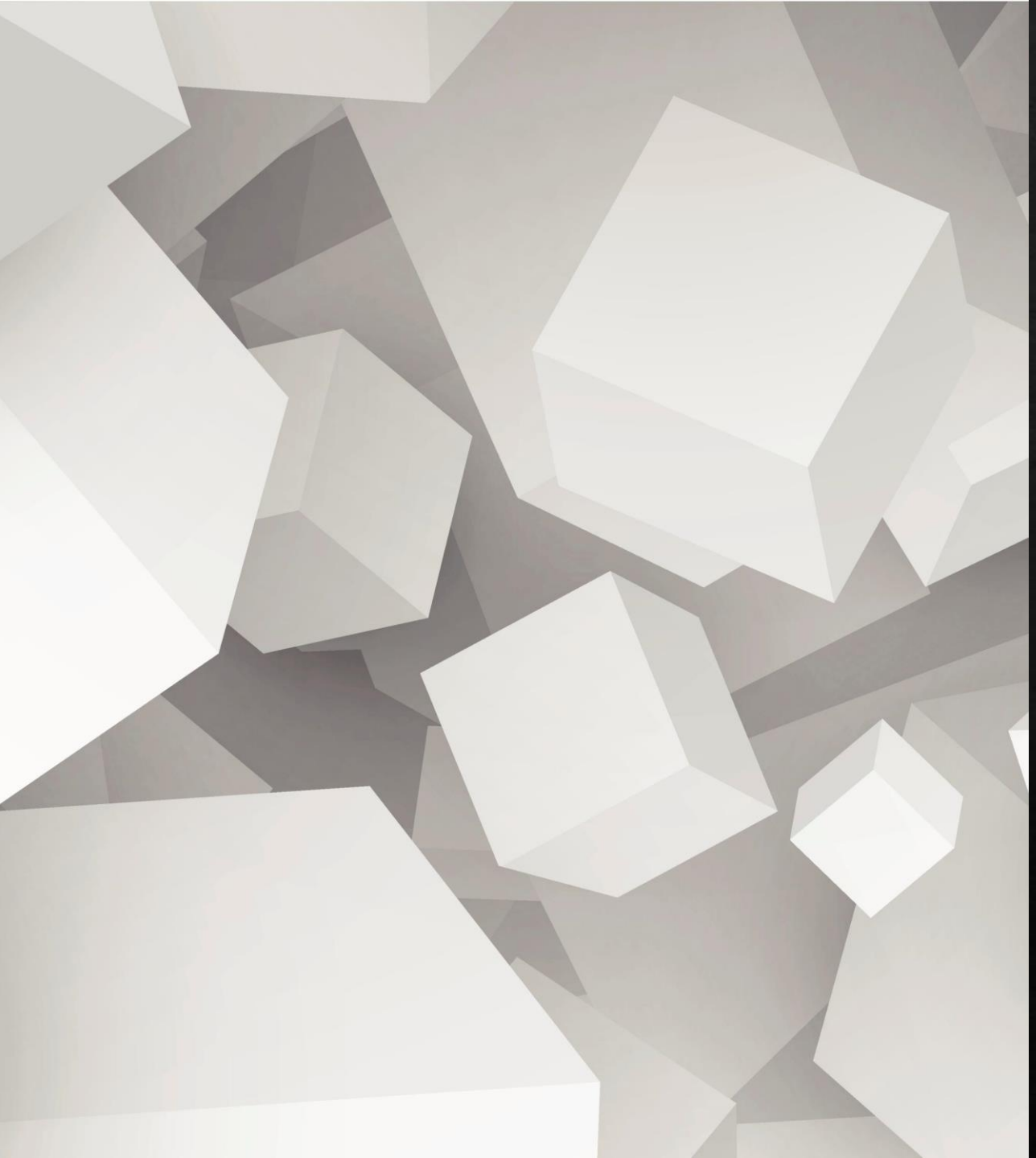
Predicting Severity of a Car Crash

Aaron Chan



Intro

- In this project, we seek to determine the **Severity of Car Crash Collisions** using machine learning. The data that we are using is for Seattle City, it comes from the SPD and was recorded by Traffic Records. Included in the GitHub repo is the metadata for the dataset as well as the dataset itself as of 10/8/2020. I have also included links to the metadata and dataset in the repository description.
- Determining the Severity of a Car Crash could be useful for car manufactures when designing what kind of safety features a car may have, or be useful for actuaries when assessing insurance risk. For this reason, we will not be looking at the exact locations where car crashes occurred, but the general features surrounding the car crash such as: Weather, Speeding, Collision Type, and State Codes to name a few.

Data

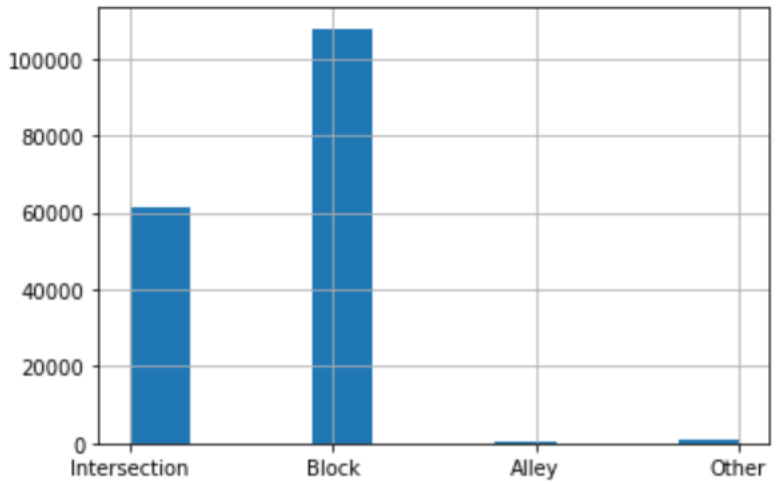
Based on the factor we want to predict, variables that might influence our model could include:

- Address Type
- Collision Type
- Pedestrian Count
- Vehicle Count
- Weather
- Road Conditions
- Speeding
- Light Conditions

Using a variety of factors could allow us to predict which factors are strongest in determining the Severity of a Car Crash. This could allow individuals to determine when they should stay more vigilant to prevent Severe accidents, or allow car companies to determine the best was to make thier cars safer (car location areas to reinforce).

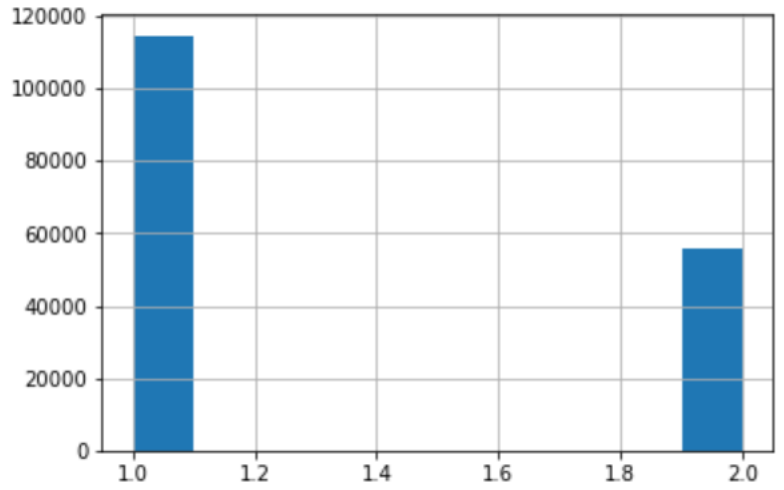
EDA

```
In [34]: # Histogram of ADDRTYPE
         df['ADDRTYPE'].hist()

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1c310cd67c8>
```



EDA

```
In [33]: # Histogram of our response variable SEVERITYCODE
         df['SEVERITYCODE'].hist()

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1c30eb68588>
```



Random Forest Model

In [40]:
```
# Create Random Forest using n descision trees
rf = RandomForestClassifier(n_estimators=100)

# Train the model using the training sets
rf.fit(X_train,y_train)

# Make predictions
y_pred = rf.predict(X_test)
```

In [48]:
```
print("Random Forest Classifier Model Accuracy:",metrics.accuracy_score(y_test, y_pred))

print('F1 Score: ',f1_score(y_test, y_pred, average='weighted'))
```

```
Random Forest Classifier Model Accuracy: 0.7294391335796532
F1 Score:  0.6990370816018231
```

Logistic Regression

Logistic Regression

In [42]:
```
# Fit Logistic Regression Model
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
```

In [47]:
```
# Logistic
LRPred = LR.predict(X_test)
yhat_prob = LR.predict(X_test)

print("Logistic Regression Model Accuracy:",metrics.accuracy_score(y_test, yhat_prob))

print('F1 Score: ',f1_score(y_test, yhat_prob, average='weighted'))

print('Logloss: ', log_loss(y_test, yhat_prob))
```

```
Logistic Regression Model Accuracy: 0.7341504897073485
F1 Score:  0.6911053349568707
Logloss:  23.291745818540427
```

Results

- It can be seen from our models above that the model that produced the best accuracy was the Logistic regression model. This is a little bit of a surprising result, since a Random Forest model goes trough many iterations and selects the best model from all of the iterations. This result could have be due to the way the data was distributed or the fact that the response variable data was not very well balanced. Further models or iterations of analysis may include Synthetic Minority Over Sampling in order to balance out skewed data, although in some cases this may not provide a better model

Conclusion

- There was a large amount of overlapping data in this dataset that I had to clean. Some columns that I excluded contained categorical data much more specific than the columns that I went with. All in All, there are many ways to play around with this dataset and selecting different columns may provide better model validity.
- As for what this model can tell us, it is possible to determine how severe a car accident might be depending on different factors.
- More effective models could allow car manufacturers to determine which variables are the most important in car safety features. ie: if angle of approach is strongly correlated to Severity, than people can determine what should be reinforced.
- The results of our model could tells individuals when to be more vigilant when driving, resulting in less fatalities.