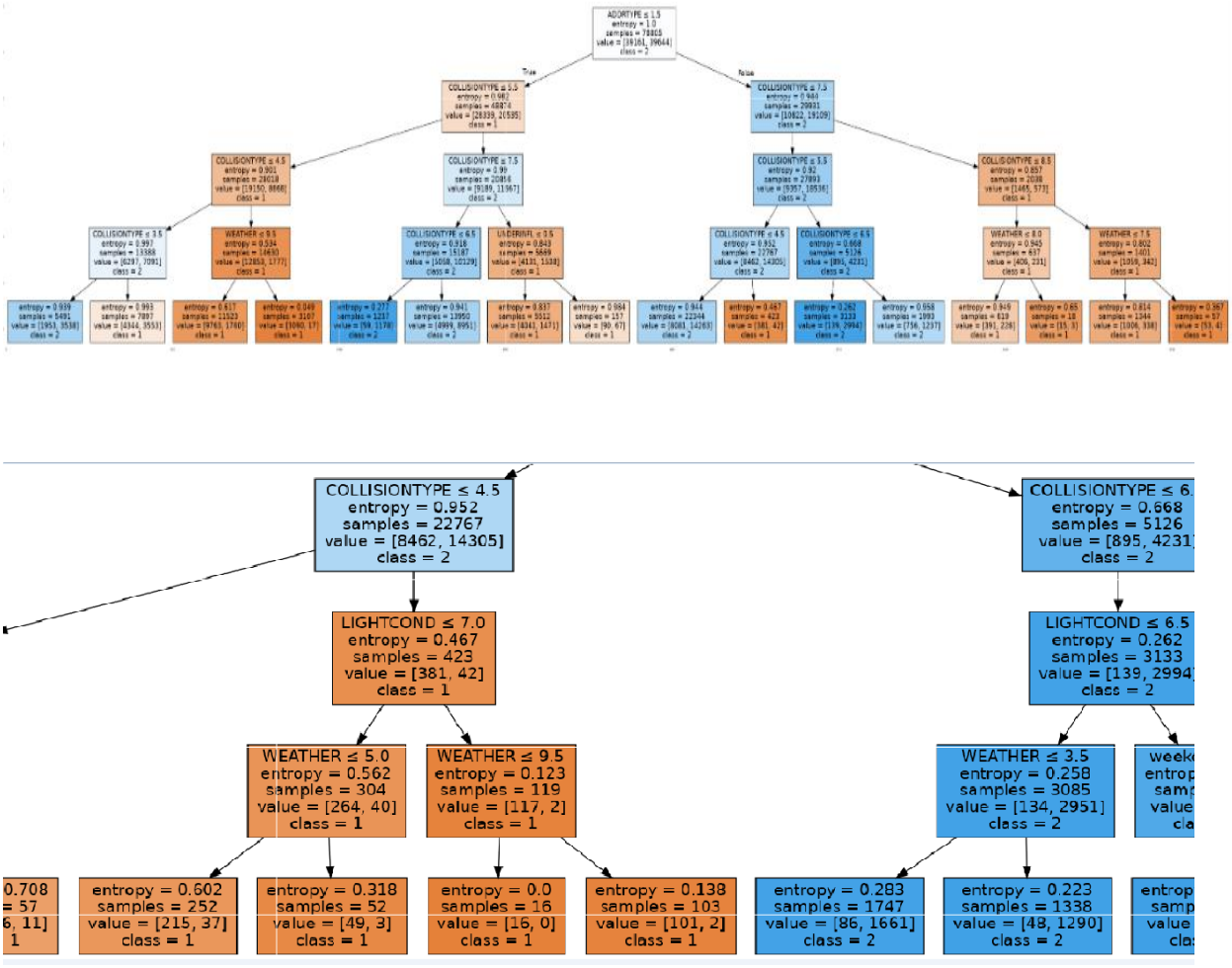


Figure 14: Decision tree (Top: entire tree with depth of 4 and bottom: zoomed portion of the tree with depth of 6)

K-Nearest Neighbourhood (KNN)Algorithm

KNN algorithm is a form of supervised machine learning algorithm. It classifies the data according to the 'K' nearest points to it and accordingly determines the category of the case. In general, larger values of K reduces effect of the noise on the classification, but make boundaries between classes less distinct [3].

To identify the best value of K, we run the model with different values of k and identify the best k with highest accuracy level. As presented in Figure 15, the correspoding accuracy for the KNN algorithm increases from 60% to the highest 68.4% where the K reaches to 11. Consequently, the increment in the value of K, reduces the accuracy of the model. Therefore, the study consider k = 11 with the highest accuracy of 68.4%. Next, the KNN model is developed based on this k value.

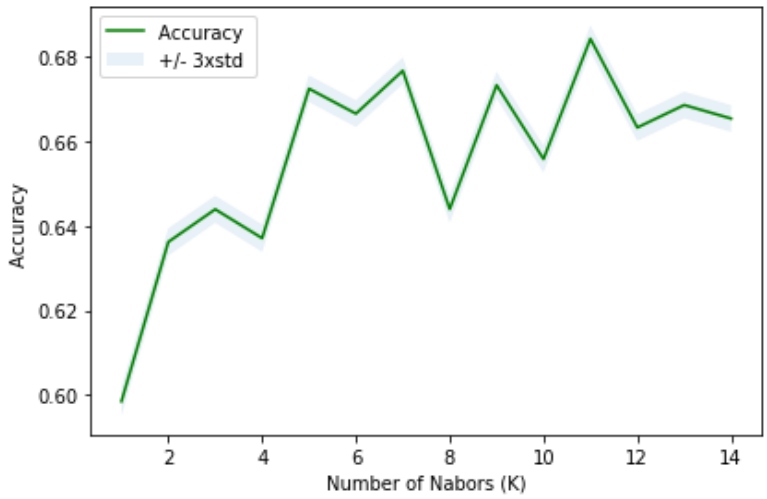


Figure 15: KNNalgorithm and various values for 'K'

The KNN model is developed using the class *sklearn.neighbors.KNeighborsClassifier*. The best value of K is selected to be 11. The weight function used in prediction follows uniform weight. That means all points in each specified neighbourhood are weighted equally. The distance metric used for the tree is Minkowski with p=2 which is equivalent to the standard Euclidean metric. The idea to use distance measure is to find the distance (similarity) between new sample and training cases and then finds the k-closest accident data to new accident data in terms of features such as weather, light condition, day of the week and so on. Finally, the classifier identifies the most appropriate algorithm (e.g. KDTree, Balltree) based on the values passed to fit the method.

Logistic Regression

The next algorithm applied to the set of data is logistic regression. It is a variation of linear regression but in contrast to linear regression which is used to predict the continues values, it predicts the categorical or discrete dependent variables [4]. We selected logistic regression algorithm because the target labels are discrete. The model is developed using class *sklearn.linear_model.LogisticRegression*. The study applies the 'Liblinear' algorithm to optimize the model. It is clarified that Liblinear algorithm is suitable for small datasets. We believe that our dataset, which is not very large does not require very fast processing such as Saga or Sag algorithms.

RESULT EVALUATION AND DISCUSSION

In this study, the derived models from previous section are evaluated to identify how accurate they perform. We examined these models using the popular metrics including confusion matrix, jaccard accuracy, f1 score and log loss for logistic regression model. The evaluation result of these models are discussed in the following sections.

1. Confusion Matrix

Confusion matrix presents the actual and predicted labels from a classification study. The reason that we applied confusion matrix is the ability that it can correctly separate and predict the classes. In our study which is a form of binary classification, we can interpret these numbers as the count of true positives, false positives, true negatives, and false negatives.

The evaluation result from confusion matrix presents that out of 33,774 car accidents in test dataset, a total of 16,548 (10485 + 6063) cases are considered property damage (Figure 16). Consequently, out of these cases, the classifier decision tree correctly predicted 10485 cases as property damage and 6063 of them as injury. In other words, among 10485 cases, the actual severity situation was property damage in test dataset and the classifier also correctly

predicted as property damage. However, while the actual label of 6063 of cases were property damage, the classifier presicted as injury. This is considered as the error of the model for property damage cases.

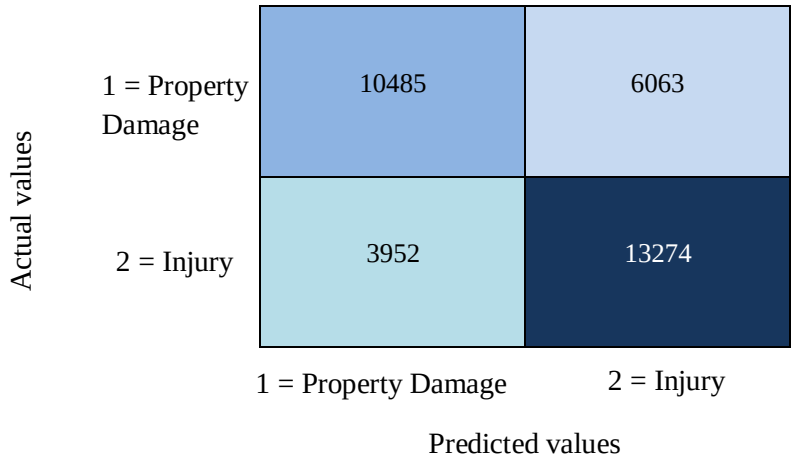


Figure 16: Confusion Matrix for Decision Tree Classifier

Moreover, total of 17226 (3952 + 13274) cases are injury. Out of these amount, the classifier decision tree correctly predicted 13274 cases as injury and only 3952 of them as property damage. In other words, among 13274 cases, the actual severity situation were injury in test dataset and the classifier also correctly predicted as injury. However, while the actual label of 3952 of cases were injury, the classifier misclassified as property damage. This is considered as the error of the model for injury cases. Based on the count of each section, we calculate precision and recall of property damage and injury (Table 3).

Table 3: Measures of accuracy

(micro-averaging : biased by class frequency and macro-averaging: taking all classes as equally important)

Class label	Precision	Recall	F1-score	Support
Property damage	0.73	0.63	0.68	16548
Injury	0.69	0.77	0.73	17226
Micro-average	0.70	0.70	0.70	33774
Macro- average	0.71	0.70	0.70	33774
Weight- average	0.71	0.70	0.70	33774

2. Precision

Basically, precision is a measure of accuracy provided that a class label has been predicted. It is calculated as the number of True Positives divided by the number of all positive results or True Positives plus the number of False Positives (False Positives are cases the model incorrectly labels as positive that are actually negative). Using the confusion matrix presnted previously, Table 3 presents the precisions related to property damage is 73% and for inury cases is 69%. These are satisfactory accuracies considering the existing dataset and lack of expert knowledge in this study.

$$precision = TP / (TP + FP)$$

3. Recall

It is the true positive rate or a measure of True Positive divided by True Positive plus False Negative (False negatives: data points labeled as negative that are actually positive) and calculated as follows. According to Table 3, the recall for property damage cases is 63% and for injury cases is 77%. It presents that our classification model is able to identify 63% and 77% of all relevant cases for property damage and injury cases respectively.

$$Recall = TP / (TP + FN)$$

4. F1 score

The F1 score is a harmonic average of the precision and recall (refer to the following formula); where an F1 score reaches its best value at 1 (perfect precision and recall) and worst value at 0 (if either the precision or the recall is zero). Since our classification resulted in good values for both recall and precision, the f1 score is consequently provided satisfactory results. In this case, 68% of cases are accurately predicted as property damage

and 73% as injury. Referring to Table 4, f1 score for the three classifiers are presnted. As shown, decision tree produces highest accuracy among the models.

$$f1\ score\ = 2\ \times \frac{precision\ \times\ recall}{precision\ + recall}$$

Table 4: F1 score values for three classifers

Algorithm	F1 score
Decision tree	70.2%
K-Nearest Neighbor	68.1%
Logistic Regression	61.2%

5. Jaccard Index

Jaccard Index is a form of accuracy metric that define 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 exactly match with the corresponding true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0. Referring to Table 5, jaccard index for the three classifiers are presnted. As shown, decision tree produces highest accuracy among the models.

Table 5: Jaccard index values for three classifers

Algorithm	Jaccard Index
Decision tree	70.5%
K-Nearest Neighbor	68.4%
Logistic Regression	61.2%

6. Logaritimic Loss (log loss)

Log loss measures the performance of a classifier where the predicted output is a probability value between 0 and 1. The output of logistic regression model of this study is the probability of car severity accidents which is either property damage or injury. The logistic regression model of this study produces a log loss accuracy of 65.6% which is the highest accuracy metrics value for logistic regression model. In the following figure, a sample of predicted probability for each test case is generated. As it is presented, first test case has 34.7% likely to be a property damage case and 65.30% is likely to be an injury case.

```
Out[27]: array([[0.34706476, 0.65293524],
                [0.43647445, 0.56352555],
                [0.34706476, 0.65293524],
                ...,
                [0.57784245, 0.42215755],
                [0.63173308, 0.36826692],
                [0.53169044, 0.46830956]])
```

Figure 17: Sample output of predicated probability for two class labels

CONCLUSION

The decision tree model predicts that the type of collision and location that collision occured are the most effective factor for predicting both types of car accidents. In addition to those factors, weather condition, road condition and attention of driver are the most influencing factors for accidents with injuries. Similarly, light condition, weather condition, and being under influence of drug or alcohol are predicted to be the most influencing factors for property damage accidents. As a result, road condition and attention of drivers are specificly predicted to affect injuries and being under influence of drug or alcohol and light condition are specificly predicted to affect property damage accidents.

The study presents that among the three classifiers, decision tree has performed better with the highest accuray of 70%. This is due to the fact that logistic regression model perform well when the training data is less, and there are large number of features. However, in this study the number of features have to be reduced due to the lack of knowledge from domain experts,

imbalanced dataset, large number of missing values and unclear understanding about the input domain. KNN has presented slightly lower accuracy from decision tree which confirm that these algorithms have approximately performed the same. However, any effort towards improving data, adding more examples or better data samples or features to the training set monotonically will increas the accuracy of the models.

The developed model in this study is able to works as an assistant to help drivers in providing the required information about road traffic, the possibilities in getting into a car accident or identifying the severity of an accident. Therefore, they have option of changing their travel time or route. It potentially results in reduced number of motor vehicle crashes, injury and fatality rate.

REFERENCES

[1] National Highway Traffic Safety Administration (NHTSA). (2020). Traffic Safetey Facts Annual Report. https://cdan.nhtsa.gov/tsftables/Fatalities%20and%20Fatality%20Rates.pdf

[2] The CRISP-DM process model (1999), http://www.crisp-dm.org/

[3] Everitt, Brian S.; Landau, Sabine; Leese, Morven; and Stahl, Daniel (2011) "Miscellaneous Clustering Methods", in Cluster Analysis, 5th Edition, John Wiley & Sons, Ltd., Chichester, UK

[4] Scikit-learn. (n.d.). *-learn 0.22.2 documentation*. scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. Retrieved October 5, 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression