PREDICTING CAR CRASH SEVERITY FOR SOUTH AUSTRALIA

Capstone Project

ABSTRACT

Prediction of Crash Severity is an old topic. This project focus on find a better way to apply for machine learning, including data understanding, cleaning, Data Balancing, one hot encoding, Normalization, PCA, Predictive Modelling, Predicting, Evaluation and what I found.

YINGMIN ZHAO

Data Science

Contents

1. Introductory

1.1. Background

The number of traffic accidents has been a rising trend globally due to increases in population and motorization. Different factors are involved in traffic crashes. If the possibility of drivers/cars getting into an accident can be predicted, the drivers would drive more carefully, change the route or travel plan. Also, the critical issue identified could be solved in advance. So how can we know the possibility of getting into a car accident? What kind factors/conditions are affecting the possibility and accident severity? This project could provide the answers/hints.

1.2. Purposes and Requirements

The purpose of Capstone project is building a prediction system for Car Accident Severity using machine learning technologies. It involves statistical modelling considering crash severity as a dependent variable which is the target of prediction. As an option, the shared dataset of Seattle city of USA can be used for it, or you can find own dataset from open data source.

I would like to research on car crash statistics of Australia instead of Seattle City. Aims to understanding followings:

1. Understanding what the factors impact on car accidents
2. The possibility of car accidents happening with the certain found factors
3. Technically, what the machine learning algorithm/model are more suitable for the project.

As expecting, the data set should include labelled severity for each accident, the supervised machine learning algorithms will be applied for it.

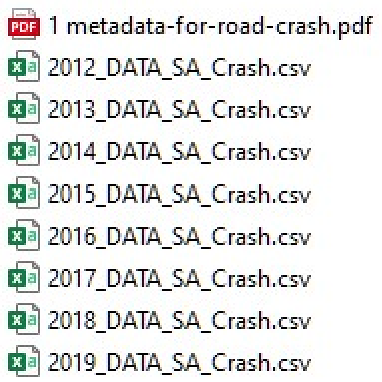
In fact, this is not a new topic, I will focus on finding the suitable features/variables and better algorithm.

2. Methodology

2.1. Data sources

I am interested in crash statistic of Australia where I am living, even though a shared dataset sample is provided by this project. I would like to run it on my own dataset. Unfortunately, I only found a good one for South Australia, not NSW's. It fully covers 2012 – 2019 and would be pretty good to support this task. (All relevant materials are shared in Appendix.)

The data set have seperate csv files for each year with metadata. There are total 127672 records and 33 attributes.

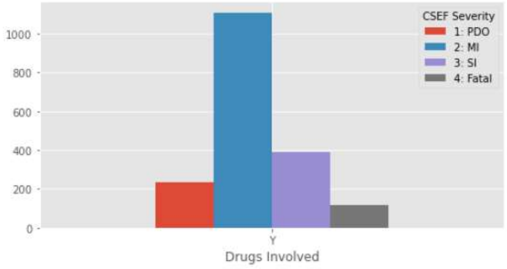


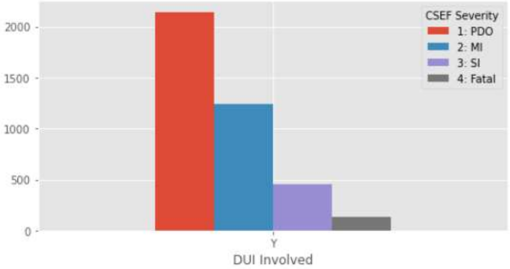
1 metadata-for-road-crash.pdf
2012_DATA_SA_Crash.csv
2013_DATA_SA_Crash.csv
2014_DATA_SA_Crash.csv
2015_DATA_SA_Crash.csv
2016_DATA_SA_Crash.csv
2017_DATA_SA_Crash.csv
2018_DATA_SA_Crash.csv
2019_DATA_SA_Crash.csv

2.1.1. Data Cleaning

2.1.1.1. feature selection

Removed “records index, Date/time and location information” which are not related to the accidents; The number of injuries is also removed, which just a result and won’t work as an independent variable; Drunk and drug involvement indicators only have small samples(<3% of 127672 rows), and the portion(vs. Severity) doesn’t show the correlation to severity, both are dropped as well. Now there are all Non-human factors as independent features.

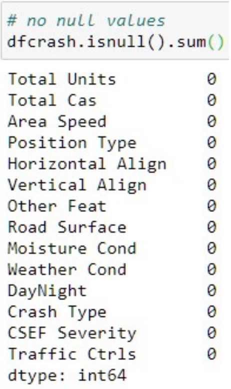




	Features	Description	Removed Columns	
1	Total Units	The total number of units involved in a road crash	REPORT_ID	DUI Involved
2	Total Cas	Total number of casualties (fatalities + treated injuries) as a result of a road crash	Stats Area	Drugs Involved
3	Area Speed	The speed limit at the time and location of the crash	Suburb	ACCLOC_X
4	Position Type	Identifying if a crash location was at an intersection or midblock	Postcode	ACCLOC_Y
5	Horizontal Align	Defines the horizontal alignment of the road at the sight of the crash	LGA Name	UNIQUE_LOC
6	Vertical Align	Defines the vertical alignment of the road at the sight of the crash	Total Fats	Entity Code
7	Other Feat	Defines other relevant features of the crash site locations	Total SI	Unit Resp
8	Road Surface	Defines the road surface type at the crash location	Total MI	Day
9	Moisture Cond	Defines the pavement surface moisture condition at the crash location	Year	Time
10	Weather Cond	Defines the weather condition at the time and location of the crash	Month	
11	DayNight	The lighting condition at the time and location of the crash		
12	Crash Type	Defines the road crash type		
13	Traffic Ctrls	Defines the traffic control at the time and location of the road crash		
	CSEF Severity	4: Fatal 3: SI = Serious Injury, 2: MI = Minor Injury, 1: PDO = Property Damage Only		

2.1.1.2. Data Formatting

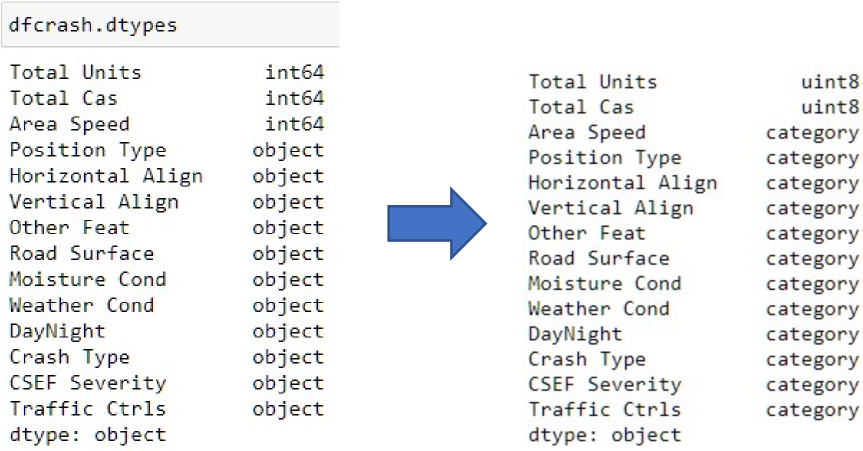
- It is perfect data without any null values.



```
# no null values
dfcrash.isnull().sum()
```

```
Total Units          0
Total Cas            0
Area Speed           0
Position Type        0
Horizontal Align     0
Vertical Align       0
Other Feat           0
Road Surface         0
Moisture Cond        0
Weather Cond         0
DayNight             0
Crash Type           0
CSEF Severity        0
Traffic Ctrls        0
dtype: int64
```

- Cast all Object type to DataFrame category type, int64 to int8 type. Area Speed is enum type so set as category.



```
dfcrash.dtypes
```

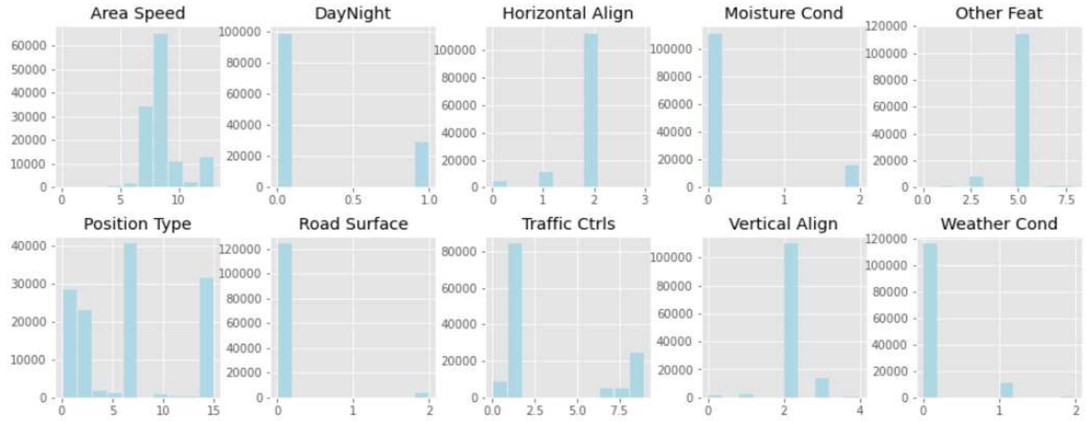
```
Total Units         int64
Total Cas           int64
Area Speed          int64
Position Type      object
Horizontal Align   object
Vertical Align     object
Other Feat         object
Road Surface       object
Moisture Cond      object
Weather Cond       object
DayNight           object
Crash Type         object
CSEF Severity      object
Traffic Ctrls      object
dtype: object
```

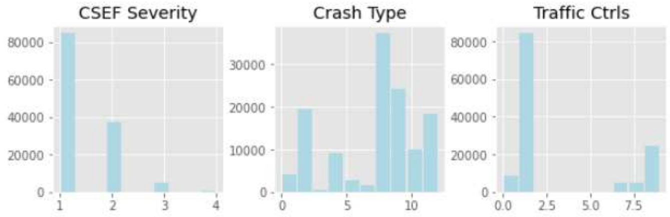
```
Total Units            uint8
Total Cas              uint8
Area Speed          category
Position Type       category
Horizontal Align    category
Vertical Align      category
Other Feat          category
Road Surface        category
Moisture Cond       category
Weather Cond        category
DayNight            category
Crash Type          category
CSEF Severity       category
Traffic Ctrls       category
dtype: object
```

2.1.2. Data Understanding and Pre-processing

2.1.2.1. Distribution of sub-categories

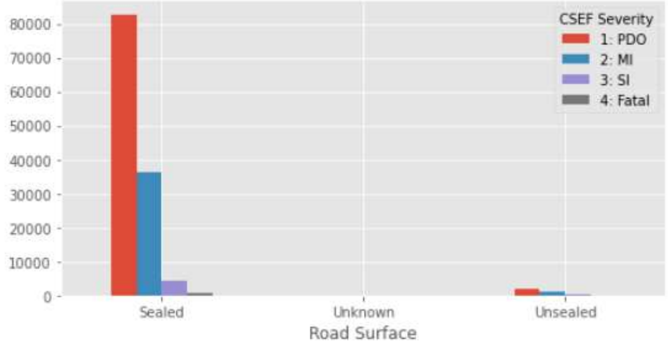
most features have imbalance sub-categories. Need furthur checking of the relationship with severity.



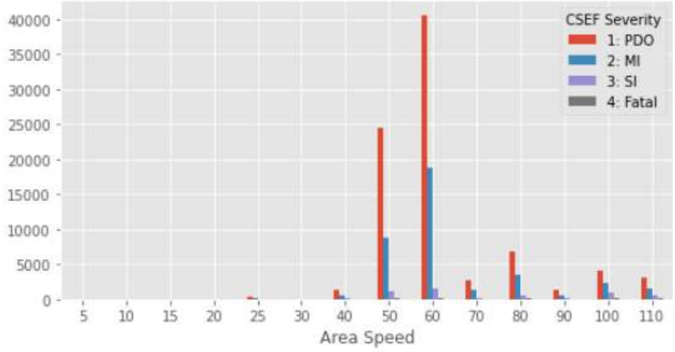


2.1.2.2. Relationship to Severity

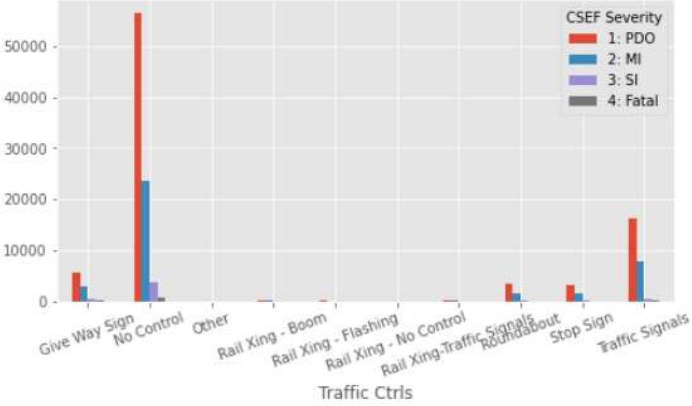
- Each categories of Road surface shows obvious tendency. It would be one of right features potentially.



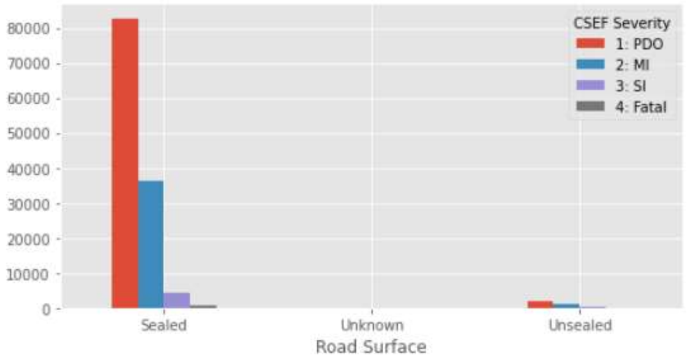
- 50&60 KM of Area Speed connected to 75% accidents, but each category has similar portion of severity.



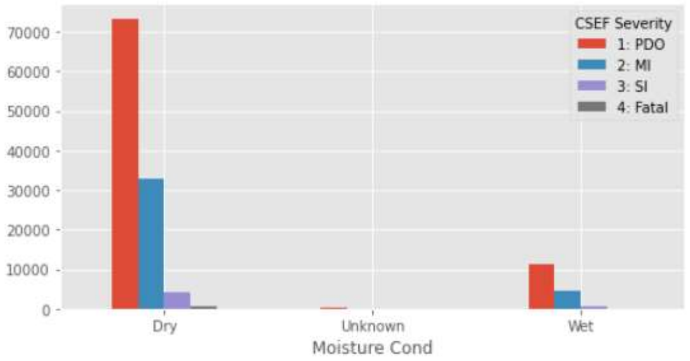
- 60% accidents happened in the area without traffic control. Main categories have similar portion of severity.



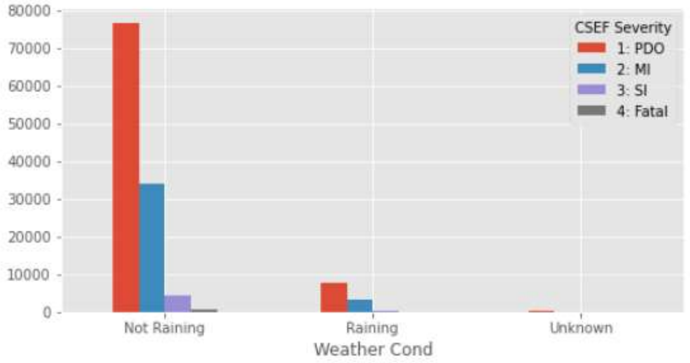
- Most accidents happened on sealed road and 60% Severity 1: POD of obvious tendency.



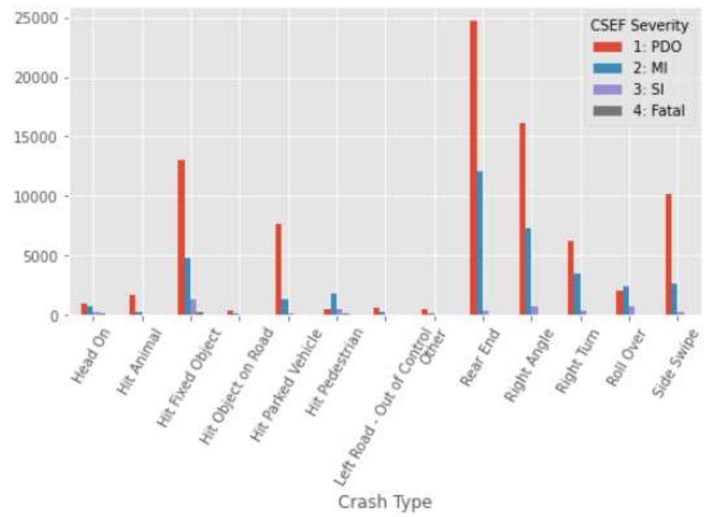
- Most accidents happened on dry road and 55% Severity 1: POD of obvious tendency.



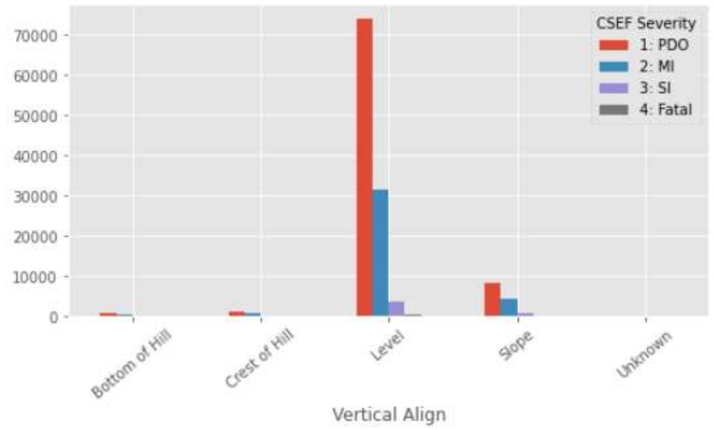
- Most accidents happened on no-raining day and 60% Severity 1: POD of obvious tendency.



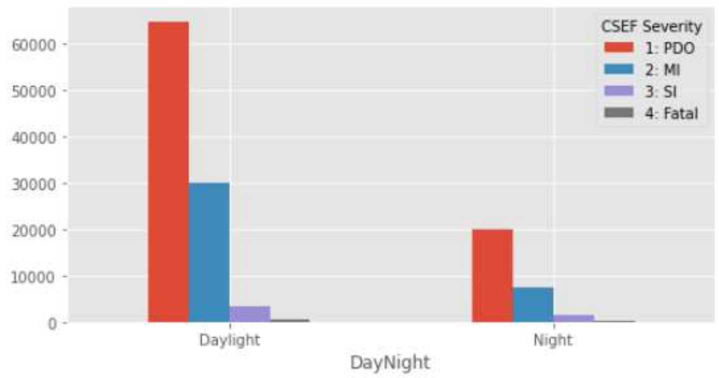
- There are many Crash type with Relative average distribution and similar portion of severity.



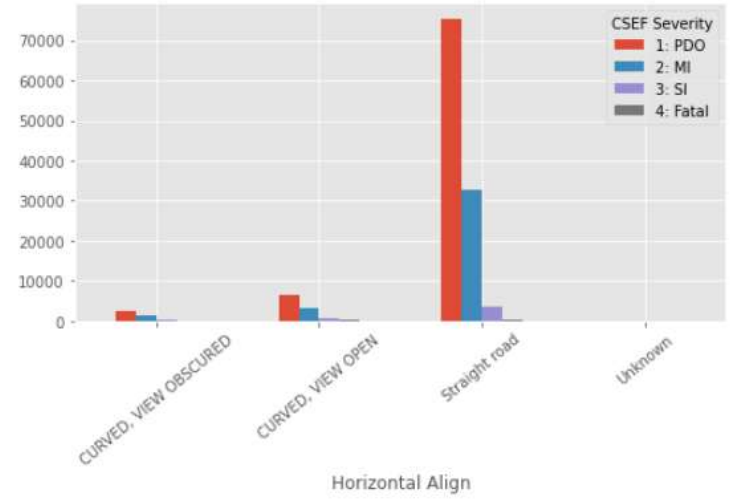
- 90% happened on level ground.



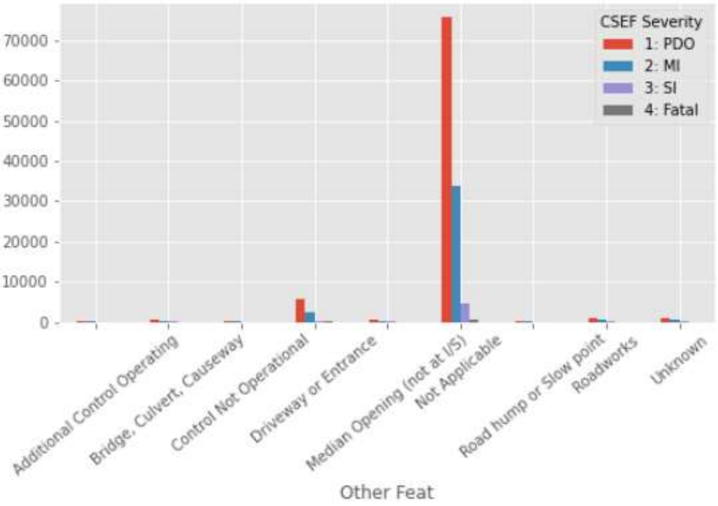
- Most happened on daytime.



- Most happened on straight road.

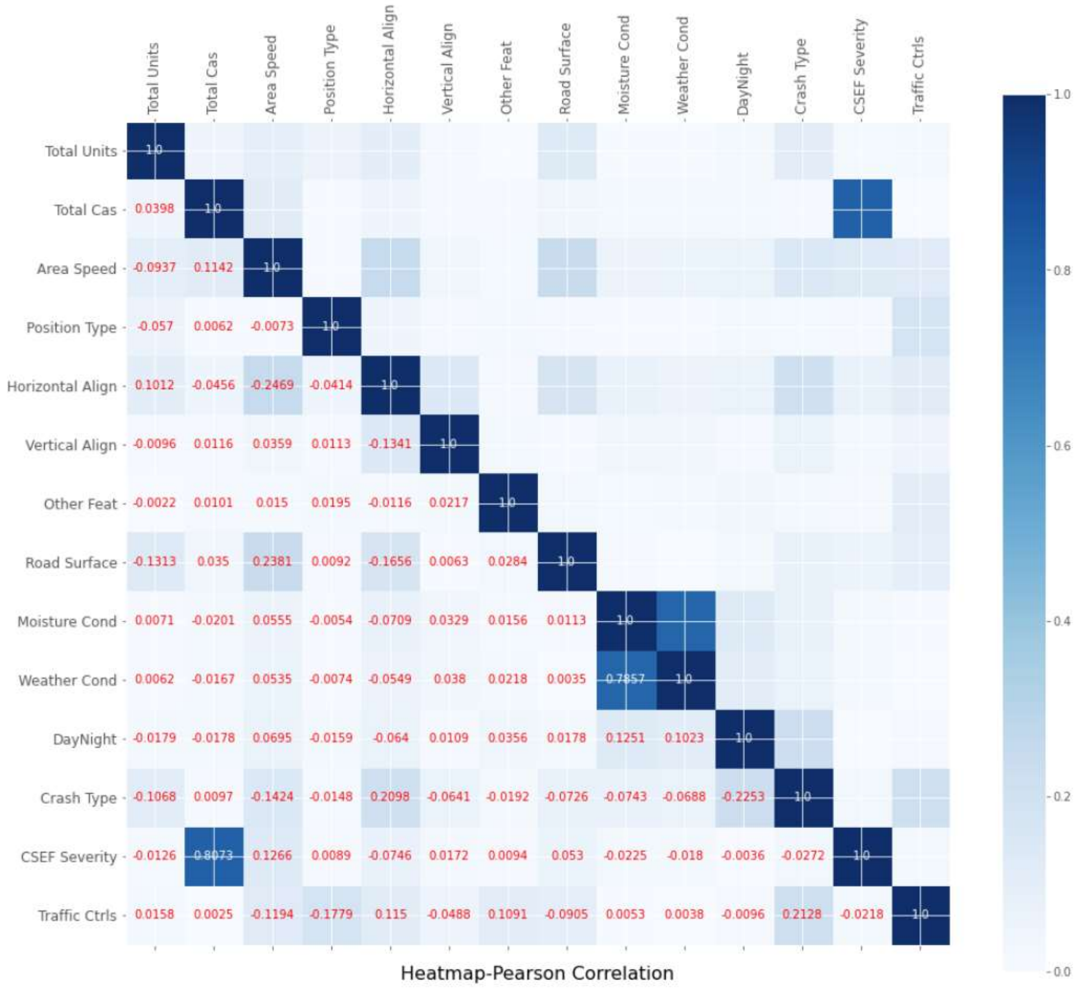


- Most happened on opening area.



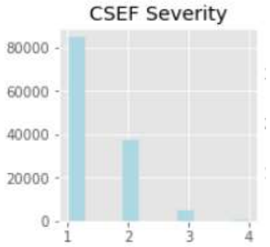
2.1.2.3. Pearson Correlation

Overall, all single feature has the very weak positive or negative correlation. However, it does not mean we will have inaccurate prediction using these dataset as Pearson correlation only shows single correlation each other. Further correlation analysis is required.



Heatmap-Pearson Correlation

2.1.2.4. Unbalanced Data



Balanced portion of severity is not expected because most are minor traffic accident. That is a good thing for our life, but it is an unfavourable factor for the research. Severity 2: Minor Injury is Mild level of imbalance – 20-40% of the data set; Severity 3: Serious Injury is Moderate level – 1-20%; Severity 4: Fatal is Extreme level – < 1%.

Following data will be used and compared with the performance.

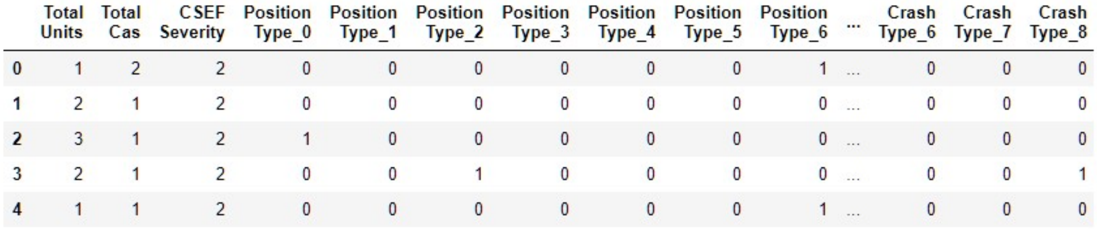
Severity	Original Data Quantity	Up-sampling Quantity	Down-sampling Quantity	Test Data Quantity
1: POD	84775	84775	37300	2000
2: MI	37300	84775	37300	2000
3: SI	4875	84775	37300	2000
4: Fatal	722	84775	37300	2000

2.1.2.5. One-Hot Encoding

Most data I got are categorical. Only Decision Tree algorithms can work with it directly, with no data transform required. Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.

Integer Encoding and One-Hot Encoding can be used to convert categorical data to numerical data. For categorical variables where no such ordinal relationship exists, the integer encoding is not enough. It would lead the model to assume a natural ordering between categories may result in poor performance or unexpected results.

So, One-Hot encoding is applied.



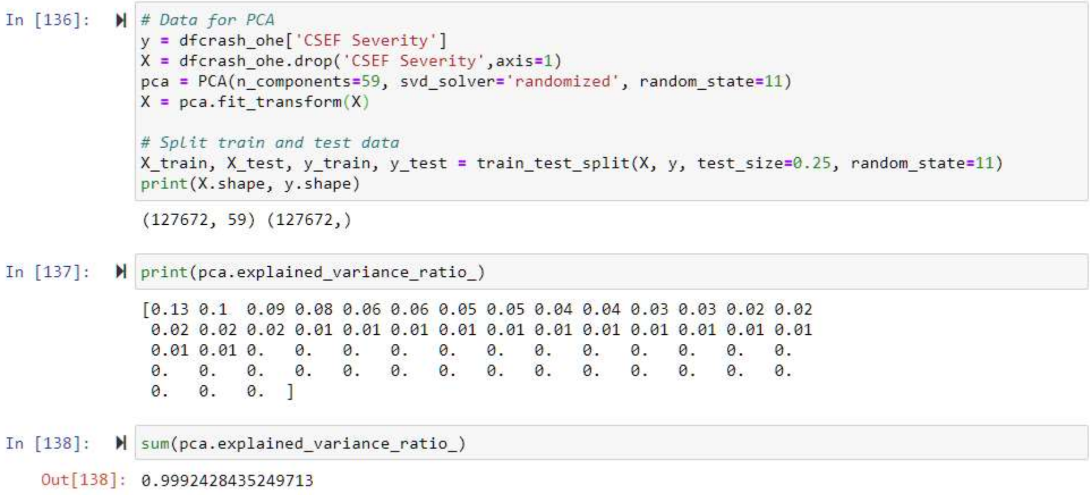
	Total Units	Total Cas	CSEF Severity	Position Type_0	Position Type_1	Position Type_2	Position Type_3	Position Type_4	Position Type_5	Position Type_6	...	Crash Type_6	Crash Type_7	Crash Type_8
0	1	2	2	0	0	0	0	0	0	1	...	0	0	0
1	2	1	2	0	0	0	0	0	0	0	...	0	0	0
2	3	1	2	1	0	0	0	0	0	0	...	0	0	0
3	2	1	2	0	0	1	0	0	0	0	...	0	0	1
4	1	1	2	0	0	0	0	0	0	1	...	0	0	0

2.1.2.6. Principal component analysis (PCA) and Dimensionality Reduction

One-hot encoding added many new binary variables/columns, total 80 variables. Also, in previous section, Pearson correlation could not find any strong relationship.

Features selection need be implemented for two purposes: 1. Measuring correlation for multiple dimensional variables; 2. Dimensionality Reduction.

I reduced Dimensionality to 59 components which cover 99.9% variance ratio.



```
In [136]:  # Data for PCA
           y = dfcrash_ohe['CSEF Severity']
           X = dfcrash_ohe.drop('CSEF Severity',axis=1)
           pca = PCA(n_components=59, svd_solver='randomized', random_state=11)
           X = pca.fit_transform(X)

           # Split train and test data
           X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=11)
           print(X.shape, y.shape)

           (127672, 59) (127672,)

In [137]:  print(pca.explained_variance_ratio_)

           [0.13 0.1  0.09 0.08 0.06 0.06 0.05 0.05 0.04 0.04 0.03 0.03 0.02 0.02
            0.02 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
            0.01 0.01 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
            0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
            0.   0.   0.  ]

In [138]:  sum(pca.explained_variance_ratio_)

Out[138]:  0.9992428435249713
```

Simultaneously, the data set is normalized by PCA; Next, Split the dataset to 75% for training and 25% for testing; the random state is set to 11 for models; Then the dataset is ready to be fed to model.

2.2. Predictive Model

This is a multiple classification problem. I would like to use several algorithms with supervised method to find the best model – K-Nearest Neighbours model, Decision Tree model, Logistic Regression model and Support Vector Machine.

2.2.1. Building Baseline

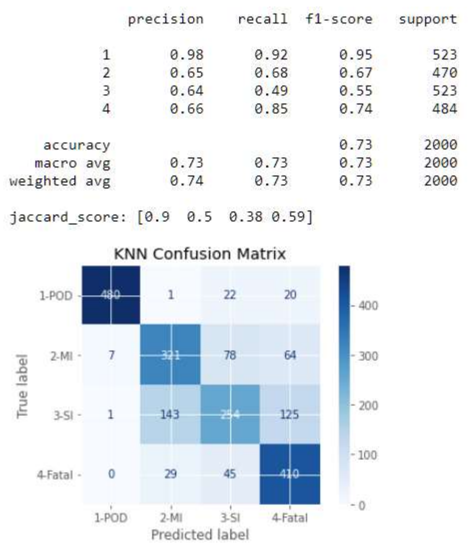
To reduced running time, I created a small dataset and built the baseline models.

Severity 1: POD – reached 95% ~ 100% accuracy.

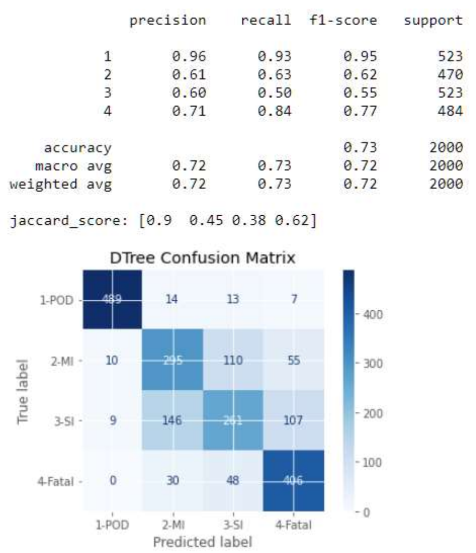
Severity 2: MI – has 65% accuracy.

Severity 1: SI – only 40% ~ 55% accuracy.

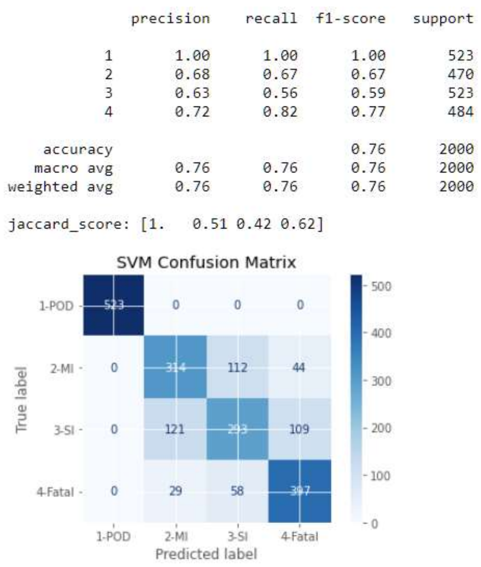
Severity 1: Fatal – has 60% ~ 77% accuracy.



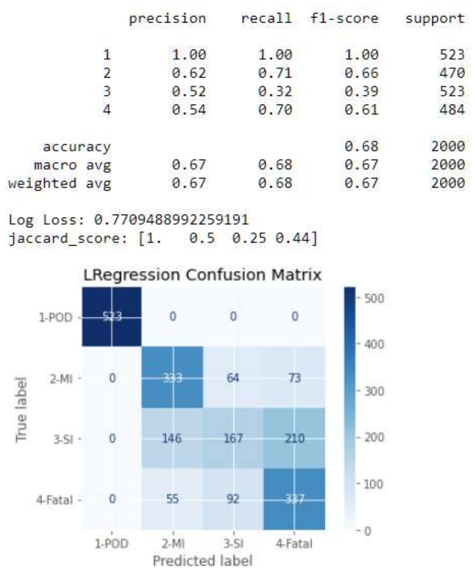
```
              precision    recall  f1-score   support

           1       0.98      0.92      0.95       523
           2       0.65      0.68      0.67       470
           3       0.64      0.49      0.55       523
           4       0.66      0.85      0.74       484

    accuracy                           0.73      2000
   macro avg       0.73      0.73      0.73      2000
weighted avg       0.74      0.73      0.73      2000

jaccard_score: [0.9  0.5  0.38 0.59]
```



```
              precision    recall  f1-score   support

           1       0.96      0.93      0.95       523
           2       0.61      0.63      0.62       470
           3       0.60      0.50      0.55       523
           4       0.71      0.84      0.77       484

    accuracy                           0.73      2000
   macro avg       0.72      0.73      0.72      2000
weighted avg       0.72      0.73      0.72      2000

jaccard_score: [0.9  0.45 0.38 0.62]
```

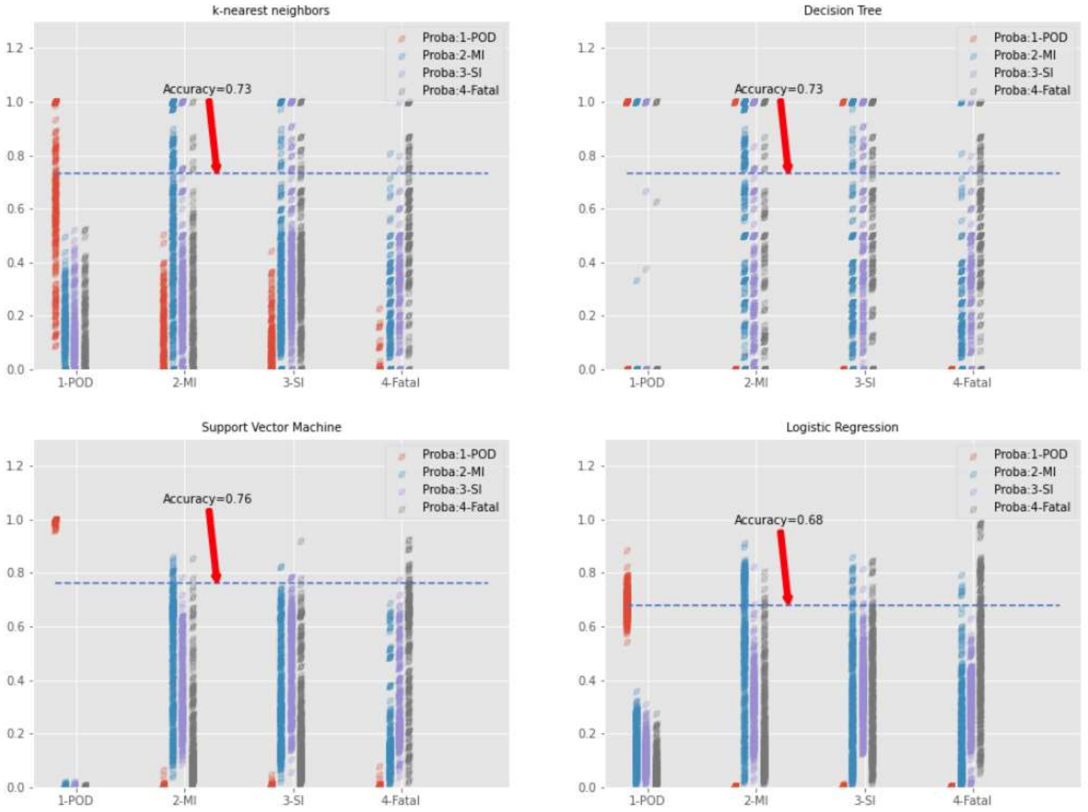


```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       523
           2       0.68      0.67      0.67       470
           3       0.63      0.56      0.59       523
           4       0.72      0.82      0.77       484

    accuracy                           0.76      2000
   macro avg       0.76      0.76      0.76      2000
weighted avg       0.76      0.76      0.76      2000

jaccard_score: [1.   0.51 0.42 0.62]
```

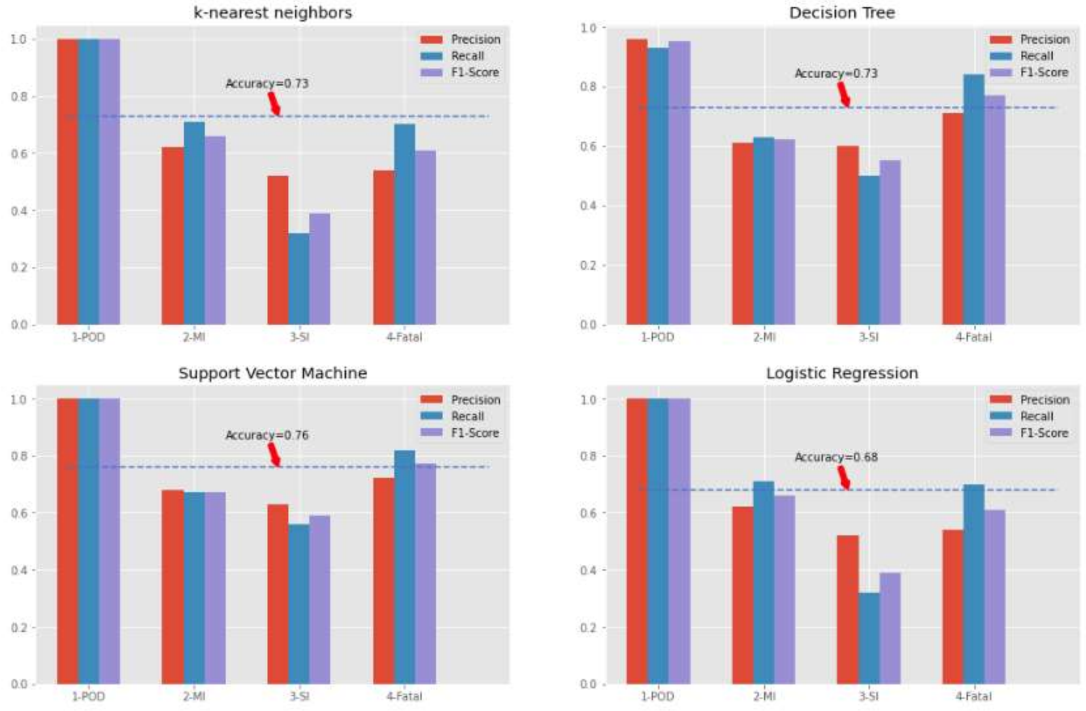


```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       523
           2       0.62      0.71      0.66       470
           3       0.52      0.32      0.39       523
           4       0.54      0.70      0.61       484

    accuracy                           0.68      2000
   macro avg       0.67      0.68      0.67      2000
weighted avg       0.67      0.68      0.67      2000

Log Loss: 0.7709488992259191
jaccard_score: [1.   0.5  0.25 0.44]
```

Comparison of Probability



Comparison of Model Evaluation



2.2.2. Cross Validation for Model Parameter Optimization

Compare randomized search and grid search for optimizing hyperparameters of SVM. All parameters that influence the learning are searched simultaneously (except for the number of estimators, which poses a time / quality trade-off).

The randomized search and the grid search explore exactly same space of parameters. The result in parameter settings is quite similar, while the run time for randomized search is drastically lower.

The performance is may slightly worse for the randomized search and is likely due to a noise effect and would not carry over to a held-out test set.

I selected randomized search cross validation as it takes less time.

K-nearest neighbours	Decision Tree	Support Vector Machine	Logistic Regression
weights='distance'	Splitter='best'	Kernel='rbf'	Solver='liblinear'
p = 1	max_features'='sqrt'	Gamma='scale'	Penalty='l2'
n_neighbors = 60	max_depth=90	Degree=5	C=5
leaf_size = 10	Criterion='entropy'	C=10	
Algorithm = 'kd_tree'			

3. Results

Based on the results, all models are ranked as following. Refer to next 3 section as details.

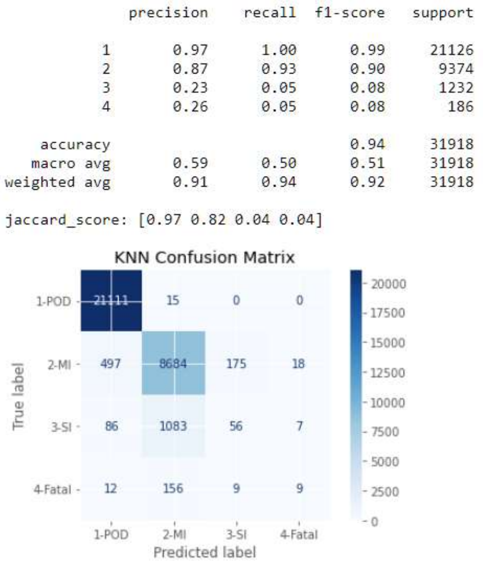
Rank1	Rank2	Rank3	Rank4
Up-sampling Data	Down-sampling Data	Baseline	Original Data
Decision Tree	Support Vector Machine	K-nearest neighbours	Logistic Regression

Average Accuracy

Dataset	Quantity	K-nearest neighbours	Decision Tree	Support Vector Machine	Logistic Regression	Remark
Baseline	8000	0.73	0.73	0.76	0.68	
Original data	127672	0.94	0.94	0.95	0.96	Poor accuracy for Severity 3&4
Down-sampling data	149200	0.82	0.84	0.84	0.68	
Up-sampling data	339100	0.85	0.86	0.85	0.67	Best

3.1. Result with Original Data(unbalanced)

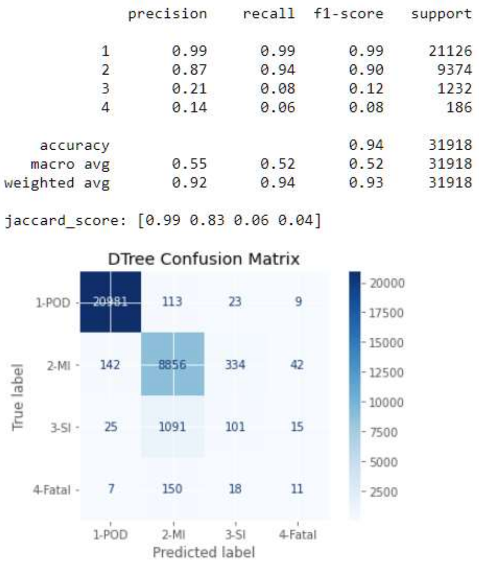
Original Data only has limited samples for Severity 3&4, Model had not enough training on it. Even though got high accuracy on Severity 1&2, Its results for reference only. If only predict Severity 1&2, could be a pretty result.



```
              precision    recall  f1-score   support

           1       0.97      1.00      0.99     21126
           2       0.87      0.93      0.90      9374
           3       0.23      0.05      0.08      1232
           4       0.26      0.05      0.08       186

    accuracy                           0.94     31918
   macro avg       0.59      0.50      0.51     31918
weighted avg       0.91      0.94      0.92     31918

jaccard_score: [0.97 0.82 0.04 0.04]
```

KNN Confusion Matrix

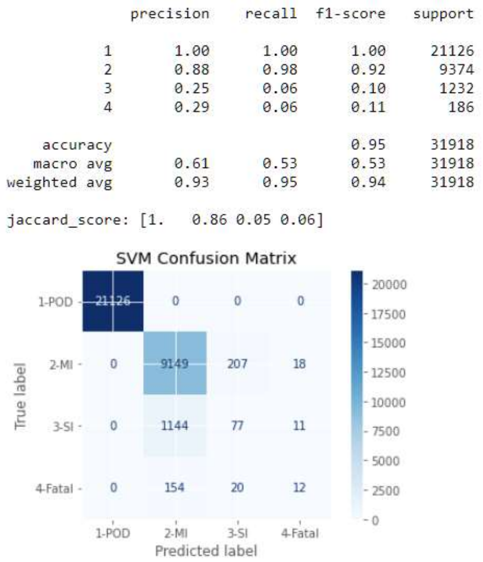
True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21111	15	0	0
2-MI	497	8684	175	18
3-SI	86	1083	56	7
4-Fatal	12	156	9	9



```
              precision    recall  f1-score   support

           1       0.99      0.99      0.99     21126
           2       0.87      0.94      0.90      9374
           3       0.21      0.08      0.12      1232
           4       0.14      0.06      0.08       186

    accuracy                           0.94     31918
   macro avg       0.55      0.52      0.52     31918
weighted avg       0.92      0.94      0.93     31918

jaccard_score: [0.99 0.83 0.06 0.04]
```

DTree Confusion Matrix

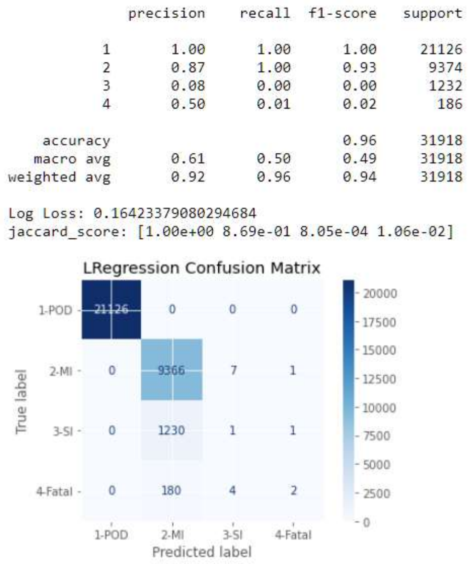
True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	20981	113	23	9
2-MI	142	8856	334	42
3-SI	25	1091	101	15
4-Fatal	7	150	18	11



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     21126
           2       0.88      0.98      0.92      9374
           3       0.25      0.06      0.10      1232
           4       0.29      0.06      0.11       186

    accuracy                           0.95     31918
   macro avg       0.61      0.53      0.53     31918
weighted avg       0.93      0.95      0.94     31918

jaccard_score: [1.   0.86 0.05 0.06]
```

SVM Confusion Matrix

True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21126	0	0	0
2-MI	0	9149	207	18
3-SI	0	1144	77	11
4-Fatal	0	154	20	12

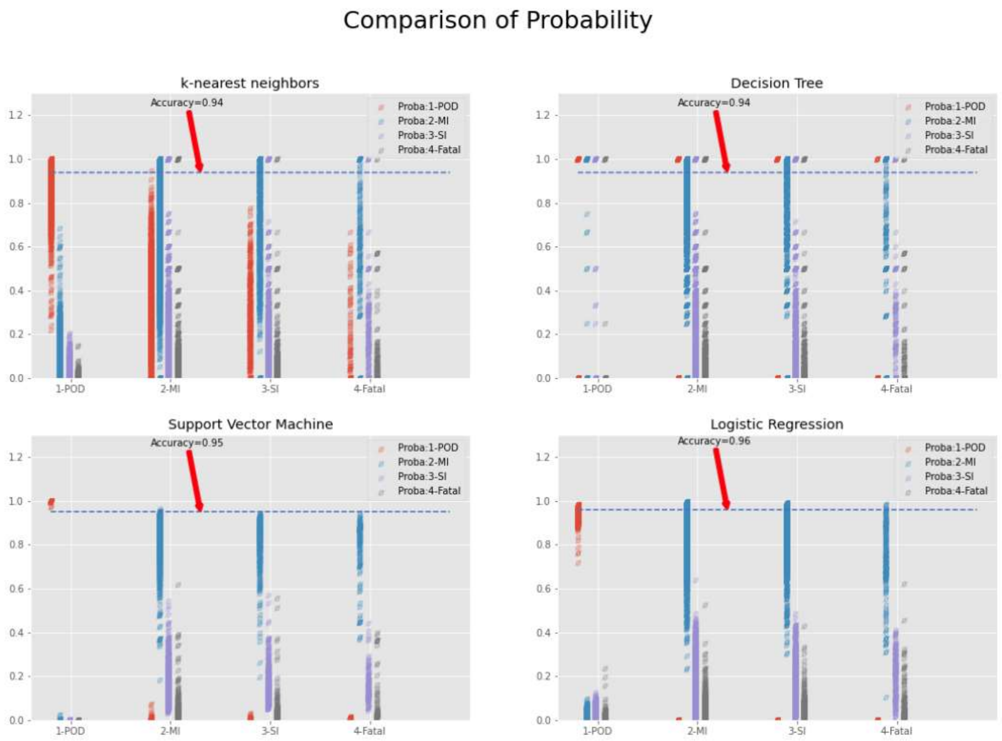


```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     21126
           2       0.87      1.00      0.93      9374
           3       0.08      0.00      0.00      1232
           4       0.50      0.01      0.02       186

    accuracy                           0.96     31918
   macro avg       0.61      0.50      0.49     31918
weighted avg       0.92      0.96      0.94     31918

Log Loss: 0.16423379080294684
jaccard_score: [1.00e+00 8.69e-01 8.05e-04 1.06e-02]
```

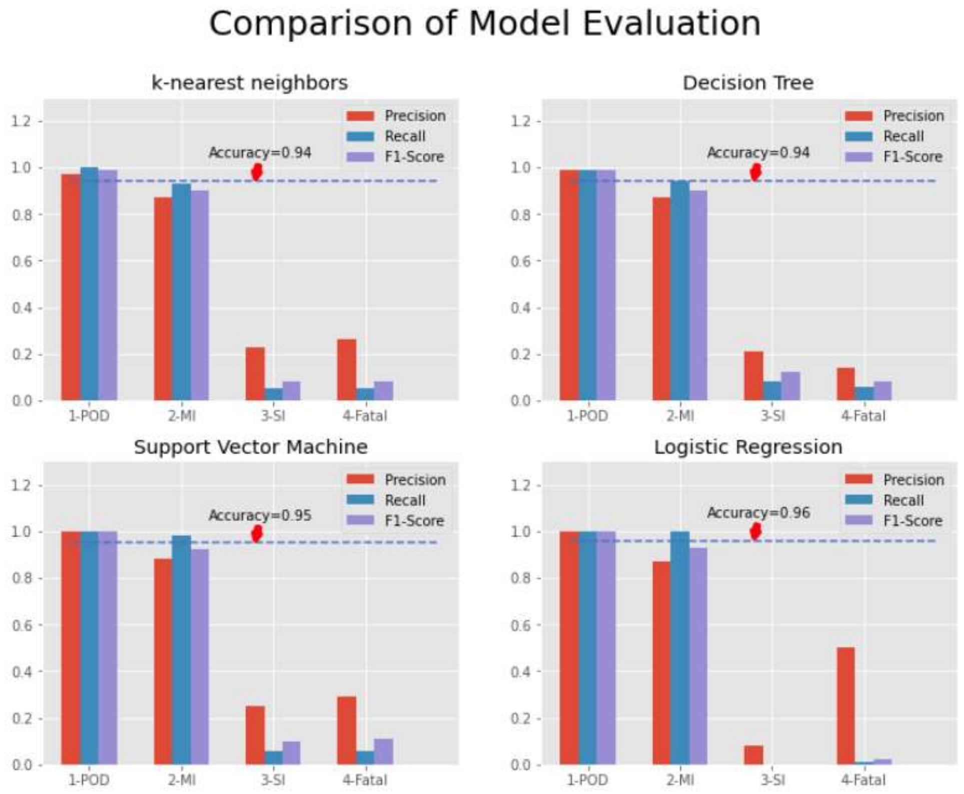
LRegression Confusion Matrix

True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21126	0	0	0
2-MI	0	9366	7	1
3-SI	0	1230	1	1
4-Fatal	0	180	4	2

Probability Scatter shows Severity 3&4 were predicted as Severity 2.



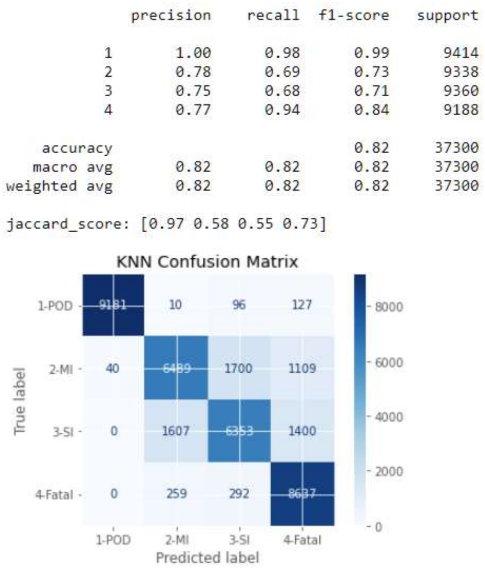
Note: The point in Probability scatter was shifted 0.09 per category on X Axes, to avoid overlap.



3.2. Result with Down-sampling Data

Regarding Down-sampling data, actually, Severity 1's is down-sampling, Severity 3&4's is up-sampling.

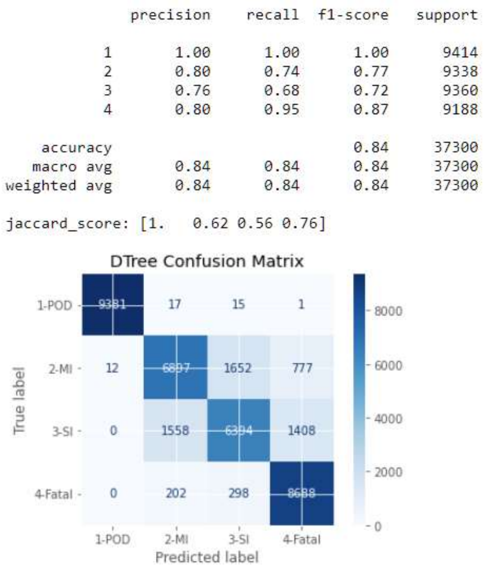
Using Down-sampling data, overall result is comparable and balanced. Decision Tree is the best model. Support Vector Machine has same accuracy – 84%.



```
              precision    recall  f1-score   support

           1       1.00      0.98      0.99      9414
           2       0.78      0.69      0.73      9338
           3       0.75      0.68      0.71      9360
           4       0.77      0.94      0.84      9188

    accuracy                           0.82     37300
   macro avg       0.82      0.82      0.82     37300
weighted avg       0.82      0.82      0.82     37300

jaccard_score: [0.97 0.58 0.55 0.73]
```

KNN Confusion Matrix

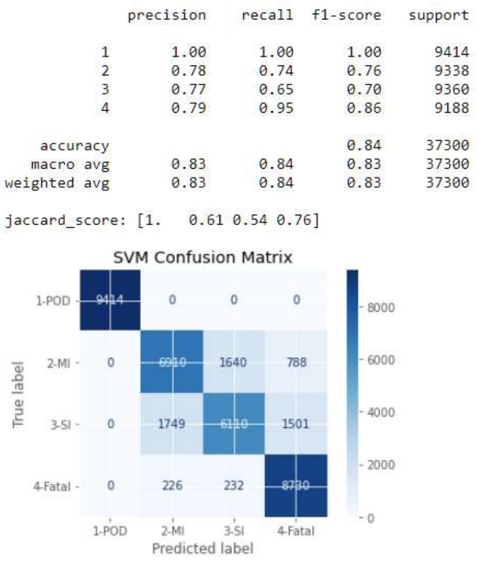
True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	9181	10	96	127
2-MI	40	6489	1700	1109
3-SI	0	1607	6353	1400
4-Fatal	0	259	292	8637



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      9414
           2       0.80      0.74      0.77      9338
           3       0.76      0.68      0.72      9360
           4       0.80      0.95      0.87      9188

    accuracy                           0.84     37300
   macro avg       0.84      0.84      0.84     37300
weighted avg       0.84      0.84      0.84     37300

jaccard_score: [1.   0.62 0.56 0.76]
```

DTree Confusion Matrix

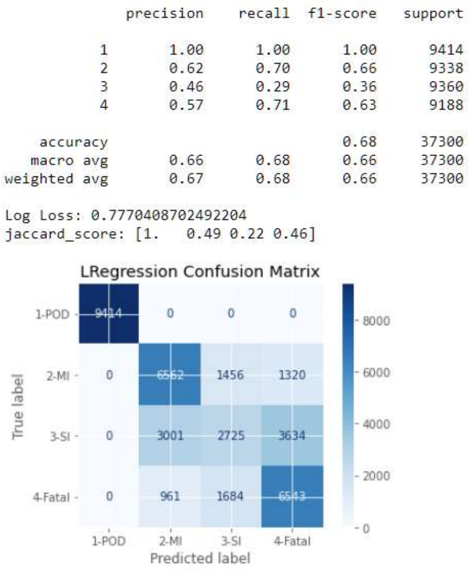
True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	9381	17	15	1
2-MI	12	6897	1652	777
3-SI	0	1558	6394	1408
4-Fatal	0	202	298	8688



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      9414
           2       0.78      0.74      0.76      9338
           3       0.77      0.65      0.70      9360
           4       0.79      0.95      0.86      9188

    accuracy                           0.84     37300
   macro avg       0.83      0.84      0.83     37300
weighted avg       0.83      0.84      0.83     37300

jaccard_score: [1.   0.61 0.54 0.76]
```

SVM Confusion Matrix

True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	9414	0	0	0
2-MI	0	6910	1640	788
3-SI	0	1749	6110	1501
4-Fatal	0	226	232	8730

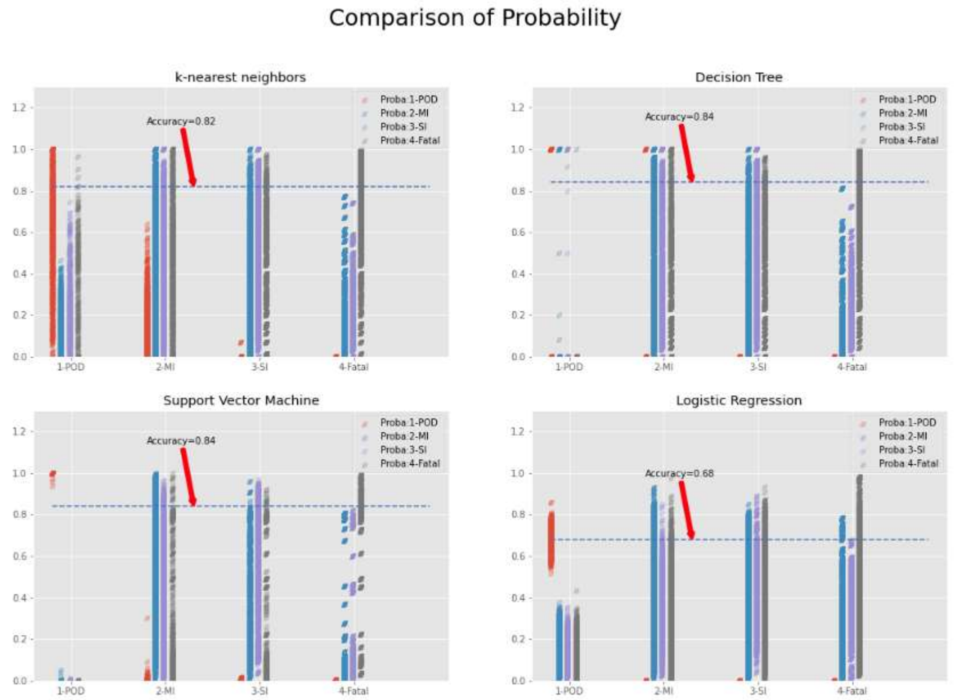


```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      9414
           2       0.62      0.70      0.66      9338
           3       0.46      0.29      0.36      9360
           4       0.57      0.71      0.63      9188

    accuracy                           0.68     37300
   macro avg       0.66      0.68      0.66     37300
weighted avg       0.67      0.68      0.66     37300

Log Loss: 0.7770408702492204
jaccard_score: [1.   0.49 0.22 0.46]
```

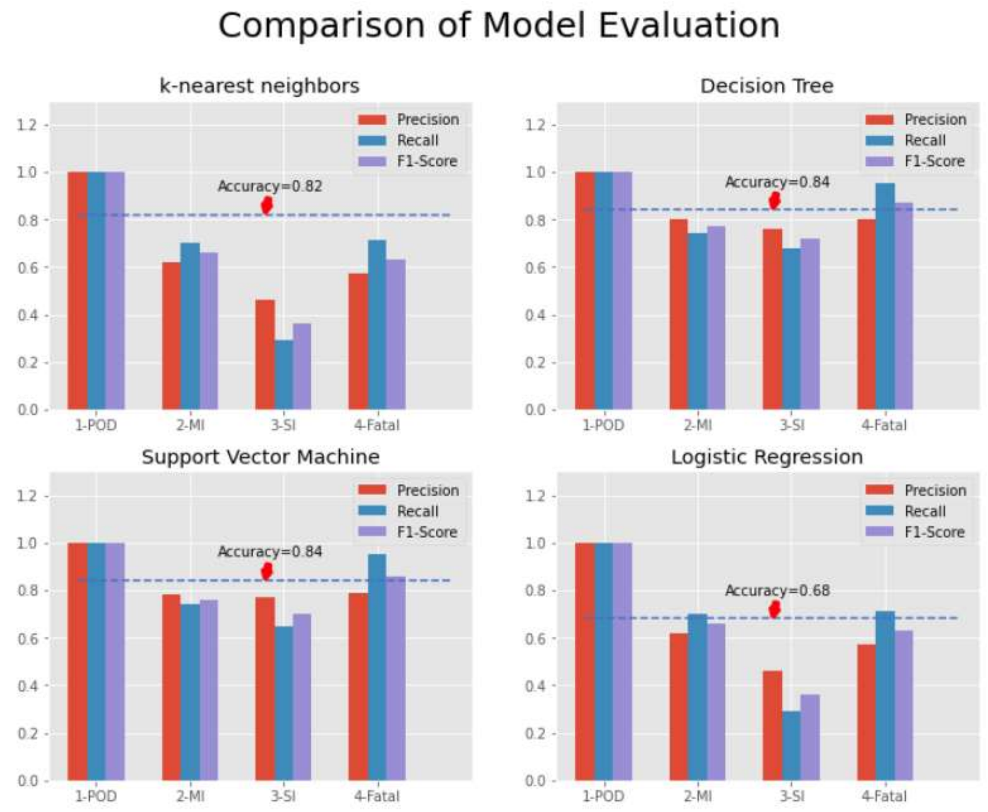
LRegression Confusion Matrix

True label \ Predicted label	1-POD	2-MI	3-SI	4-Fatal
1-POD	9414	0	0	0
2-MI	0	6562	1456	1320
3-SI	0	3001	2725	3634
4-Fatal	0	961	1684	6543

The distribution of probability for Severrity 2&3 is similar – Probability of Severity 2 is higher. It cause low predicting accuracy on Severity 3.

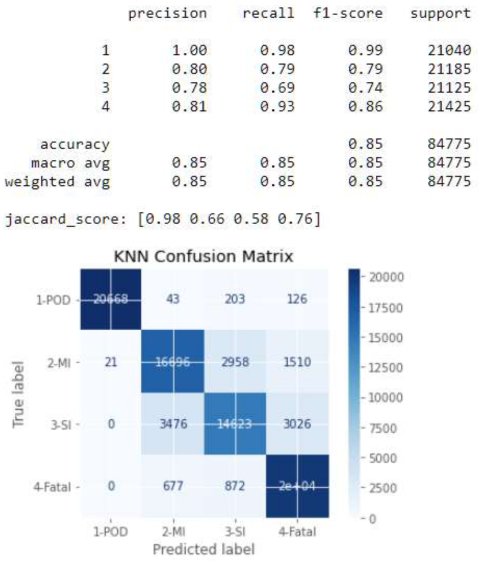


Note: The point in Probability scatter was shifted 0.09 per category on X Axes, to avoid overlap.



3.3. Result with Up-sampling Data

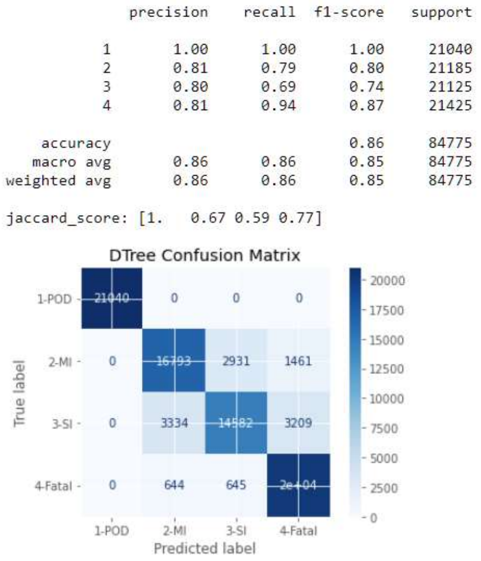
Using Up-sampling data, overall result is most comparable and balanced. Same as Down-sampling data, Decision Tree also is the best model – 86% Accuracy. Support Vector Machine reached 85% accuracy.



```
              precision    recall  f1-score   support

           1       1.00      0.98      0.99     21040
           2       0.80      0.79      0.79     21185
           3       0.78      0.69      0.74     21125
           4       0.81      0.93      0.86     21425

    accuracy                           0.85     84775
   macro avg       0.85      0.85      0.85     84775
weighted avg       0.85      0.85      0.85     84775

jaccard_score: [0.98 0.66 0.58 0.76]
```

KNN Confusion Matrix

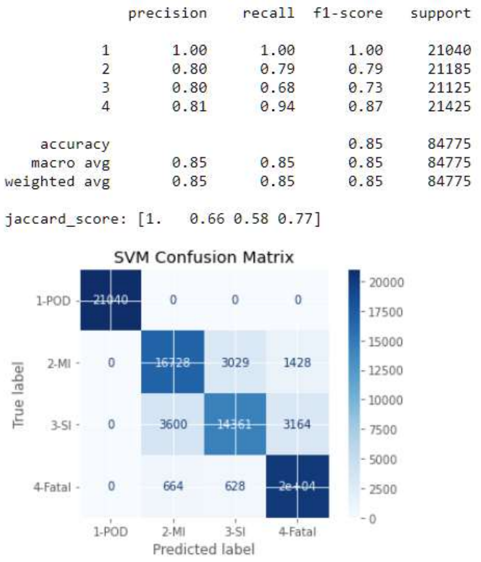
True label	1-POD	2-MI	3-SI	4-Fatal
1-POD	20668	43	203	126
2-MI	21	16696	2958	1510
3-SI	0	3476	14623	3026
4-Fatal	0	677	872	2e+04



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     21040
           2       0.81      0.79      0.80     21185
           3       0.80      0.69      0.74     21125
           4       0.81      0.94      0.87     21425

    accuracy                           0.86     84775
   macro avg       0.86      0.86      0.85     84775
weighted avg       0.86      0.86      0.85     84775

jaccard_score: [1.   0.67 0.59 0.77]
```

DTree Confusion Matrix

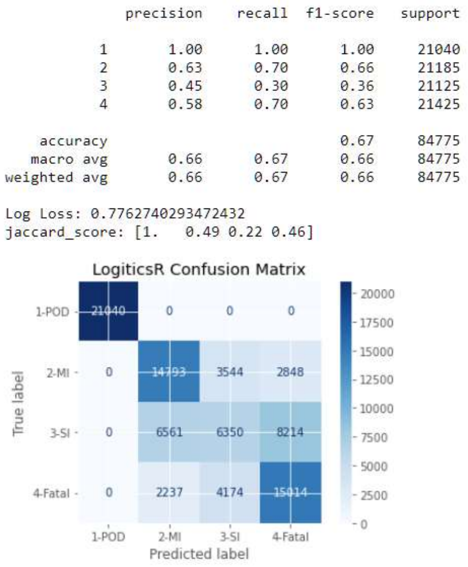
True label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21040	0	0	0
2-MI	0	16793	2931	1461
3-SI	0	3334	14582	3209
4-Fatal	0	644	645	2e+04



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     21040
           2       0.80      0.79      0.79     21185
           3       0.80      0.68      0.73     21125
           4       0.81      0.94      0.87     21425

    accuracy                           0.85     84775
   macro avg       0.85      0.85      0.85     84775
weighted avg       0.85      0.85      0.85     84775

jaccard_score: [1.   0.66 0.58 0.77]
```

SVM Confusion Matrix

True label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21040	0	0	0
2-MI	0	16728	3029	1428
3-SI	0	3600	14361	3164
4-Fatal	0	664	628	2e+04

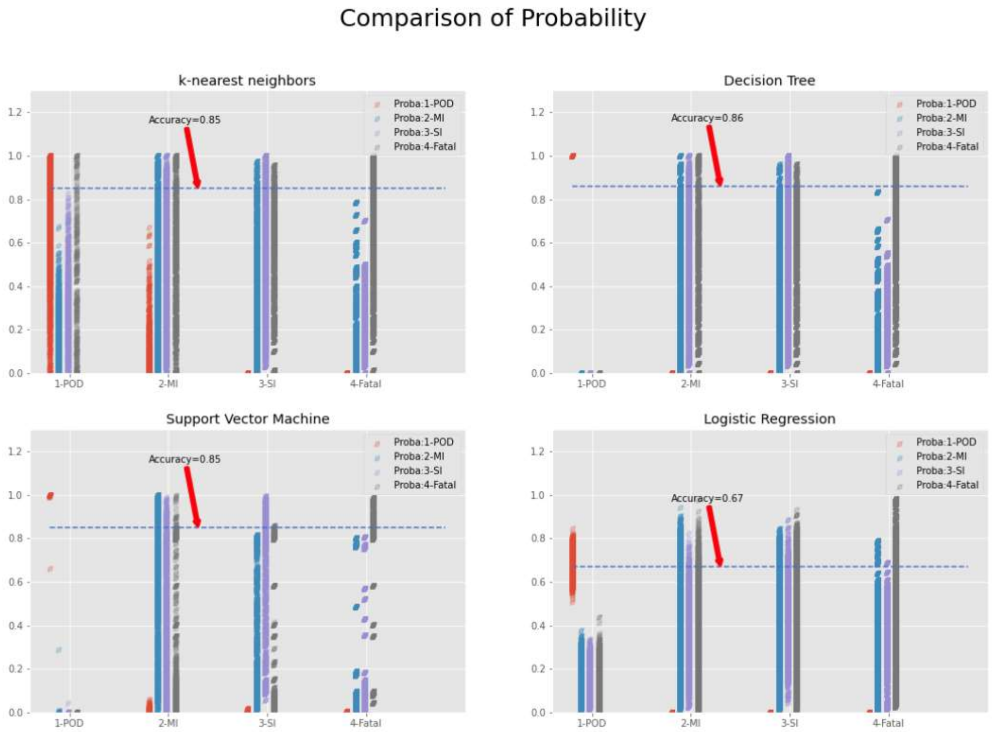


```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     21040
           2       0.63      0.70      0.66     21185
           3       0.45      0.30      0.36     21125
           4       0.58      0.70      0.63     21425

    accuracy                           0.67     84775
   macro avg       0.66      0.67      0.66     84775
weighted avg       0.66      0.67      0.66     84775

Log Loss: 0.7762740293472432
jaccard_score: [1.   0.49 0.22 0.46]
```

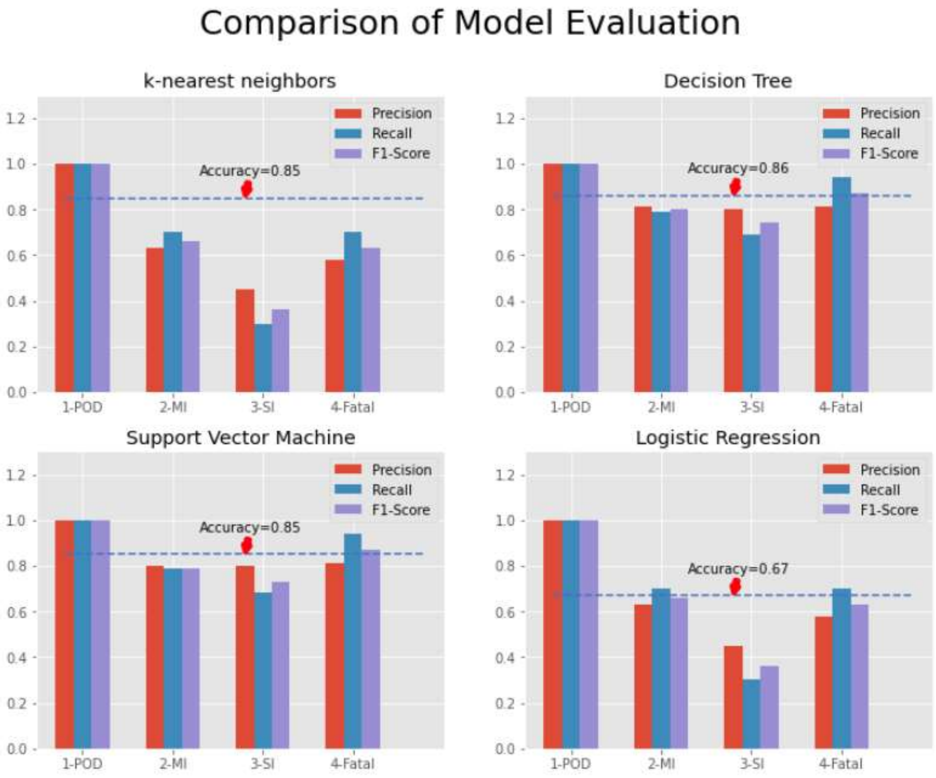
LogiticsR Confusion Matrix

True label	1-POD	2-MI	3-SI	4-Fatal
1-POD	21040	0	0	0
2-MI	0	14793	3544	2848
3-SI	0	6561	6350	8214
4-Fatal	0	2237	4174	15014

Same as Down-sampling data, the distribution of probability for Severrity 2&3 is similar – Probability of Severity 2 is higher. It cause low predicting accuracy on Severity 3.



Note: The point in Probability scatter was shifted 0.09 per category on X Axes, to avoid overlap.



4. Conclusion & Discussion

Decision Tree model have the best performance for multiple classification with Up-sampling data; Its 86% average accuracy is a quality result for this project; The result of Logistic Regression and K-Nearest Neighbours model are acceptable; Logistic Regression is the worst one, which solve the problem based on binary classification algorithm.

And all models have the similar trend compared with Baseline, Down-sampling data and Up-sampling data. More sampling for training, more accurate.

Every single feature has the weak correlation, even though checked the multi-dimension correlation using PCA, but it was invisible for features.

Down-sampling improved 11% from baseline, using Up-sampling got 2% more improvement. Following items can be applied for further optimization.

- Measure and visualize Multi-dimension correlation
- Seeking other algorithm for multiple classification.
- Deep optimization for model parameters.
- Reduce the running time for Cross Validation and predicting.

5. References

- IBM Data Science Professional Certificate Course
- Python Lib
  - SciKit-Learn Lib
  - Pandas Lib
  - Matplotlib

6. Appendix

Data Source and Metadata

https://data.sa.gov.au/data/dataset/road-crash-data