Predicting Car Accident Severity

IBM Data Science Certificate

GeoTsa

23/09/2020

Predicting car accident severity as an element of a “Safe System”

- The number of people killed in road crashes around the world continues to increase
- UN and WHO policy framework and approach: **"Safe System"**: safe vehicles, safe infrastructure, safe road use (speed, sober driving, belts and helmets)
- Could predicting severity be another element?

Predicting, not describing and analysing

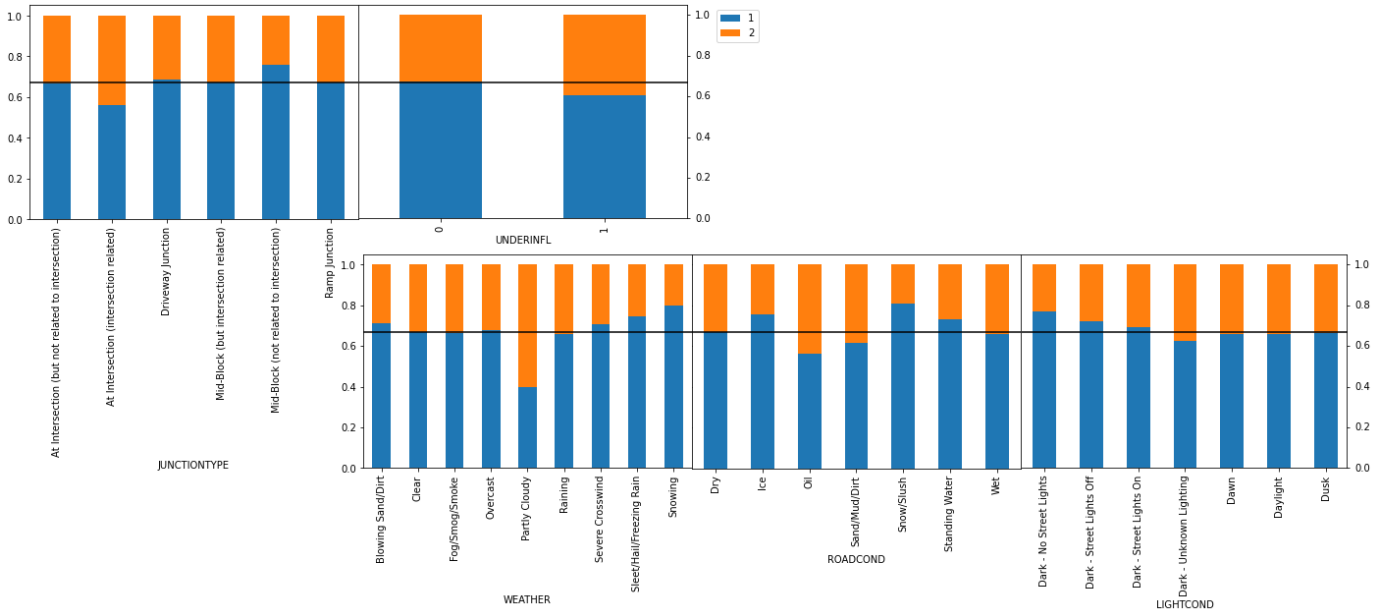
- We are not interested in effects or long term causes and conditions
- We are interested in the direct causes and conditions:

  location coordinates — date — time — junction type — inattention ivolved — driver under drugs/alcohol — influence — weather, road and light conditions — speeding involved — "pedestrian right of way" not granted

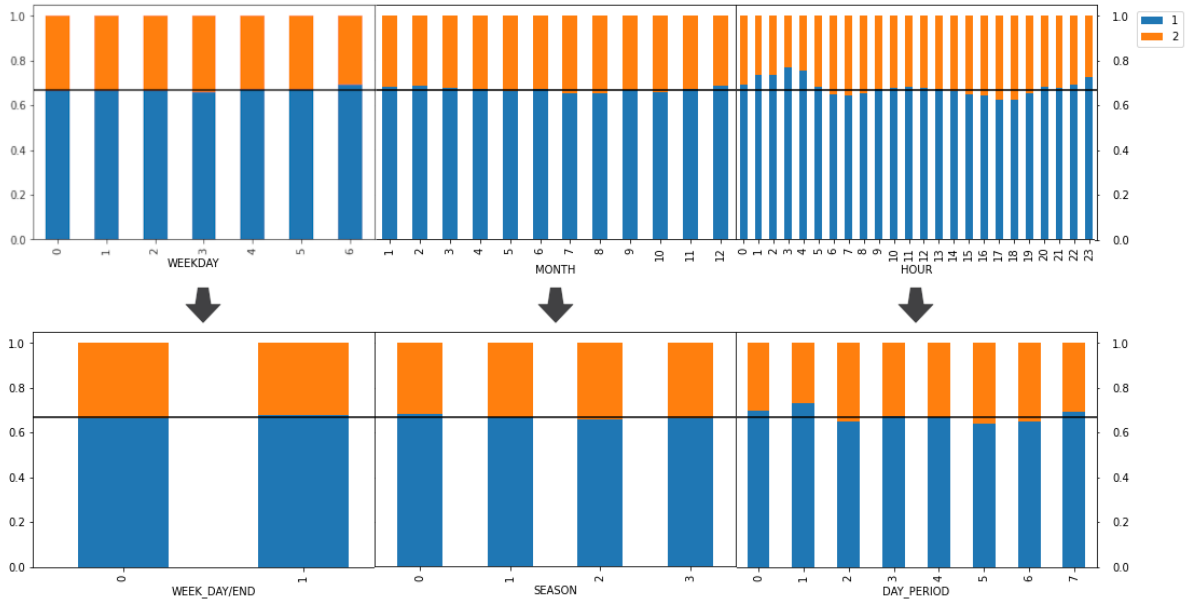
Data acquisition and cleaning

- Our dataset has been updated weekly by the Seattle Department's of Transport "Traffic Management Division"
- Its data come from the Seattle Police Department Traffic Records and record all types of collisions from 2004 to May 2020.
- In total, 194,673 rows, 36 features and 1 target variable (severity: 0/1)
- Duplicate, NaN and highly correlated features were dropped.
- Cleaned data contains 164,726 rows and 11 columns.

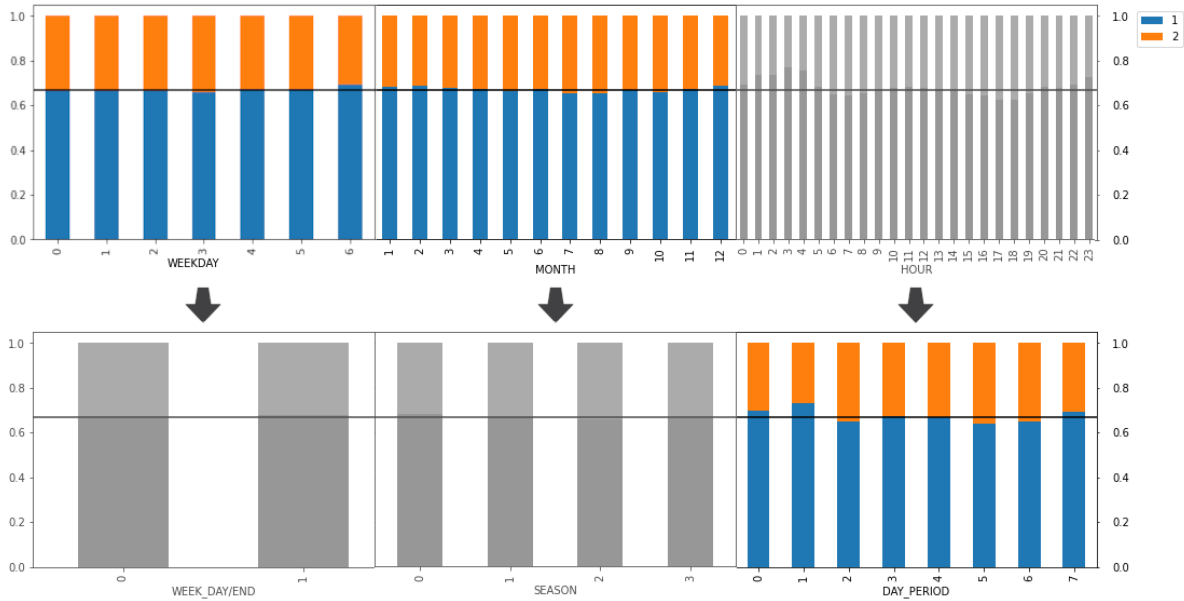
Bivariate analysis



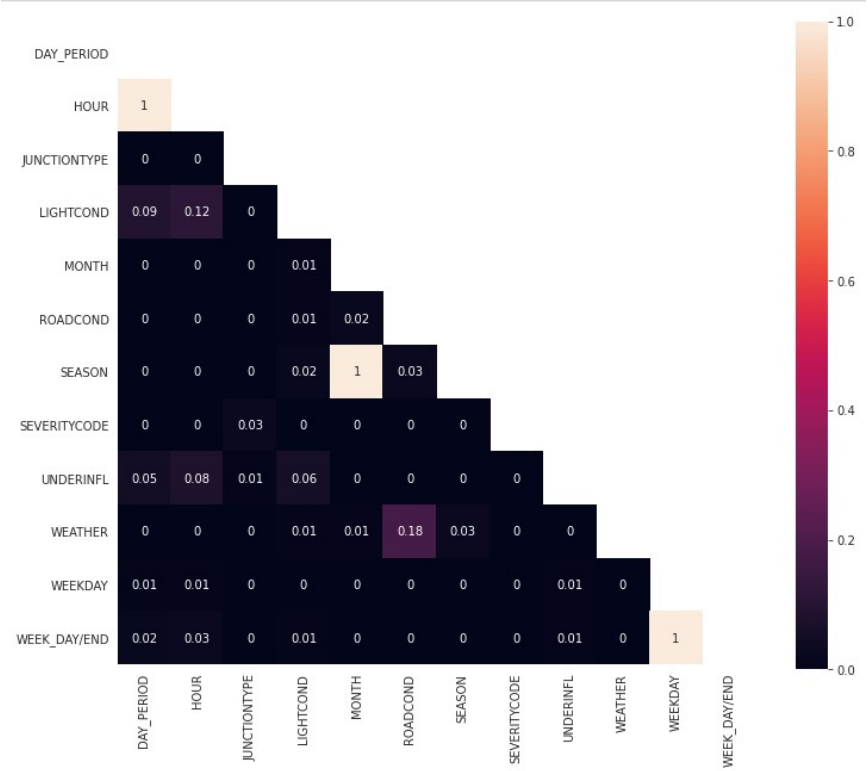
… and some feature extraction...



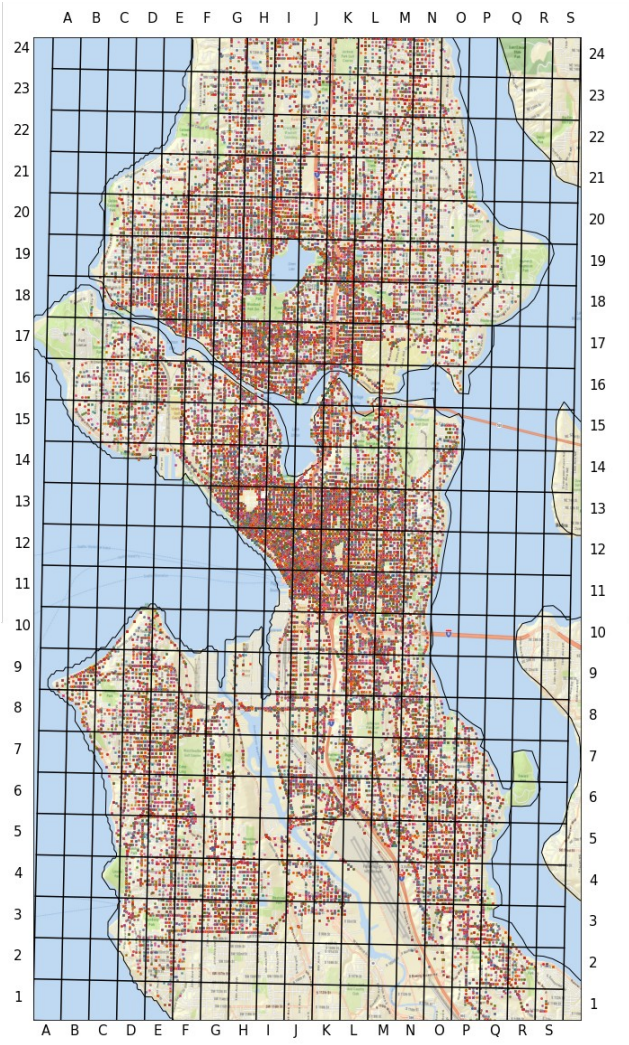
… and feature selection …



… based on Correlation Analysis



- Crammer’s V Correlation
- Based on Pearson's chi-squared statistic (Harald Cramér, 1946)
- Measure of association between two categorical variables [0, 1]
- We base our feature selection on:
  - variable with less Crammer's V Correlation with the rest of the variables
  - variable with more Crammer's V Correlation with the target variable (SEVERITYCODE)



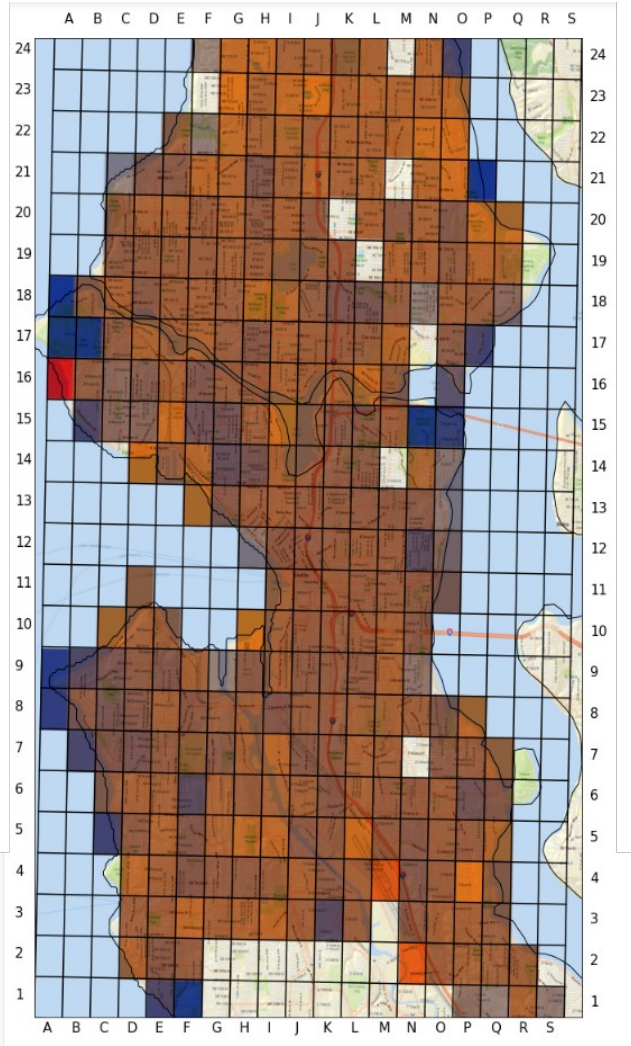
How to involve Location Attribute

1) Visualize and Grid the map of the accidents

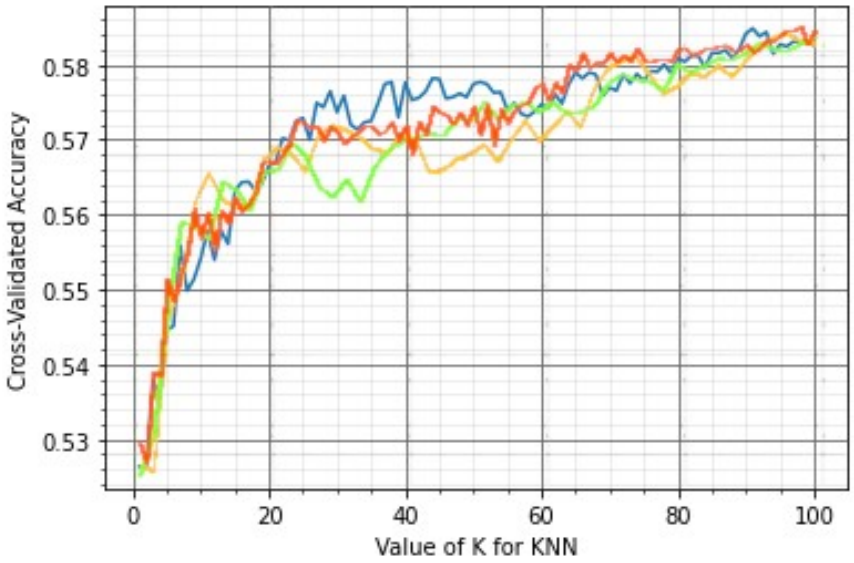
2) Calculate the probability for each section

3) Create 2 new features:
- XY: these cartesian sections
- Distance from city center (bins)

4) Create 4 feature sets:
- with both these features (both)
- only with the one of two (xy, dist)
- with none of these features (none)

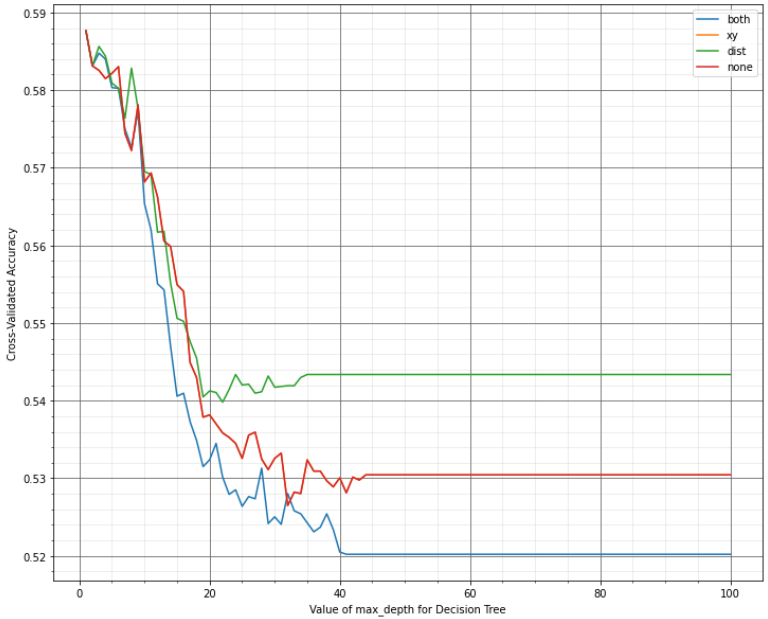


K-nearest neighbor classifiers (knn)



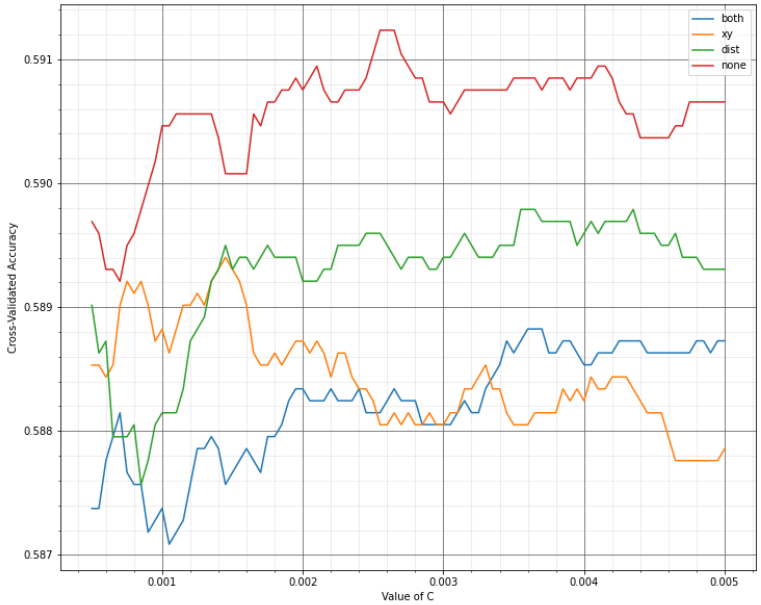
- Test for k=1...100 and evaluate based on performance (accuracy) with elbow method
- both: k = 32
- xy: k = 24
- dist: k = 23
- none: k = 30

Decision tree algorithm



- Test for max_depth=1...100
- Performance decrease
- Overfitting

Logistic Regression Algorithm



- Test for C∈(0,1] and evaluate based on performance (accuracy)
- Test thoroughly for C∈[0.0005,0.005]
- both: C = 0.00360
- xy: C = 0.00145
- dist: C = 0.00355
- none: C = 0.00255

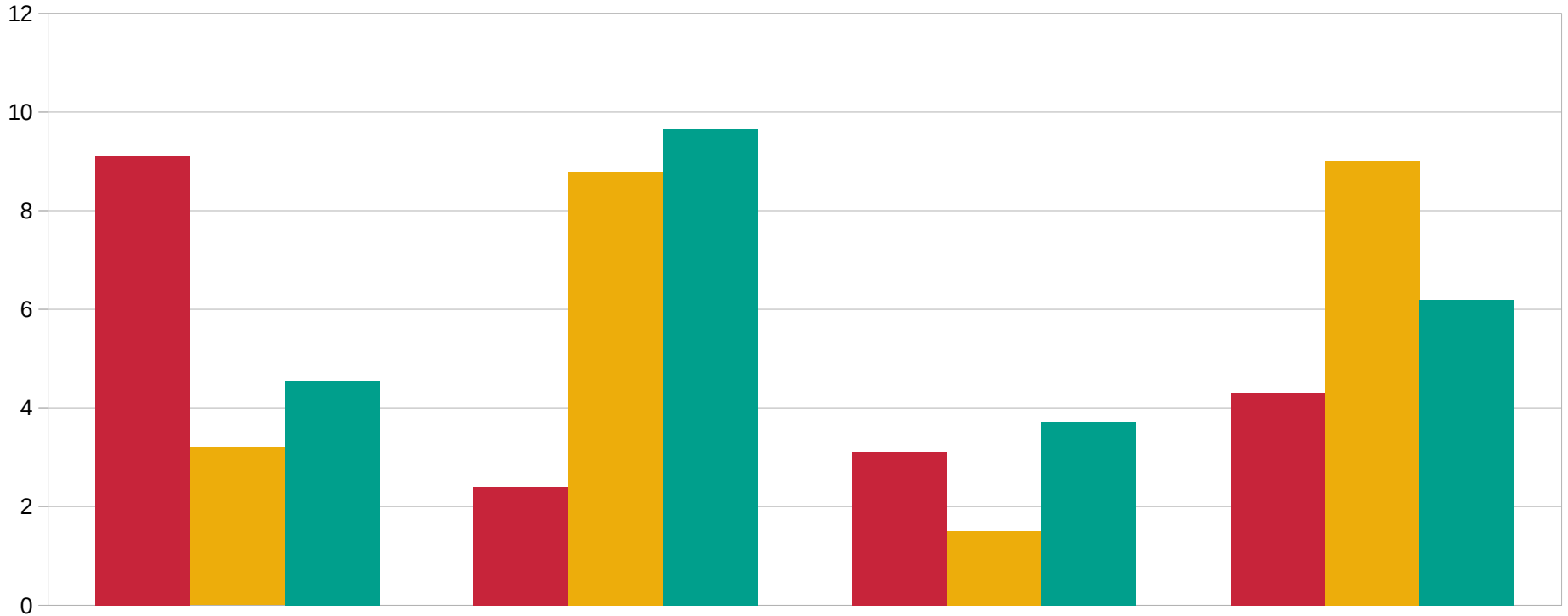
Classifiers Evaluation

Algorithm	Feature Set	Accuracy	Jaccard	F1-score	LogLoss
KNN	both	0.59	0.59	0.59	NA
KNN	xy	0.57	0.57	0.56	NA
KNN	dist	0.57	0.57	0.57	NA
KNN	none	0.57	0.57	0.57	NA
SVM	both	0.59	0.59	0.59	NA
SVM	xy	0.59	0.59	0.59	NA
SVM	dist	0.59	0.59	0.59	NA
SVM	none	0.59	0.59	0.59	NA
Logistic Regression	both	0.59	0.59	0.59	0.67
Logistic Regression	xy	0.59	0.59	0.59	0.67
Logistic Regression	dist	0.59	0.59	0.59	0.67
Logistic Regression	none	0.6	0.6	0.6	0.67

- Evaluate with hold-out test dataset
- Performance metrics:
  - Accuracy
  - Jaccard co-ef
  - F1-score
  - LogLoss
- Logistic Regression

Prediction possible, WORK NEEDED

- Very low performance: 60%
(Blind Guess performance is 50%)
- Need for better exploration of all variables/features,
especially for **location** parameter
- Work on Clustering based on location and severity


12
10
8
6
4
2
0