PREDICTION OF ACCIDENT SEVERITY USING MACHINE LEARNING MODELS

A case study to predict the severity of an accident based on external factors such as road condition, weather condition, etc, using supervised machine learning models.

Capstone Project – IBM Data Science Coursera

By: Parthiv Lakshman

CHAPTER 1

Introduction and Business understanding

With increasing traffic on roads each year, there is an increase in the number of accidents which occur each day across the globe. The role of human factor is a major influence in accidents, but so is the importance of external phenomenon such as road conditions, weather conditions, etc.

There is a need to have a system in place which would predict the severity of a probable accident based on these external phenomena. This would be a great tool for drivers as it would alert them of impending danger. This would decrease the probability of accidents by a certain margin.

Such a system would make use of the real time incoming data from various sources such as weather station, satellites, etc. and use machine learning models to predict the severity of an accident that might happen if the driver is not cautious. This would also contribute to efficient traffic flow, reducing the amount of travel time, and in turn reduce pollution to an extent.

CHAPTER 2

Understanding the Data

The dataset used is the accident severity data of Seattle from the SDOT Traffic Management Division, obtained from Coursera. The dataset contains a parameter “Severity” which is out target variable or predictor. This describes how severe the accident is or would be.

The current dataset has only two levels of severity: Level 1 – Property damage, Level 2 – Injury. We would use the external factors of road conditions, weather conditions and lighting conditions as dependent variables for the prediction of our accident severity.

As shown in Fig 1, the dataset consists of 194673 rows and 38 columns. The dataset would be reduced to the necessary dependent variables and wrangled before using the machine learning models.

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGH
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Dayli
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Lights
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Dayli
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Dayli
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Dayli

Fig 1 : Original dataset

CHAPTER 3

Data Preparation

In this section, the uploaded data is put through filtering and wrangling procedures in order to obtain the final dataset to be fed to the machine learning models. A new dataset is created from the original dataset which has only those columns that are necessary for the model, reducing the number of columns to 6, as shown in Fig 2.

```
df = pd.DataFrame()
df = df1[['SEVERITYCODE','ROADCOND','LIGHTCOND','WEATHER','PERSONCOUNT','VEHCOUNT']].copy()
df.shape
```

```
(194673, 6)
```

Fig 2 : Reduced dataset

After this, all rown that have missing information, such as “Unknown” and “Other” as conditions, have been removed, since these do not state any significant condition. These values would contribute to inaccurate prediction of severity.

After this, the predictor variables, which are categorical values, have been converted to numerical values using one-hot encoding (shown in Fig 3). The categorical values have been studied and the ones which have less than 1% influence on the outcome, have been removed or dropped (as shown in Fig 4). For example, for Light conditions, values after “Dark-No street lights” have been dropped.

	SC	Persons	Vehicles	Dark	Dawn	Daylight	Dusk	Dry	Ice	Snow/Slush	Wet	Clear	Fog/Smog/Smoke	Overcast	Raining	Snowing
0	2	2	2	0	0	1	0	0	0	0	1	0	0	1	0	0
1	1	2	2	1	0	0	0	0	0	0	1	0	0	0	1	0
2	1	4	3	0	0	1	0	1	0	0	0	0	0	1	0	0
3	1	3	3	0	0	1	0	1	0	0	0	1	0	0	0	0
4	2	2	2	0	0	1	0	0	0	0	1	0	0	0	1	0

Fig 3 : One-hot Encoding

```
SEVERITYCODE  LIGHTCOND
1             Daylight                      0.586028
              Dark - Street Lights On       0.257030
              Unknown                       0.097187
              Dusk                          0.029893
              Dawn                          0.012673
              Dark - No Street Lights       0.009086
              Dark - Street Lights Off      0.006669
              Other                         0.001382
              Dark - Unknown Lighting       0.000053
2             Daylight                      0.675050
              Dark - Street Lights On       0.253512
              Dusk                          0.034047
              Dawn                          0.014431
              Unknown                       0.010596
              Dark - No Street Lights       0.005850
              Dark - Street Lights Off      0.005534
              Other                         0.000911
              Dark - Unknown Lighting       0.000070
Name: LIGHTCOND, dtype: float64
```

Fig 4 : Statistical analysis of Light condition values contributing to the accident severity

The columns have also been renamed, and the final dataset consists of 169957 rows and 16 columns.

CHAPTER 4

Modeling – Classification Algorithms

In this phase, various methods and machine learning algorithms have been built using supervised machine learning techniques. Decision trees, K-Nearest neighbours, Support Vector Machine and Logistic Regression models have been used. The same dataset has ben applied for all the different models to see how the predictions differ from model to model, and also as a basis for evaluating the models.

The training and test sets have been split by 70 / 30 percent split.

1. Decision Tree

Decision trees are built by splitting dataset into distinct nodes, where one node contains the data category. The categorical valuesa re now the binary classifiers, which are used to predict the severity category, which are nodes of the decision tree.

Only the first 5 results are presented below in Fig 5. Along with it, the accuracy of both training and test sets have been shown.

```
Initial classification :    [1 1 1 1 1]
Predicted classification :  [1 1 1 1 1]
Training set accuracy :  0.7219023443081811
Test set accuracy :  0.7273083862869695
```

Fig 5 : Decision tree results

2. K-Nearest neighbours

This is a classification algorithm that takes a set of labelled points and uses this labelling information to label other near points. This algorithm classifies the labels based on their similarity to nearby clusters.

First, to find the best value of k (number of nearest neighbours), i.e. the one with the highest accuracy; the prediction accuracy has been evaluated with the dataset as shown in Fig 6; which gave k=4 as the best value to use. Values from 1 to 9 have been used for the evaluation. The model then uses the value 4.

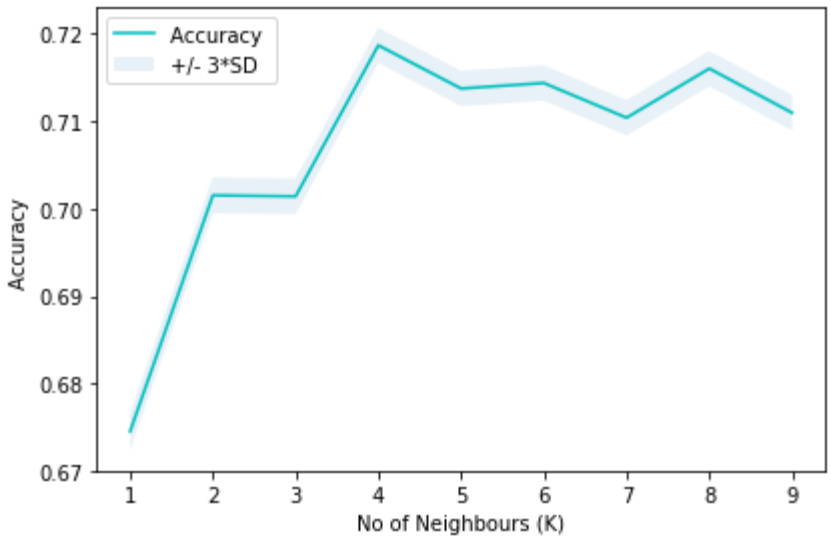

Fig 6 : Finding best k

```
Initial classification :    [1 1 1 1 1]
Predicted classification :  [1 2 1 2 1]
Training set accuracy :  0.715354420059007
Test set accuracy :  0.7186004550090217
```

Fig 7 : Results of KNN classifier

This shows the accuracy of KNN classifier is slightly less than decision trees for our dataset.

3. Support Vector Machine

This is a supervised algorithm that can classify cases by determining a separator. This is also the model which has the highest computation time. The results are shown in Fig 8.

```
Initial classification :    [1 1 1 1 1]
Predicted classification :  [1 1 1 1 1]
Training set accuracy :  0.7216417722263783
Test set accuracy :  0.7256413273711462
```

Fig 8 : Results of SVM classifier

4. Logistic Regression

This is a statistical machine learning algorithm for classifying records of a dataset based on the values of input fields. The results are shown in Fig 9.

```
Initial classification :    [1 1 1 1 1]
Predicted classification :  [1 1 1 1 1]
Training set accuracy :  0.6740075145626172
Test set accuracy :  0.6801992625715855
```

Fig 9 : Results of Logistic regression classifier

This model has the lowest accuracy compared to the previous models.

CHAPTER 5

Model Evaluation

The Jaccard index, F1-Score and Log loss have been used to evaluate the models. The log loss is used only for the logistic regression. Later a table shows the comparison of the accuracies of the various models.

1. Jaccard Index

```
print('Jaccard similarity for Decision tree       : ', JCS(y_test,tree_pred))
print('Jaccard similarity for KNN                 : ', JCS(y_test,knn_pred))
print('Jaccard similarity for SVM                 : ', JCS(y_test,svm_pred2))
print('Jaccard similarity for Logistic Regression : ', JCS(y_test,lr_pred))
```

```
Jaccard similarity for Decision tree       :  0.7273083862869695
Jaccard similarity for KNN                 :  0.7186004550090217
Jaccard similarity for SVM                 :  0.7256413273711462
Jaccard similarity for Logistic Regression :  0.6801992625715855
```

2. F1-Score

```
F1 Score for Decision tree      :
                precision    recall  f1-score   support

           1       0.72      0.97      0.82    114274
           2       0.77      0.22      0.34     55683

   micro avg       0.72      0.72      0.72    169957
   macro avg       0.75      0.59      0.58    169957
weighted avg       0.74      0.72      0.67    169957

F1 Score for KNN                :
                precision    recall  f1-score   support

           1       0.70      0.96      0.81    114274
           2       0.68      0.16      0.26     55683

   micro avg       0.70      0.70      0.70    169957
   macro avg       0.69      0.56      0.54    169957
weighted avg       0.70      0.70      0.63    169957

F1 Score for Logistic Regression :
                precision    recall  f1-score   support

           1       0.68      0.97      0.80    114274
           2       0.54      0.08      0.14     55683

   micro avg       0.68      0.68      0.68    169957
   macro avg       0.61      0.52      0.47    169957
weighted avg       0.63      0.68      0.58    169957

F1 Score for SVM                :
                precision    recall  f1-score   support

           1       0.72      0.96      0.82    114274
           2       0.76      0.23      0.35     55683

   micro avg       0.72      0.72      0.72    169957
   macro avg       0.74      0.60      0.59    169957
weighted avg       0.73      0.72      0.67    169957
```

3. Log loss

```
print('Log loss of Logistic Regression : ', LL(Y_t, lr_pred))
Log loss of Logistic Regression :  23.22337713038452
```

Comparison of model accuraccy:

Method	Jaccard Index	F1 Score	Log loss
Decision Tree	0.73	0.67	X
KNN	0.72	0.63	X
SVM	0.73	0.58	X
Logistic regression	0.68	0.67	23

As per Jaccard score, Decision tree and SVM are the best models. According to F1 score, Decision tree and Logistic regression are best.