Business Understanding:

Imagine the ability to anticipate whether an auto accident is likely and its severity that would allow the ability to save lives. The objective of the project is to predict the likelihood and severity of an accident so that the driver can adjust driving behavior and/or journey plan.

Data understanding:

The dataset to be used is the example data set provided in the capstone course. The data is both categorical and numerical. Data is available for accident location, event description, weather, and road conditions. Data is missing for several attributes.

Location will be the primary attribute used to determine likelihood of an accident occurring. Location is available in both description (intersection/block) and gps coordinates. To clearly communicate results, a map will be used.

Data Preparation:

Data was pulled from the course website. There were only 2 severity codes: 1 for personal property damage and 2 for bodily injury. Several attributes were missing data and/or had mixed entries that that meant the same thing (e.g. 0 and N, and 1 and Y). This data was cleaned up using pandas to make all attributes consistent. Attributes that were not relevant in predicting the severity were dropped. There were over 200,000 data points. After the data was cleaned up, there were 189,339 data points and broken up per the table below.

SEVERITYCODE	
1	132221
2	57118

To provide some insight into geographic location of the accidents, clusters were created using scipy and kmeans. Folium was used to show the areas the accidents clustered in. The graph below shows the clusters generated.

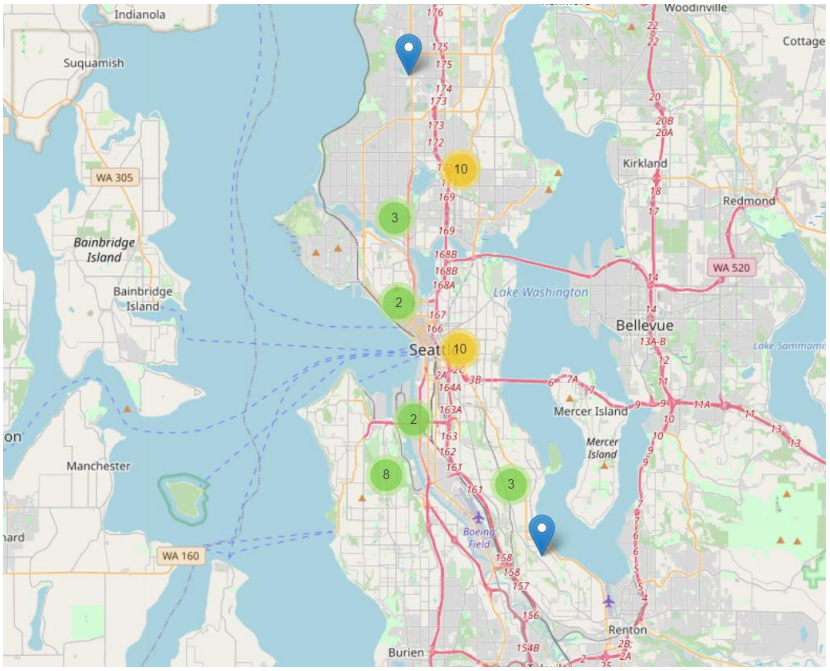

Figure 1: Map showing clustered data

The data was unbalanced as shown in graph below. To prevent a biased model, oversampling and under sampling were both investigated with results being very similar since over 50,000 data points are available for severity 2 at a minimum. The graph below shows the data after it was oversampled.

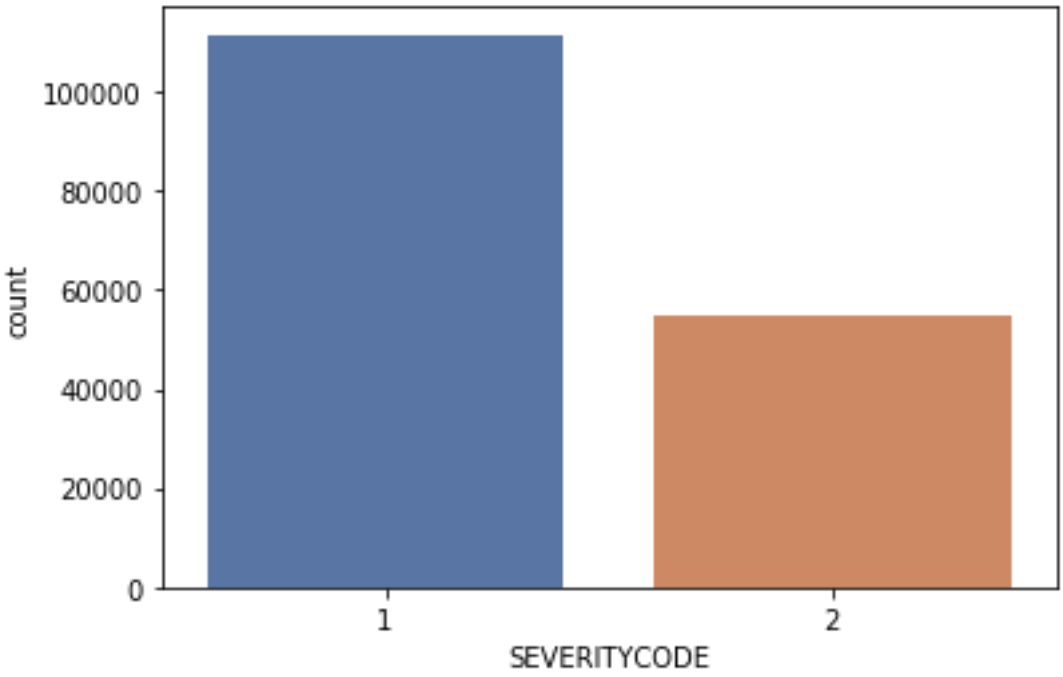

Figure 2: Unbalanced data

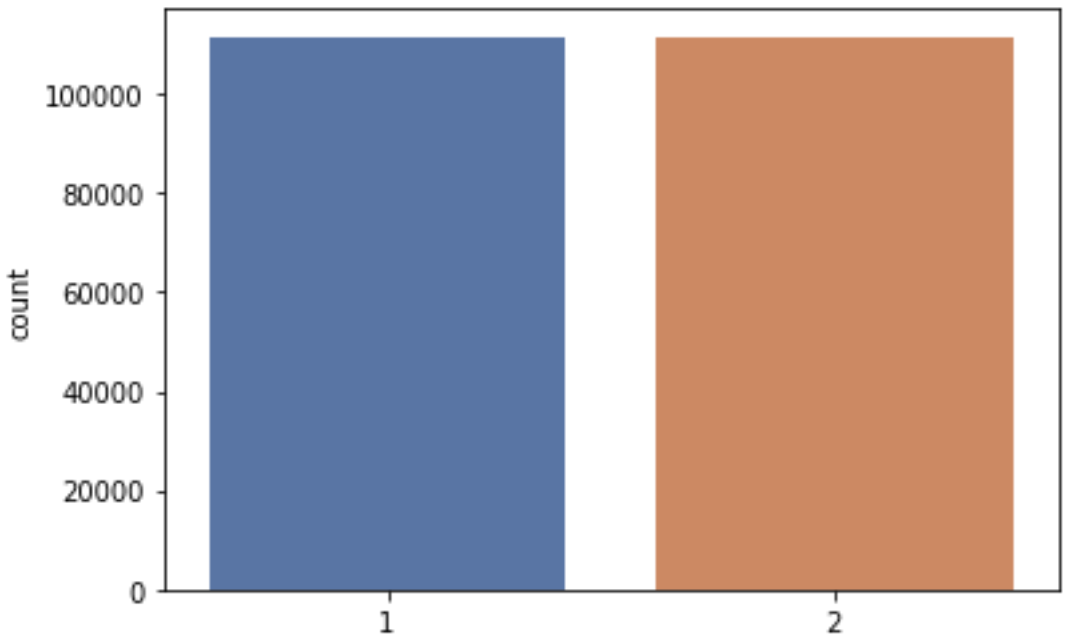

Figure 3:Balanced data by oversampling

Modeling:

The data was then encoded using sklearn's preprocessing laberencoder as some of the attributes were categorical. The oversampled/undersampled data was then split into training and testing sets and run trough the K Nearest Neighbors, Decision Tree, Logistic Regression, Naïve Bayes, and Random Forest algorithms. For the KNN model, the optimal K was determined to be K=6.

Evaluation:

The various metrics to compare the algorithms are demonstrated below. The models were very similar. Confusion matrices were also plotted and shown below.

	Jaccard	F1-score	Recall	Precision
K Neighbors	0.689362	0.694387	0.724108	0.71193
DecisionTree	0.706268	0.659339	0.72509	0.768958
LogisticRegression	0.704081	0.673087	0.727056	0.744422
Naive Bayes	0.69427	0.663835	0.717369	0.72101
Random Forest	0.692533	0.705986	0.730525	0.718708

Figure 4: Table showing model comparison metrics

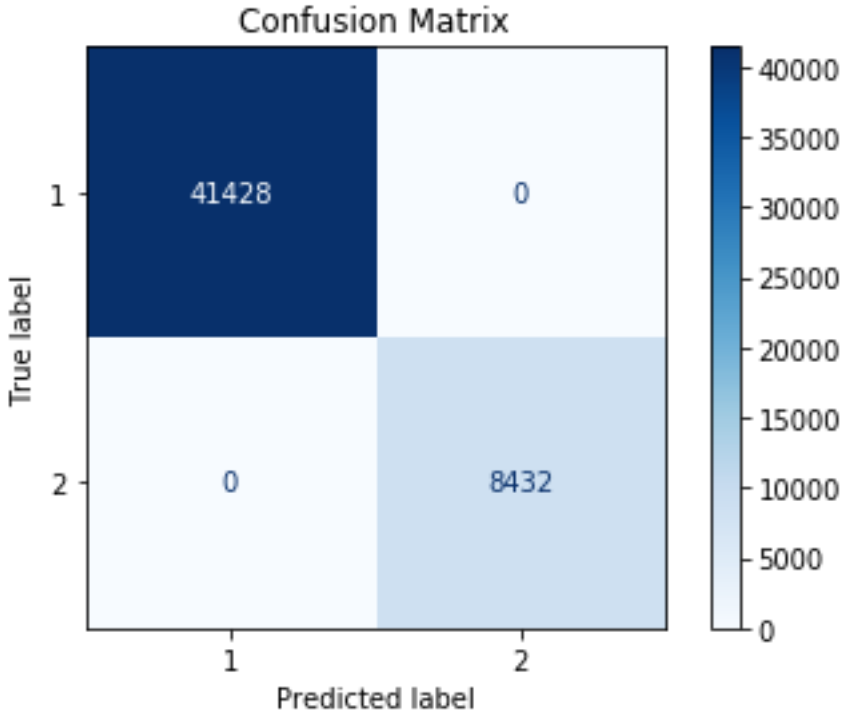

Figure 5:Confusion matrix for the KNN model

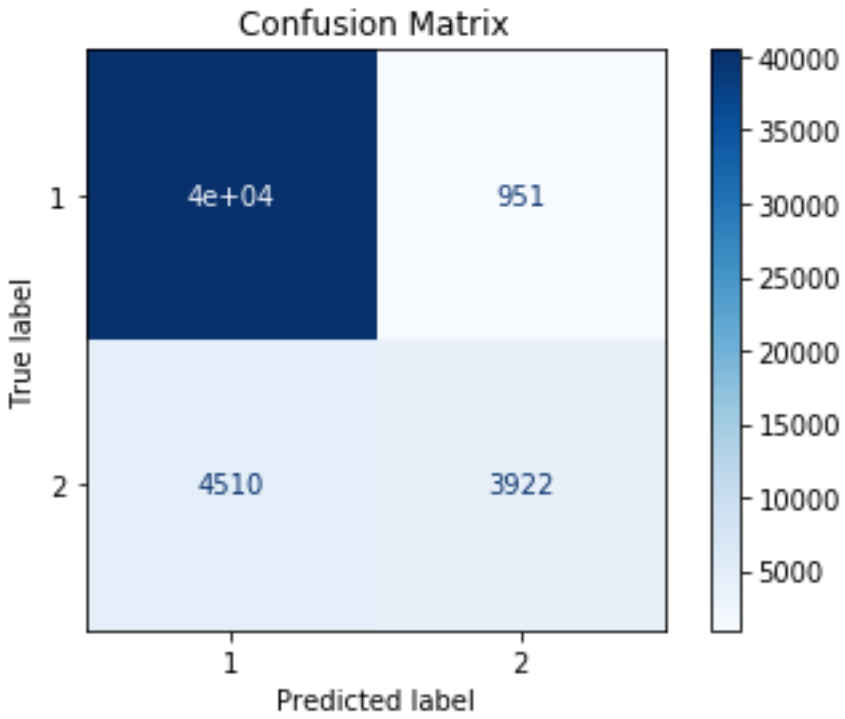

Figure 6: Confusion matrix for the logistic regression model

Deployment:

The plan for deployment would be to create an app that uses a weather API to provide information on the weather and road conditions. The user would have to upload conditions related to the car but imagine if the data from the car's computer could be auto uploaded. If possible, traffic conditions source could be used to provided additional information. With all the data available, the app would then run the data through the model to provide the user with guidance on what should be done to avoid risk and hazards and saving lives.