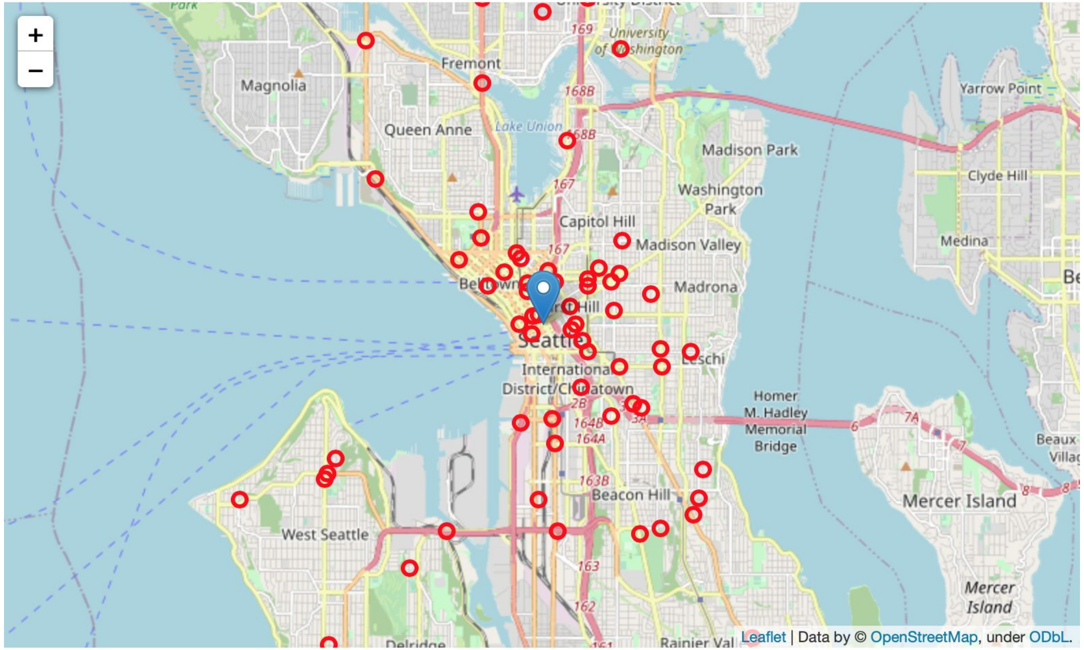
Predicting Severity of Collisions in Seattle

Relevance

- Seattle Police Department has collected data on collisions over the years;Unpredictable factors such as weather, road conditions, and light conditions are more difficult to regulate
- Collisions drastically impact traffic flow and the safety of drivers
- Audience: SPD and the Emergency Medical Services to predict the severity of the collisions based on the conditions present and deploy the necessary protocols.

Location of Collisions



Data Acquisition and Cleaning

- Data collected from the Seattle Geo Data open source website; dataset contains records of collision from 2004 to present
    - Consists of 194,673 observations and 38 features
- Features that weren't used to develop the machine learning models were removed
- Cleaned dataset contained 17 features.
- The target variable for the study was determined to be the severity of the collisions, which consisted of two severity codes: property damage (1) or injury (2).

Dealing with Imbalanced Dataset

- After removing irrelevant features from the dataset, the distribution of the target variable was determined.

```
1    136485
2     58188
Name: SEVERITYCODE, dtype: int64
```

- The dataset was heavily biased and therefore would cause the models to provide biased predictions
- Resampling was established, specifically undersampling from the majority class which was severity code 1.

Determining Independent Features

- Determined the breakdown of different kinds of conditions for the light of day, road conditions, and weather feature
- Features such as the road condition and weather were ultimately chosen since they seemed to have more impact on the severity of the collisions compared to the light condition

```
Dry               76076
Wet               29266
Unknown            6819
Ice                 690
Snow/Slush          561
Other                71
Standing Water       62
Sand/Mud/Dirt        46
Oil                  36
Name: ROADCOND, dtype: int64
```

```
Clear                        67977
Raining                      20494
Overcast                     16865
Unknown                       6842
Snowing                        503
Other                          434
Fog/Smog/Smoke                 359
Sleet/Hail/Freezing Rain        66
Blowing Sand/Dirt               32
Severe Crosswind                15
Partly Cloudy                    4
Name: WEATHER, dtype: int64
```

Feature Preparation

- After determining independent features for the study, each feature was extracted from the dataset.
- One hot encoding was utilized to convert the categorical variables in each feature to binary variables
- The features were then normalized to standardize them and were used to train the chosen classification models one at a time.

Classification Models for Each Feature

- K Nearest Neighbor
  - Classification of cases based on similarity with other cases
  - The best number of nearest neighbors to consider for the model was found by determining the accuracy of the predictions
- Decision Tree
  - Determination of the depth of the decision tree model by determining the accuracy of the predictions
- Logistic Regression
  - Using a small inverse of the regularization parameter in order to avoid overfitting the model

K Nearest Neighbor Model Development

```
from sklearn.neighbors import KNeighborsClassifier
```

```
# Finding the best k for the model
Ks=15
mean_acc=np.zeros((Ks-1))
std_acc=np.zeros((Ks-1))
for n in range(1,Ks):

    #Train Model and Predict
    kNN_model = KNeighborsClassifier(n_neighbors=n).fit(X1_train,y1_train)
    yhat = kNN_model.predict(X1_test)
    mean_acc[n-1]=np.mean(yhat==y1_test);
    std_acc[n-1]=np.std(yhat==y1_test)/np.sqrt(yhat.shape[0])
mean_acc
```

```
array([0.55686618, 0.54513623, 0.5637816 , 0.55305832, 0.56340956,
       0.55491848, 0.56651712, 0.55846373, 0.56546668, 0.56067403,
       0.5647445 , 0.56159317, 0.56305942, 0.55990809])
```

Figure: model development for the weather feature

Decision Tree Model Development

```
from sklearn.tree import DecisionTreeClassifier
Ks = 10
mean_acc1 = np.zeros((Ks-1))
for n in range(1,Ks):
    #Train Model and Predict
    tree_define = DecisionTreeClassifier(criterion="entropy", max_depth = n).fit(X1_train,y1_t
rain)
    y_hat2 = tree_define.predict(X1_test)
    mean_acc1[n-1]=np.mean(y_hat2==y1_test);
mean_acc1
```

```
array([0.54415144, 0.54415144, 0.54526753, 0.54520188, 0.53977459,
       0.54870336, 0.54410767, 0.5469745 , 0.54835321])
```

Figure: model development for the weather feature

Logistic Regression Model Development

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import log_loss
LR1 = LogisticRegression(C=0.1, solver='liblinear').fit(X1_train,y1_train)
yhat_test3 = LR1.predict(X1_test)
yhat_prob = LR1.predict_proba(X1_test)
print("LogLoss: : %.5f" % log_loss(y1_test, yhat_prob))
```

```
LogLoss: : 0.67225
```

Figure: model development for the weather feature

Metric Evaluation for Each Feature

Algorithm	Jaccard	F1-Score	LogLoss
KNN	0.563782	0.563638	NA
Decision Tree	0.546209	0.543917	NA
LogisticRegression	0.548922	0.494079	0.672254

Figure: report for the Weather feature

Algorithm	Jaccard	F1-Score	LogLoss
KNN	0.584121	0.582869	NA
Decision Tree	0.559042	0.492537	NA
LogisticRegression	0.556416	0.494928	0.671102

Figure: report for the Road Condition feature

Discussion

- K Nearest Neighbor was the most accurate model in terms of predicting the target variable when referring to the Jaccard Index and F1-Score
- The road condition feature was a slightly better predictor compared to the weather feature when comparing the Log Loss metric
    - The Log Loss metric measures how far the prediction is from the actual label
- Accuracy values for the features are not as high as expected, which might be caused by the random resampling of the data used before feature extraction.

Conclusion and Recommendations

- Analyzed impact of features on the prediction of the severity of collisions in Seattle such as property damage or injury to parties involved
- Three classification models developed for each feature and were evaluated using three evaluation metrics
- Additional model development necessary using different algorithms to improve the prediction of the target variable
    - However, models in the study suggest that road conditions are a better predictor of predicting potential collisions
- Recommendation: allocate resources to warn drivers about road conditions so they can plan accordingly.