Capstone Project – Car accident severity

Elida Axundzada

November 14, 2020

Introduction

As a driving school, we would like to create a mobile application for young drivers. The purpose of this application is to increase the safety on the road based on various criteria. The young drivers could verify the risk of accident before hitting the road. To create our application, we should develop an accurate prediction model.

We will use various variables to train our machine learning model. Indeed, we should verify the correlation between the car accident severity and the chosen variables before making our application available.

Data acquisition

Data source

The city of Seattle has a dataset on all collisions collected since 2004 to present, as provided by the Seattle Police District and recorded by Traffic Records. It has 194,673 observations with 37 attributes. It has been downloaded as a csv file – DataCollisions.csv from this link: https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv

The description of the attributes is below:

Attribute	**Data Type**	**Data Length**	**Description**
ADDRTYPE	Text	12	Collision address type: Alley, Block, Intersection
COLDETKEY	Long		Secondary key for the incident.
COLLISIONTYPE	Text	300	Collision type.
CROSSWALKKEY	Long		A key for the crosswalk at which the collision occurred.
EXCEPTRSNCODE	Text	10	
EXCEPTRSNDESC	Text	300	
FATALITIES	Double		The number of fatalities in the collision. This is entered by the state.
HITPARKEDCAR	Text	1	Whether or not the collision involved hitting a parked car. (Y/N)
INATTENTIONIND	Text	1	Whether or not collision was due to inattention. (Y/N)
INCDATE	Date		The date of the incident.

Attribute	Data Type	Data Length	Description
INCDTTM	Text	30	The date and time of the incident.
INCKEY	Long		A unique key for the incident.
INJURIES	Double		The number of total injuries in the collision. This is entered by the state.
INTKEY	Double		Key that corresponds to the intersection associated with a collision.
JUNCTIONTYPE	Text	300	Category of junction at which collision took place.
LIGHTCOND	Text	300	The light conditions during the collision.
LOCATION	Text	255	Description of the general location of the collision.
OBJECTID	ObjectID		ESRI unique identifier.
PEDCOUNT	Double		The number of pedestrians involved in the collision. This is entered by the state.
PEDCYLCOUNT	Double		The number of bicycles involved in the collision. This is entered by the state.
PEDROWNOTGRNT	Text	1	Whether or not the pedestrian right of way was not granted. (Y/N)
PERSONCOUNT	Double		The total number of people involved in the collision.
ROADCOND	Text	300	The condition of the road during the collision.
SDOT_COLCODE	Text	10	A code given to the collision by SDOT.
SDOT_COLDESC	Text	300	A description of the collision corresponding to the collision code.
SDOTCOLNUM	Text	10	A number given to the collision by SDOT.
SEGLANEKEY	Long		A key for the lane segment in which the collision occurred.
SERIOUSINJURIES	Double		The number of serious injuries in the collision. This is entered by the state.

Attribute	Data Type	Data Length	Description
SEVERITYCODE	Text	100	A code that corresponds to the severity of the collision: [3 – fatality, 2b - serious injury, 2 – injury, 1 – prop, damage, 0 – unknown]
SEVERITYDESC	Text		A detailed description of the severity of the collision.
SHAPE	Geometry		ESRI geometry field.
SPEEDING	Text	1	Whether or not speeding was a factor in the collision. (Y/N)
ST_COLCODE	Text	10	A code provided by the state that describes the collision. For more information about these codes, please see the State Collision Code Dictionary.
ST_COLDESC	Text	300	A description that corresponds to the state's coding designation.
UNDERINFL	Text	10	Whether or not a driver involved was under the influence of drugs or alcohol.
VEHCOUNT	Double		The number of vehicles involved in the collision. This is entered by the state.
WEATHER	Text	300	A description of the weather conditions during the time of the collision.

Feature selection

For our analyse, to help in predicting the possibility and severity of an accident/collision {SEVERITYCODE}, the following attributes will be used:

- {ADDRTYPE}: collision address type as source of location information
- {WEATHER}: weather during the time of collision
- {LIGHTCOND}: light conditions during the collision
- {ROADCOND}: road condition during the collision

Methodology

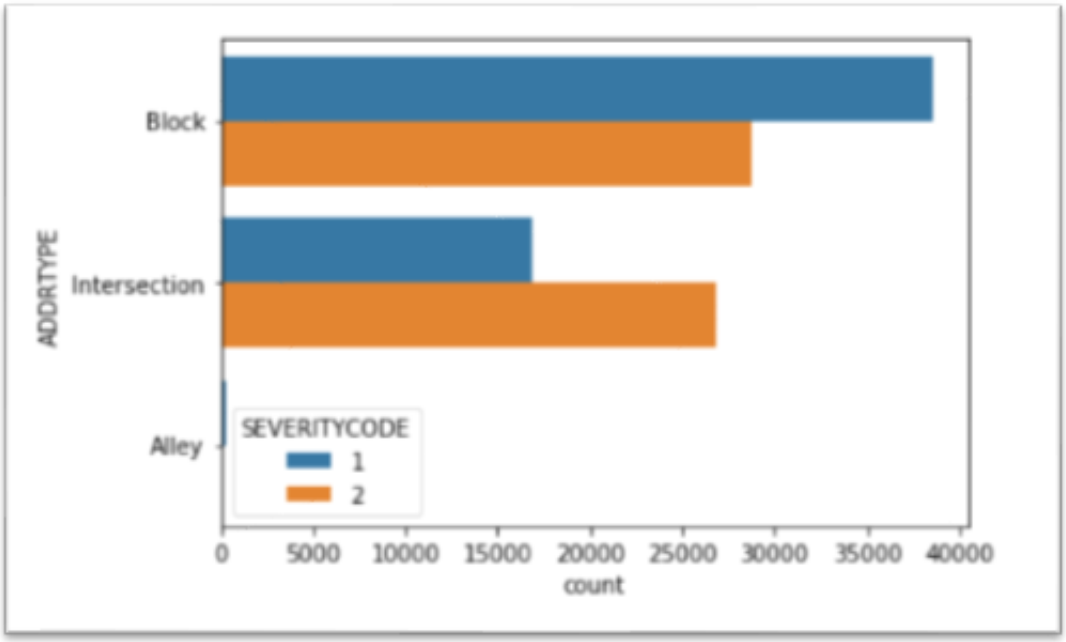
Data cleaning

For the chosen features, there are NaN values, they were dropped for better processing of sklearn. Moreover, there are unknown data on WEATHER, LIGHTCOND and ROADCOND, and these were dropped as well.

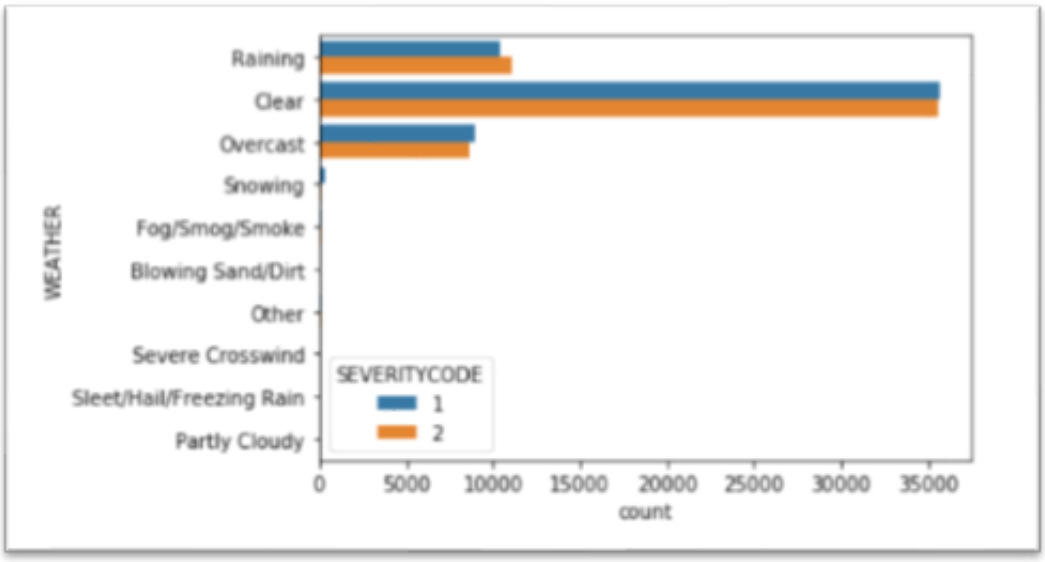
Exploratory analysis

Separate exploratory analysis is done on the independent variables.

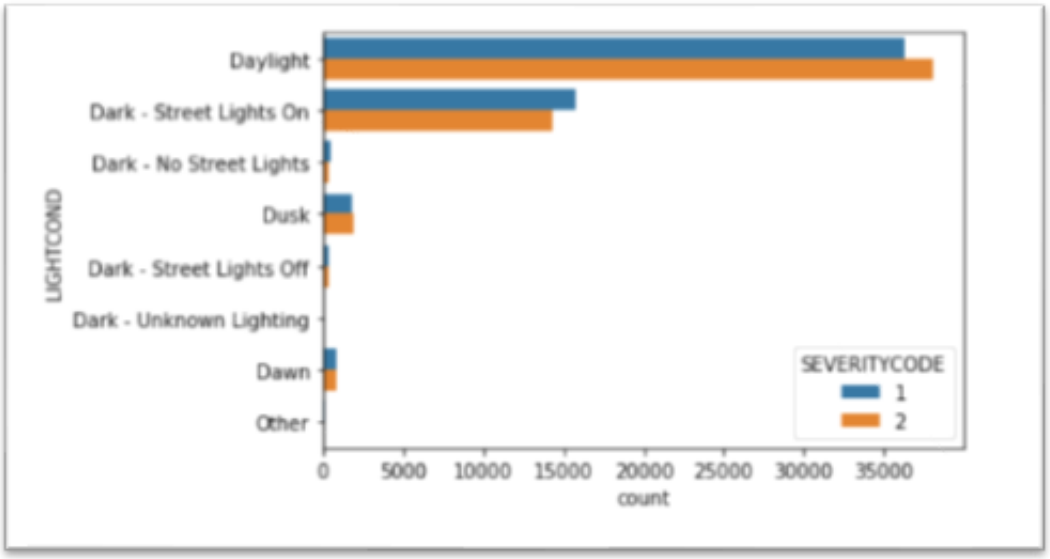
- Severity Code and Address Type: Severity Code 1 collisions along Blocks are more than on Intersections. The collisions of Severity Code 2 are almost the same on both Blocks and Intersections.



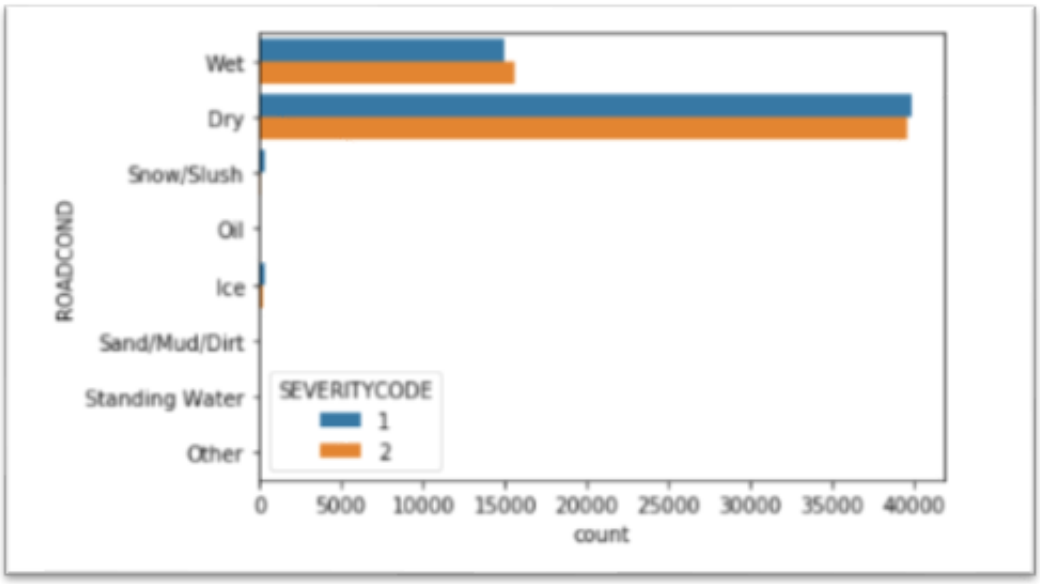
- Severity Code and Weather: most number of collisions happen on Clear weather.



- Severity Code and Light Conditions: most number of collisions happen on daylight.

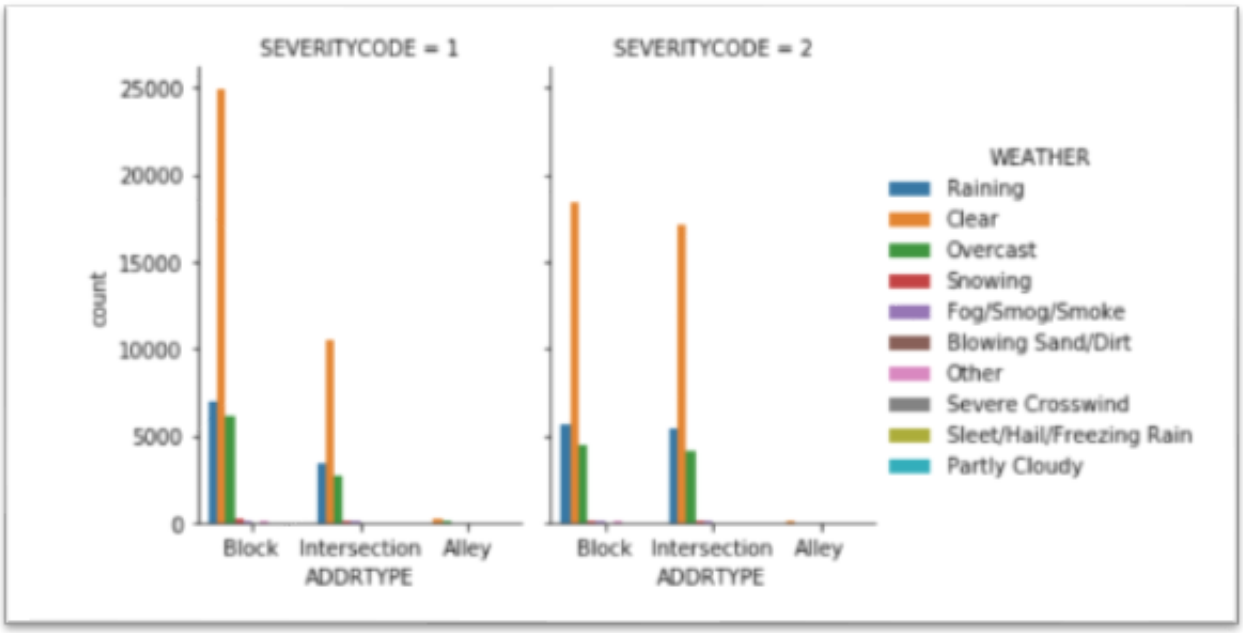


- Severity Code and Road Conditions: most number of collisions happen on dry roads.

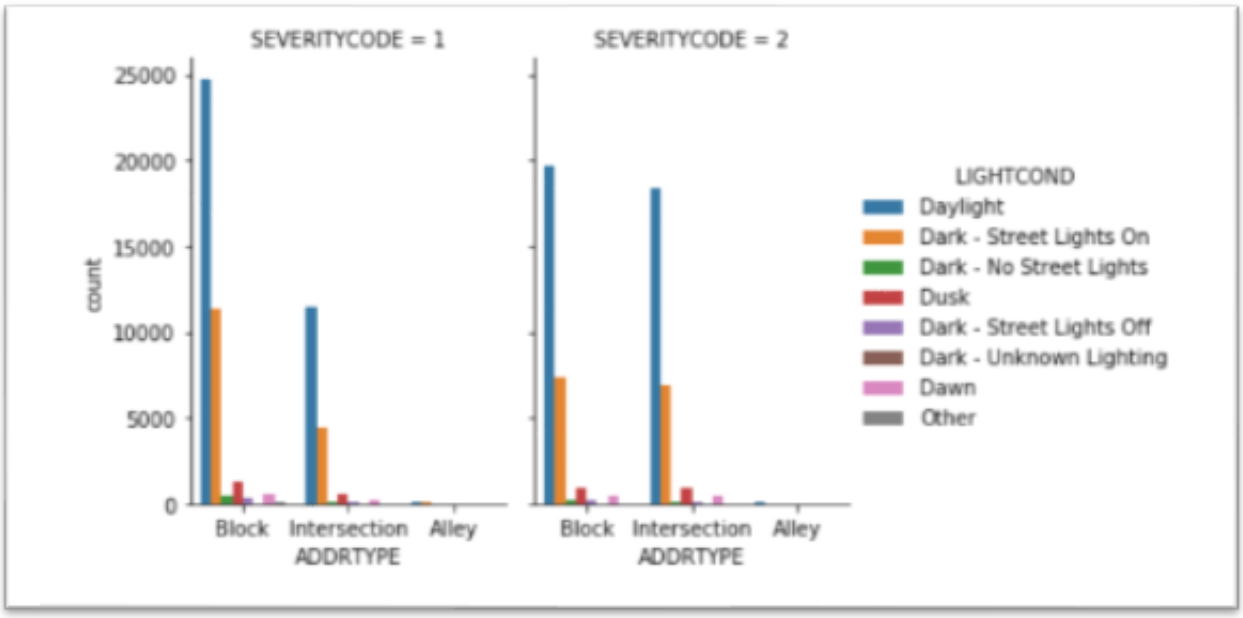


Correlations between variables are observed.

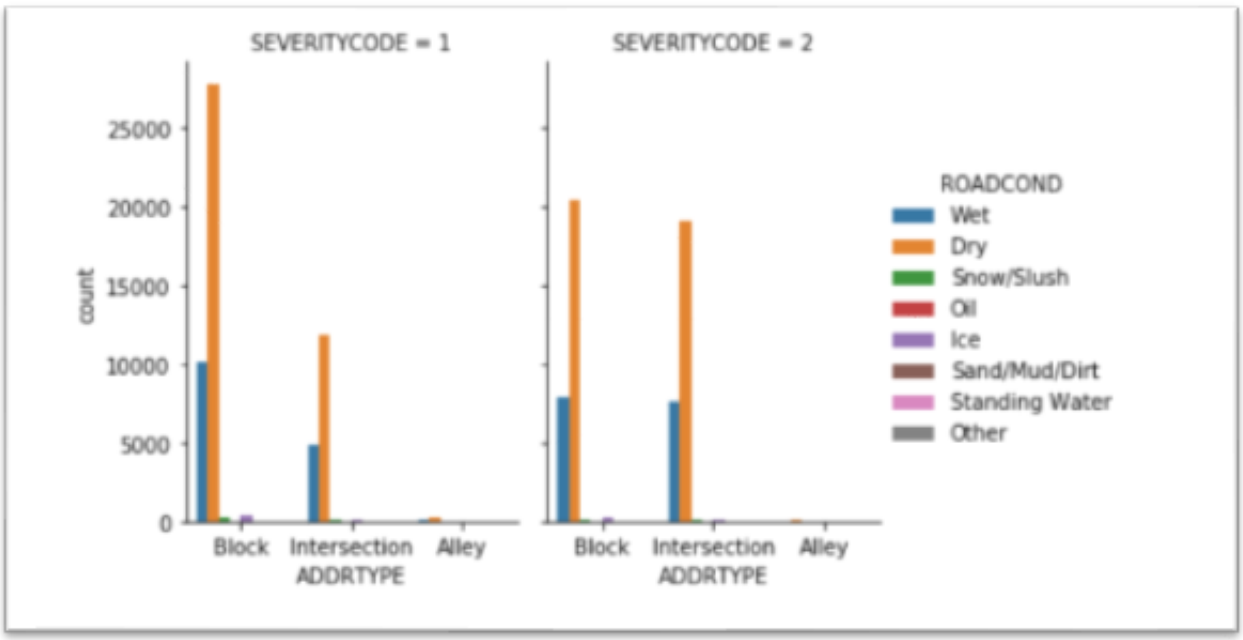
- Severity Code, Address type and weather: most number of collisions happen on blocks and on a clear weather.



- Severity Code, Address type and light conditions: most number of collisions happen daylight, mostly on blocks.



- Severity Code, Address type and road conditions: most number of collisions happen on dry roads, mostly on blocks.



Machine learning model

3.3.1 One-hot encoding

All selected independent variables are transformed to the format that can be fit to the machine learning model by using one-hot encoding. Indeed, the one-hot encoding help us to convert categorial features into numerical, that works better with classification and regressions algorithms. Each independent variable is transformed to a list

- One-hot encoding for each categorical feature

```
df3["ADDRTYPE"] = df3["ADDRTYPE"].astype('category')
df3["ADDRTYPE_CAT"] = df3["ADDRTYPE"].cat.codes
df3["WEATHER"] = df3["WEATHER"].astype('category')
df3["WEATHER_CAT"] = df3["WEATHER"].cat.codes
df3["LIGHTCOND"] = df3["LIGHTCOND"].astype('category')
df3["LIGHTCOND_CAT"] = df3["LIGHTCOND"].cat.codes
df3["ROADCOND"] = df3["ROADCOND"].astype('category')
df3["ROADCOND_CAT"] = df3["ROADCOND"].cat.codes
df4 = df3.drop(columns = ['ADDRTYPE', 'WEATHER', 'ROADCOND', 'LIGHTCOND'])
df4.head()
```

- Definition of the feature set (X) and the label (Y)

```
Feature = df4[['ADDRTYPE_CAT','WEATHER_CAT','LIGHTCOND_CAT','ROADCOND_CAT']]
Feature.head()
X = Feature
X[0:5]
```

```
Y = df4['SEVERITYCODE'].values
Y[0:5]
```

```
array([1, 1, 1, 1, 1], dtype=int64)
```

- Normalization of the data

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

```
array([[-0.78936715,  1.67946553,  0.59533255,  1.61357969],
       [-0.78936715, -0.71057727,  0.59533255, -0.62623701],
       [-0.78936715, -0.71057727,  0.59533255, -0.62623701],
       [-0.78936715, -0.71057727,  0.59533255, -0.62623701],
       [-0.78936715, -0.71057727,  0.59533255, -0.62623701]])
```

3.3.2 Data split

We are divided the dataset into a training set and a test set for better understanding.

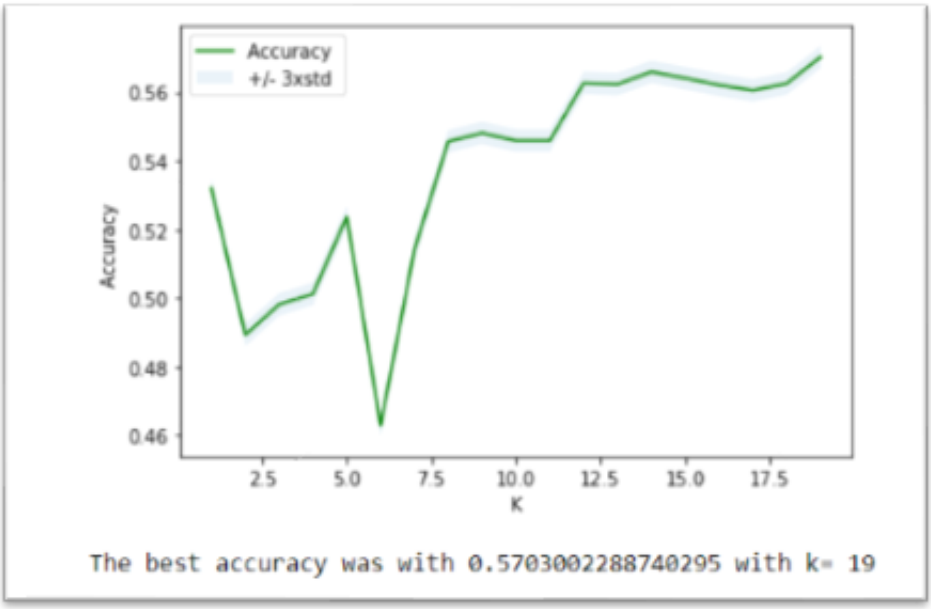
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=4)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

```
Train set: (89131, 4) (89131,)
Test set: (22283, 4) (22283,)
```

3.3.3 Machine Learnings

We have used K-Nearest Neighbor, Decision tree and Logistic Regression for our machine learnings to predict the severity.

- K-Nearest Neighbor (KNN)



The best accuracy for the KNN was 0.57 with k=19 using the test data.

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=19, p=2,
                     weights='uniform')
```

- Decision tree

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=4,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

- Logistic Regression

```
LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='liblinear', tol=0.0001, verbose=
0,
                   warm_start=False)
```

3.3.4 Model evaluation

We evaluate our model using test set.

```
yhat_knn = model_KNN.predict(X_test)
jaccard_knn = jaccard_similarity_score(Y_test, yhat_knn)
f1_score_knn = f1_score(Y_test, yhat_knn, average='weighted')
```

```
yhat_dt = model_DecisionTree.predict(X_test)
jaccard_dt = jaccard_similarity_score(Y_test, yhat_dt)
f1_score_dt = f1_score(Y_test, yhat_dt, average='weighted')
```

```
yhat_lg = model_LR.predict(X_test)
yhat_lg_prob = model_LR.predict_proba(X_test)
jaccard_lg = jaccard_similarity_score(Y_test, yhat_lg)
f1_score_lg = f1_score(Y_test, yhat_lg, average='weighted')
logloss_lg = log_loss(Y_test, yhat_lg_prob)
```

3.3.5 Results

We have plotted the results of our model evaluation using the test data. The accuracy of the Logistic Regression is based on the Logistic Loss (0.675). The result is not good as we expected because the accuracy of the models is not very high.

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.5703	0.5635	NA
Decision Tree	0.5896	0.5844	NA
LogisticRegression	0.5896	0.5845	0.675

Conclusion

We are chosen the collision address type, weather condition, road condition, light condition as our independent variables to predict the severity of a collision. Our analysis shows us that most of the collisions occurs on the following conditions: clear weather, dry road and daylight.

Moreover, there are more severity code 1 collisions along Blocks than on Intersections. And, the collisions of severity code 2 is almost the same on both Block and Intersections.

Futures directions

For futures research, it could be interesting to increase the accuracy of our model by considering other features for analysis.

The model accuracy could maybe be improved by grouping independent variables into groups to reduce model bias (for example: good-moderate-bad driving conditions, which is influenced by LIGHTCOND, WEATHER, ROADCOND). To have more accurate results, it is also recommended to balance data.