Add a waring feature to Navigation Apps

Background

•Given the dataset of accident records in Seattle, Washington district.

•A good way of utilizing it to benefit the local residents is to provide reference warning messages.

•The behavioral habit nowadays is to use navigation apps for direction. Hence navigation apps like Google Maps would be the best platforms to provide these messages.

Data Acquisition and Cleaning

- The "Data-Collisions.csv" file which contains the accident data is provided by the course
- The record rows which are lack of selected variables' information (N/A) were dropped
- The record rows which contains unhelpful information(Unknown or Other) were dropped
- The record rows which their INCDTTM column do not contain the formatted information were dropped

Data Acquisition and Cleaning

- Combining categories of variables(WEATHER, ROADCOND, LIGHTCOND)
- Change the combined categorical variables into Boolean numbers(WEATHER, ROADCOND)
- Change the the combined categorical variables into numbers using Label Encoder(LIGHTCOND)
- Parse the time string in order to get the timestamp representing time of day

Preprocessing data before trained

	X	Y	VEHCOUNT	INCDTTM	SEVERITYCODE	Precipitation	Slippery	LIGHTCOND
0	-122.323148	47.703140	2	24840.0	2	0	1	2
1	-122.347294	47.647172	2	39300.0	1	1	1	0
2	-122.334540	47.607871	3	8400.0	1	0	0	2
3	-122.334803	47.604803	3	5160.0	1	0	0	2
4	-122.306426	47.545739	2	240.0	2	1	1	2

Normalize train/test set separately after splitting

If normalize the whole dataset before splitting, the data that will be used as test set were already get affected by the data in training set when normalize.

Normalize training data

```
[170]: from sklearn import preprocessing
       normalized_train_X= preprocessing.StandardScaler().fit(X_train).transform(X_train)
       normalized_train_X[0:5]
```

```
t[170]: array([[-0.09885777,  0.53794131,  0.05269116, -1.47744574, -0.49857387,
                -0.62934785,  0.54896495],
               [-0.90574397,  1.26741473,  0.05269116, -0.70118405, -0.49857387,
                -0.62934785,  0.54896495],
               [-0.82366081,  0.52569259,  3.50592418, -0.5847448 , -0.49857387,
                -0.62934785,  0.54896495],
               [-0.95118484,  0.10493691,  0.05269116, -0.60137898, -0.49857387,
                 1.5889464 ,  0.54896495],
               [ 0.04200691, -0.8267832 ,  0.05269116, -0.64573679, -0.49857387,
                -0.62934785,  0.54896495]])
```

Normalize testing data

```
[171]: normalized_test_X = preprocessing.StandardScaler().fit(X_test).transform(X_test)
       normalized_test_X
```

```
t[171]: array([[-1.55770701, -1.3110814 ,  0.04409666, ...,  2.01399401,
                 1.60480383, -1.56361257],
               [ 0.24069887,  2.02164248,  0.04409666, ..., -0.49652581,
                 1.60480383, -1.56361257],
               [ 0.58751476, -1.24974848,  0.04409666, ..., -0.49652581,
                -0.62312912,  0.54692996],
               ...,
               [ 0.48184422,  1.00352869,  1.77623281, ..., -0.49652581,
                -0.62312912,  0.54692996],
               [-2.62010016, -0.74551183,  0.04409666, ..., -0.49652581,
```

Conclusion

- This model can predict the severity of road accidents so that it can help the residents reduce chances of getting fatal or severe injury.
- The navigation apps should be able to access other APIs so that they can get weather info etc. to guarantee that all required variables other than X/Y coordinates can be provided when using this model to provide predictions of severity of accidents.
- To evaluate the model performance in real life, not only the prediction accuracy counts, additional factors should be taken into considerate. (Personal situation and decision)