Predicting the Severity Level of Traffic Accidents

Prediction on the severity of an accident can be useful

- An accident is inevitable, but alert can be send out to other drivers for precaution.
- A prediction on the severity of an accident can help other traffic participants to decide how they should react in face with a ”small” or ”big” accident.
- Such information can also be integrated into the on board GPS system to rearrange the route when necessary.

Data acquisition and cleaning

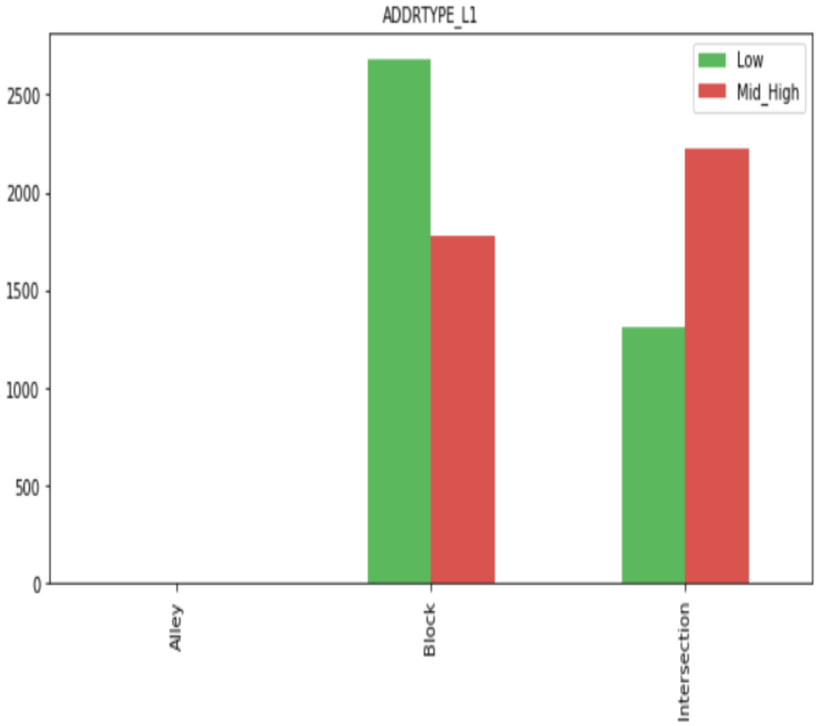
- Data is downloaded from SDOT’s data sharing page: SDOT GIS datasets/collisions
- This dataset records registered accidents happened in Seattle from 2004 to present. There are more than 2 million cases with 39 featrues in the original list, and each case is labelled with a severity code.
- Duplicate, meaningless, highly similar or highly correlated features were dropped.
- Cleaned data contains 17 features.

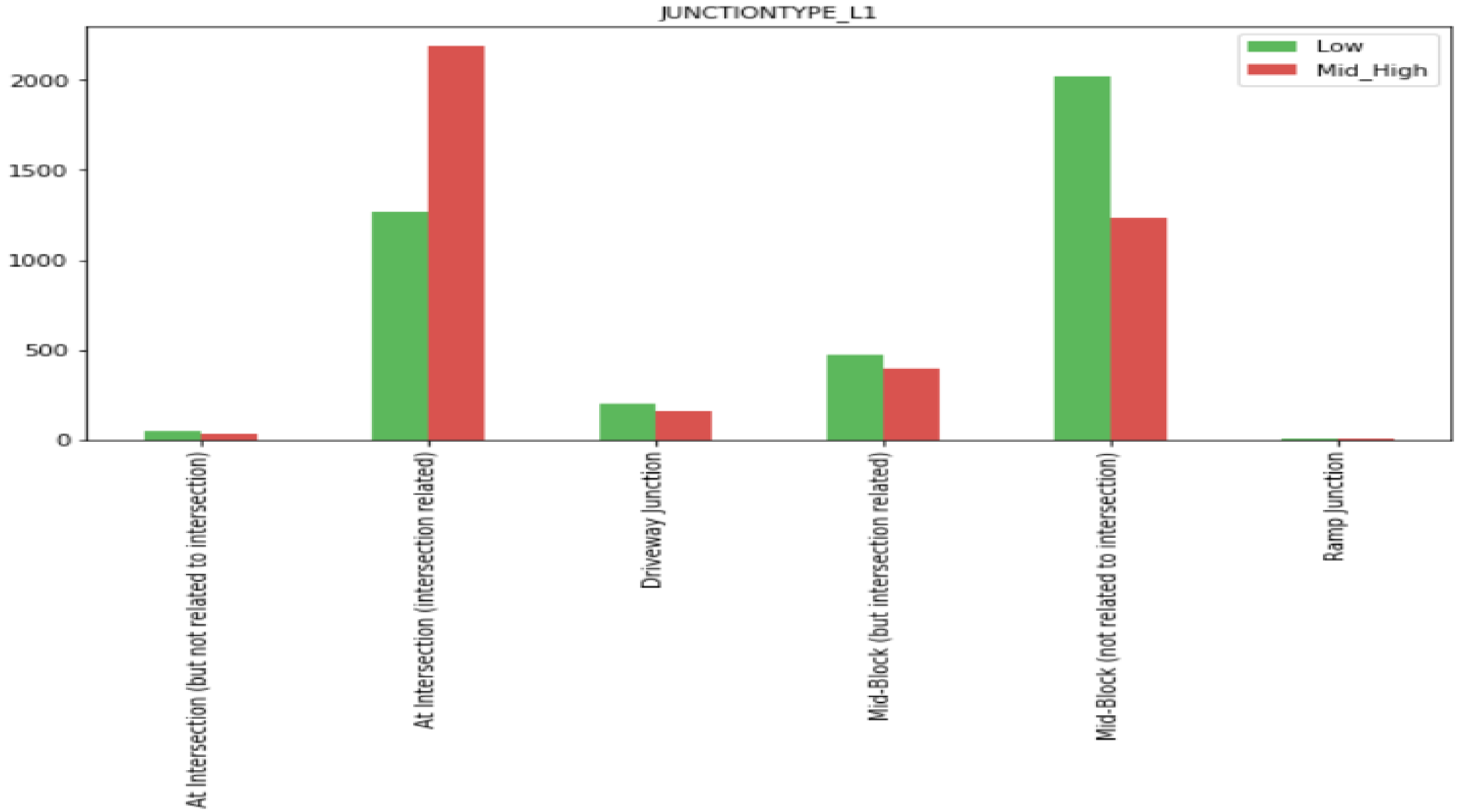
Data balancing

- As this is typically a classification problem, balance between each category is important in avoiding a biased model.
- After analysis, I decided to take out samples based on the table below. Then in the modeling section, I'll try to do 2 rounds of classification to separate 3 categories.

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

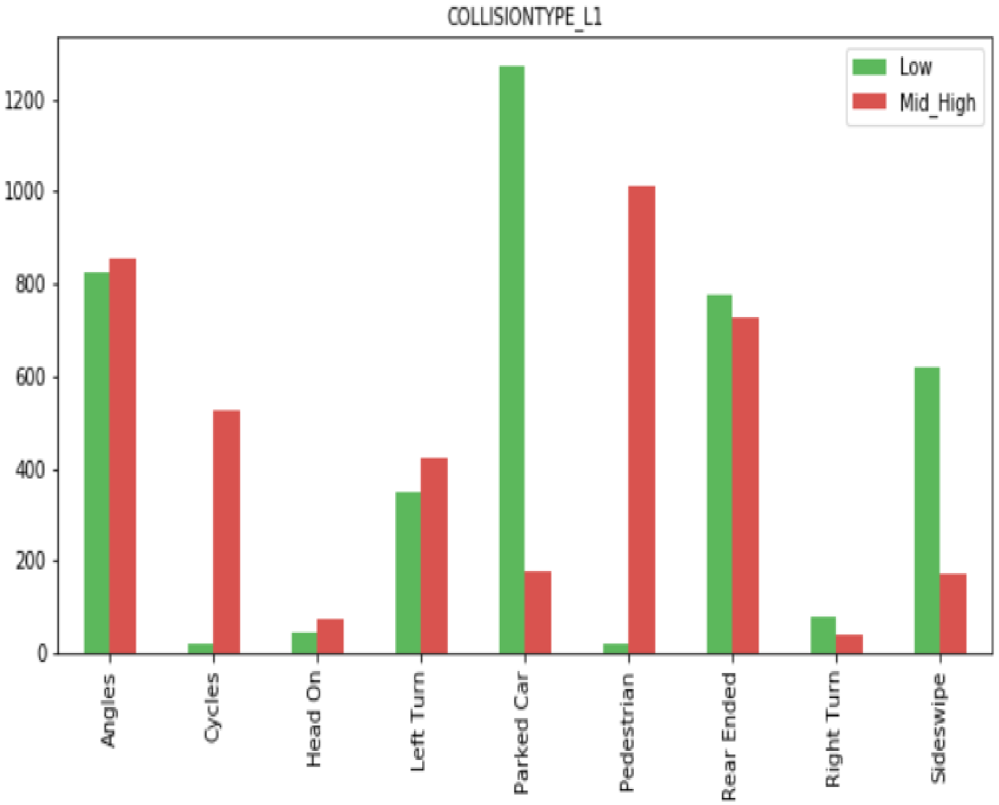
There are more higher severity cases at the intersection

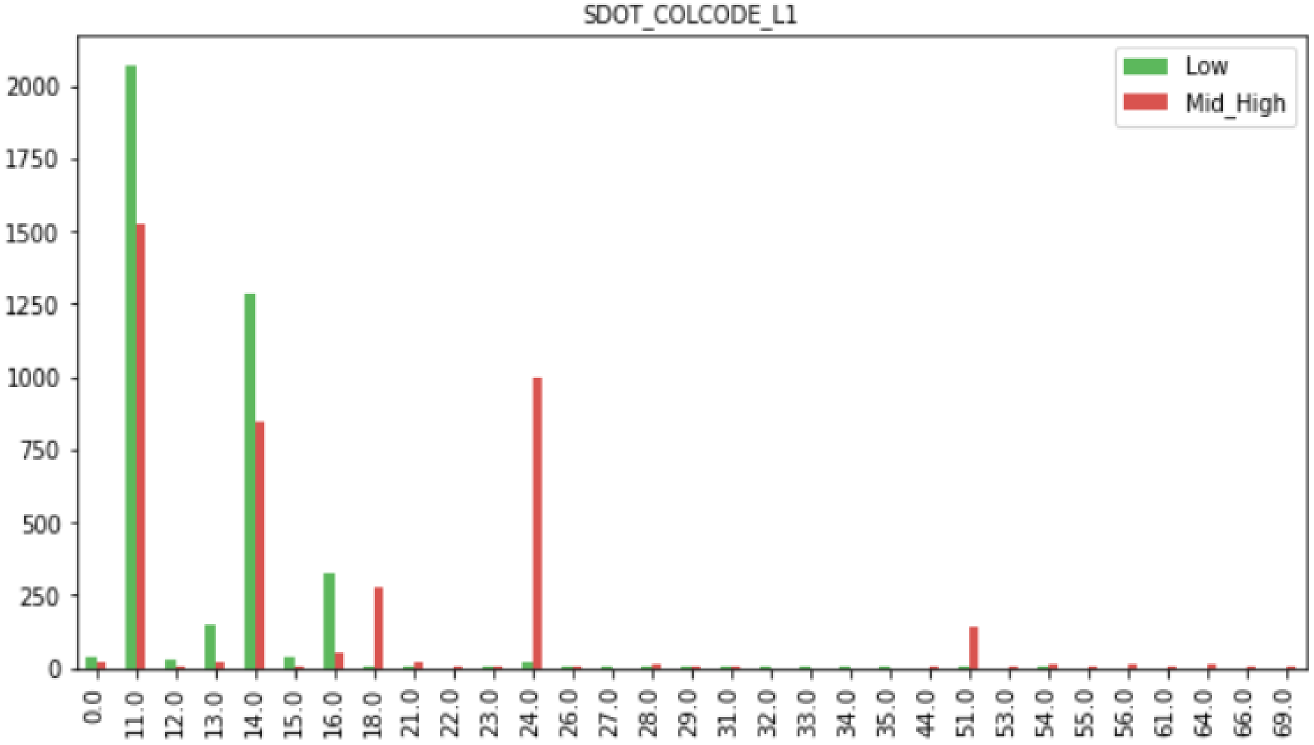




- When an accident is happened at an intersection, it’s more likely to have a higher severity level. Reason could be that the traffic condition is more compliaced there.

When pedestrian or cyclist involved, higher severity

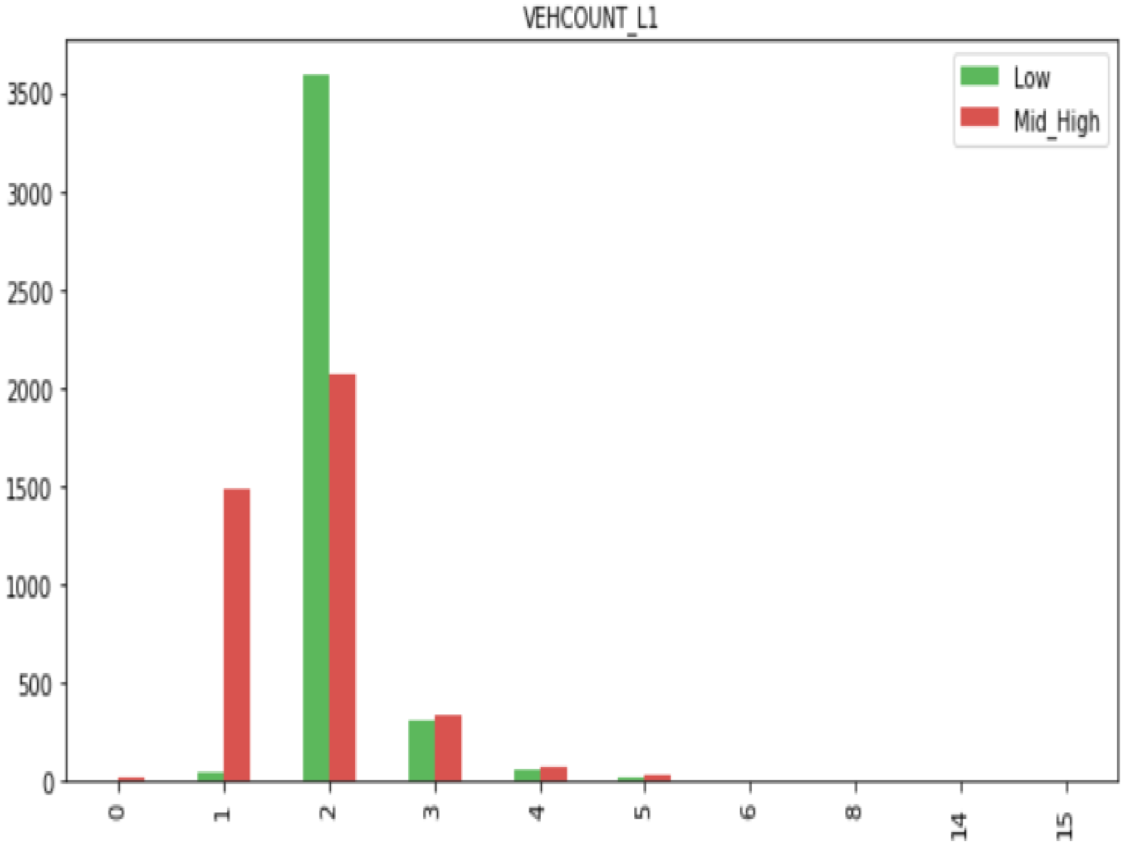




- When pedestrians or cyclists are hit by a vehicle, they’ll probably be injured, thus a higher severity.

** code-24 = hit pedestrians, code-18, 51 = hit cyclists*

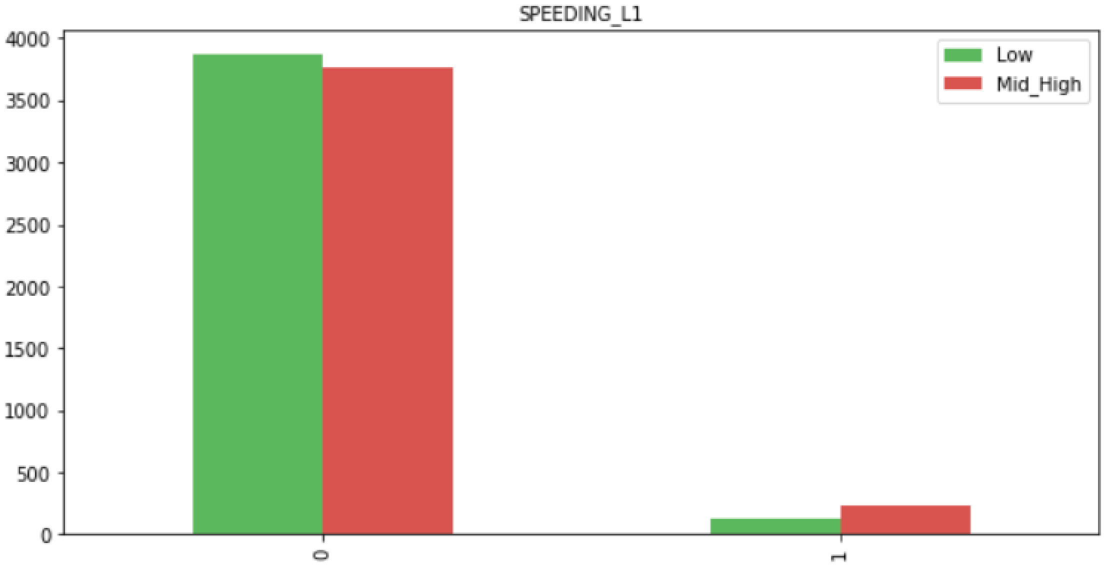
A single vehicle accident usually means higher severity

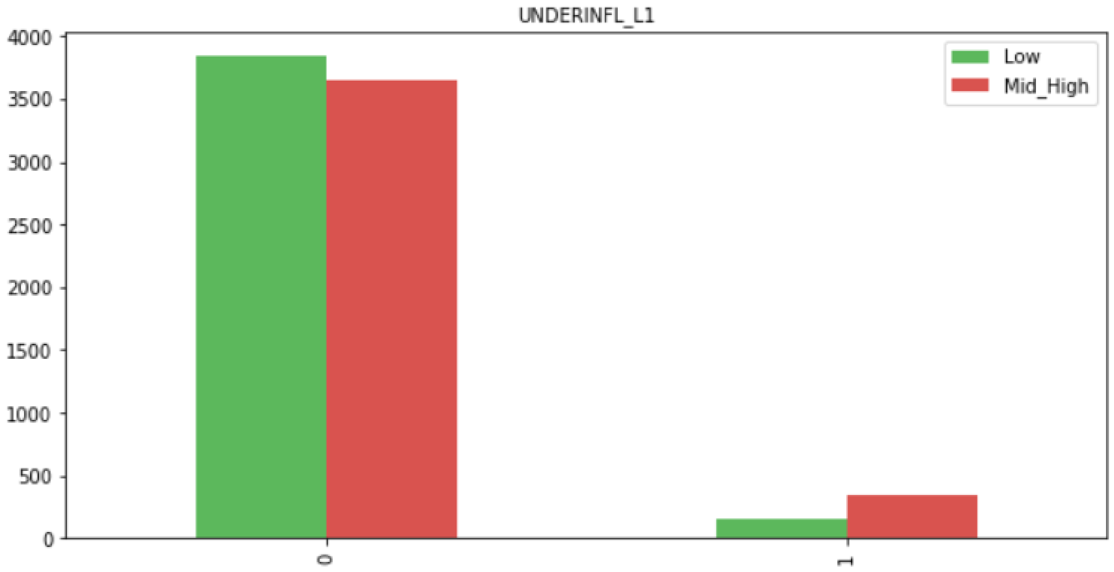


After a quick statistics in the original list, there are around 13000 cases with VEHCOUNT=1 and a certain collision type. Most of them involve pedestrians or cyclists (>99%), and 89% of those are labelled with code 2, 2b or 3.

- A single vehicle accident is probably a dirver vs non-driver case (pedestrians or cyclists) and that type of collision usually has a higher severity.

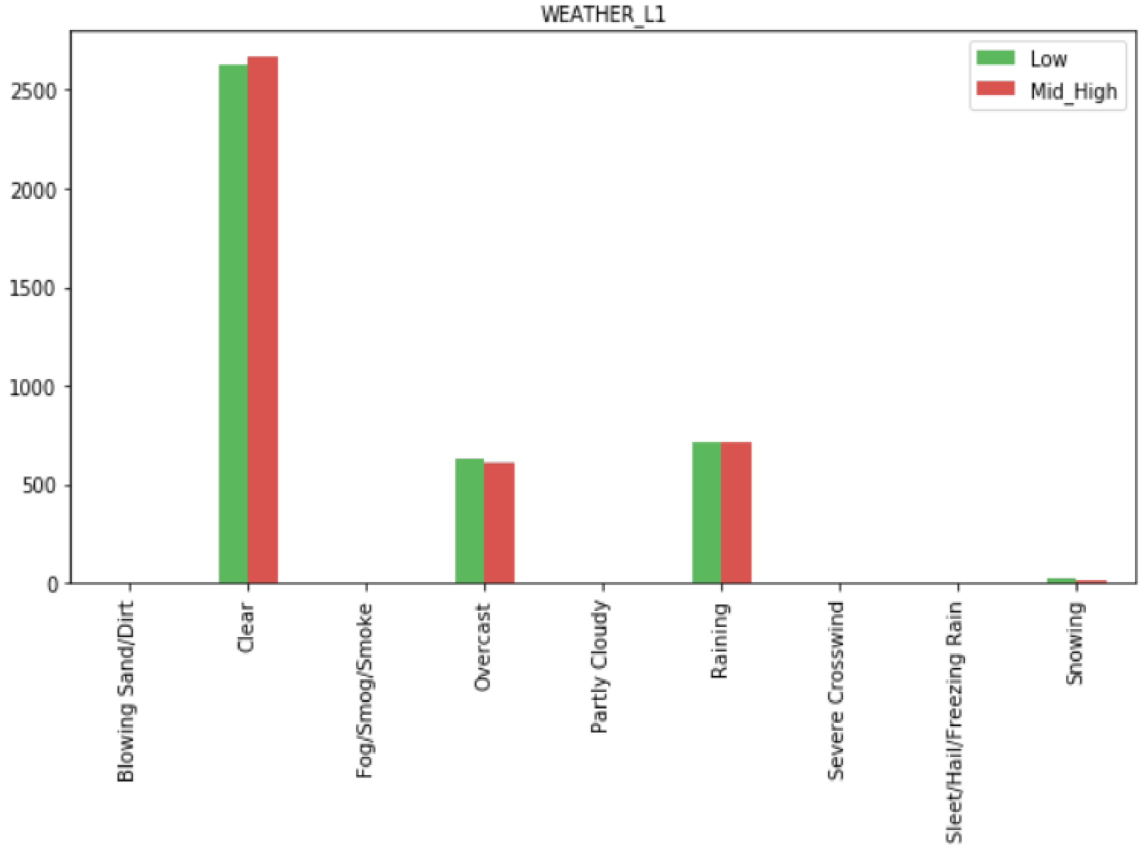
Subjective mistakes result in more higher severity cases

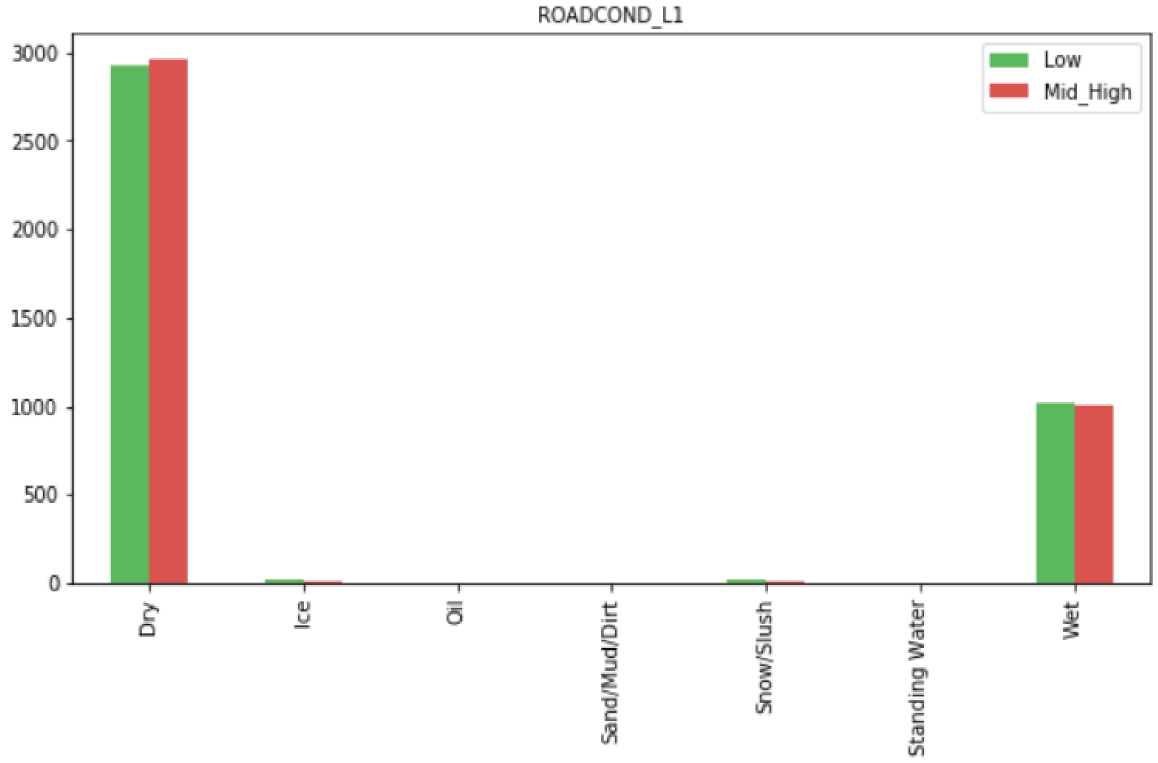




- If the accident is due to the driver’s subjective mistake, i.e. speeding, taking drugs or drinking alcohol, will result in more cases with a higher severity.

Wheather condition dosn’t have a noticable impact

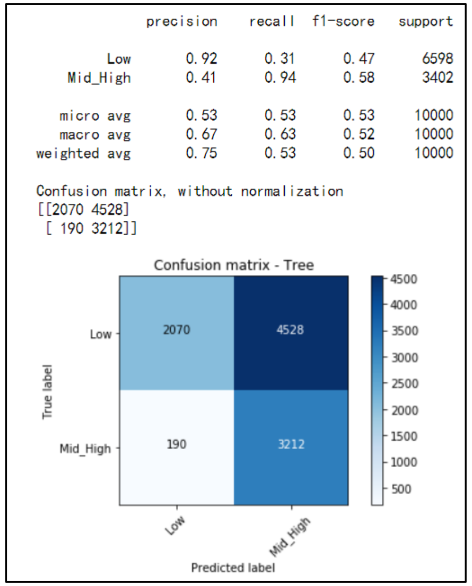




- Although it might be a common sense that wheather condition may have impact on the accidents, but the data shows us that it could just be a misunderstanding.

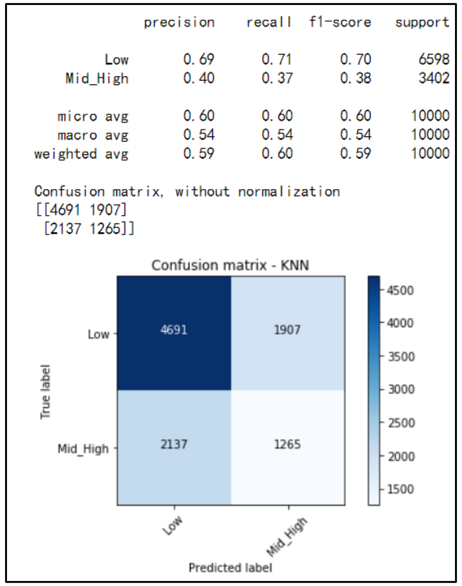
Model performance on different feature selections (KNN)

All features



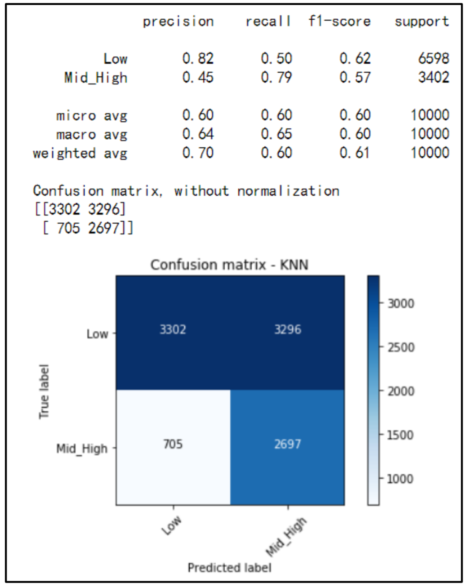
```
             precision    recall  f1-score   support

        Low       0.92      0.31      0.47      6598
   Mid_High       0.41      0.94      0.58      3402

  micro avg       0.53      0.53      0.53     10000
  macro avg       0.67      0.63      0.52     10000
weighted avg      0.75      0.53      0.50     10000

Confusion matrix, without normalization
[[2070 4528]
 [ 190 3212]]
```

w/o numerical



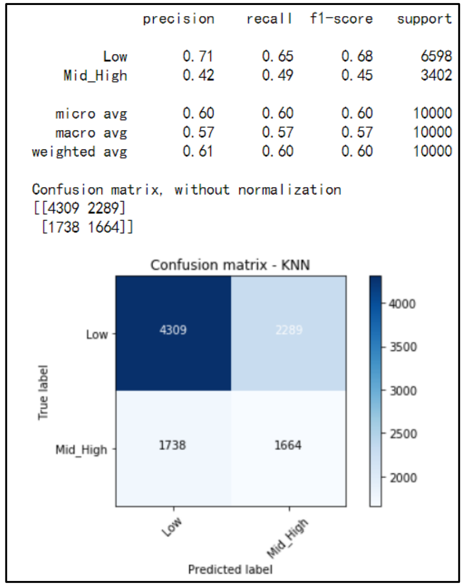
```
             precision    recall  f1-score   support

        Low       0.69      0.71      0.70      6598
   Mid_High       0.40      0.37      0.38      3402

  micro avg       0.60      0.60      0.60     10000
  macro avg       0.54      0.54      0.54     10000
weighted avg      0.59      0.60      0.59     10000

Confusion matrix, without normalization
[[4691 1907]
 [2137 1265]]
```

w/o categorical



```
             precision    recall  f1-score   support

        Low       0.82      0.50      0.62      6598
   Mid_High       0.45      0.79      0.57      3402

  micro avg       0.60      0.60      0.60     10000
  macro avg       0.64      0.65      0.60     10000
weighted avg      0.70      0.60      0.61     10000

Confusion matrix, without normalization
[[3302 3296]
 [ 705 2697]]
```

only one-hot

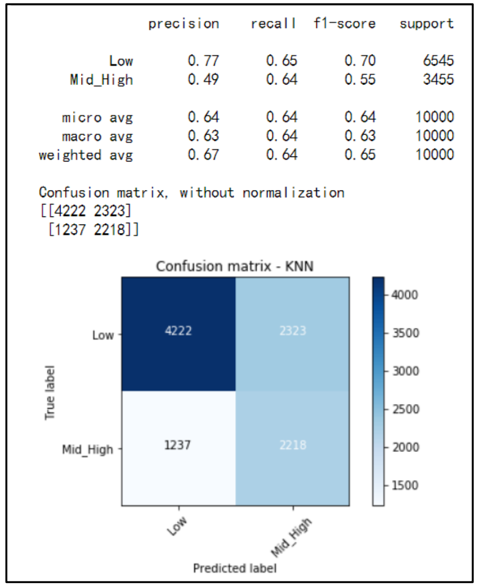


```
             precision    recall  f1-score   support

        Low       0.71      0.65      0.68      6598
   Mid_High       0.42      0.49      0.45      3402

  micro avg       0.60      0.60      0.60     10000
  macro avg       0.57      0.57      0.57     10000
weighted avg      0.61      0.60      0.60     10000

Confusion matrix, without normalization
[[4309 2289]
 [1738 1664]]
```

- When training the model with different selection of features, the performance varies quite a lot.
- I decided to use only one-hot features for training of other models, since the result looks more balanced.

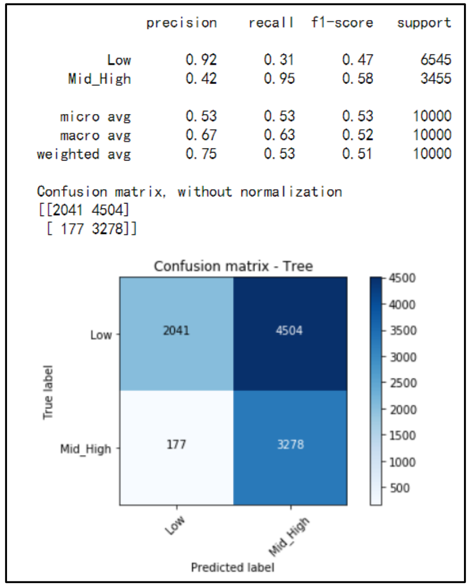
Model performance on level-1 (Low vs Mid_High)

KNN



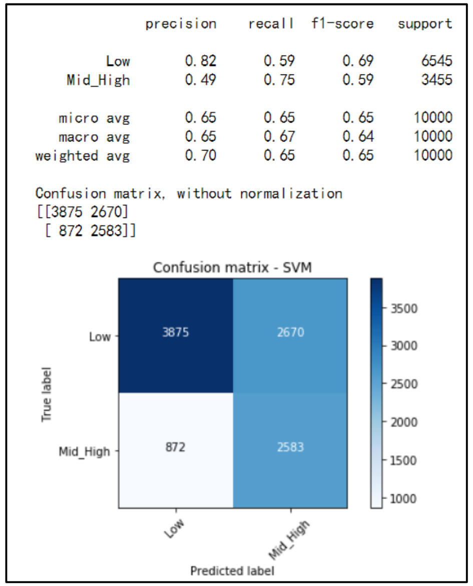
```
              precision    recall  f1-score   support

         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]]
```

Decision Tree



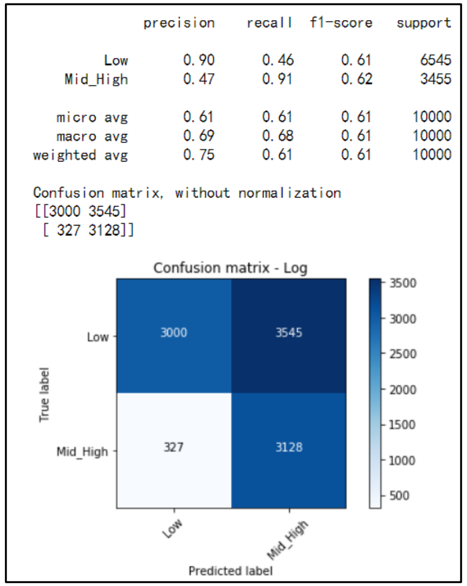
```
              precision    recall  f1-score   support

         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]]
```

SVM



```
              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]]
```

Logistic Regression

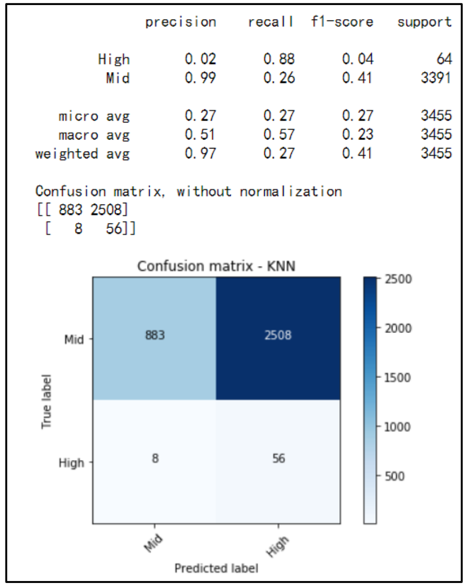


```
              precision    recall  f1-score   support

         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]]
```

- If we gauge model performance based on accuray then the overall result of each model is SVM(0.65)>KNN(0.64)>Log(0.6)>D.T(0.53)
- But if paying more attention to ”recall”, easy to find that they all did poorly on Low cases, there must be some way to improve that.

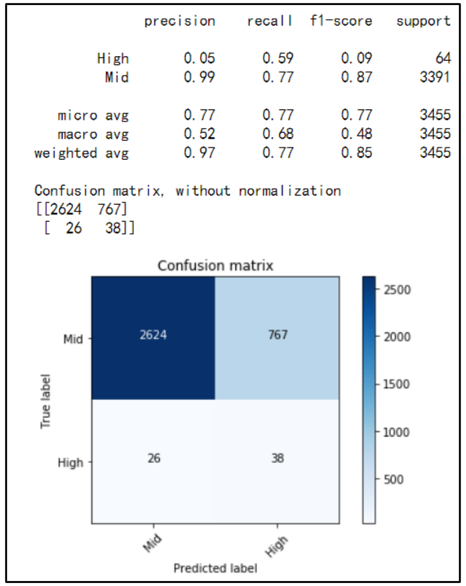
Model performance on level-2 (Mid vs High)

KNN



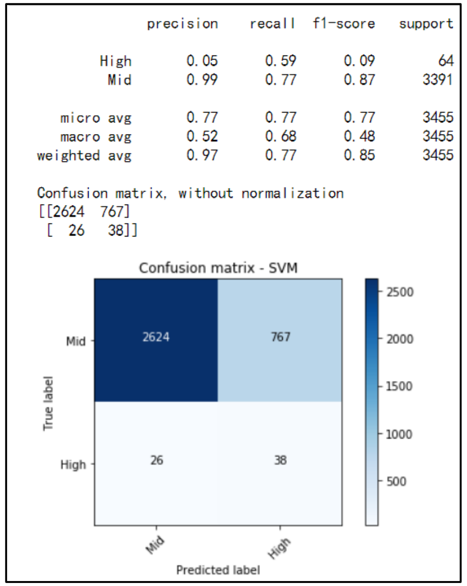
```
             precision    recall  f1-score   support

        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]]
```

Decision Tree



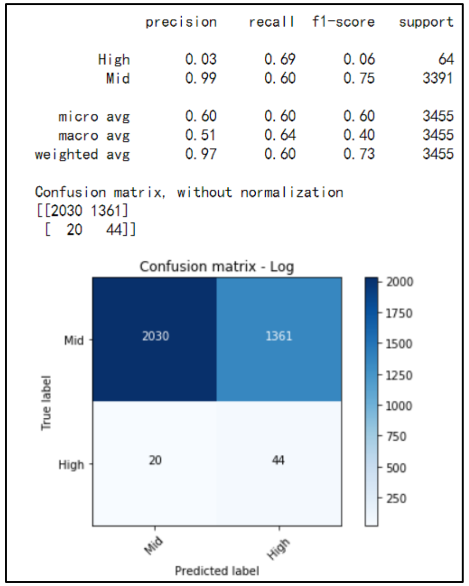
```
             precision    recall  f1-score   support

        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       0.05      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]]
```

Logistic Regression



```
             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]]
```

- 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.

Conclusion and future directions

- When training with only one-hot, models will have a more balanced performance, but the recall of Low category is not very good.
- It should be possible to train models with different selection of features to achieve unique specialties. Then combine them together to get a higher accuracy of the prediction.
- Code 2, 2b, 3 are not so easy to be separated, we can consider to only devide severity level into Low and High for code-1 and the rest.