IBM Capstone Project
Seattle Car Accident Severity

Jason Rutherford
September 13th, 2020

Introduction/Business Problem

Did you know that road crashes are the leading cause of death in the United States for people aged between 1 and 54? The United States is one of the busiest countries in the world in terms of road traffic with nearly 280 million vehicles in operations.

They are numerous factors that determine the severity of accidents such as irresponsible driving (speeding, distracted and under the influence of alcohol/dugs), time of day/day of week and environmental conditions (weather, season, road surface and lighting conditions).
We can gain insight and solutions from the numerous factors that affect accident severities by using data from past accidents.

We will be using machine learning to build multiple models that can predict the severity of a future accident base on the similarity of their initial conditions to those of other accidents from historical data.

Data

A comprehensive dataset of 194,673 accidents occurring from 2004 to May 2020 in the Seattle city area was obtained from the Seattle Police Department and recorded by Traffic Records and include Collisions at intersection or mid-block of a segment.

The data also has 37 columns describing the details of each accident including the weather conditions, collision type, date/time of accident and location (latitude and longitude).

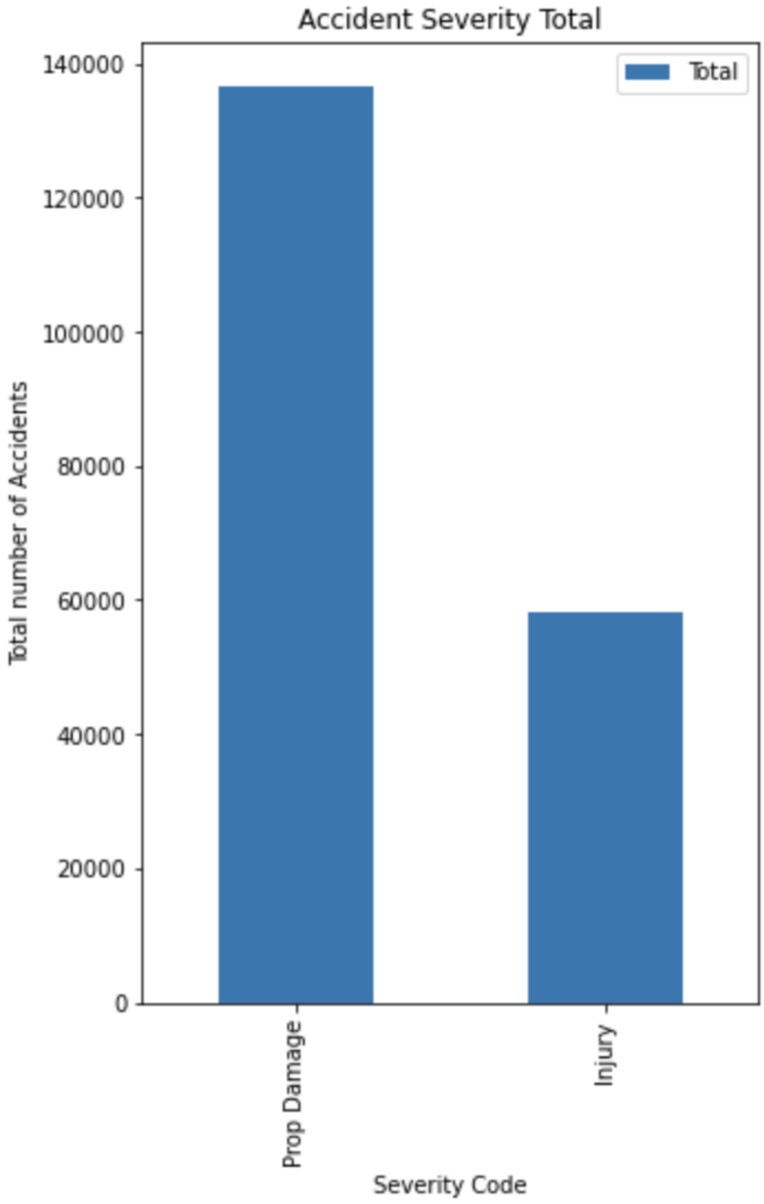
An additional document was provided with the description of each column given; the data is labeled but unbalanced.

We will be using a predictive analytic approach to determine the severity of an accident base on its attributes such as location, weather condition, speeding, light conditions and road condition. The attribute **SEVERITYCODE** (dependent variable) will be used to determine the severity of an accident.

According to the additional document, the possible values for **SEVERITYCODE** are:

- 0 - Unknown
- 1 - Prop Damage
- 2 - Injury
- 2b - Serious Injury
- 3 – Fatality

Unfortunately, the data set only provided two values for the **SEVERITYCODE** (1 & 2).



A bar graph showing the amount of accidents base on the severity

As mentioned above, the dataset has almost 40 attributes, but we want to focus only a set of attributes that has useful information being able to predict the severity of a future accident. Within the data, the following columns was selected:

SEVERITYCODE – This variable corresponds to the severity of the collision

COLLISIONTYPE – This variable determines the collision type.

UNDERINFL – This variable determines whether or not a driver involved was under the influence of drugs or alcohol.

INATTENTIONIND – This variable determines whether or not collision was due to inattention.

WEATHER – This variable determines the description of the weather conditions during the time of the collision.

ROADCOND – This variable determines the condition of the road during the collision.

LIGHTCOND – This variable determines the light conditions during the collision.

SPEEDING – This variable determines whether or not speeding was a factor in the collision.

During data analysis, it was discovered the data set had a lot of missing values for certain columns:

SPEEDING

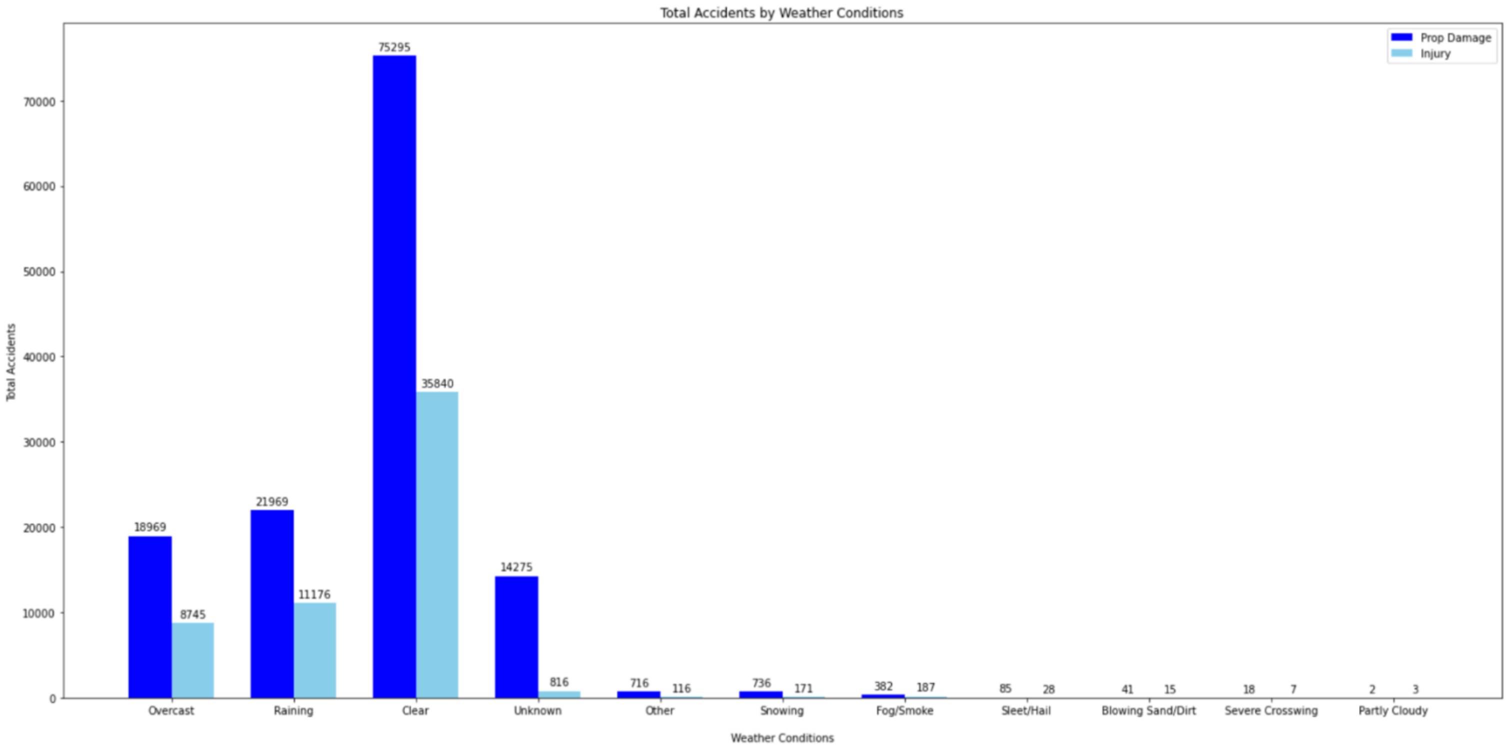
- In the SPEEDING column, 95.21% of the values were unknown.

INATTENTIONIND

- In the INATTENTIONIND column, 84.69% of the values were unknown.

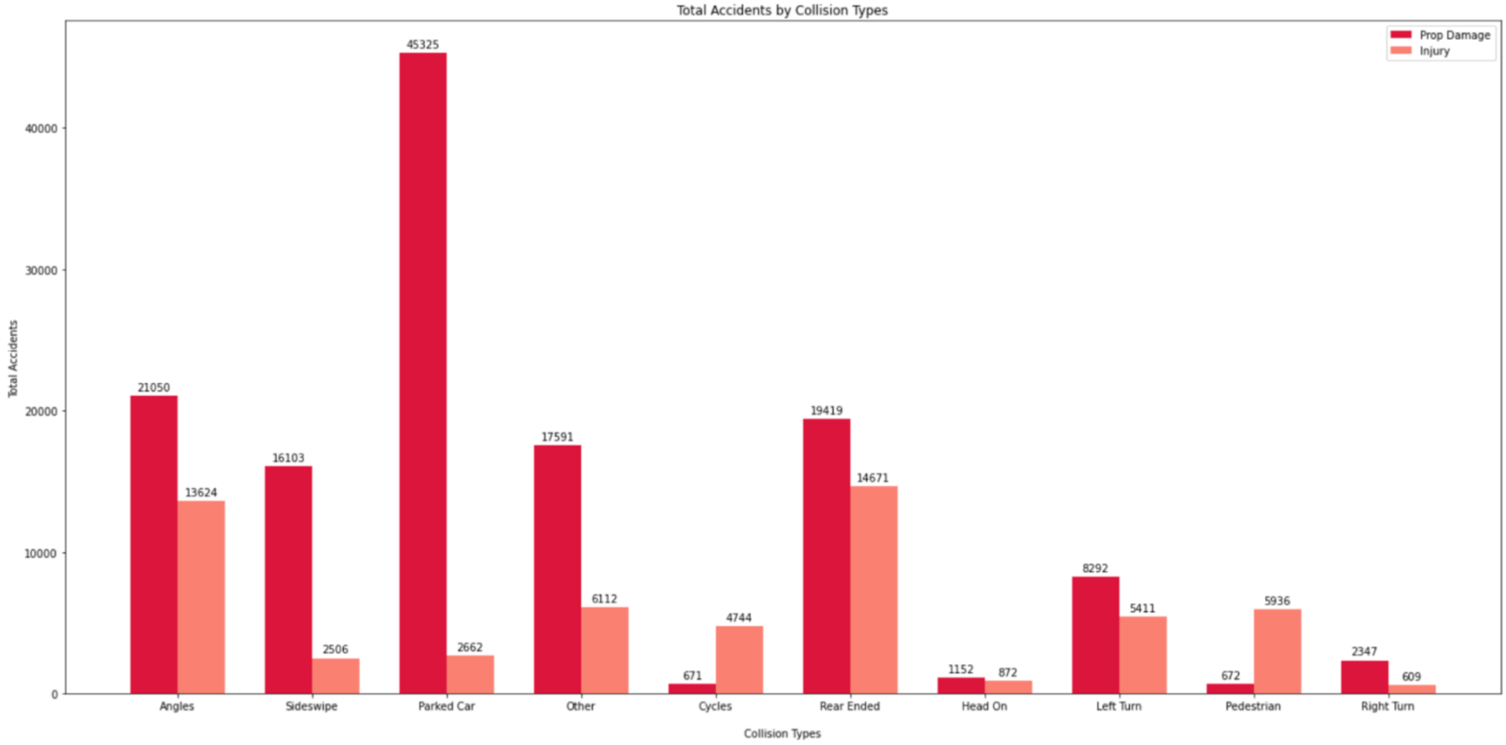
So, both columns had to be removed to further increase the accuracy of the model and the analysis of the data.

Using the severity code for each accident, we can detect the amount of accidents for each type of value in each column.

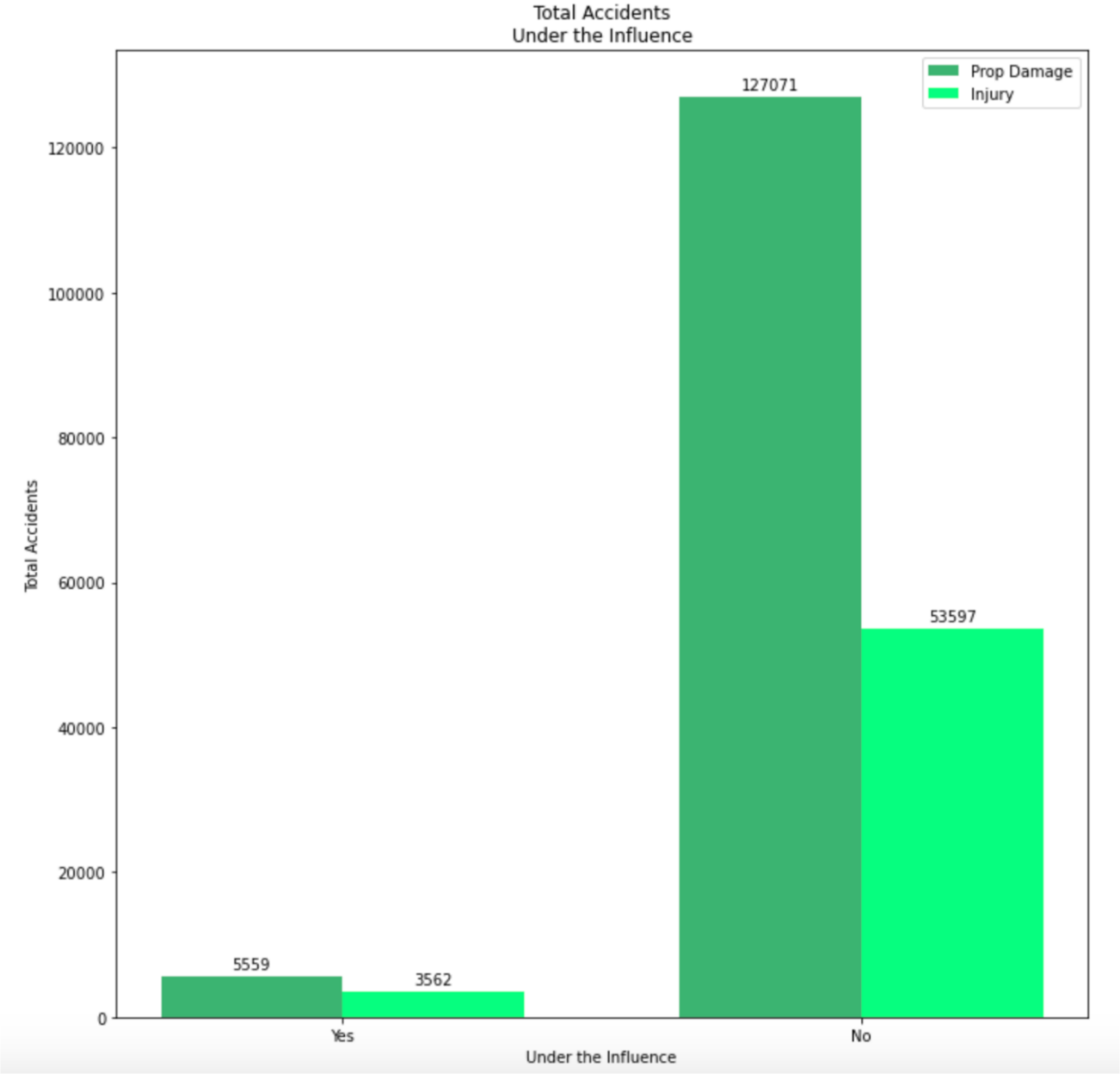


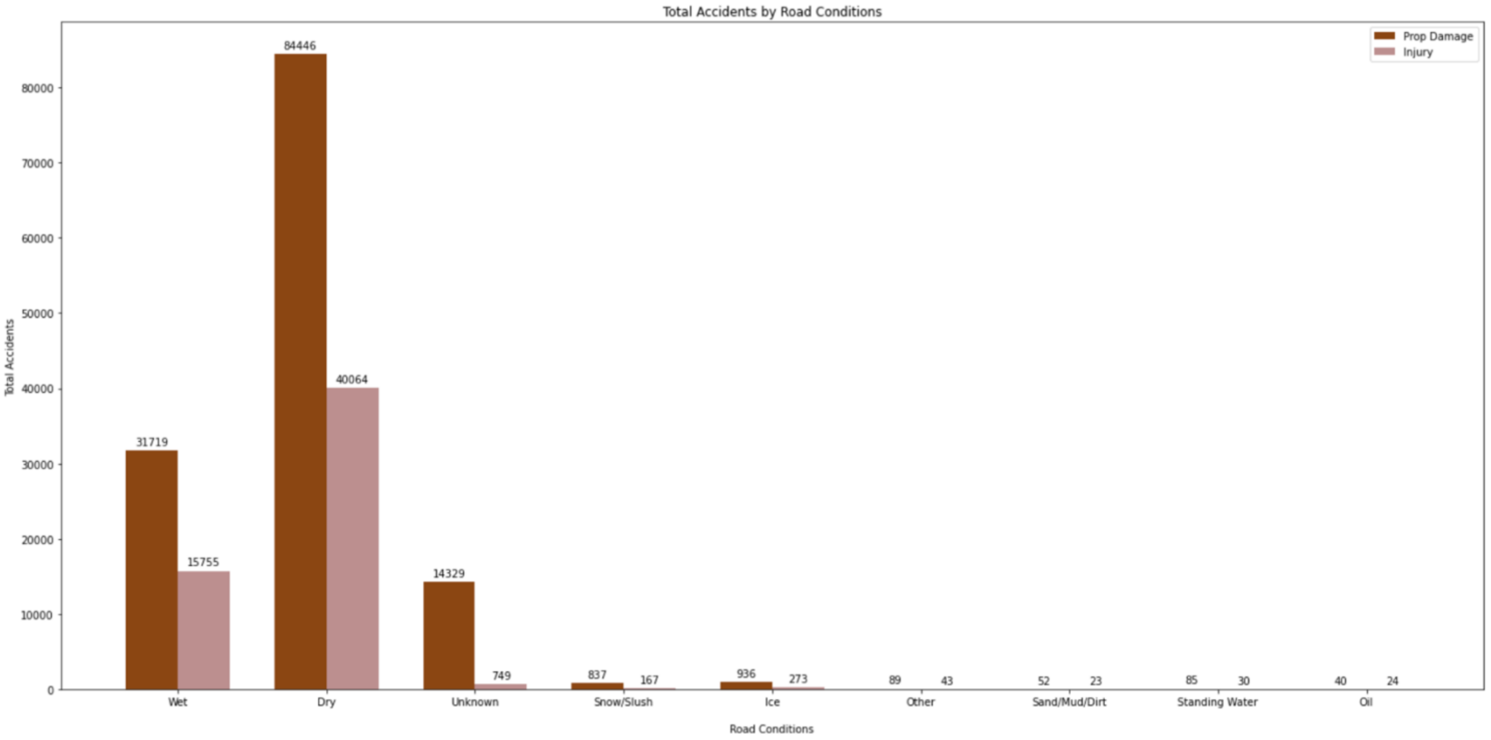
A bar graph that shows the amount of accidents for each type of weather condition base on the severity of the accident

A bar graph that shows the amount of accidents for each type of collision base on the severity of the accident



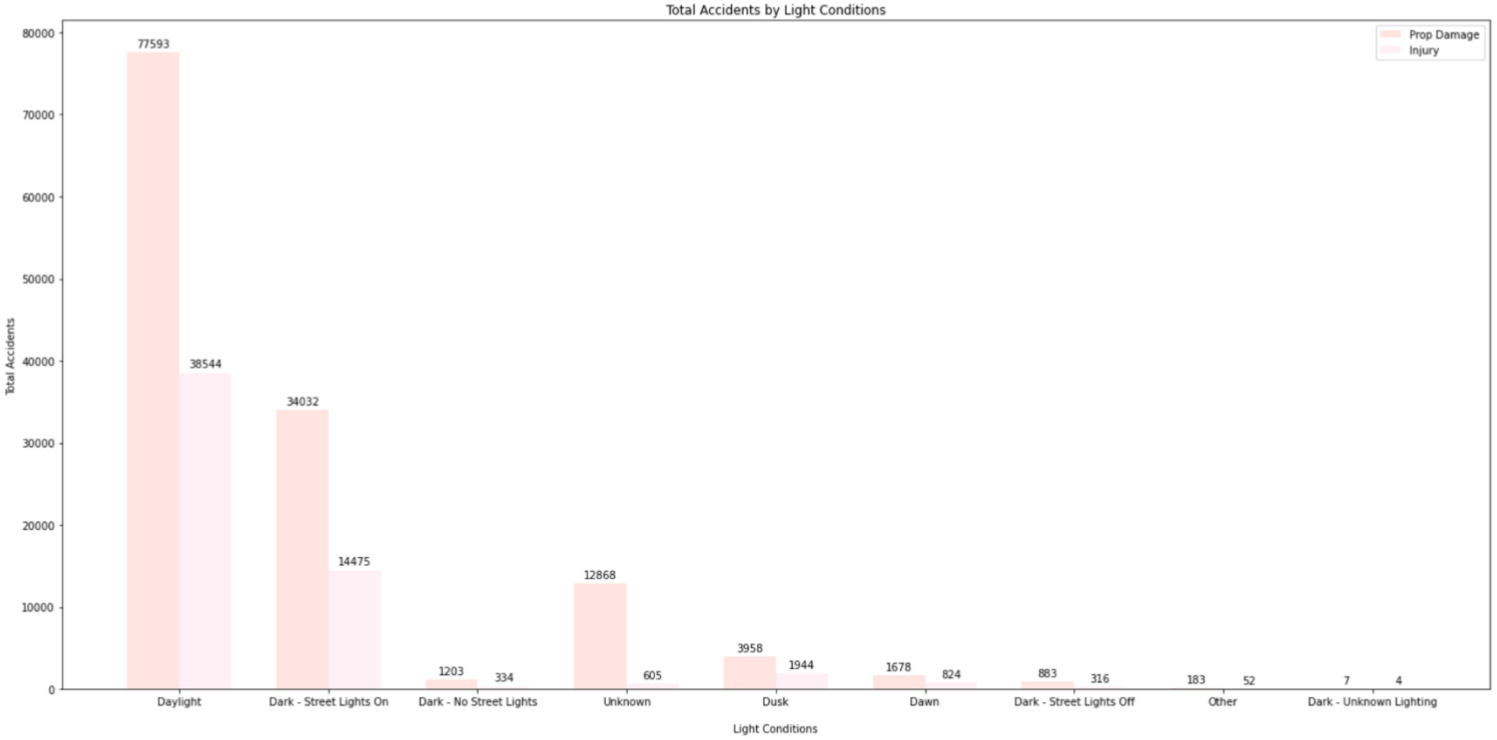
A bar graph that shows the amount of accidents whether the driver was under the influence of alcohol or drugs base on the severity of the accident





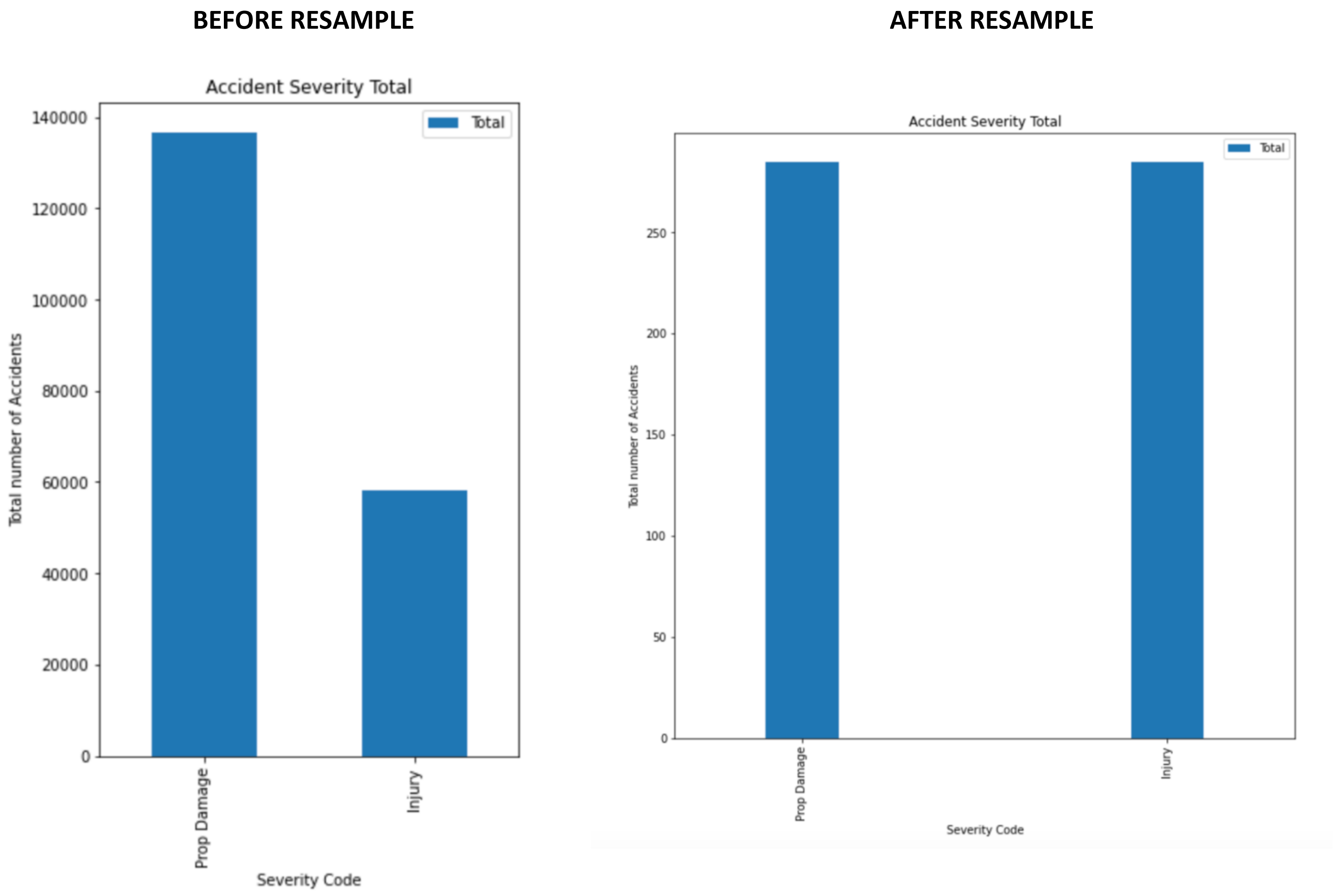
A bar graph that shows the amount of accidents for each type of road condition base on the severity of the accident

A bar graph that shows the amount of accidents for each type of light condition base on the severity of the accident



Base on the bar graphs above, the difference for each type of value has a huge gap which makes the data unbalanced. In order to correct this, we will remove all rows in the data set that doesn’t have a value for any column, then we will downsize the data by resampling it.

The dataset was reduced to 695 rows from 194,673 rows by dropping all rows without a value in every column.

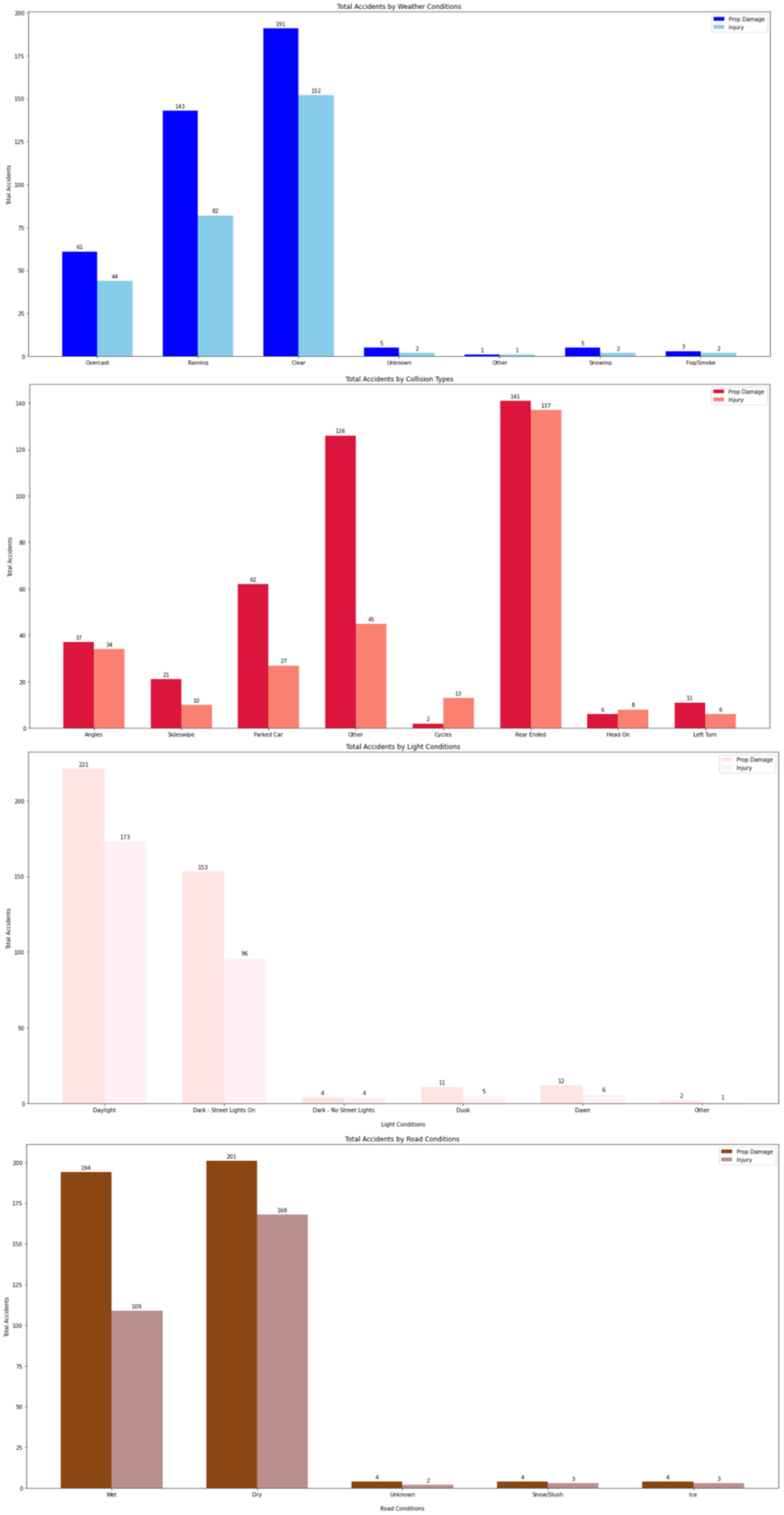


BEFORE RESAMPLE

AFTER RESAMPLE

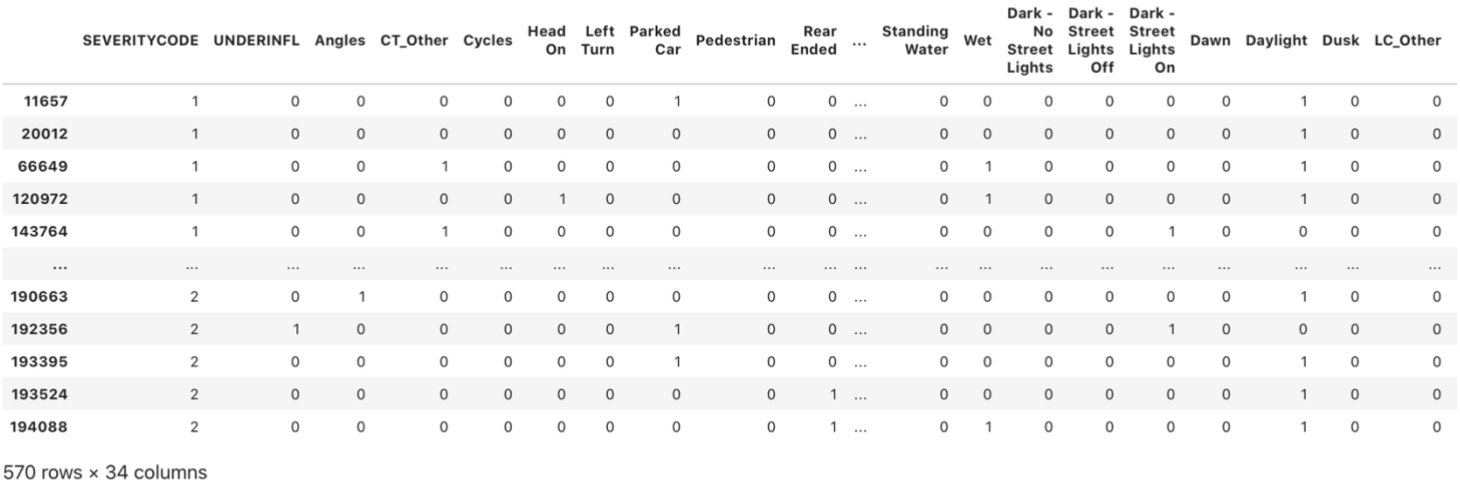
A bar graph that shows the amount of accidents for each type of severity. As you can see, after the resample, the amount of accidents for both severities are now equal making the data balanced.

The dataset was reduced to 570 rows from 695 rows after resampling the data, making both severities balanced to furture accurate the model.



After removing unwanted rows and resampling the data, you can see that they are missing types of values and the numbers have reduced.

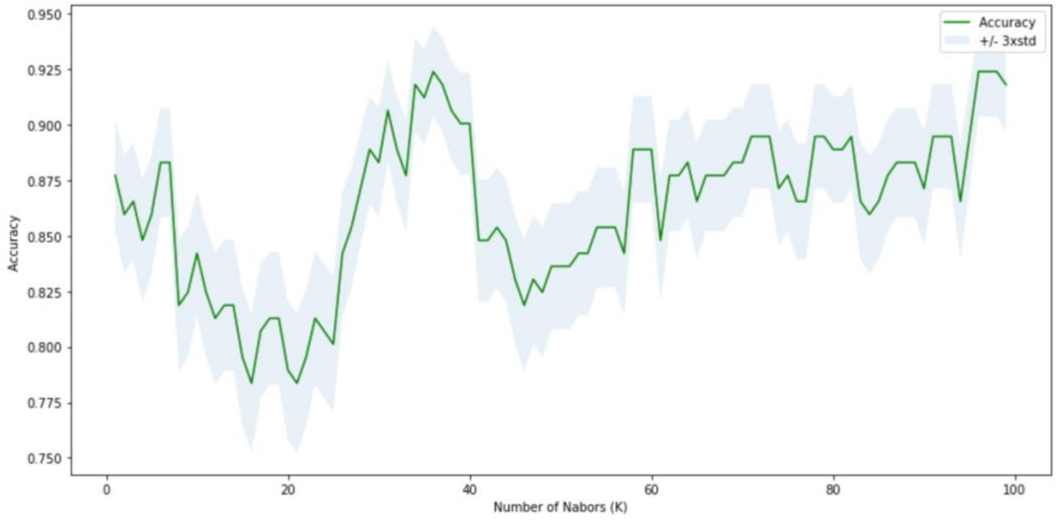
Now the module itself is only able to process a dataset if the values are numbers. Given our dataset, the values for each type columns contains numbers and words. We will be using the “One Hot Encoding” method to relpace those values with 0s and 1s, so the model will be able to process our dataset. So each unique value will have their own column with values of 0s and 1s.



	SEVERITYCODE	UNDERINFL	Angles	CT_Other	Cycles	Head On	Left Turn	Parked Car	Pedestrian	Rear Ended	...	Standing Water	Wet	Dark - No Street Lights	Dark - Street Lights Off	Dark - Street Lights On	Dawn	Daylight	Dusk	LC_Other
11657	1	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	1	0	0
20012	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0
66649	1	0	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	1	0	0
120972	1	0	0	0	0	1	0	0	0	0	...	0	1	0	0	0	0	1	0	0
143764	1	0	0	1	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
190663	2	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0
192356	2	1	0	0	0	0	0	1	0	0	...	0	0	0	0	1	0	0	0	0
193395	2	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	1	0	0
193524	2	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	1	0	0
194088	2	0	0	0	0	0	0	0	0	1	...	0	1	0	0	0	0	1	0	0

570 rows × 34 columns

Results



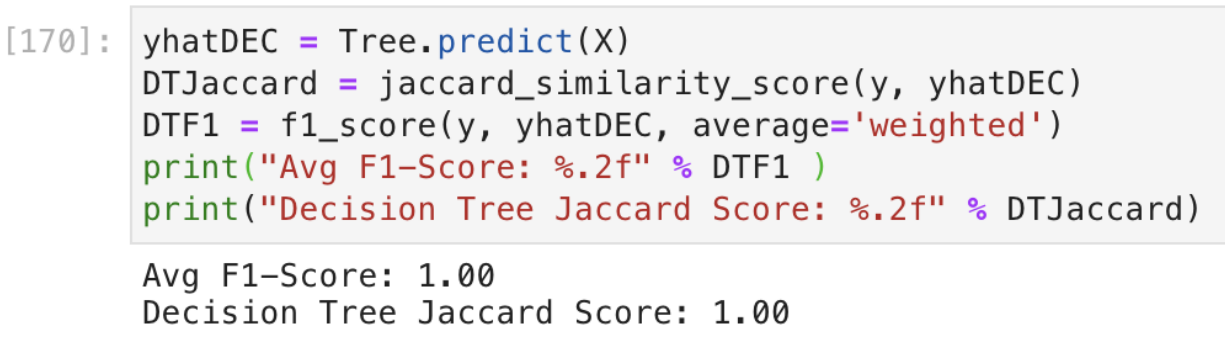
```
Best Accuracy: 0.9239766081871345 , K = 36
```

K Nearest Neighbor Model

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. The best accuracy of the KNN model is when K equals 36 with an accuracy of 92.39%.

Decision Tree Model

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

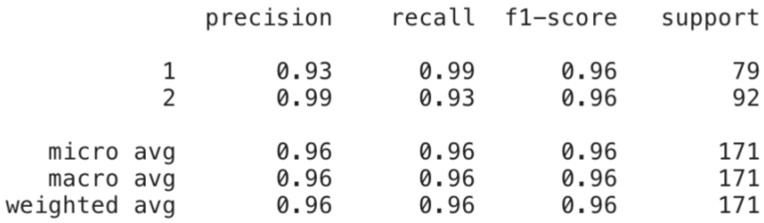


```
[170]: yhatDEC = Tree.predict(X)
       DTJaccard = jaccard_similarity_score(y, yhatDEC)
       DTF1 = f1_score(y, yhatDEC, average='weighted')
       print("Avg F1-Score: %.2f" % DTF1 )
       print("Decision Tree Jaccard Score: %.2f" % DTJaccard)
```

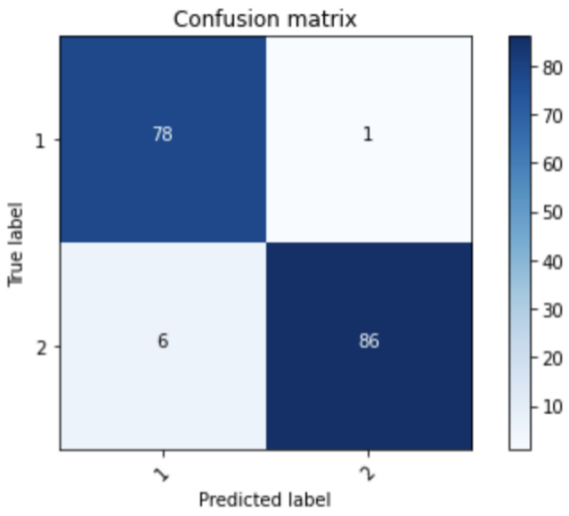
```
Avg F1-Score: 1.00
Decision Tree Jaccard Score: 1.00
```

Support Vector Machine

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

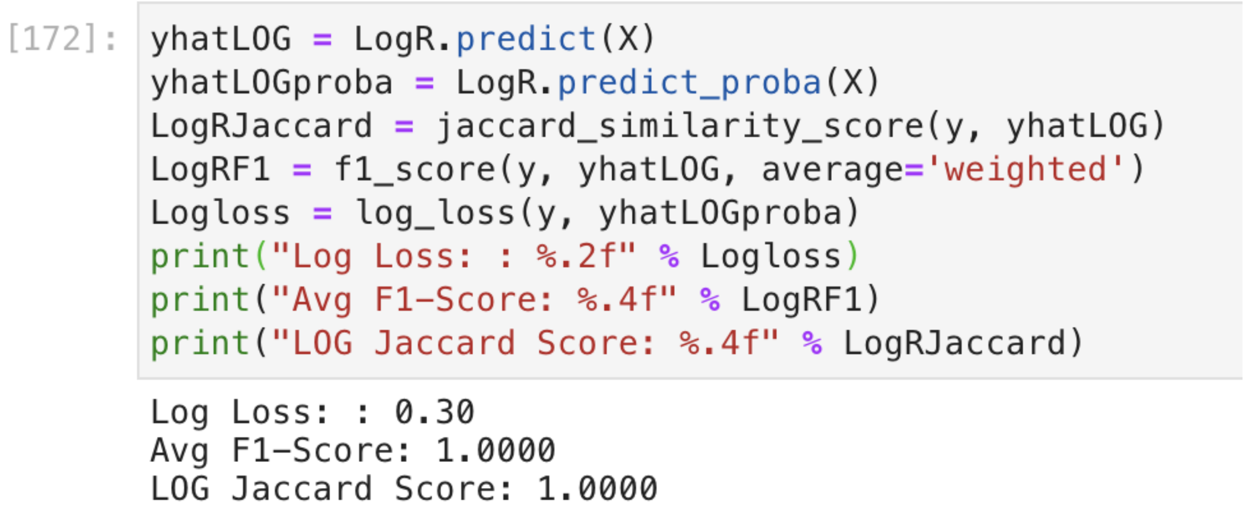


```
              precision    recall  f1-score   support

           1       0.93      0.99      0.96        79
           2       0.99      0.93      0.96        92

   micro avg       0.96      0.96      0.96       171
   macro avg       0.96      0.96      0.96       171
weighted avg       0.96      0.96      0.96       171
```



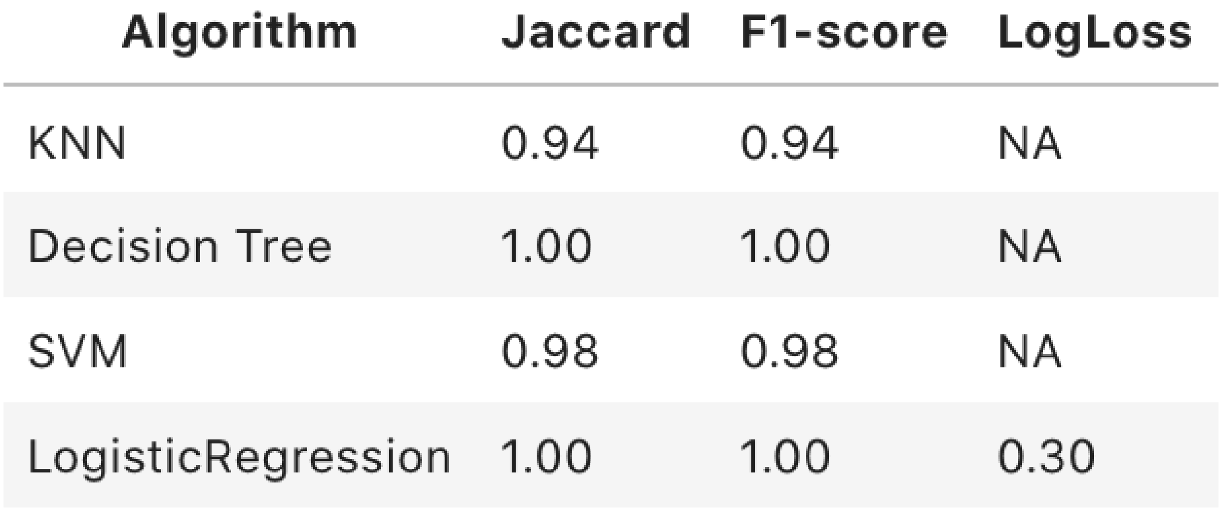
Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model.



```
[172]: yhatLOG = LogR.predict(X)
       yhatLOGproba = LogR.predict_proba(X)
       LogRJaccard = jaccard_similarity_score(y, yhatLOG)
       LogRF1 = f1_score(y, yhatLOG, average='weighted')
       Logloss = log_loss(y, yhatLOGproba)
       print("Log Loss: : %.2f" % Logloss)
       print("Avg F1-Score: %.4f" % LogRF1)
       print("LOG Jaccard Score: %.4f" % LogRJaccard)
```

```
Log Loss: : 0.30
Avg F1-Score: 1.0000
LOG Jaccard Score: 1.0000
```



Algorithm	Jaccard	F1-score	LogLoss
KNN	0.94	0.94	NA
Decision Tree	1.00	1.00	NA
SVM	0.98	0.98	NA
LogisticRegression	1.00	1.00	0.30

Discussion

The dataset after analysis was reduced by 194,103 rows to 570 rows to accurate the models with correct and valid data. The columns were reduced by 37 to 6 to focus on more accurate causes of severities. The columns were later increase to 34 to give the modules more values to predict the severity of an accident.

The first analysis was to understand the data with visualization and remove unnecessary columns that wasn't accurate enough. We also realized majority of the data had missing data/values that was needed for specific columns. It was decided to better accurate the model, every row in the dataset that had at least one missing value was removed. It did reduce the data drastically, but even with 570 rows of accurate data, it was enough information needed for the modules.

The second analysis was to downsize the data by resampling it. We only had two severity values and there was a huge gap in terms of how many accidents were in each severity, which made the data unbalanced. We resample the data by reducing the amount of accidents in severity (1) to match the exact amount in severity (2).

We also had to modify the dataset by removing the old columns and adding the values from those old columns as new columns with values of (0s & 1S), 0 meaning false and 1 meaning true. With the modified dataset, we were able to run multiple predictive models by splitting up a portion of the dataset, to allow the models test and run the data to result the accuracy of predicting the severity of an accident.

Conclusion

In this study, the information provided by the Seattle Police Department was well labeled with additional information for each attribute to better understand how each column can affect the severity an accident. They are many benefits of handling accident severity data, in a sense of knowing a specific area reputation of accidents to strategize against it. Building models that can predict the severity of an accident to effectively allocate the amount of resources needed. Base on the results of the models, all models were able to accurately predict base on the 30% of dataset used for testing. The best predictive model is the Logistics Regression model with a Jaccard and F1-score of 1.00. These types of models can be used on any dataset with the right steps to further improve different type of locations for a better understanding of driving safely and preventing more economic cost on the severities of accidents. The project can be further improved with more rows of data and accurate values in each column that affect the severity of an accident.

References

https://matplotlib.org/gallery/lines_bars_and_markers/barchart.html

https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn/

https://github.com/jason-ghub/capstone_project/blob/main/Capstone%20Project%20Notebook%20-%20JR.ipynb