Predicting the Severity Level of Traffic Accidents

Chenhao Xi

September 10, 2020

1. Introduction

1.1 Background

An accident on the road is inevitable, it will of course bring trouble or damage to the people directly involved, or even worse. While it will also cause problem to other participants in the traffic (i.e. a big jam), but for those people it's still possible to avoid negative impact if they can be warned beforehand.

A prediction on the severity of an accident popped out could be such a warning, which can help other drivers to decide on whether to change their route or not. Since changing the route also costs extra resources, a reasonable choice is to re-route in face of a "big" accident, while keeping on current track if it's only a "small" one as it may be settled in a relatively short time.

1.2 Problem

Then the problem now is how to predict the severity of an accident. Things which may have an influence on that could be the collision type, light condition, road condition, etc.;

The target of this project is to find out features which have the most influence on the severity and train a model to predict the severity of future accidents.

1.3 Interest

The result of the prediction can be integrated into an on-board GPS application, on receiving necessary information of an accident, a prediction of the severity will be given. For a severe accident, the GPS app shall automatically re-route and push to the driver as preferred choice; for a less severe one, only to remind the driver and re-route on demand. And maybe this information could also be sent to a traffic control system, where the traffic heading into the accident location can be adjusted by the traffic lights.

2. Data acquisition and cleaning

2.1 Data sources

The data is originally from Seattle Department of Transportation (SDOT), which covers the collision happened in Seattle from 2004-01-01 to 2020-05-20, each case is labeled with a severity code, and contains a variety of attributes of the case. There is also an explanation document "Metadata" for the meaning of each column.

2.2 Data cleaning & Feature selection

By checking with missing data in the list, we can find there are several columns missing many values. But before removing any rows or columns, it's better to understand the meaning of each column. Since if one column is not suitable or necessary to be kept as a feature, then any missing value in that column doesn't matter at all.

So that we can start by reading the file "Metadata" to rule out unnecessary columns first, then deal with missing values in the remaining columns. It’s easy to find out that a couple of the columns are duplicated peers such as “SEVERITYCODE” and “SEVERITYDESC”, they are actually equal to each other; or some are identifications of a case, such as “INCKEY”, “COLDETKEY” and so on, those columns should just be excluded from the data set.

And there are three columns which are quite interesting, "INJURIES", "SERIOUSINJURIES" and "FATALITIES". After a quick analysis, it's easy to find that there is a strong logic between these columns and the label "SEVERITYCODE". Where the one-hot logics are like this:

severity code	**INJURIES**	**SERIOUSINJURIES**	**FATALITIES**
0	drop		
1	0	0	0
2	1	0	0
2b	x	1	0
3	x	x	1

But a couple of exceptions of code-2b are found having the combination of “x00”, while I also noticed that for these samples, it says “Unmatched” in column “STATUS”. So that, in conclusion, it should be reasonable to drop all these columns, since they play like dummies to the labels. And all rows marked as “Unmatched” shall also be dropped before column drops.

But before dropping those columns, they can still be useful in some logic test. (i.e., there should be a logic between (PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, INJURIES, SERIOUSINJURIES, FATALITIES) Where it's only reasonable when:

PERSONCOUNT >= PEDCOUNT+PEDCYLCOUNT

&

PERSONCOUNT >= INJURIES+SERIOUSINJURIES+FATALITIES.

Or otherwise the number in column "PERSONCOUNT" doesn't make sense, and that row should be dropped.

Also, there are several columns contain the value "Unknown" or 'Other' which is not quite useful, the rows will also be dropped.

Column to be dropped	**Reason to drop**
X	not explained in Metadata
Y	not explained in Metadata
INCKEY	case id, no meaning
COLDETKEY	case id, no meaning
REPORTNO	case id, no meaning
STATUS	drop "Unmatched" rows first, then drop the column
INTKEY	a combination of other columns
LOCATION	too detailed address info
EXCEPTRSNCODE	not explained in Metadata
EXCEPTRSNDESC	not explained in Metadata
SEVERITYDESC	duplicated column

INCDATE	duplicated column
SDOT_COLDESC	duplicated column
SDOTCOLNUM	case id, no meaning
ST_COLDESC	duplicated column
SEGLANEKEY	too many missing values, and meaning is unclear
CROSSWALKKEY	too many missing values, and meaning is unclear
INJURIES	dummy column
SERIOUSINJURIES	dummy column
FATALITIES	dummy column

For a better understanding of missing data of remaining columns, replace empty value with "N" in column "INATTENTIONIND", "PEDROWNOTGRNT" and "SPEEDING" as only "Y" is marked out in the columns. In column "UNDERINFL" there are both "Y/N" and "1/0", for standardization, replace "Y/N" into "1/0" respectively. Now we can drop entitle rows with missing value in any column.

The next step of data cleaning is to do some conversion of certain values for standardization: replace "Y/N" into "1/0" in column "INATTENTIONIND", "PEDROWNOTGRNT" and "SPEEDING".

2.3 Data balancing

There are 4 labels (categories) in the dataset (1,2,2b,3), but the number of samples within each category are not balanced as the statistics shows:

Code	**Total Samples (around)**
1	100000
2	50000
2b	2000
3	200

If all categories are to be used as individual label with balanced samples, then the result will be around 200 from each category and 800 in total. But if comparing to the size of the original data, that's quite a waste. Instead, we can consider combining code-2b and code-3, then we pick out 2000 samples from code-(2b+3) and code-2, then 4000 from code-1. So that it will be possible to carry out two rounds of binary classification, hopefully, we can separate these three categories (Low, Mid, High) with a reasonable accuracy.

Code	**Category**	**Sample Qty.**
1	Low	4000
2	Mid	2000
2b + 3	High	2000

3. Exploratory Data Analysis

All potential features will be checked, by counting the occurrence of each unique value in categories of "Low", "Mid" and "High". While the comparison is between "Low" vs "Mid_High", and "Mid" vs "High". (L1 and L2 respectively)

Binary classification level-1: Low vs Mid_High

Binary classification level-2: Mid vs High

3.1 relationship between location and severity

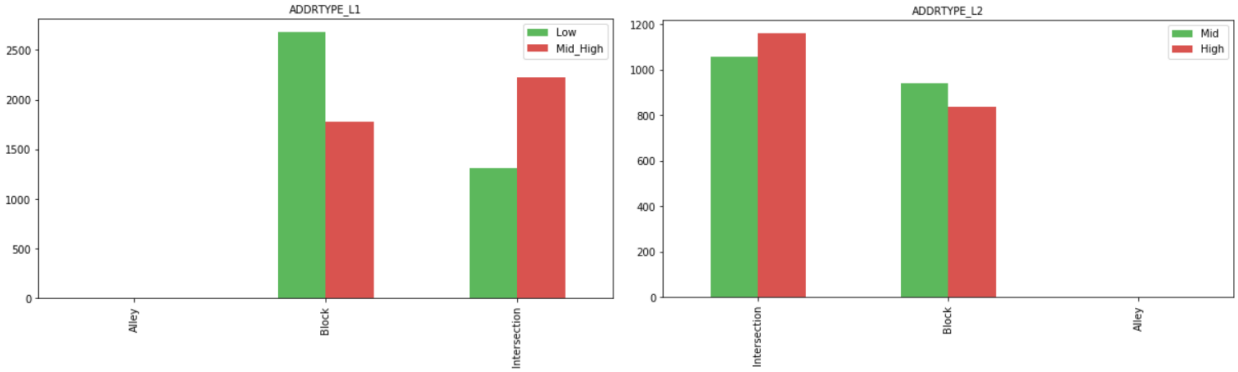


Fig 3.1a Severity dstribution on address type

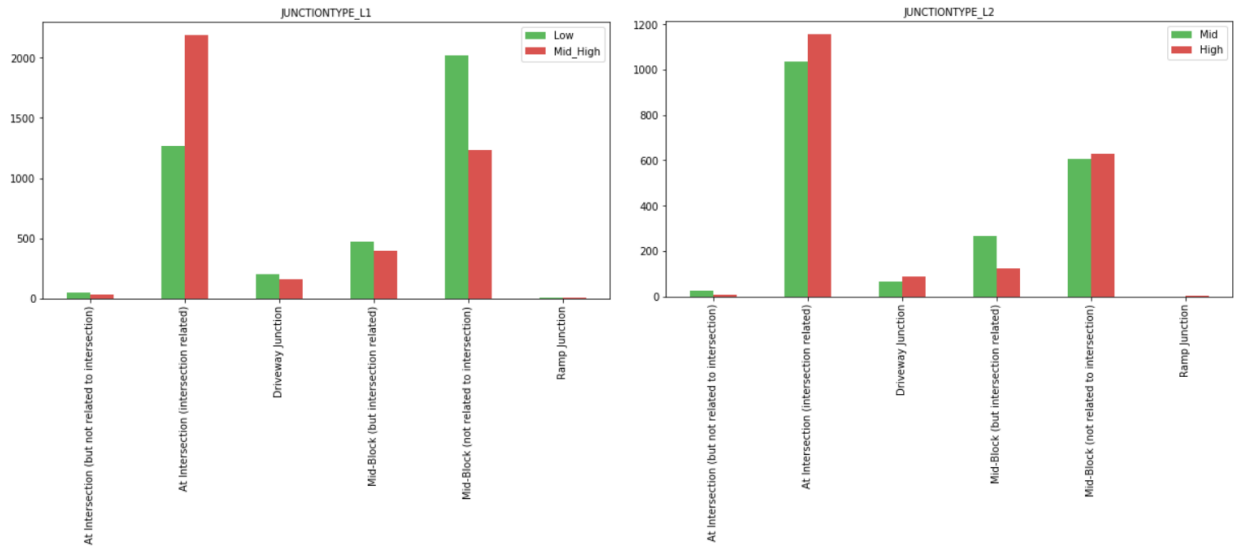


Fig 3.1b Severity dstribution on junction type

We can find that when the accident is happened at a "Block", it has a higher probability to have a lower severity. The reason could be that the traffic condition there is usually less complicated than at an intersection.

3.2 relationship between collision type and severity

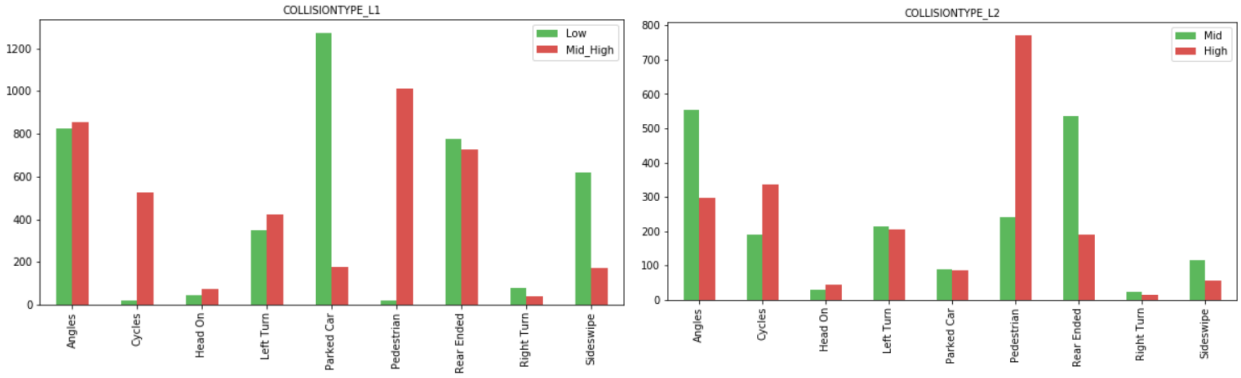


Fig 3.2a Severity dstribution on collision type

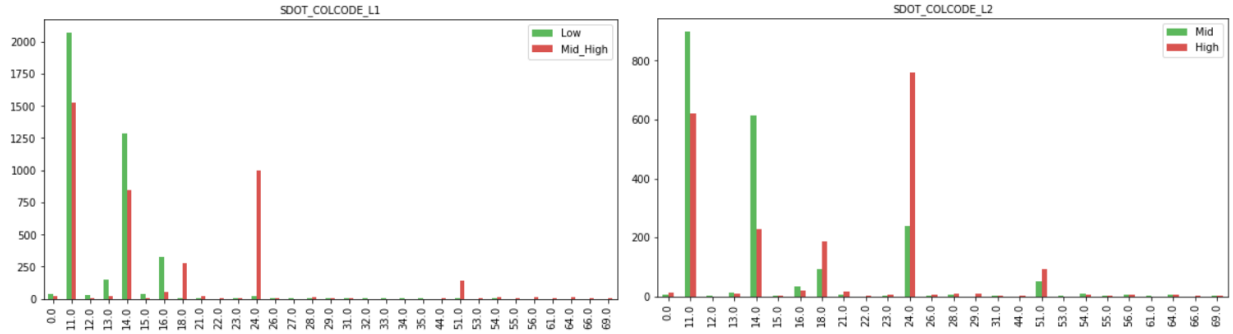


Fig 3.2b Severity dstribution on collision type (SDOT_COLCODE)

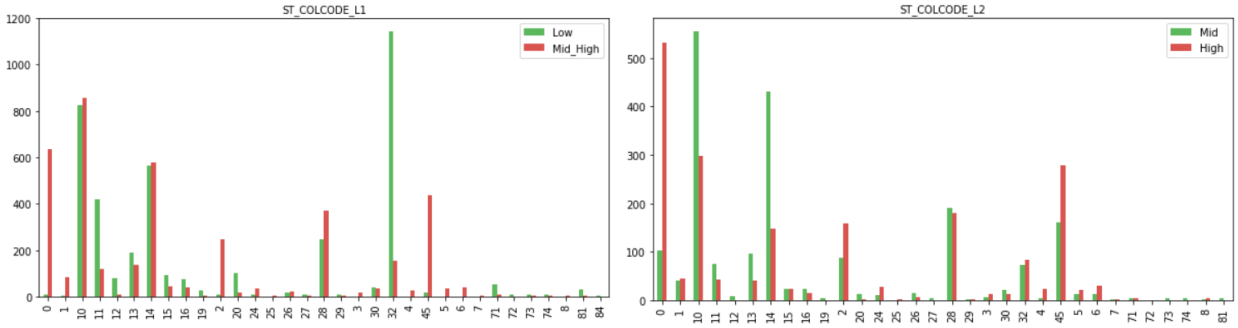


Fig 3.2c Severity dstribution on collision type (ST_COLCODE)

The charts show that if the accident is just a sideswipe, it's usually a lower severity. but when it's a head-on, then the opposite. And if it involves cycles or pedestrians, very likely it'll have a higher severity since someone is probably injured.

Fig 3.2b and Fig 3.2c can be considered as breakdowns of Fig 3.2a from different perspectives, their patterns are in consistent with Fig 3.2a, which can be verified by reading the file “Metadata”.

3.3 relationship between numerical features and severity

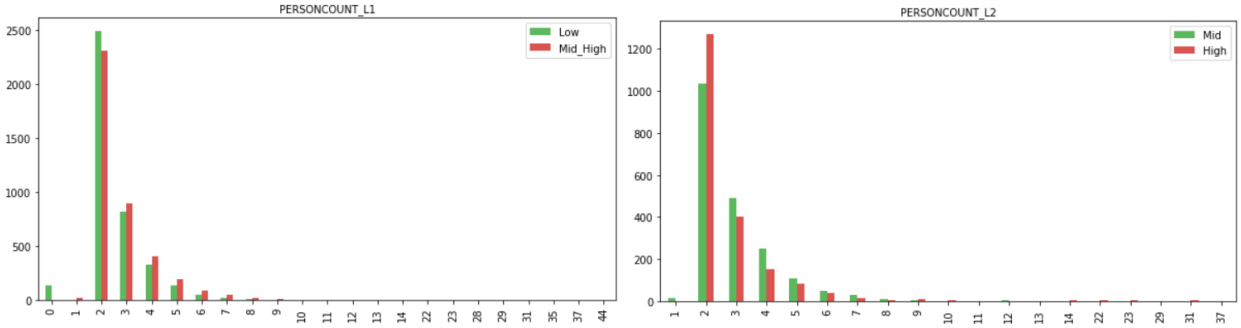


Fig 3.3a Severity dstribution on person count

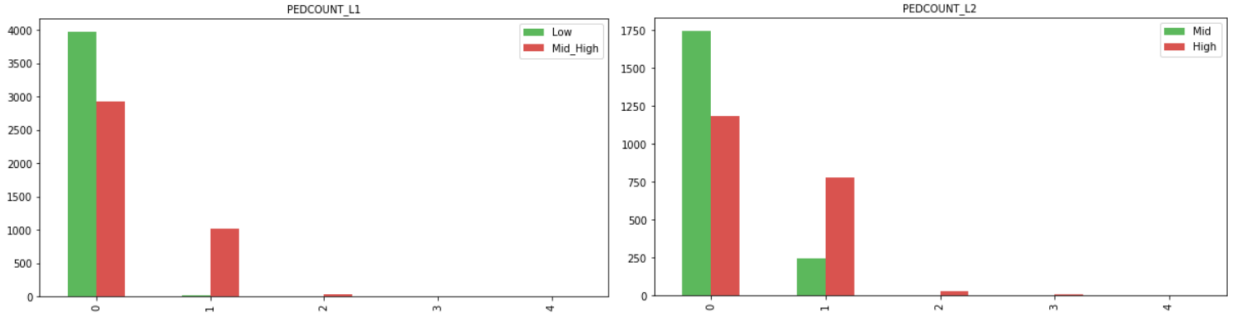


Fig 3.3b Severity dstribution on pedestrian count

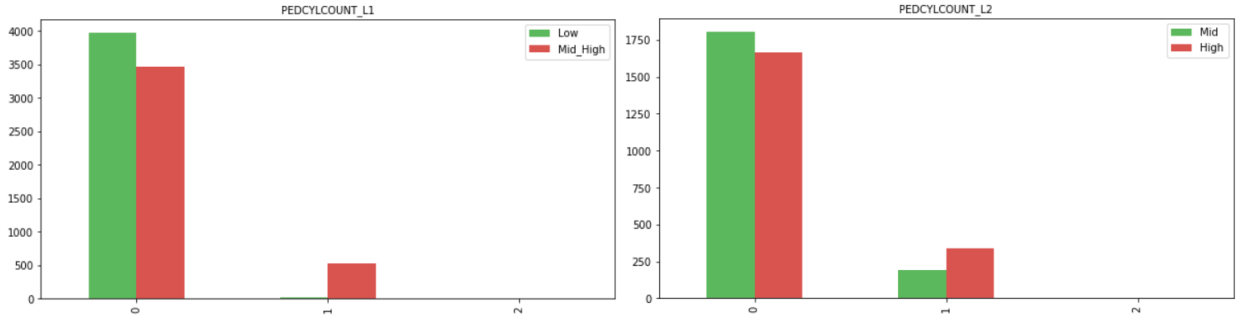


Fig 3.3c Severity dstribution on cyclist count

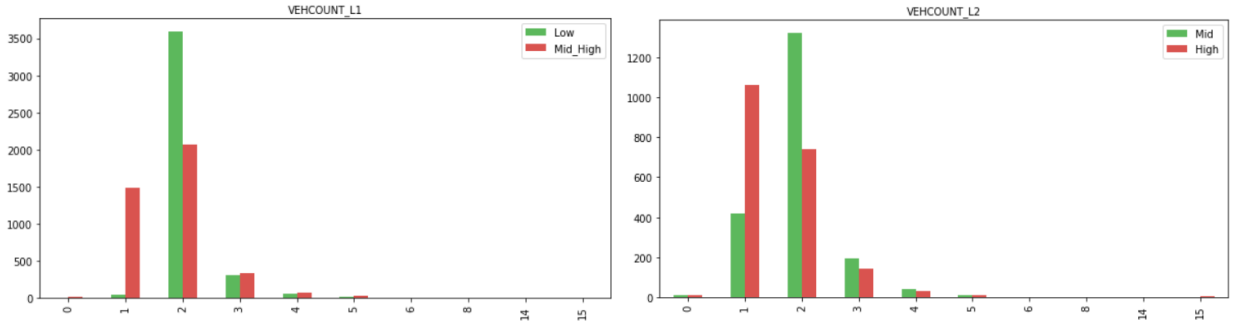


Fig 3.3d Severity dstribution on vehicle count

In Fig 3.3a, L1 comparison, as the number of involved people increases, it's more likely to have a higher severity. since more people means a higher possibility to have people injured. While in L2, when it's

certain there are injuries, 2-person cases outstand as an exception, the reason is that there are more cases of driver against non-driver (cyclist, pedestrian) in this category, let's verify below:

Pedestrian	**771**
Cycles	**336**
Angles	299
Left Turn	206
Rear Ended	189
Parked Car	86
Sideswipe	55
Head On	43
Right Turn	15

This is also reflected in Fig 3.3d that a single vehicle accident is more likely to have a high severity, because in those cases, the car probably hit cycles or pedestrians.

3.4 relationship between subjective mistakes and severity

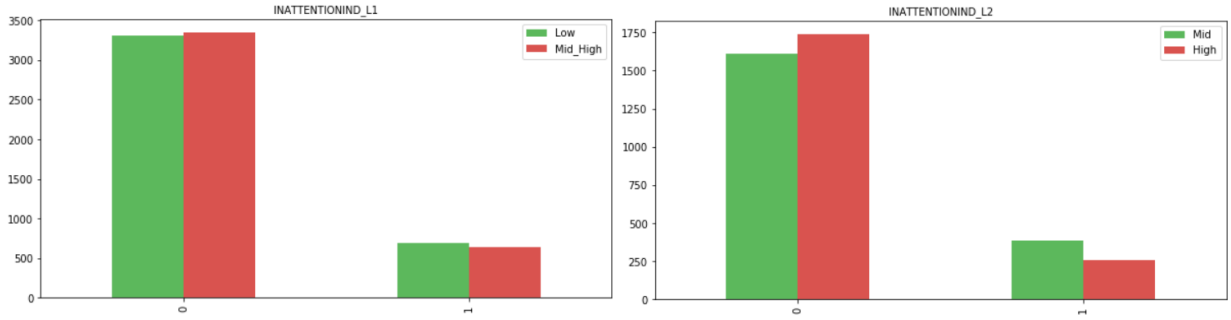


Fig 3.4a Severity dstribution on whether due to inattention

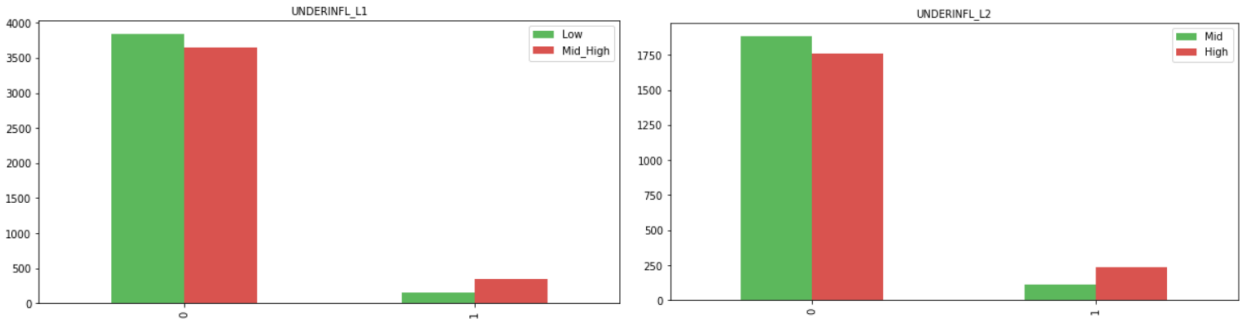


Fig 3.4b Severity dstribution on whether under influence of drug/alcohol

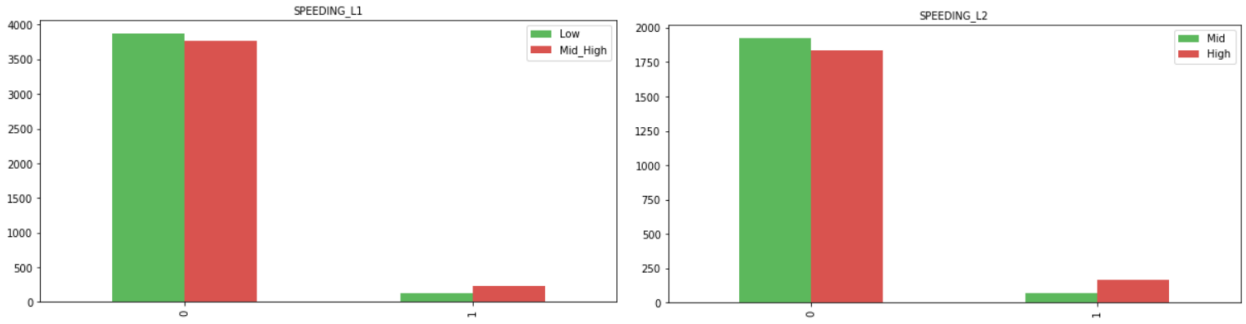


Fig 3.4c Severity dstribution on whether speeding

Generally, if an accident is due to the driver's subjective mistake (under influence of drug/alcohol, or speeding), more likely it will have a higher severity.

3.5 relationship between traffic conditions and severity

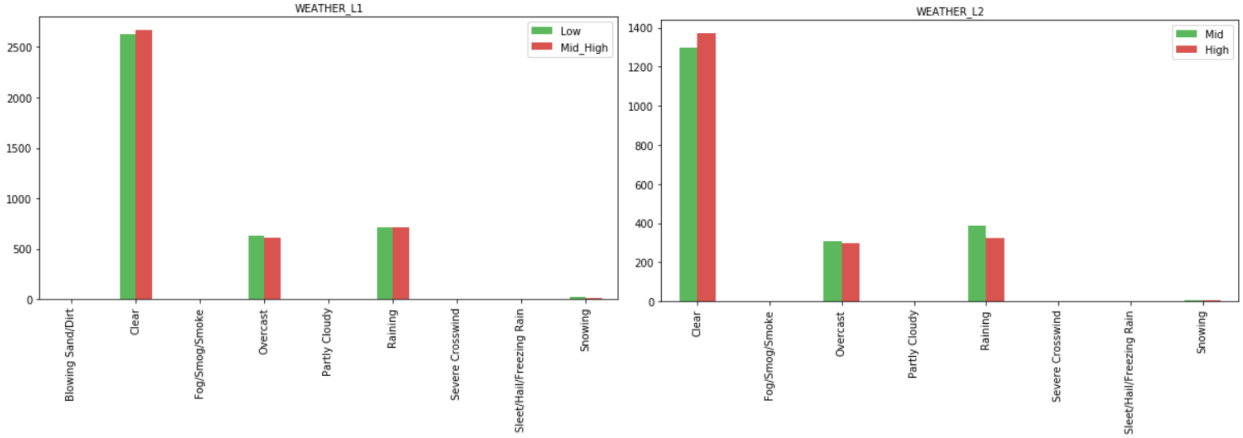


Fig 3.5a Severity dstribution on wheather condition

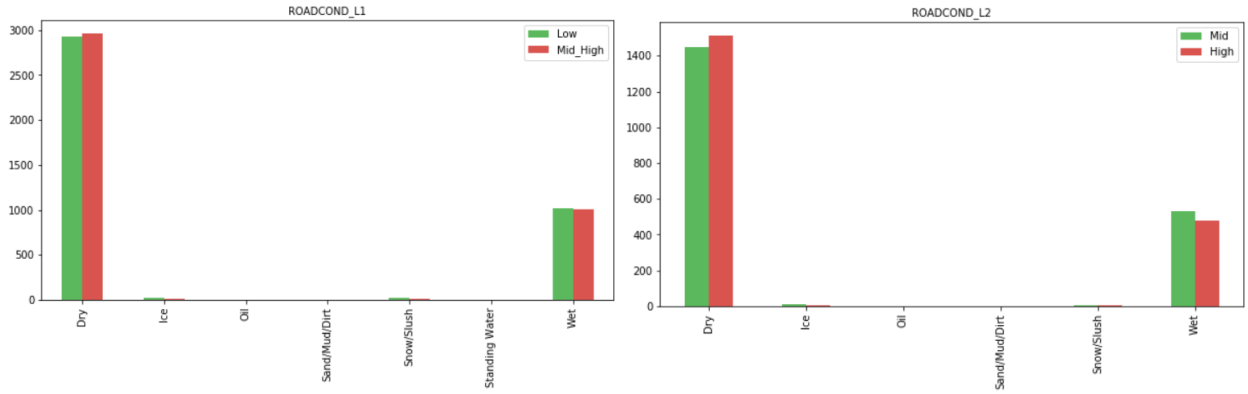


Fig 3.5b Severity dstribution on road condition

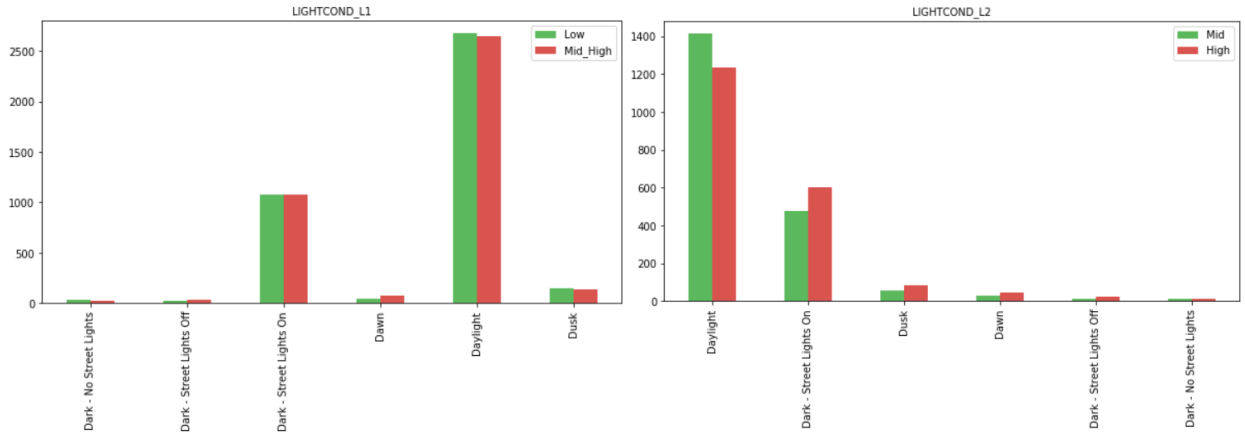


Fig 3.5c Severity dstribution on light condition

Well, the result of weather and road condition are a bit counter intuition, that a "Dry" condition means a higher severity. I can only guess that the vehicle speed tends to be higher when the weather is better, which result in a higher severity when there is an accident. But the light condition shows what's expected.

3.6 other features

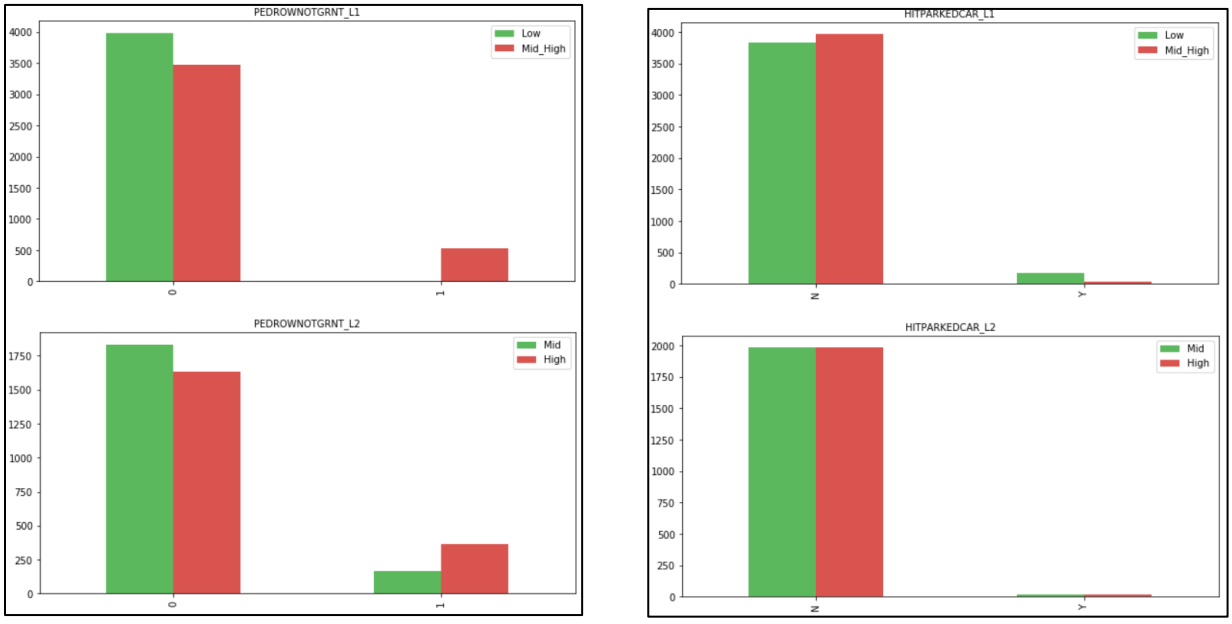


Fig 3.6 Severity dstribution on other features

Obviously, if the pedestrian is walking in the wrong place, the cases usually have a high severity. But the difference is not quite clear whether hitting a parked car.

4. Predictive Modeling

This is a typical classification problem, KNN, decision tree, SVM and logistic regression will be trained and evaluated. And for KNN and decision tree models, different values of parameter K and depth will be tried to find the best parameter.

4.1 train models with different selection of features

After data cleaning and other processes, there are 22 features left, among which, 9 are converted one-hot features. While there are three different types of features: numerical, categorical and one-hot.

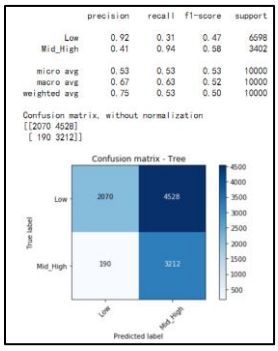
Numerical features: ['PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT']

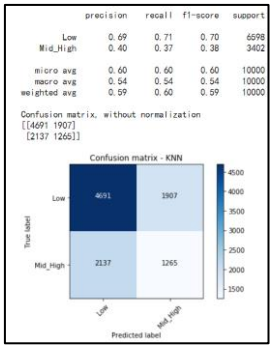
Categorical features: ['SDOT_COLCODE', 'ST_COLCODE']

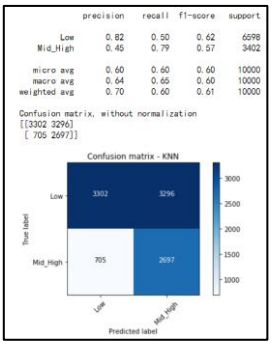
One-hot features: ['ADDRTYPE', 'INATTENTIONIND', 'UNDERINFL', 'LIGHTCOND', 'PEDROWNOTGRNT', 'SPEEDING', 'Trafficcon', COLLISIONTYPE' (in one-hot format)]

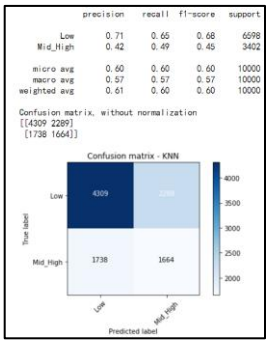
The training and evaluation is based on the randomly selected 8000 samples, and the validation is performed on the rest of the samples (10000 random samples).

Model performance on different feature selections (KNN):









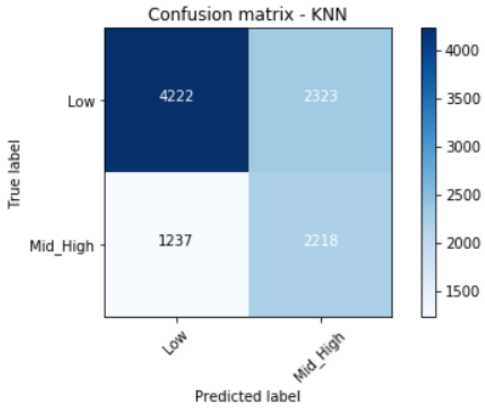
All features w/o numerical w/o categorical only one-hot

After several tries, it seems that when only using one-hot features, the models have a more balanced performance, in regard of F1 score in each category.

Model performance on level-1 classification:

	precision	recall	f1-score	support
Low	0.77	0.65	0.70	6545
Mid_High	0.49	0.64	0.55	3455
micro avg	0.64	0.64	0.64	10000
macro avg	0.63	0.64	0.63	10000
weighted avg	0.67	0.64	0.65	10000

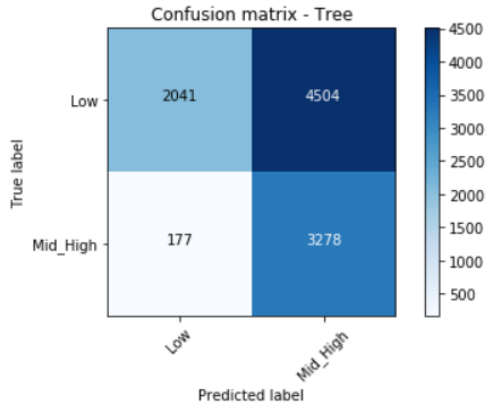
Confusion matrix, without normalization
[[4222 2323]
 [1237 2218]]



KNN

	precision	recall	f1-score	support
Low	0.92	0.31	0.47	6545
Mid_High	0.42	0.95	0.58	3455
micro avg	0.53	0.53	0.53	10000
macro avg	0.67	0.63	0.52	10000
weighted avg	0.75	0.53	0.51	10000

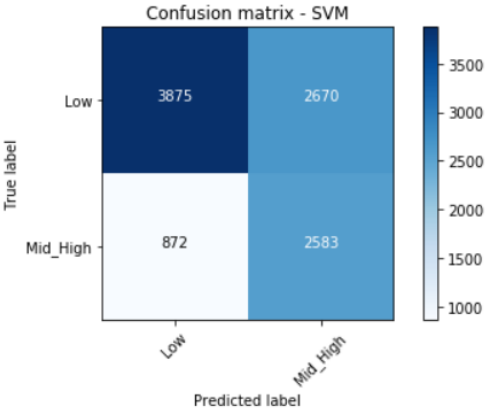
Confusion matrix, without normalization
[[2041 4504]
 [177 3278]]



decision tree

	precision	recall	f1-score	support
Low	0.82	0.59	0.69	6545
Mid_High	0.49	0.75	0.59	3455
micro avg	0.65	0.65	0.65	10000
macro avg	0.65	0.67	0.64	10000
weighted avg	0.70	0.65	0.65	10000

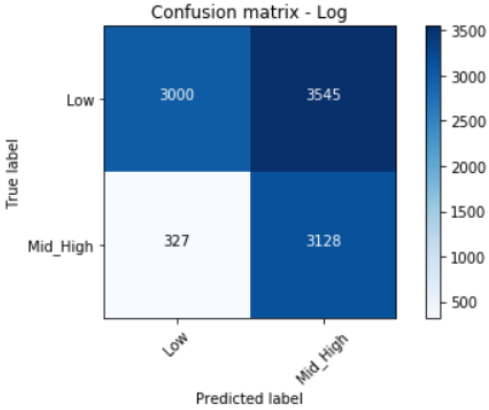
Confusion matrix, without normalization
[[3875 2670]
 [872 2583]]



SVM

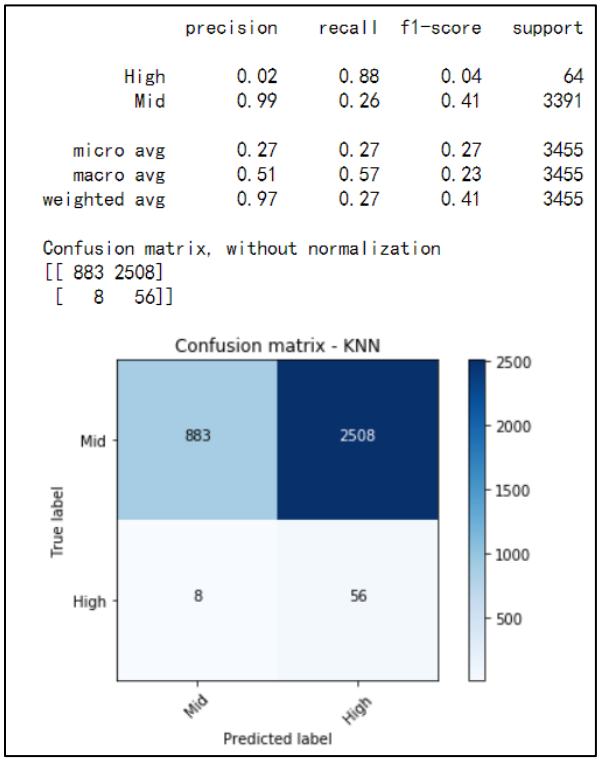
	precision	recall	f1-score	support
Low	0.90	0.46	0.61	6545
Mid_High	0.47	0.91	0.62	3455
micro avg	0.61	0.61	0.61	10000
macro avg	0.69	0.68	0.61	10000
weighted avg	0.75	0.61	0.61	10000

Confusion matrix, without normalization
[[3000 3545]
 [327 3128]]



logistic regression

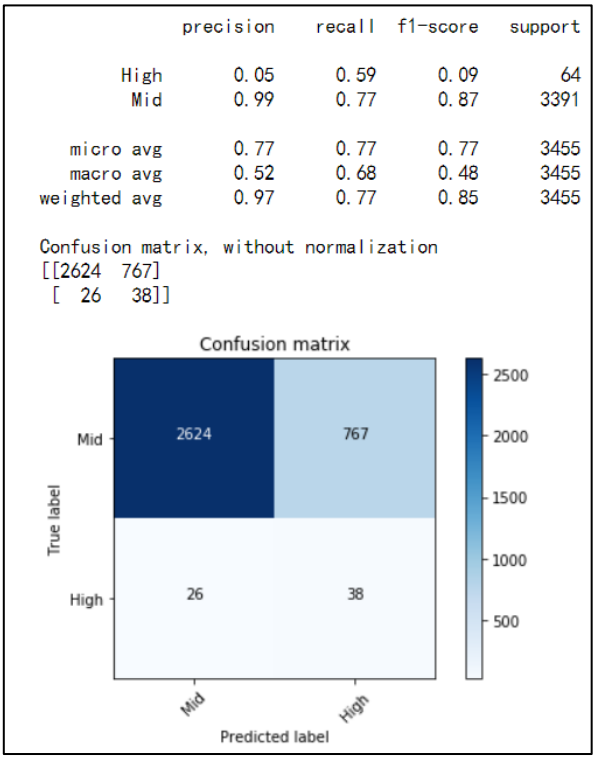
Model performance on level-2 classification:



	precision	recall	f1-score	support
High	0.02	0.88	0.04	64
Mid	0.99	0.26	0.41	3391
micro avg	0.27	0.27	0.27	3455
macro avg	0.51	0.57	0.23	3455
weighted avg	0.97	0.27	0.41	3455

Confusion matrix, without normalization
[[883 2508]
 [8 56]]

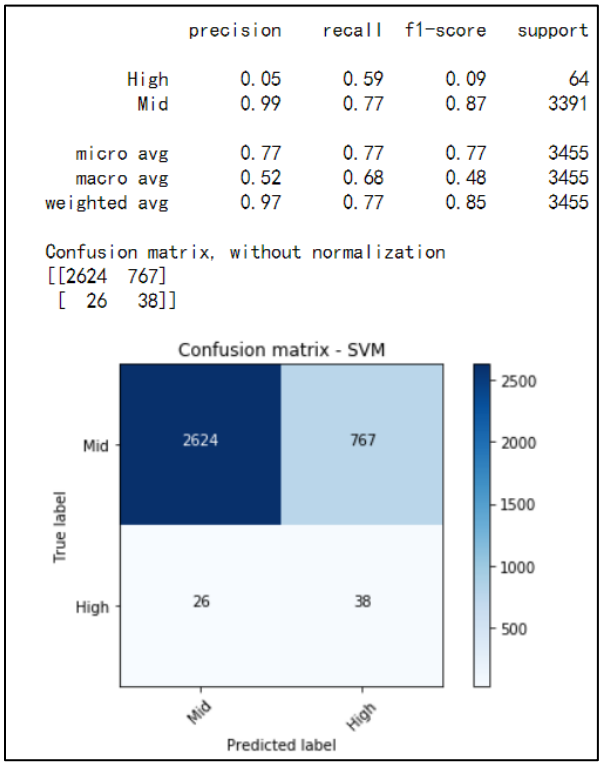
KNN



	precision	recall	f1-score	support
High	0.05	0.59	0.09	64
Mid	0.99	0.77	0.87	3391
micro avg	0.77	0.77	0.77	3455
macro avg	0.52	0.68	0.48	3455
weighted avg	0.97	0.77	0.85	3455

Confusion matrix, without normalization
[[2624 767]
 [26 38]]

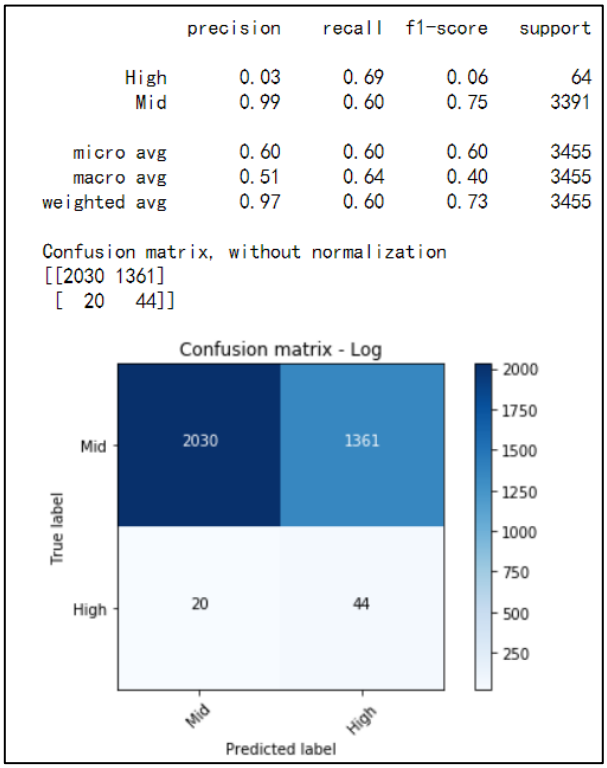
decision tree



	precision	recall	f1-score	support
High	0.05	0.59	0.09	64
Mid	0.99	0.77	0.87	3391
micro avg	0.77	0.77	0.77	3455
macro avg	0.52	0.68	0.48	3455
weighted avg	0.97	0.77	0.85	3455

Confusion matrix, without normalization
[[2624 767]
 [26 38]]

SVM



	precision	recall	f1-score	support
High	0.03	0.69	0.06	64
Mid	0.99	0.60	0.75	3391
micro avg	0.60	0.60	0.60	3455
macro avg	0.51	0.64	0.40	3455
weighted avg	0.97	0.60	0.73	3455

Confusion matrix, without normalization
[[2030 1361]
 [20 44]]

logistic regression

As above, we can find that in L2 prediction, the performance is quite poor. But considering there are very few samples of category High, maybe it won't be necessary to separate Mid and High.

5 Conclusions

In previous sections, I've tried to analyze the relationship between different features of an accident and its severity level. We can see that the collision type probably has the largest influence on the severity level. And other conditions also matter, for instance the location, and the driver's subjective mistakes. Then in the modeling section, I tried to train 4 types of classifier models (KNN, Decision Tree, SVM, Logistic Regression), with different selections of features and parameters. As a result, I found that if only use one-hot features, most models have a quite balanced performance.

At beginning, I planned to do 2 rounds of binary classification to separate all three categories (Low, Mid, High). But when looking at the L2 prediction performance, it seems that code-2 and code-2b cases probably have very similar features, considering they both involve human injuries. In addition, that the proportion of High severity cases is pretty low in the complete sample space (2000/152000~1.3%). So that it should be OK to just use level-1 prediction to separate code-1 and the others.

While the overall accuracy is around 60%, there must be a large room for improvement.

6. Future directions

When looking at the performance matrix, it's easy to find that most models have a high precision on predicting Low category, while it's almost a guessing game for Mid_High category. But during the tries with different feature selections, some of the models does give a better precision on Mid_High prediction.

In further investigation, maybe we can find a way to take advantage of the characters of different models to have a combined prediction with a higher overall accuracy. For example, train SVM with a certain selection of features to make it good at predicting Low category, then train KNN with another selection of features to specialize in Mid_High category, put on different weights on each model according to their skills, at last having a combined result.