Capstone Project - Car Accident Severity Prediction

Applied Data Science Capstone by IBM/Coursera

Table of Contents

Introduction: Business Problem

The object of this project is predicting the severity of an accident by time, driving condition and location. The target of this project is car drivers, the project will inform them about under which conditions they need to be more caution. If the drivers are more caution under dangerous conditions, there will be less accidents occur.

In addition, this project can also help Seattle Department of Transportation providing signs at locations where accidents occur frequently.

Data

Based on definition of our problem, the factors that will influence our analysis are:

- Type of location of the accident: Represented by ADDRTYPE, e.g. block, intersection.
- Severity: The target of this project, represented by SEVERITYCODE, 1 means property damage only, 2 means injury collision, 2b means serious injury, and 3 means fatality
- Date: The date and time of the accident, represented by INCDTTM
- Condition: The weather condition WEATHER, e.g. clear, raining, the road condition ROADCOND, e.g. wet, dry and the light condition LIGHTCOND during the collision e.g. daylight, dark.

The dataset used in this project is provided by Seattle Department of Transportation, the link is: https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv

Methodology

Data Cleaning

Not all the factors are used for the analysis, we only keep the factors mentioned in Data section, which are SEVERITYCODE, ADDRTYPE, INCDTTM, WEATHER, ROADCOND and LIGHTCOND, the head of our data looks like

	SEVERITYCODE	ADDRTYPE	INCDTTM	WEATHER	ROADCOND	LIGHTCOND
0	2	Intersection	3/27/2013 2:54:00 PM	Overcast	Wet	Daylight
1	1	Block	12/20/2006 6:55:00 PM	Raining	Wet	Dark - Street Lights On
2	1	Block	11/18/2004 10:20:00 AM	Overcast	Dry	Daylight
3	1	Block	3/29/2013 9:26:00 AM	Clear	Dry	Daylight
4	2	Intersection	1/28/2004 8:04:00 AM	Raining	Wet	Daylight

Missing Values

The first step of data cleaning is dealing with the missing values, count the number of missing values under the factors we are interested in, we have:

```
SEVERITYCODE       0
ADDRTYPE        1926
INCDTTM            0
WEATHER         5081
ROADCOND        5012
LIGHTCOND       5170
dtype: int64
```

We can find that there are only few missing values (less than 5%) in each factor, therefore we can drop the missing values in all features. Then, we removed the values such as 'unknown' and 'other' under WEATHER, LIGHTCOND and ROADCON, which are not helpful for our prediction.

Convert Data Type

Check the types of the data in interested features, we have:

```
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   SEVERITYCODE  169239 non-null  int64
 1   ADDRTYPE      169239 non-null  object
 2   INCDTTM       169239 non-null  object
 3   WEATHER       169239 non-null  object
 4   ROADCOND      169239 non-null  object
 5   LIGHTCOND     169239 non-null  object
dtypes: int64(1), object(5)
```

We can find that the data type of INCDTTM is object, we converted it into datetime64 using pandas.to_datetime() function, the updated types of data are:

```
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   SEVERITYCODE  169239 non-null  int64
 1   ADDRTYPE      169239 non-null  object
 2   INCDTTM       169239 non-null  datetime64[ns]
 3   WEATHER       169239 non-null  object
 4   ROADCOND      169239 non-null  object
 5   LIGHTCOND     169239 non-null  object
dtypes: datetime64[ns](1), int64(1), object(4)
```

Then, we extracted new factors hour of a day: HOUR, day of a week: DAY and month of a year: MONTH from INCDTTM, then we dropped the original INCDTTM column using dt.hour, dt.day_name() and dt.month functions. The head of the updated data looks like:

	SEVERITYCODE	ADDRTYPE	WEATHER	ROADCOND	LIGHTCOND	HOUR	DAY	MONTH
0	2	Intersection	Overcast	Wet	Daylight	14	Wednesday	March
1	1	Block	Raining	Wet	Dark - Street Lights On	18	Wednesday	December
2	1	Block	Overcast	Dry	Daylight	10	Thursday	November
3	1	Block	Clear	Dry	Daylight	9	Friday	March
4	2	Intersection	Raining	Wet	Daylight	8	Wednesday	January

Balancing the Dataset

We checked the number of data under each category of each factor, we found that under WEATHER, the categories ‘Blowing Sand/Dirt’, ‘Severe Crosswind’ and ‘Partly Cloudy’ and under ROADCOND, the categories ‘Sand/Mud/Dirt’ and ‘Oil’ have only a few data, which may affect our prediction. Therefore, we removed them for better prediction.

```
Clear                       108374
Raining                      32510
Overcast                     26809
Snowing                        821
Fog/Smog/Smoke                 547
Sleet/Hail/Freezing Rain       106
Blowing Sand/Dirt               42
Severe Crosswind                25
Partly Cloudy                    5
Name: WEATHER, dtype: int64
```

```
Dry               120978
Wet                46135
Ice                 1072
Snow/Slush           829
Standing Water       101
Sand/Mud/Dirt         64
Oil                   60
Name: ROADCOND, dtype: int64
```

Then, we checked whether our data set is balanced by checking the number of data under each category of target SEVERITYCODE, we have:

```
1    113571
2     55472
Name: SEVERITYCODE, dtype: int64
```

We can see that our target SEVERITYCODE is not balanced, the data under class 1 is two times the size of class 2. We need to balance the data by downsampling class 1 to the same size as class 2 using resample from sklearn. Then we have

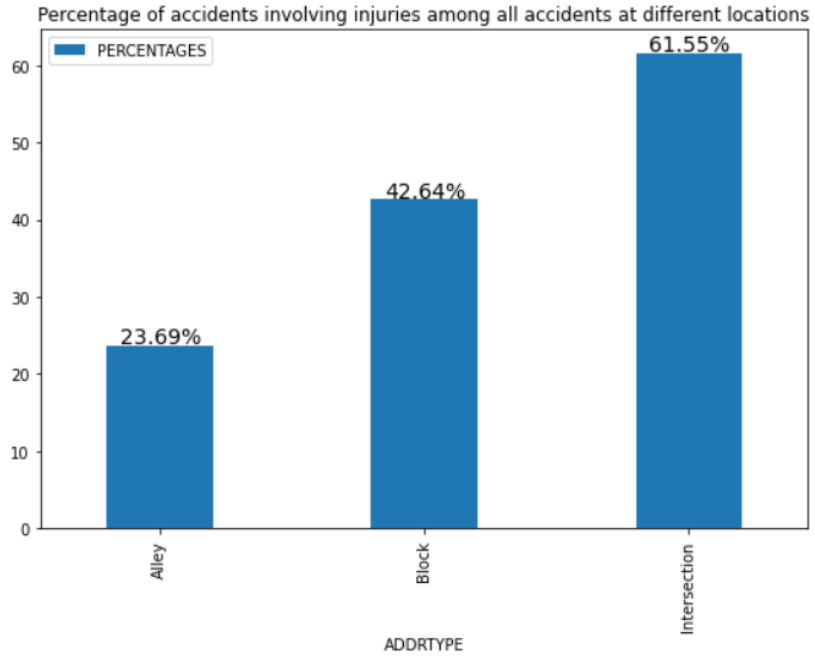
```
2    55472
1    55472
Name: SEVERITYCODE, dtype: int64
```

Explanatory Data Analysis

In this section, we are analyzing under which condition the accidents are more severe by comparing the percentage of accident with SEVERITYCODE 2 among all accidents under each condition. Firstly, we compare different address types:

ADDRTYPE	PERCENTAGES
Alley	23.69
Block	42.64
Intersection	61.55

Visualize the percentages, we have

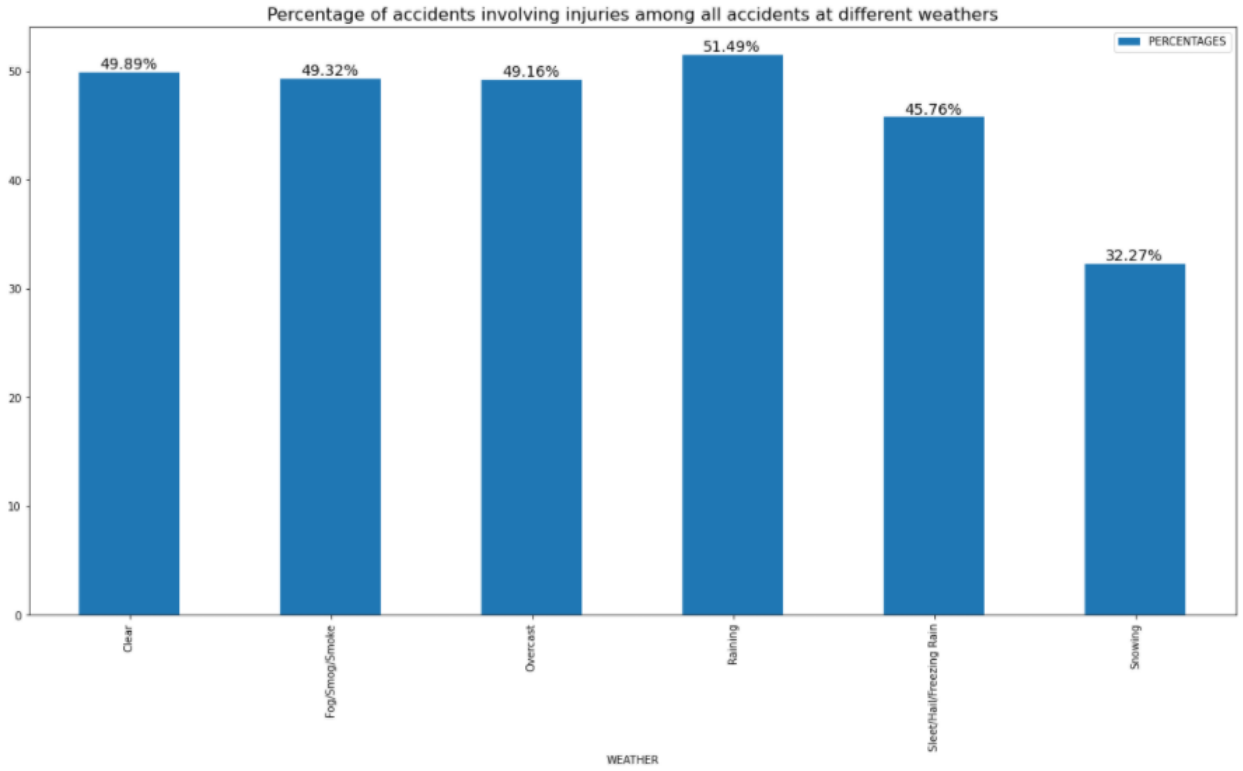


We can see that type of an address is correlated to the severity of an accident. At intersections, the accidents tend to be most severe while at alleys, the accidents tend to be less severe.

Then, we compare different kinds of weather:

WEATHER	PERCENTAGES
Clear	49.89
Fog/Smog/Smoke	49.32
Overcast	49.16
Raining	51.49
Sleet/Hail/Freezing Rain	45.76
Snowing	32.27

Visualizing the data, we have:

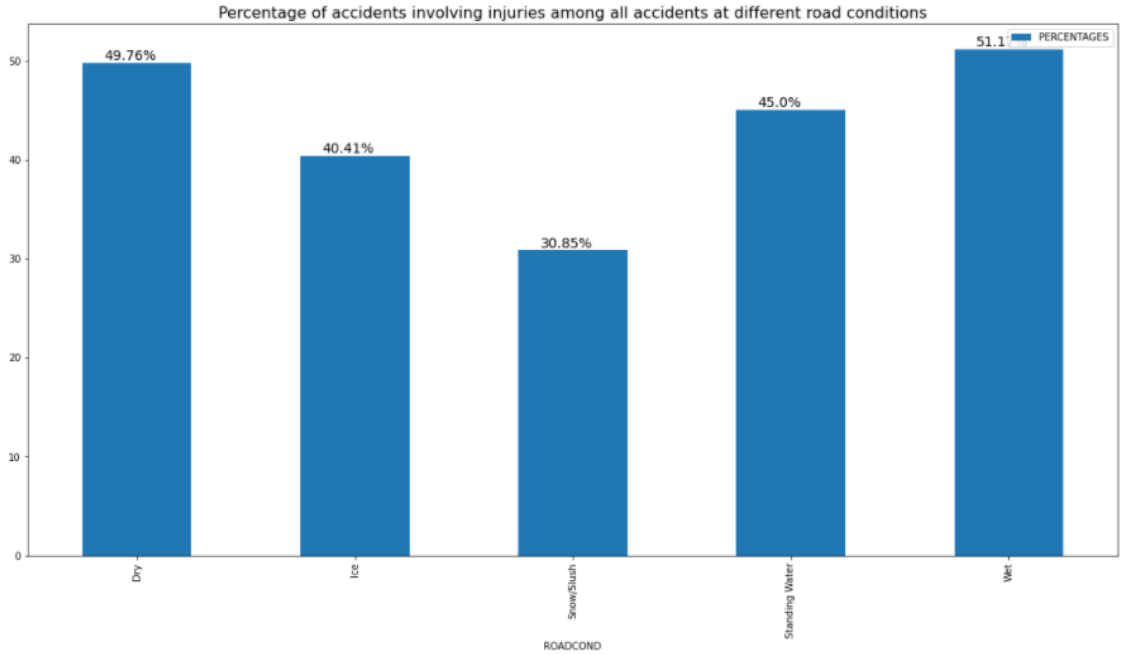


From the data above, we can infer that the weather condition is also correlated to severity of the accident, the accidents tend to be less severe when there is freezing rain or snowing. A possible reason is that people tend to drive slower under these weather conditions.

Then, we compare different kinds of road conditions:

ROADCOND	PERCENTAGES
Dry	49.76
Ice	40.41
Snow/Slush	30.85
Standing Water	45.00
Wet	51.17

Visualizing it, we have:

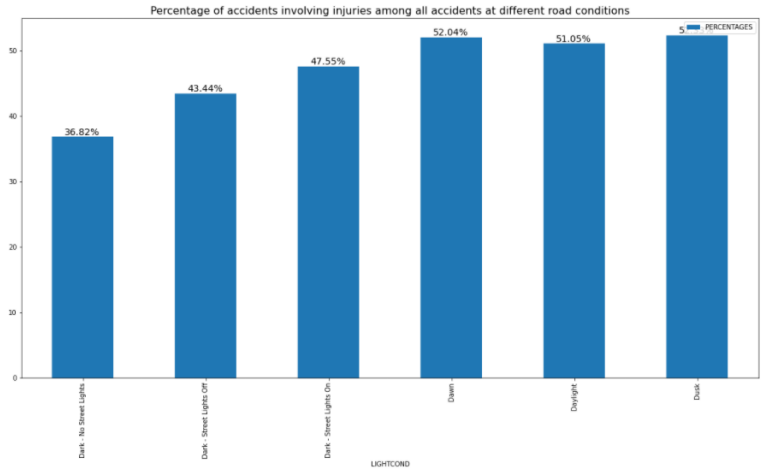


We can infer that the accidents are less severe when there is snow or ice on the road.

Then, we compared different light conditions:

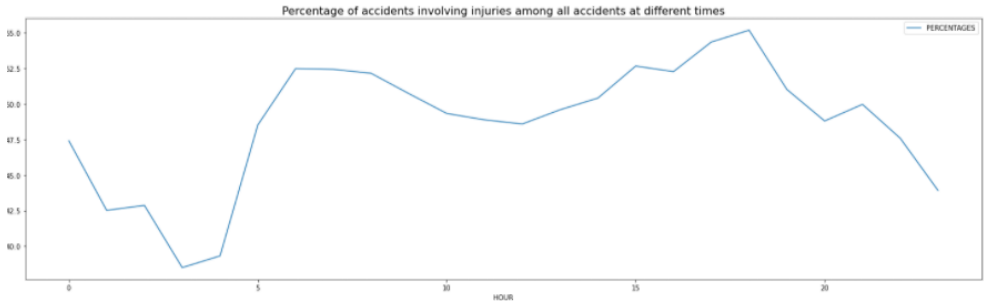
LIGHTCOND	PERCENTAGES
Dark - No Street Lights	36.82
Dark - Street Lights Off	43.44
Dark - Street Lights On	47.55
Dawn	52.04
Daylight	51.05
Dusk	52.33

Visualizing the data,



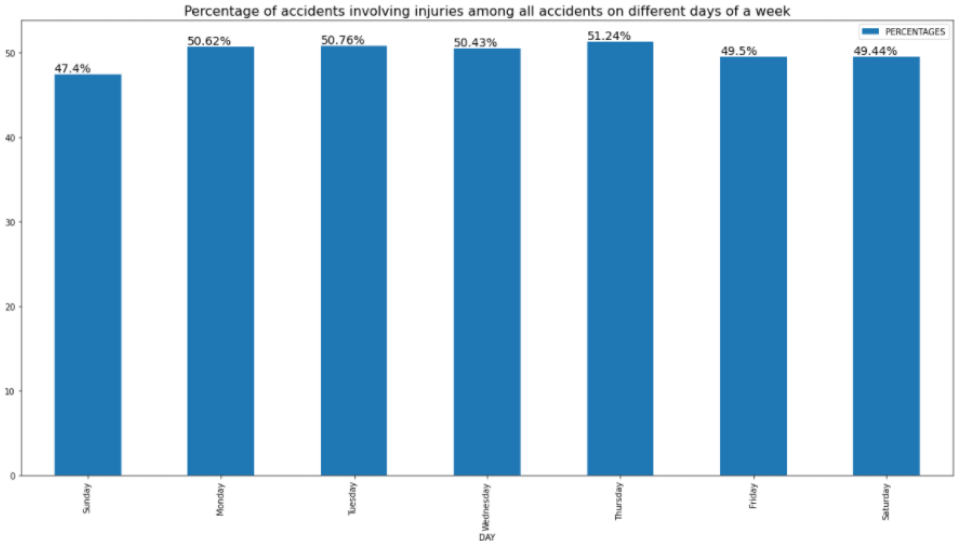
We can infer that the accidents tend to be less severe when the light is more insufficient.

Then, we compare different times of a day:



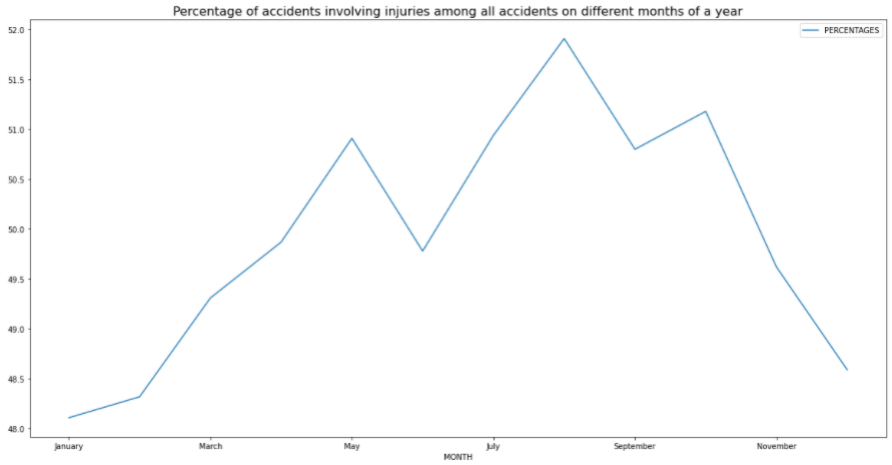
We can infer that the accident tends to be more severe at rush hours, probably because there more cars on the road at rush hours.

Then, we compare different days of a week:



We can infer that the accidents are slightly less severe on weekends.

Finally, we compare different months of a year:



We can infer that the accidents are slightly less severe in the winters

Feature Selection and Preprocessing

Before we feed our data into machine learning models, we need to select which features are we need and convert the values to the format which can be fed into machine learning models.

In explanatory data analysing step, we analyzed how different features related to the severity of an accident. Some of the features such as weather condition and address type are strongly correlated to the severity of accidents, while other features such as day of a week or month of a year have only a little correlation with the severity of accidents. In addition, feature weather condition, feature month of a year and feature road condition are overlapping with each other. For example, snowing only happen in winter, and the road condition will be snow when the weather condition is snowing. Feature light condition also overlapping with feature time of a day, when the light condition is night, the time of a day will definitely after 21.

Therefore, we only select address type ADDRTYPE, weather condition WEATHER and light condition LIGHTCOND as features.

All three features we selected need to be converted to the format that can be fed into machine learning models, we use one-hot encoding to label them. Firstly, split the data into X (features) and Y (target), then we encode the features in X using one-hot encoding, the head of training data looks like:

	ADDRTYPE_Block	ADDRTYPE_Intersection	WEATHER_Fog/Smog/Smoke	WEATHER_Overcast	WEATHER_Raining	WEATHER_Sleet/Hail/Freezing Rain	WEATHER_Snowing	LIGHTCOND_Dark - Street Lights Off	LIGHTCOND_Dark - Street Lights On	LIGHTCOND_Dawn	LIGHTCOND_Daylight	LIGHTCOND_Dusk
178895	0	1	0	0	1	0	0	0	0	0	1	0
176892	0	1	0	0	0	0	0	0	1	0	0	0
72113	1	0	0	0	0	0	0	0	1	0	0	0
42683	1	0	0	0	0	0	0	0	0	0	1	0
194212	0	1	0	0	0	0	0	0	0	0	1	0

For train/test split, we used 70% of the data for training and 30% of the data for testing, we have 77660 data points in training set and 33284 data points in testing set.

Machine Learning Models

Because the task of this project is to classification and our target only have two values, we use decision tree and logistic regression as our machine learning models.

Decision Tree

For decision tree, we used the tree.DecisionTreeClassifier() function from sklearn to train the model, the random state was set to 0.

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier

collision_tree = tree.DecisionTreeClassifier(random_state=0)
clf = collision_tree.fit(X_train,Y_train)
Y_DT_pred = collision_tree.predict(X_test)
```

Logistic Regression

For logistic regression, we used the LogisticRegression() function from sklearn.linear_model to train the model, the C value is set to 1 and the solver is set to ‘liblinear’

```
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression(C = 1, solver='liblinear')
LR.fit(X_train, Y_train)
Y_LR_pred = LR.predict(X_test)
Y_LR_prob = LR.predict_proba(X_test)
```

Results and Evaluation

We used F-1 score to test the accuracy of our decision tree model and our logistic regression model, F-1 score is the harmonic mean of recall and precision, the equations for calculating F-1 score is

- Recall = True Positive / (True Positive + False Negative)
- Precision = True Positive / (True Positive + False Positive)
- F-1 score = 2 * (Precision * Recall) / (Precision + Recall)

The best value of F-1 score is 1 and the worst value is 0.

The F-1 score of our decision tree model is:

```
f1_score(Y_test, Y_DT_pred, average='macro')
```

```
0.5860910942444686
```

The F-1 score of our logistic regression model is:

```
f1_score(Y_test, Y_LR_pred, average='macro')
```

```
0.5865646443755439
```

Also, we used logistic loss to evaluate the accuracy of our logistic regression model, which represent the error for the predictions in classification problems. The value of logistic loss is between 0 and 1.

The logistic loss of our logistic regression model is:

```
log_loss(Y_test, Y_LR_prob)
```

```
0.6745473339536407
```

Discussion and Conclusion

At the beginning of this project, we first cleaned the data. We dropped the missing values, dropped the values with insufficient data, changed the data type of features to our desired data type, then made the data balanced by downsampling the majority class to match the minority class.

After we cleaned the data, we visualized the data. We compared the probabilities of injury in accidents under different address types, road conditions, weather conditions, light conditions, time of days, day of weeks and month of years. Then we decided to use address type, weather condition and light condition as features for our machine learning model.

After we cleaned and analyzed the data, we encoded the features using one-hot encoding, then we feed the data to two machine learning model: decision tree and logistic regression.

We used F-1 score to test the accuracy of our decision tree model and logistic regression model and we also used logistic loss to test the accuracy of our logistic regression model.

However, the result is not as good as we expected, the accuracy of both models is not very high, possibly due to the lack of features. Therefore, we conclude that location, condition, and time provided by the dataset have some impact on the severity of an accident.