Capstone Project - Car accident severity

Introduction

Road safety is one of the main concerns for every government in the world. Approximately 1.35 million people die in road crashes each year, on average 3,700 people lose their lives every day on the roads, while 20-50 million people suffer non-fatal injuries (resulting in long-term disabilities). For United States, the road crash is also a security risk for drivers and pedestrian. More than 38,000 people die every year in crashes on U.S. roadways. The traffic fatality rate is 12.4 deaths per 100,000 inhabitants and the terrible traffic security condition also cause 4.4 million people suffering from serious injure. The economic and societal damages of this single problem cost every citizen 871 billion dollars. To ease the harsh traffic security issue and offer a solution for the relevant department in the U.S. government, a prediction for the severity of car accidents and a detailed data analysis is crucial for the policy maker.

With the relevant data, I could use data science techniques to explore the incident severity in Python. The core aim of this report is to locate the key factor that lead to an incident in Seattle, as well as to offer some basic descriptions about the Seattle traffic incidents. To achieve this target, this report will present the answers to several issues. I shall start with the location of the incident on the map and the hot areas for the incident; then I will turn to the exploratory analysis for the collision type, location type and severity (the incident scale); in the end, I would build up a model to analyze the factors which may affect the incident severity.

Data Section

I. Data Description

To explore the problems, I utilize some database to acquire data.

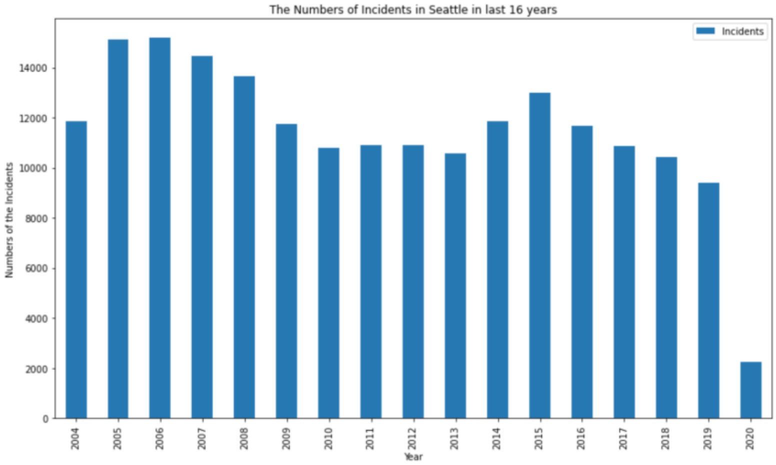
- The example dataset comes from the web page of Coursera Course *Applied Data Science Capstone*. The .csv file contains 19,4673 pieces of collision records of Seattle. The data for this capstone project is offered by SDOT Traffic Management Division and recorded by Traffic Records Group. It covers the annual collisions data from 2004 to present. The time frequency of this dataset is weekly and it shows the traffic collision records in Seattle.The example datasets contain 194,673 pieces of records starting from 2004. The attributes in the datasets covers the weahter condition, road condition, collision type and fatality.
- The other relevant data comes from the database of Seattle government. The main data source is from Seattle Department of Transportation. The basic background information is offered by this department.

II. Data Exploration

This sub-section presents the exploratory data analysis for the example dataset from Coursera Course *Applied Data Science Capstone*. After the primary data cleaning procedure, the remaining useful attributes covers the collision type, address type, severity description, collision scales and potential factors of the collision (weather, speed, road condition, light condition and so on). The following data description comes from all remained data attributes.

Since 2004, the number of collisions has remained on a high level. This figure keeps around 10,000. The following bar chart shows that there is only a fluctuation in last 16 years. In recent 5 years, the number of incidents has decreased. But the incident scale has no significant change in this time period.

Figure 1 The numbers of traffic incidents (Annually) in Seattle in last 16 years



The incident scale is demonstrated by the numbers of involved pedestrians, bicycles and vehicles (the main objects on the road). The annual statistics of the incident scale is as follows.

Figure 2 The scales of traffic incidents (Annually) in Seattle in last 16 years & The numbers of involved vehicles (Annually) in Seattle in last 16 years

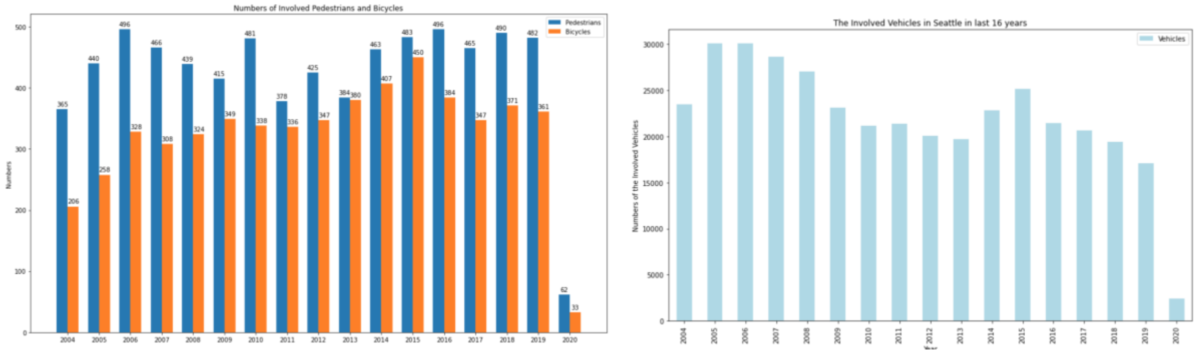


Figure 3 infers that the number of involved pedestrians has been on a high position while the number of involved bicycles has been gyrating up since 2004. However, the number of the involved vehicles has been in a decreasing trend with some small fluctuations.

In this report, I would analysis and predict the target variable SEVERITYCODE and also search for the relevant factors which affect the target variables. In the following table, I remove all the irrelevant data attribute away and list the rest necessary data columns.

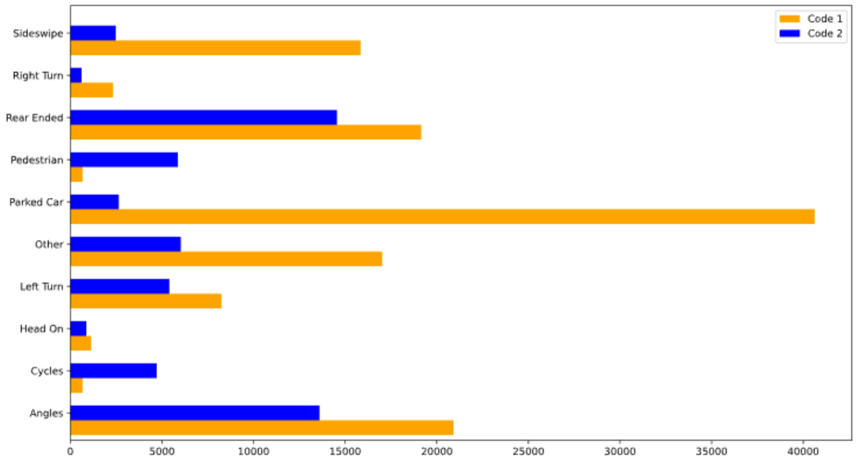
Table 1 The used variables in this report

Variables	Description
Longitude	The geographic data for incident locations
Latitude	
OBJECTID	The series number of the incidents
SEVERITYCODE	**The code to describe the severity of each collision**
ADDRTYPE	Whether the collision happened in block or intersection
JUNCTIONTYPE	The decription of the collision places
ROADCOND	The road status when collisions happened
LIGHTCOND	The light condition when collisions happened
WEATHER	The weather condition when collisions happened
PERSONCOUNT	The number of people involved in the incidents
VEHCOUNT	The number of vehicles involved in the incidents
UNDERINFL	Whether the driver was under alcohol or drugs influence or not
INATTENTIONIND	Whether the collision was caused by inattention

Exploratory data analysis

To analyze the main factors for the collision, this section provides the primary exploratory analysis for the dataset. It shows how the collision type, weather, junction type and other attributes affect the

Figure 3 The severity distribution for different collision type



collision severity. Figure 3 is about the collision severity of different collision type, inferring that property damage only collision happens more frequently than injury collision among all collision types. Figure 4 shows how the collision severity distributes at different areas of roads.

Figure 4 The collision severity distribution in different areas of the roads

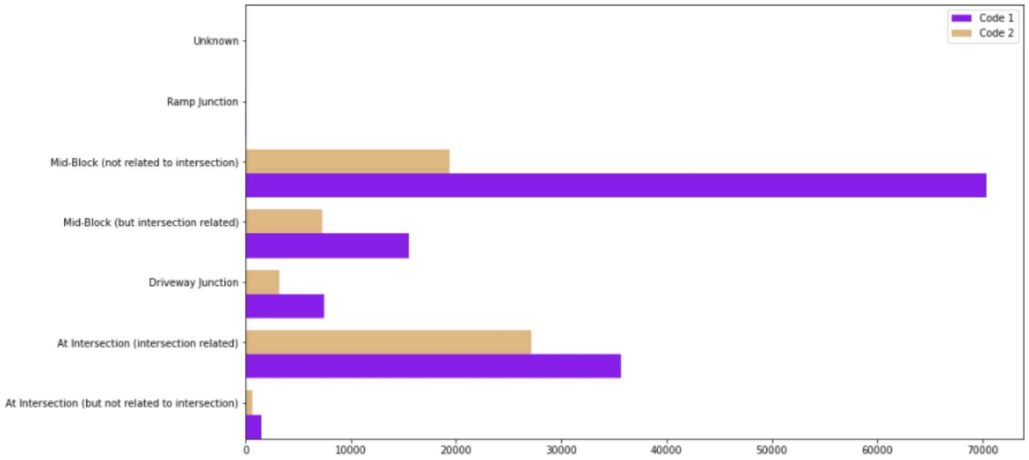


Figure 5, 6 and 7 separately provide a glimpse of the collision severity happened under different external circumstance. From the figures we could find that all the mentioned factors, including collision type, junction type and the external conditions (weather, road and light conditions) have different level influence on the collision severity. Then we could turn to machine learning algorithm section.

Figure 6 The collision severity distribution under different weather conditions

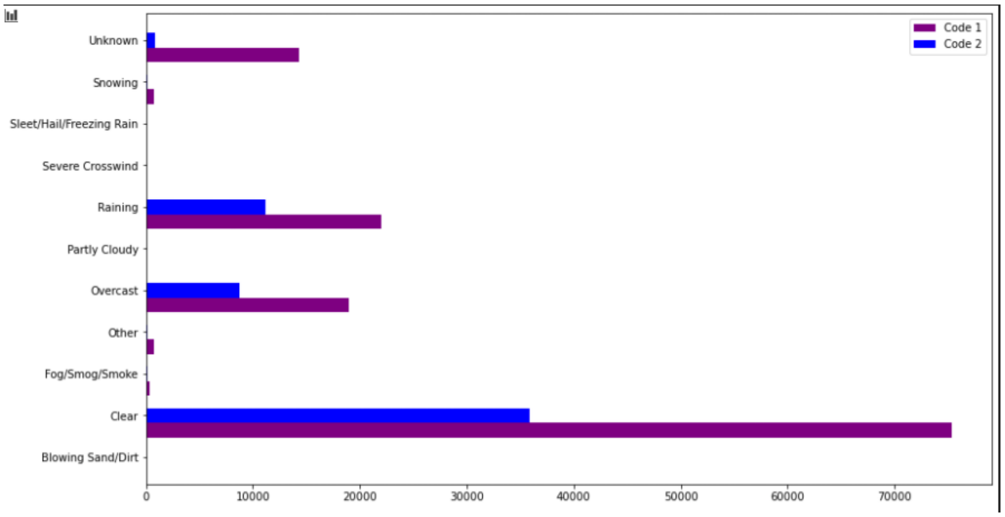


Figure 7 The collision severity distribution under different light conditions

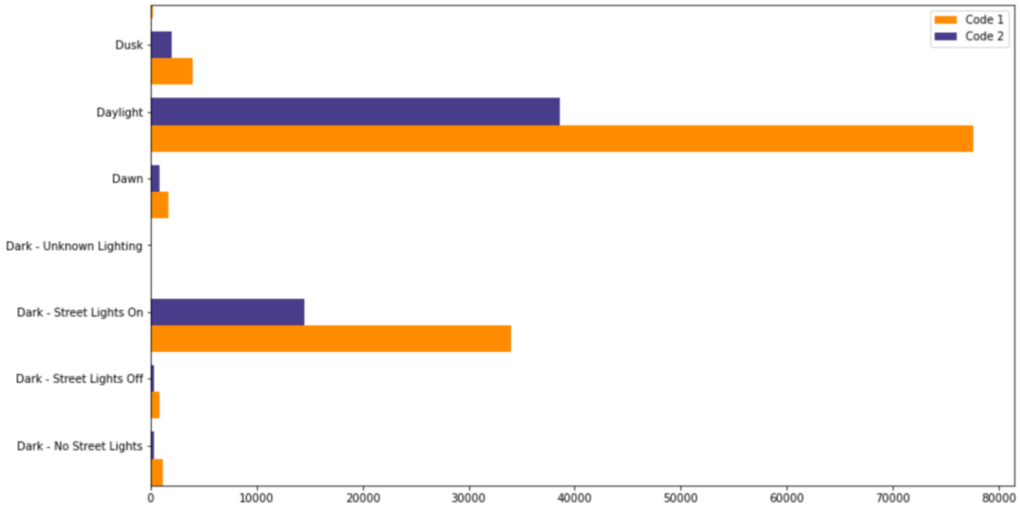
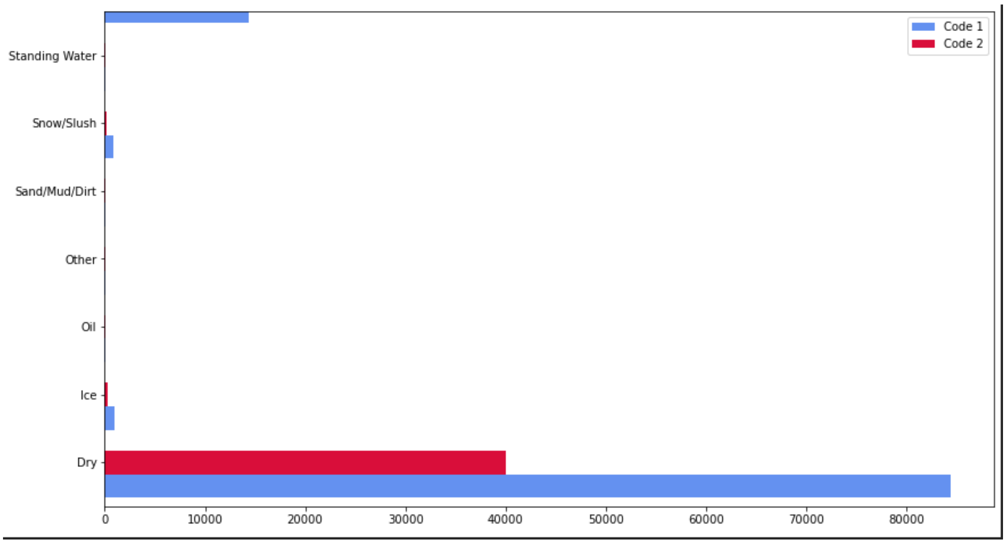


Figure 8 The collision severity distribution under different light conditions



The upper exploratory analysis results demonstrate how the recorded factors in the dataset affect the collision severity separately. The following section would apply several popular machine learning algorithms and build up models to analysis and predict the collision severity.

Methodology

1. Data Pre-processing

With the help of python library *pandas*, I finished the data cleaning process through several procedure. This step makes the dataset analyzable and easy to apply the machine learning algorithms.

i. Dropping all the irrelevant variables and attributes

Here I dropped some unclear and irrelevant variables among 37 attributes in the datasets. “OBJECTID”, “INCKEY”, “COLDETKEY” and some other variables are not closely related to the collision severity, while the other variables, such as “EXCEPTRSNCODE”, have no clear explanations in the offered data instructions. For the further analysis, the mentioned data variables need to be removed.

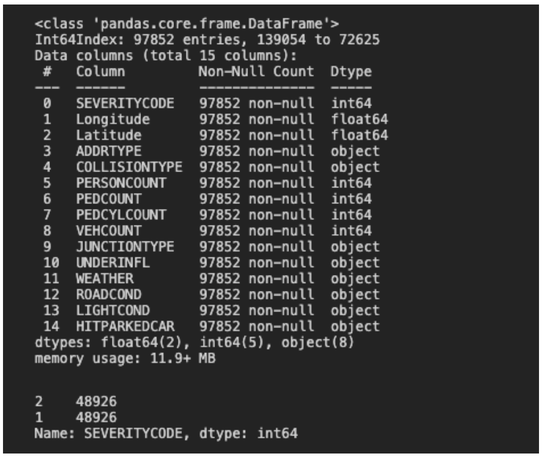
ii. Dealing with missing data

I use the codes *dataframe.drop()* and *dataframe.replace()* to replace the typo and meaningless values like “unknown” or “Other” in the dataset, as well as drop the columns with too many missing values. For the columns with few missing values (missing value less than 20%), I dropped all the missing values in these columns. After this procedure, the dataset still remains 15 columns and 143,747 rows inside.

iii. Dataset balancing procedure

As the requirement of the algorithms, the clean dataset need to be balanced before being used in any algorithm. Otherwise, there will be some biased results coming out through the algorithms. There are several methods that could help the python users to balance the datasets, such as randomly moving the values from the larger data group to the smaller group or applying the data balancing library to re-balance. Here I used the first method. I randomly sampled out 48,926 rows from the larger group and re-group these values into the smaller group (the balanced dataset is shown in Figure 9).

Figure 9 The information of the re-balanced dataset

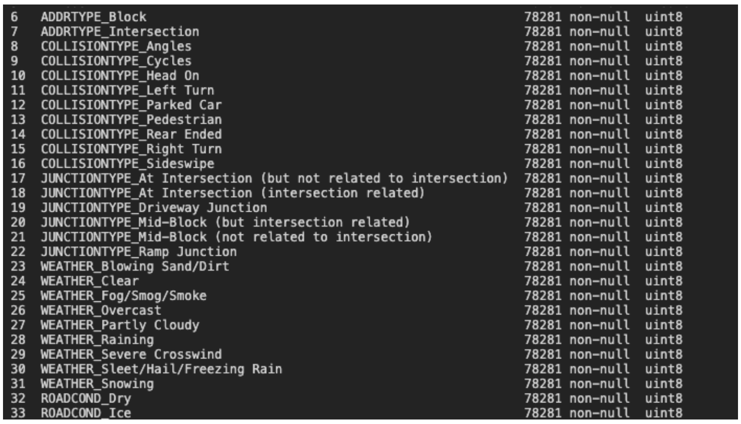


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 97852 entries, 139054 to 72625
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SEVERITYCODE   97852 non-null  int64  
 1   Longitude      97852 non-null  float64
 2   Latitude       97852 non-null  float64
 3   ADDRTYPE       97852 non-null  object 
 4   COLLISIONTYPE  97852 non-null  object 
 5   PERSONCOUNT    97852 non-null  int64  
 6   PEDCOUNT       97852 non-null  int64  
 7   PEDCYLCOUNT    97852 non-null  int64  
 8   VEHCOUNT       97852 non-null  int64  
 9   JUNCTIONTYPE   97852 non-null  object 
 10  UNDERINFL      97852 non-null  object 
 11  WEATHER        97852 non-null  object 
 12  ROADCOND       97852 non-null  object 
 13  LIGHTCOND      97852 non-null  object 
 14  HITPARKEDCAR   97852 non-null  object 
dtypes: float64(2), int64(5), object(8)
memory usage: 11.9+ MB


2    48926
1    48926
Name: SEVERITYCODE, dtype: int64
```

iv. Creating the dummies variables from the categorial attributes

The cleaned dataset from upper procedure have several categorial columns inside, which could not be used directly in the machine learning process. In this step, I apply the *get.dummies()* function in the *pandas* library to create the dummy variables for the further model building procedure. The dummy variables cover different circumstance under each factor, which could offer a numerical description for the remained factors (including the junction type, collision type, weather and so on). The information for the final cleaned dataset is as follows (Figure 10).

Figure 10 The cleaned dataset with dummy variables inside



```
6   ADDRTYPE_Block                                           78281 non-null  uint8
7   ADDRTYPE_Intersection                                    78281 non-null  uint8
8   COLLISIONTYPE_Angles                                     78281 non-null  uint8
9   COLLISIONTYPE_Cycles                                     78281 non-null  uint8
10  COLLISIONTYPE_Head On                                    78281 non-null  uint8
11  COLLISIONTYPE_Left Turn                                  78281 non-null  uint8
12  COLLISIONTYPE_Parked Car                                 78281 non-null  uint8
13  COLLISIONTYPE_Pedestrian                                 78281 non-null  uint8
14  COLLISIONTYPE_Rear Ended                                 78281 non-null  uint8
15  COLLISIONTYPE_Right Turn                                 78281 non-null  uint8
16  COLLISIONTYPE_Sideswipe                                  78281 non-null  uint8
17  JUNCTIONTYPE_At Intersection (but not related to intersection)  78281 non-null  uint8
18  JUNCTIONTYPE_At Intersection (intersection related)      78281 non-null  uint8
19  JUNCTIONTYPE_Driveway Junction                           78281 non-null  uint8
20  JUNCTIONTYPE_Mid-Block (but intersection related)        78281 non-null  uint8
21  JUNCTIONTYPE_Mid-Block (not related to intersection)     78281 non-null  uint8
22  JUNCTIONTYPE_Ramp Junction                               78281 non-null  uint8
23  WEATHER_Blowing Sand/Dirt                                78281 non-null  uint8
24  WEATHER_Clear                                            78281 non-null  uint8
25  WEATHER_Fog/Smog/Smoke                                   78281 non-null  uint8
26  WEATHER_Overcast                                         78281 non-null  uint8
27  WEATHER_Partly Cloudy                                    78281 non-null  uint8
28  WEATHER_Raining                                          78281 non-null  uint8
29  WEATHER_Severe Crosswind                                 78281 non-null  uint8
30  WEATHER_Sleet/Hail/Freezing Rain                         78281 non-null  uint8
31  WEATHER_Snowing                                          78281 non-null  uint8
32  ROADCOND_Dry                                             78281 non-null  uint8
33  ROADCOND_Ice                                             78281 non-null  uint8
```

Then I separated the independent variables into dataset A as well as dependent variable (SEVERITYCODE) into dataset B. These two datasets are ready for the machine learning process.

v. Creating the training and testing subsets for the algorithms and data normalization

With the A and B datasets waiting to be analyzed, I use 8:2 scale to split two datasets separately into training and testing datasets. Both of the training datasets and testing datasets will be applied into the models in the following sections. The data normalization process is to make the performance of the following algorithms better.

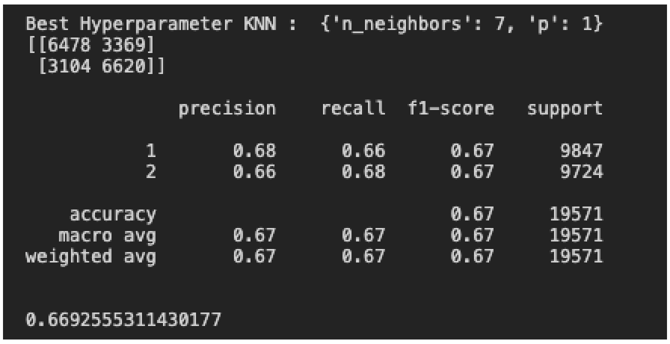
2. Applying machine learning algorithms

The following sub-section presents the results of training and testing 4 algorithm models. With the result compared, we could have a clear view on the accuracies. The explanation about each algorithm will also be offered.

a. K Nearest Neighbors Classifier

This algorithm a non-parametric method proposed by Thomas Cover used for classification and regression. It uses the variables to classify the new cases through the majority of its "neighbors". It could rely on labeled input data to learn a function that produces an appropriate output when given new unlabeled data. In this case, the output is as follows.

Table 1 The KNN output

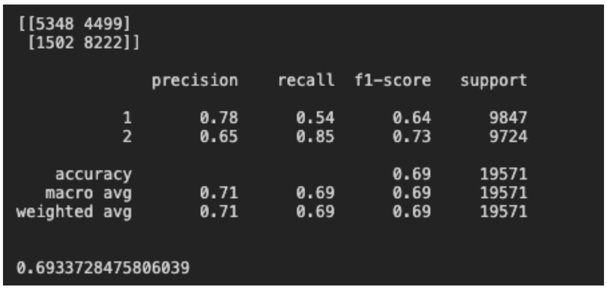


```
Best Hyperparameter KNN :  {'n_neighbors': 7, 'p': 1}
[[6478 3369]
 [3104 6620]]

              precision    recall  f1-score   support

           1       0.68      0.66      0.67      9847
           2       0.66      0.68      0.67      9724

    accuracy                           0.67     19571
   macro avg       0.67      0.67      0.67     19571
weighted avg       0.67      0.67      0.67     19571


0.6692555311430177
```

b. Logistic Regression

The logistic regression model is used to model the probability of a certain class. In this case, we can extend the logistic regression to model the impacts of independent variables (the different factors such as weather, road condition) on the dependent variable (collision severity). It works as a classifier that could estimate discrete values. Moreover, through fitting the existing data into the logistic regression function, this model could predict the probability of occurrence of collision severity. The output of the Logistic Regression Model is offered below.

Table 2 The LC output

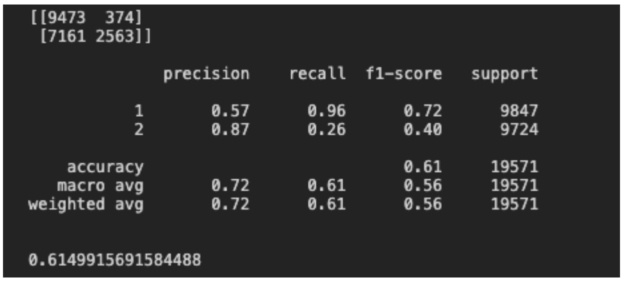


```
[[5348 4499]
 [1502 8222]]
              precision    recall  f1-score   support

           1       0.78      0.54      0.64      9847
           2       0.65      0.85      0.73      9724

    accuracy                           0.69     19571
   macro avg       0.71      0.69      0.69     19571
weighted avg       0.71      0.69      0.69     19571

0.6933728475806039
```

c. Naive Bayes classifiers

This model belongs to a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.

Table 3 The NC output

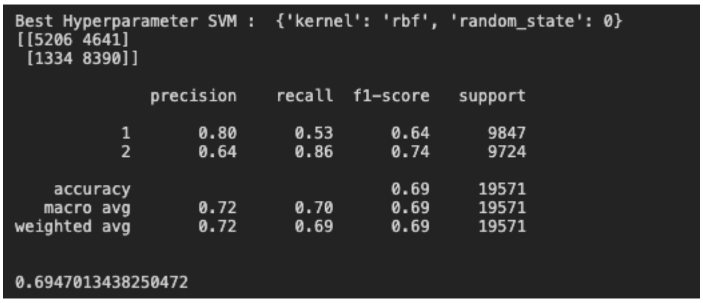


```
[[9473  374]
 [7161 2563]]
              precision    recall  f1-score   support

           1       0.57      0.96      0.72      9847
           2       0.87      0.26      0.40      9724

    accuracy                           0.61     19571
   macro avg       0.72      0.61      0.56     19571
weighted avg       0.72      0.61      0.56     19571

0.6149915691584488
```

d. Support-Vector Machines Classifier

SVMs, also support-vector networks are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). The output is as follows.

Table 4 The SVM output



```
Best Hyperparameter SVM :  {'kernel': 'rbf', 'random_state': 0}
[[5206 4641]
 [1334 8390]]
              precision    recall  f1-score   support

           1       0.80      0.53      0.64      9847
           2       0.64      0.86      0.74      9724

    accuracy                           0.69     19571
   macro avg       0.72      0.70      0.69     19571
weighted avg       0.72      0.69      0.69     19571


0.6947013438250472
```

3. The comparison for accuracy of each algorithms

   Table 7 The comparison of the accuracy

Algorithms	KNN	LC	NC	SVM
Accuracy (%)	66.93	69.34	61.50	69.47

From Table 7, we could find that the highest accuracy among the models is still below 70%. The accuracy values fluctuated from 60% to 70%. That means the models could offer the prediction of the collision severity with 60%~70% accuracy. It also shows that SVM (Support-Vector Machines Classifier) is the best model that could offer a better prediction for the collision severity

Conclusion

In this report, I run 4 separate machine learning algorithms on the Seattle Collision dataset to predict the collision severity. With the output and accuracy, I also compared all these models. It is quite obvious that SVM model has the best accuracy and could present the better prediction. Meanwhile, based on the research procedure, I also find that the collision type, address type and external condition (weather, road and light) affect the collision severity. They have various correlation with the collision severity.

Further Suggestions

Through the research in this report, I could also find several issues with the dataset offered by Seattle traffic department. Here I provide some suggestions for the further research.

1. The traffic department should find some solutions to reduce the missing values of the collision datasets. This could increase the accuracy of machine learning algorithms;

2. The dataset description document could offer more explanations about the data attributes;