**IBM Data Science Capstone: Car Accident Severity Report**

***Introduction | Business Understanding***

In an effort to reduce the frequency of car collisions in a community, an algorithm must be developed to predict the severity of an accident given the current weather, road and visibility conditions. When conditions are bad, this model will alert drivers to remind them to be more careful**.**

***Data Understanding***

Our predictor or target variable will be 'SEVERITYCODE' because it is used measure the severity of an accident from 0 to 5 within the dataset. Attributes used to weigh the severity of an accident are 'WEATHER', 'ROADCOND' and 'LIGHTCOND'.

***Balancing the Dataset***

Our target variable SEVERITYCODE is only 42% balanced. In fact, severity code in class 1 is nearly three times the size of class 2.

We can fix this by down sampling the majority class.

[Alt Text](https://res.cloudinary.com/practicaldev/image/fetch/s--ab3RblIA--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/owt8a0ptc68nzkntehue.png)

Perfectly balanced.

***Methodology***

Our data is now ready to be fed into machine learning models.

We will use the following models:

*K-Nearest Neighbor (KNN)*

KNN will help us predict the severity code of an outcome by finding the most similar to data point within k distance.

*Decision Tree*

A decision tree model gives us a layout of all possible outcomes so we can fully analyze the consequences of a decision. It context, the decision tree observes all possible outcomes of different weather conditions.

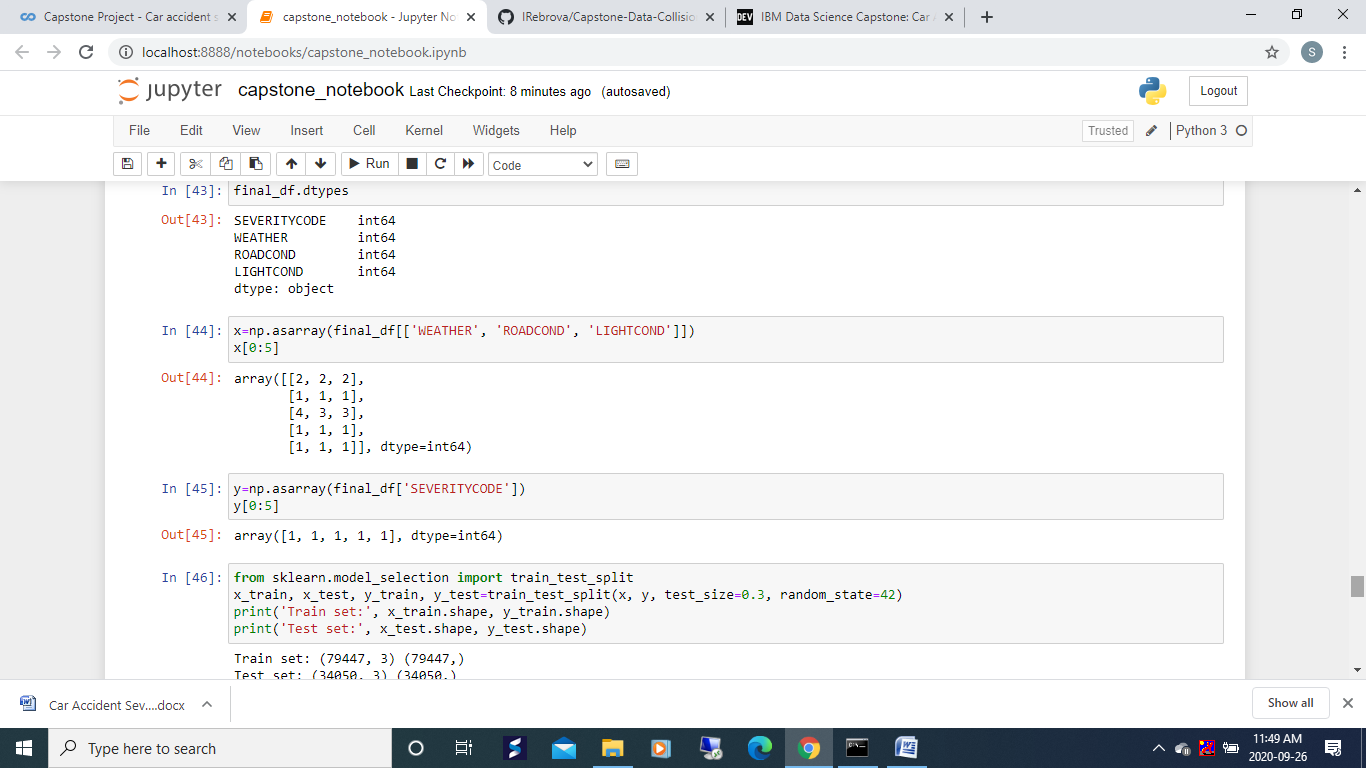
***Logistic Regression***

Because our dataset only provides us with two severity code outcomes, our model will only predict one of those two classes. This makes our data binary, which is perfect to use with logistic regression.

Let's get started!

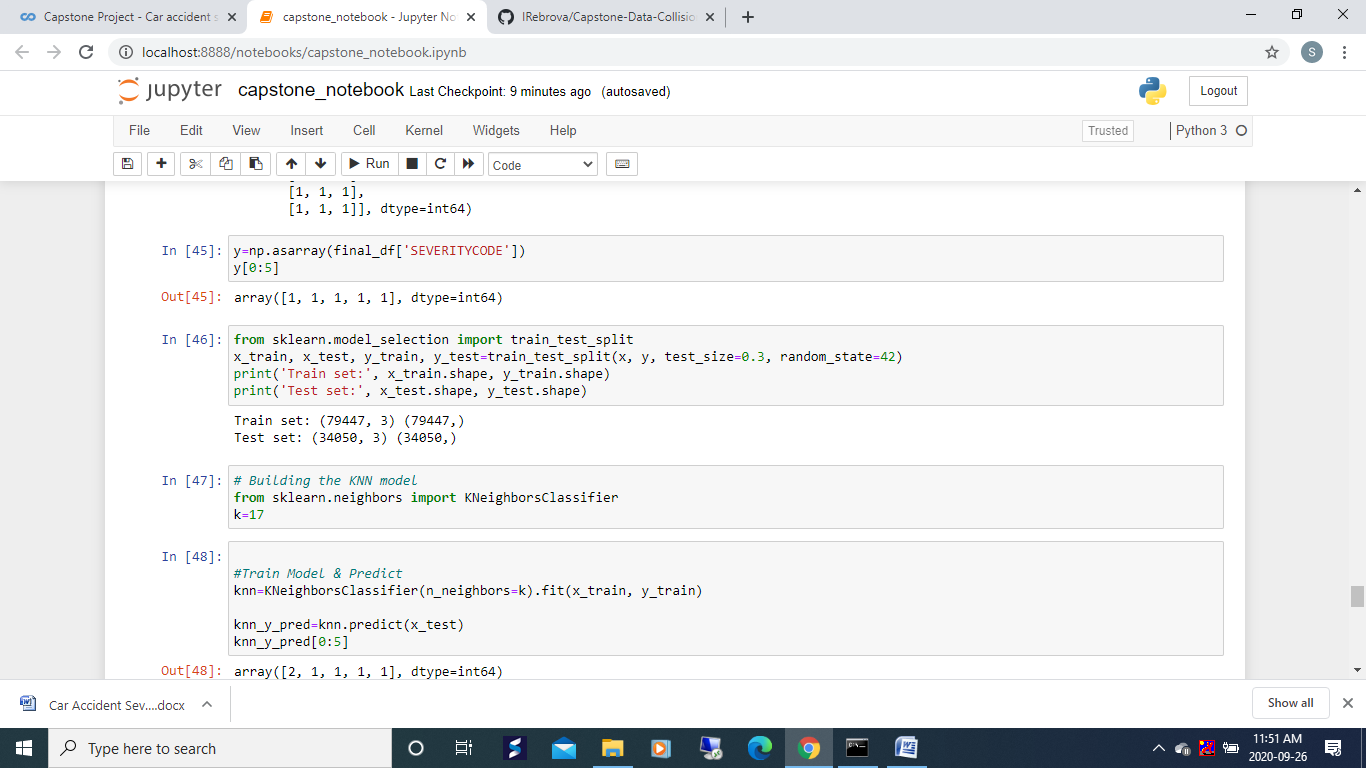
Initialization

**Define X and y**

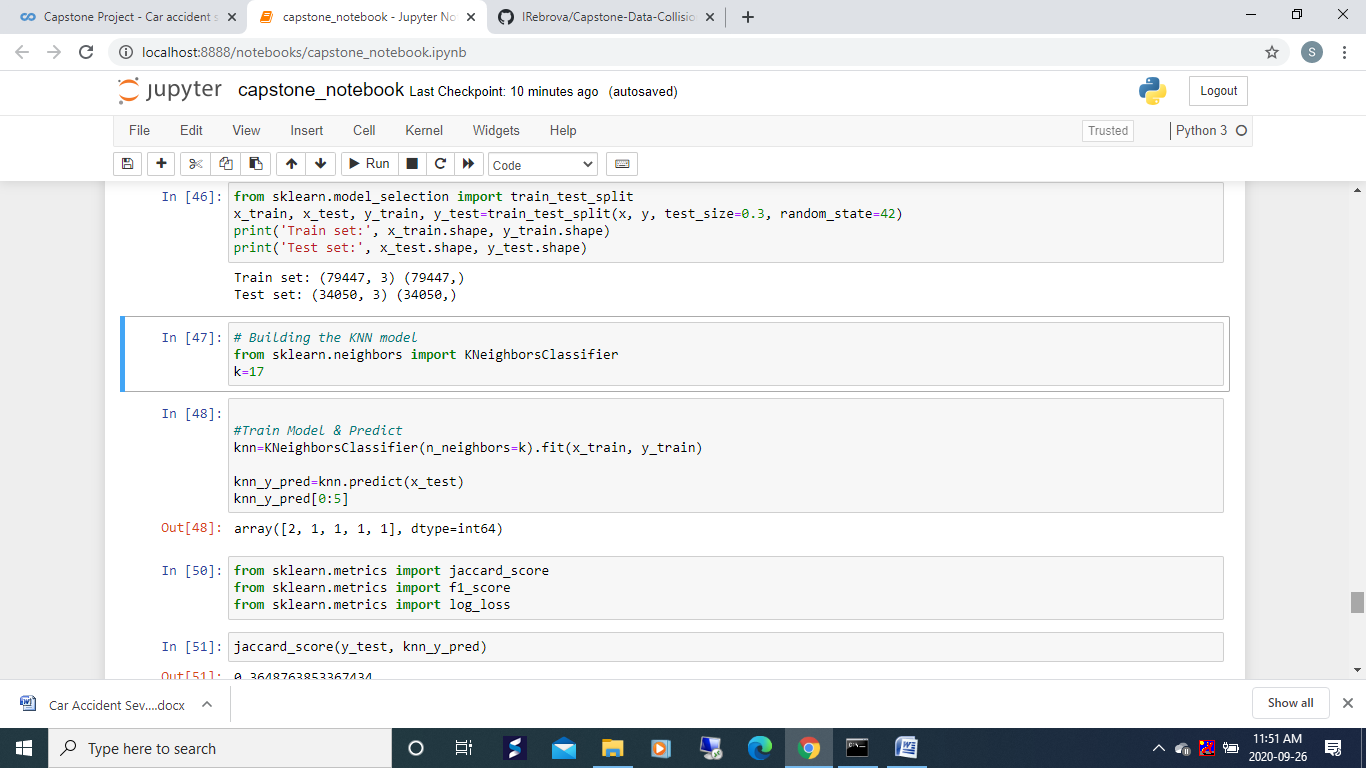


***Train/Test Split***

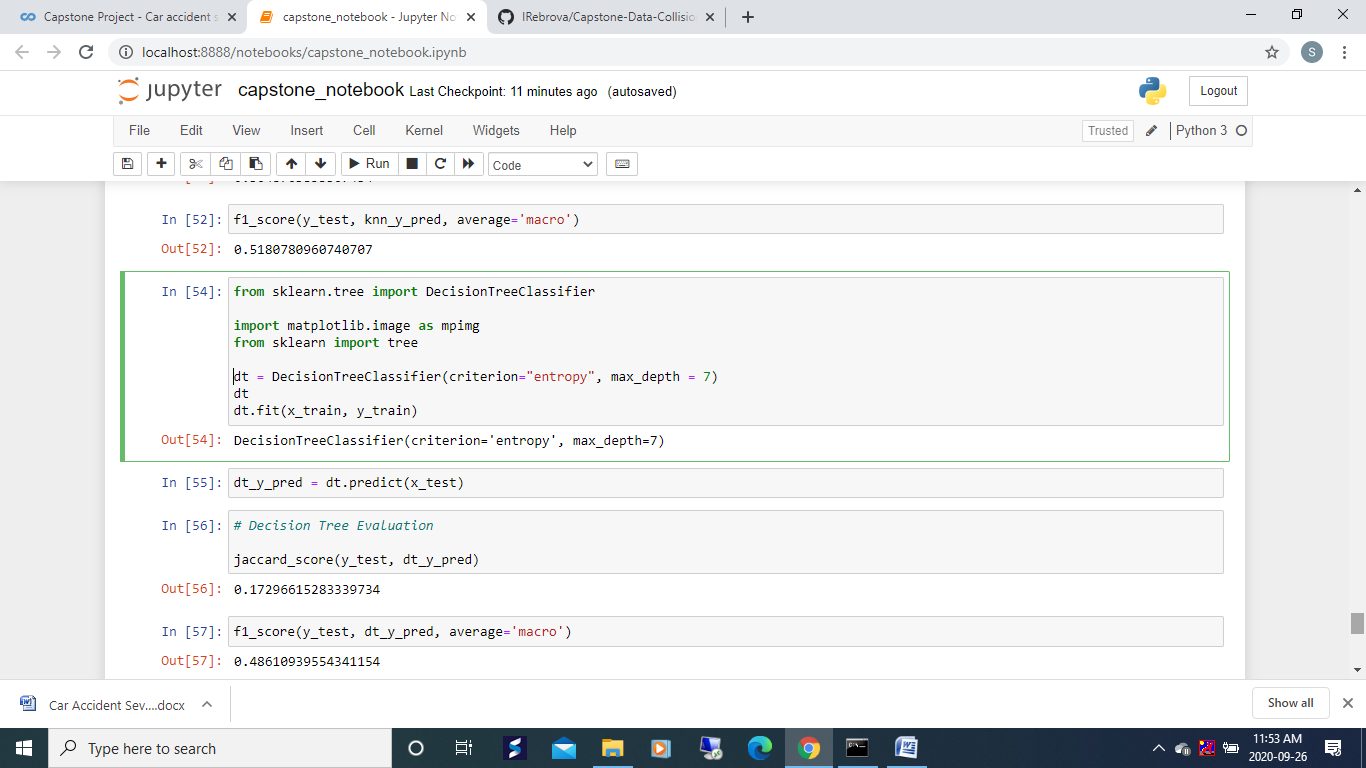
We will use 30% of our data for testing and 70% for training.



Here we will begin our modeling and predictions...



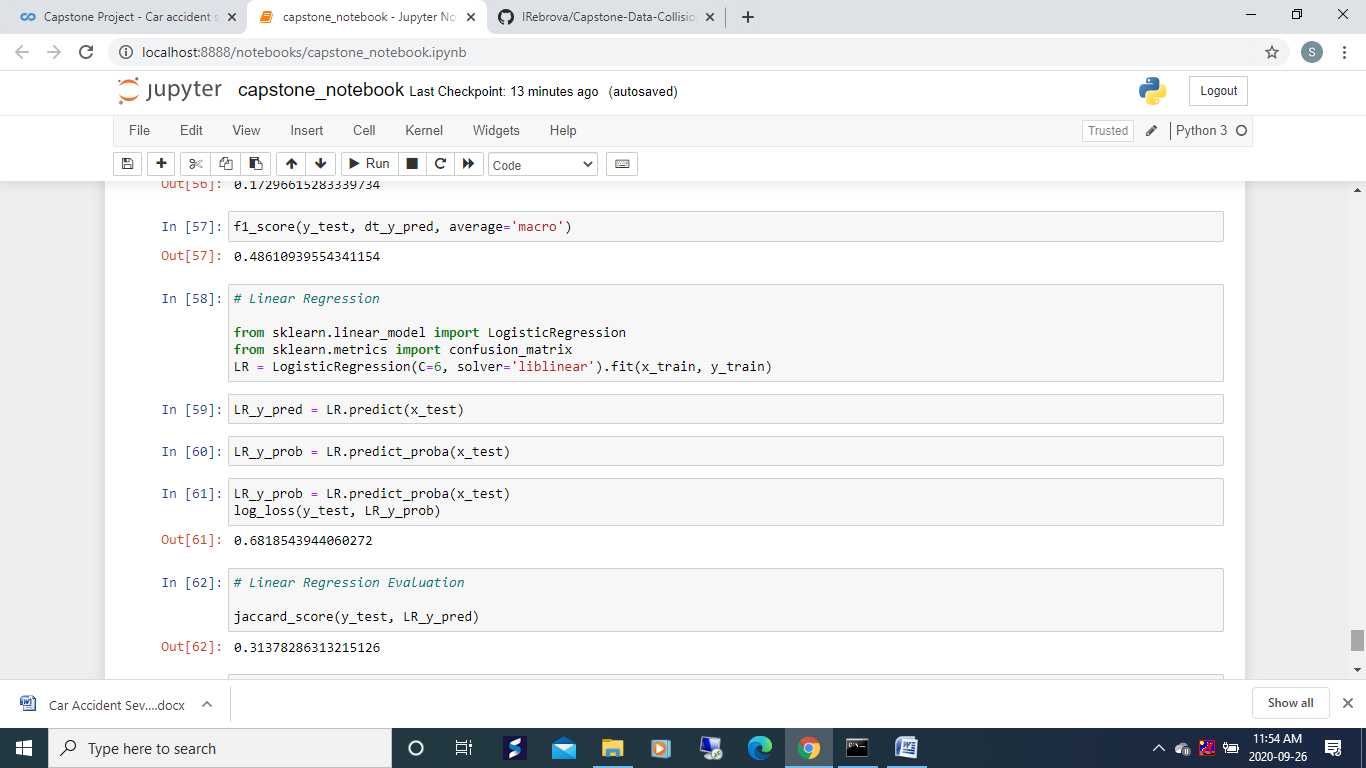
Decision tree

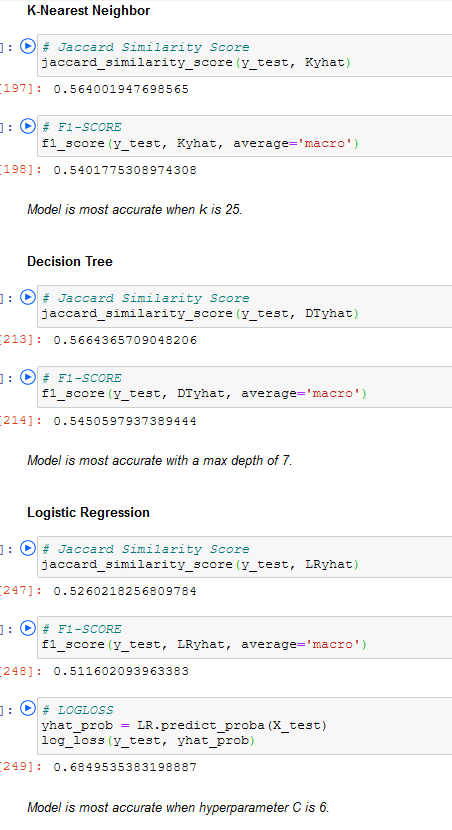


Results & Evaluation

Now we will check the accuracy of our models.

Linear regression



[](https://res.cloudinary.com/practicaldev/image/fetch/s--lYC6RzUx--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/i/6cp9rajstv6hwewyfp0h.png)

***Discussion***

In the beginning of this notebook, we had categorical data that was of type 'object'. This is not a data type that we could have fed through an algorithm, so label encoding was used to created new classes that were of type int8; a numerical data type.

After solving that issue we were presented with another - imbalanced data. As mentioned earlier, class 1 was nearly three times larger than class 2. The solution to this was down sampling the majority class with sklearn resample tool. We down sampled to match the minority class exactly with 58188 values each.

Once we analyzed and cleaned the data, it was then fed through three ML models; K-Nearest Neighbor, Decision Tree and Logistic Regression. Although the first two are ideal for this project, logistic regression made most sense because of its binary nature.

Evaluation metrics used to test the accuracy of our models were jaccard index, f-1 score and logloss for logistic regression. Choosing different k, max depth and hyper parameter C values helped to improve our accuracy to be the best possible.

***Conclusion***

Based on historical data from weather conditions pointing to certain classes, we can conclude that particular weather conditions have a somewhat impact on whether or not travel could result in property damage (class 1) or injury (class 2).