**Car Severity Accident Report**

**Introduction | Business Understanding****[¶](https://render.githubusercontent.com/view/ipynb?commit=e643e37bc943dd24ed7363c3f244813392cc3cf8&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f6a6572656d79626465762f436f7572736572612d43617073746f6e652f653634336533376263393433646432346564373336336333663234343831333339326363336366382f49424d25323043617073746f6e6525323050726f6a6563742833292e6970796e62&nwo=jeremybdev%2FCoursera-Capstone&path=IBM+Capstone+Project%283%29.ipynb&repository_id=289949679&repository_type=Repository" \l "Introduction-%7C-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. In an application, drivers will be alerted of the severity level when conditions are above code 0.

## Data Understanding[¶](https://render.githubusercontent.com/view/ipynb?commit=e643e37bc943dd24ed7363c3f244813392cc3cf8&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f6a6572656d79626465762f436f7572736572612d43617073746f6e652f653634336533376263393433646432346564373336336333663234343831333339326363336366382f49424d25323043617073746f6e6525323050726f6a6563742833292e6970796e62&nwo=jeremybdev%2FCoursera-Capstone&path=IBM+Capstone+Project%283%29.ipynb&repository_id=289949679&repository_type=Repository" \l "Data-Understanding)

Our predictor or target variable will be 'SEVERITYCODE' because it is used to measure the severity of an accident from 0 to 4 within the dataset. Attributes used to weigh the severity of an accident are 'WEATHER', 'ROADCOND' and 'LIGHTCOND'.

Severity codes are as follows:

* 0 : Little to no Probability (Clear Conditions)
* 1 : Very Low Probability - Chance or Property Damage
* 2 : Low Probability - Chance of Injury
* 3 : Mild Probability - Chance of Serious Injury
* 4 : High Probability - Chance of Fatality

### **Extract Dataset & Convert**

In it's original form, this data is not fit for analysis. For one, there are many columns that we will not use for this model. Also, most of the features are of type object, when they should be numerical type.

We must use label encoding to covert the features to our desired data type.

## Methodology [¶](https://render.githubusercontent.com/view/ipynb?commit=e643e37bc943dd24ed7363c3f244813392cc3cf8&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f6a6572656d79626465762f436f7572736572612d43617073746f6e652f653634336533376263393433646432346564373336336333663234343831333339326363336366382f49424d25323043617073746f6e6525323050726f6a6563742833292e6970796e62&nwo=jeremybdev%2FCoursera-Capstone&path=IBM+Capstone+Project%283%29.ipynb&repository_id=289949679&repository_type=Repository" \l "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.

## Results & Evaluation

Now we will check the accuracy of our models.

In [112]:

**from** **sklearn.metrics** **import** jaccard\_similarity\_score

**from** **sklearn.metrics** **import** f1\_score

**from** **sklearn.metrics** **import** log\_loss

#### K-Nearest Neighbor

In [197]:

*# Jaccard Similarity Score*

jaccard\_similarity\_score(y\_test, Kyhat)

Out[197]:

0.564001947698565

In [198]:

*# F1-SCORE*

f1\_score(y\_test, Kyhat, average='macro')

Out[198]:

0.5401775308974308

*Model is most accurate when k is 25.*

#### Decision Tree

In [213]:

*# Jaccard Similarity Score*

jaccard\_similarity\_score(y\_test, DTyhat)

Out[213]:

0.5664365709048206

In [214]:

*# F1-SCORE*

f1\_score(y\_test, DTyhat, average='macro')

Out[214]:

0.5450597937389444

*Model is most accurate with a max depth of 7.*

#### Logistic Regression

In [247]:

*# Jaccard Similarity Score*

jaccard\_similarity\_score(y\_test, LRyhat)

Out[247]:

0.5260218256809784

In [248]:

*# F1-SCORE*

f1\_score(y\_test, LRyhat, average='macro')

Out[248]:

0.511602093963383

In [249]:

*# LOGLOSS*

yhat\_prob = LR.predict\_proba(X\_test)

log\_loss(y\_test, yhat\_prob)

Out[249]:

0.6849535383198887

*Model is most accurate when hyperparameter C is 6.*

## Discussion [¶](https://render.githubusercontent.com/view/ipynb?commit=e643e37bc943dd24ed7363c3f244813392cc3cf8&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f6a6572656d79626465762f436f7572736572612d43617073746f6e652f653634336533376263393433646432346564373336336333663234343831333339326363336366382f49424d25323043617073746f6e6525323050726f6a6563742833292e6970796e62&nwo=jeremybdev%2FCoursera-Capstone&path=IBM+Capstone+Project%283%29.ipynb&repository_id=289949679&repository_type=Repository" \l "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 downsampling the majority class with sklearn's resample tool. We downsampled 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 the 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 hyperamater 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)*