

# LOOK BEFORE YOU LEAP

Leveraging predictive models to improve  
Automotive Safety and Travel time

# Introduction

Recent growing amount of vehicles and traffic has been slowing down intensive road users, which costs over 10% of the logistic drivers' working hours. Human-based reports have high labor cost with significant delays, and current automatic incident detection algorithms have difficulties in reaching high accuracy due to unexpected circumstances and expensive costs.

**Our objective is to find an optimal traffic predicting algorithm using real-time and comparatively inexpensive Twitter data to avoid congestion and optimize time cost for many people in daily lives.**

# Member Roles

## *Kamil*

Generation of prediction scores from various ML models, cross validating predictions on data and implementing hyperparameter tuning to all ML models.

Implementation of multithreaded algorithms for data and prediction processing. Research into State of the art models and papers and algorithm evaluation on datasets.

Testing and evaluation of ML Models on different sets of predictors, and eliminating models which yielded no results or were incompatible. Implementation of the XGBoost model into the project.

## *Hannah*

Implementation of collecting the datasets, feature selection from traffic accident record and retrieval of tweets through Snscape, Tweepy, and geocoding through ArcGIS.

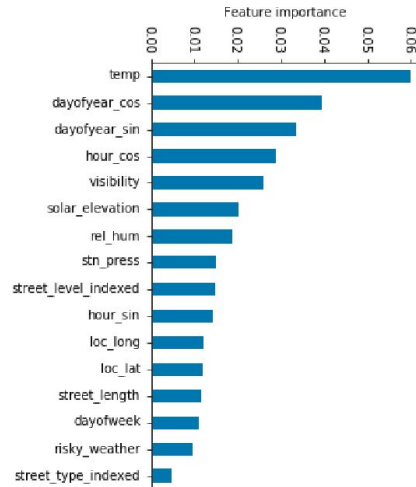
Processing the tweet texts through various NLP and Vader methods (sentiment analysis) and converted the tweet dataframe into a frequency table to integrate into traffic accident records.

Merged the two datasets based on the distance between instances and used SMOTE and random instance generation to balance the class.

# State of the art

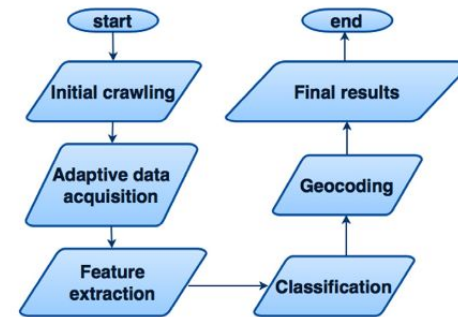
“High-Resolution Road Vehicle Collision Prediction for the City of Montreal” (Antoine Hébert et al., 2019)

Predicted risk of accidents with high spatiotemporal resolution over a large area and time, allowing for useful identification of significant risk factors. This extended the state of the art by dynamically combining three large datasets, doing feature engineering for imbalanced classes, and implementing BRF in Spark.



“From Twitter to detector: Real-time traffic incident detection using social media data” (Gu et al., 2016)

Processed real-time tweets in order to retrieve possible traffic accident areas via NLP methods



**Fig. 1.** The work flow of Twitter data acquisition, processing and analytics.

The methodology was applied in two regions, Pittsburgh and Philadelphia Metropolitan Areas, and the discrepancy between public Twitter accounts and individual accounts was measured.

# Approach

## 1. Processing of tweets from public Twitter accounts to retrieve possible accident-prone geolocations

SNScrape, Tweepy - 511NY

Language Processing through NLTK packages

Geocode - retrieve location through ArcGIS platform

	tweet	separate camel cases	replace acronyms	remove stopwords	longitude	latitude
12	Cleared: Construction on #NY30 Both directions...	Cleared Construction on NY30 Both directions...	cleared construction on new york 30 both di...	cleared construction new york 30 directions ne...	-1.045049	53.880315
13	Incident on #Q48Bus from Roosevelt Ave: Main S...	Incident on Q48Bus from Roosevelt Ave Main S...	incident on q48bus from roosevelt ave main s...	incident q48bus roosevelt ave main st roosevel...	-73.588960	40.678550
14	Incident on #US9 Both directions at County Rou...	Incident on US9 Both directions at County Rou...	incident on U.S. route 9 both directions at...	incident U S route 9 directions county route 3...	-73.917929	40.868738
16	Updated: Incident on #BM3 Both directions at 5...	Updated Incident on BM3 Both directions at 5...	updated incident on bm3 both directions at 5...	updated incident bm3 directions 57 st 2nd ave ...	-74.007140	40.714550
17	Updated: Incident on #M101Bus at Second Ave ht...	Updated Incident on M101Bus at Second Ave ht...	updated incident on m101bus at second ave ht...	updated incident m101bus second ave https co u...	-73.988656	40.727283
19	Cleared: Construction on #92Bus Both direction...	Cleared Construction on 92Bus Both direction...	cleared construction on 92bus both direction...	cleared construction 92bus directions scotland...	-73.976250	40.621120
20	Cleared: Construction on #Q32Bus at E 60th Str...	Cleared Construction on Q32Bus at E 60th Str...	cleared construction on q32bus at e 60th str...	cleared construction q32bus e 60th street lexi...	-73.970674	40.764026
21	Incident on #BM3 Both directions at 57 St and ...	Incident on BM3 Both directions at 57 St and ...	incident on bm3 both directions at 57 st and ...	incident bm3 directions 57 st 2nd ave https co...	-74.007140	40.714550
24	Incident on #Q15Bus from Roosevelt Ave: Main S...	Incident on Q15Bus from Roosevelt Ave Main S...	incident on q15bus from roosevelt ave main s...	incident q15bus roosevelt ave main st roosevel...	-73.588960	40.678550
25	Incident on #BrooklynBridge EB at Brooklyn Bou...	Incident on Brooklyn Bridge EB at Brooklyn Bo...	incident on brooklyn bridge east bound at...	incident brooklyn bridge east bound brooklyn b...	29.769650	-24.868210

*Text processing (left to right) - separation of camel cases, acronym replacements, removal of stopwords, and location retrieval through geocoding : returning longitude and latitude values*

# Approach

## 2. Addition of Twitter data as a new feature set to the previous traffic record

The existing traffic record was compared with the given Twitter geolocations. For each location (instance) in traffic record, the level of traffic is added as a new feature by calculating the distance between each location and its proximity to neighboring accident-prone areas.

	longitude	latitude	Frequency	polarity	neutral
11	-73.958048	40.619211	2	0.0000	1.000
10	-73.968886	40.672875	2	0.0000	1.000
9	-73.977580	40.752125	1	0.1027	0.877
15	-73.800325	40.703240	1	0.4588	0.800
14	-73.934931	40.713819	1	0.0000	1.000
13	-73.946861	40.590935	1	0.0000	1.000
12	-73.949580	40.650100	1	0.1027	0.877
0	-74.068302	40.600070	1	0.0000	1.000
1	-74.046500	40.606030	1	0.1027	0.833
7	-74.007328	40.654323	1	0.0000	1.000
6	-74.009464	40.762307	1	0.5267	0.672

*Location frequency table (twitter features) retrieved from previous twitter collection*

COLLISION_ID	VEHICLE TYPE CODE 1	VEHICLE TYPE CODE 2	distance	severity	polar sentiment	neutral sentiment
4371841	Sedan	NaN	10000	0.0	0.000000	0.000000
4371752	Sedan	NaN	206	42.0	0.000102	0.999867
4371339	Taxi	Sedan	1346	4.0	0.000000	1.000000
4371325	Sedan	NaN	61	52.0	0.103920	0.817364
4371303	Sedan	NaN	333	55.0	0.075074	0.883583
...	...	...	...	...	...	...

*Twitter features added to traffic record based on distance between instances (long, lat values) : distance, severity, neutral sentiment, target*

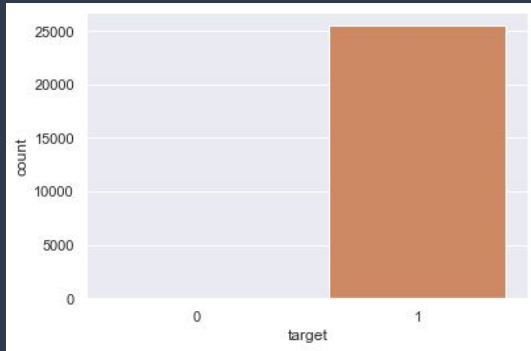
# Approach

## 3. Addressing Data Imbalance

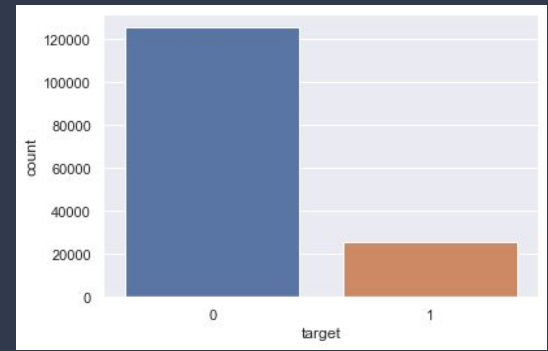
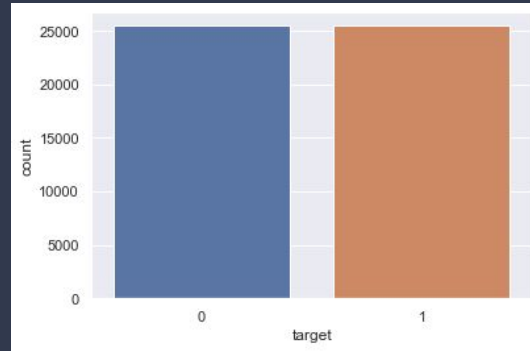
Given the large imbalance between classes (accident > non-accident instances), we carefully rebalanced these classes through random sampling of the negative class (SMOTE)

## 4. Evaluation on ML Algorithms

Previous work has shown that road accident risk is best predicted by models built with either decision tree-based algorithms or deep learning. We focused on classical machine learning techniques, Random Forest and its balanced variants, along with alternative machine learning models.

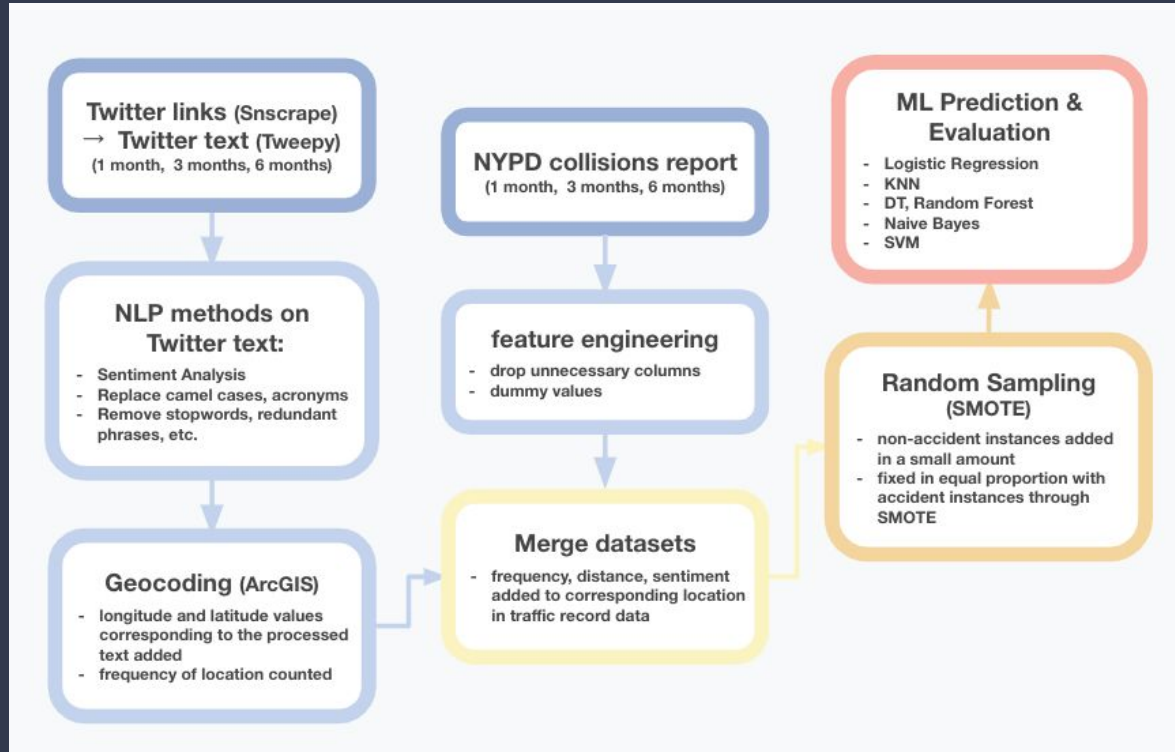


*Before and after SMOTE oversampling of the minority class :  
the result (right) shows equal number of target 0s and 1s after oversampling*



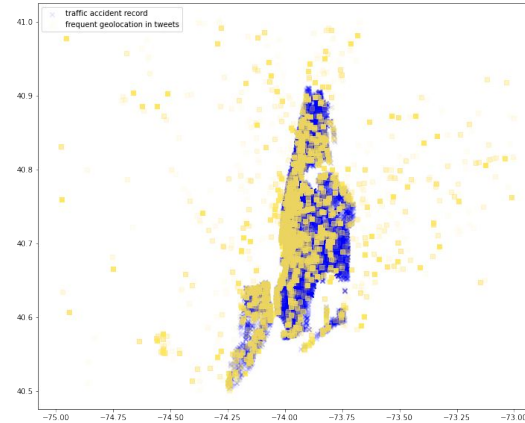
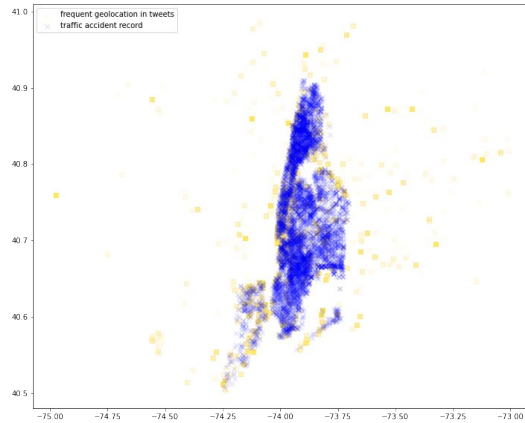
*Larger portion of negative instances (target 0)  
added to SMOTE data*

# Approach : summary





# Experiments & Evaluation



*Scatterplot of traffic accident record (blue) and twitter geocoded locations (yellow)*

# Experiments & Evaluation

	1 month			3 months			6 months		
ML models	accident 1:1	twitter smote 1:1	twitter negative boost 4:1	accident 1:1	twitter smote 1:1	twitter negative boost 4:1	accident 1:1	twitter smote 1:1	twitter negative boost 4:1
kNN	1	1	1	1	1	1	0.967749	0.999054	0.971667
SVC	1	1	1	1	1	1	0.865009	0.899182	0.958956
Gaussian NB	1	0.998102	0.993450	1	0.999667	0.999841	0.907693	0.988164	0.992940
Decision Tree	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Random Forest	1	1	1	1	1	1	0.957593	0.999741	0.988900
Logistic Regression	0.547829	0.892255	0.904576	0.581027	0.957114	0.925400	0.484491	0.973014	0.941
XGBoost	1	1	1	1	1	1	0.951163	0.999283	0.998666

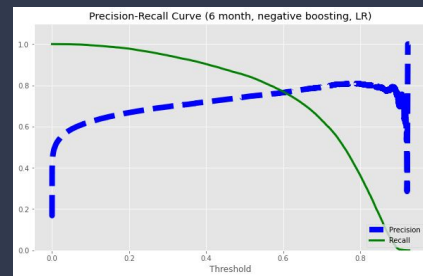
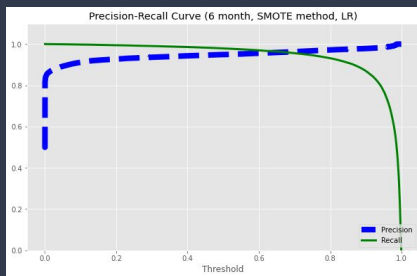
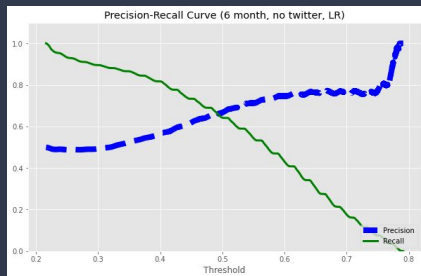
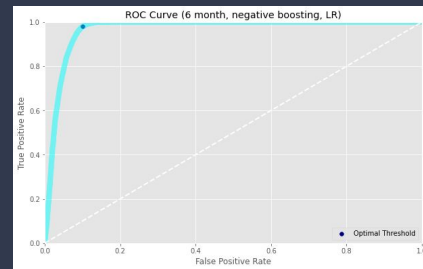
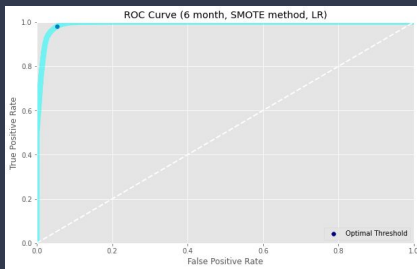
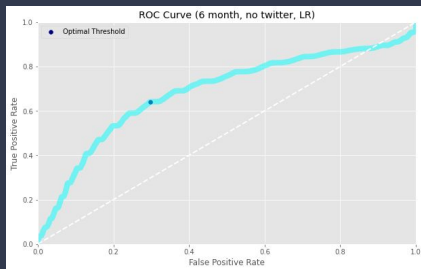
*Table 1. Accuracy of prediction through different machine learning models performed on extracted data :  
1 month, 3 months, and 6 months record of accident record, twitter dataset with smote method and negative instance boosting (4:1 proportion)*

# Experiments & Evaluation

	3 months			6 months		
	no twitter 1:1	smote 1:1	negative boost 4:1	no twitter 1:1	smote 1:1	negative boost 4:1
<i>precision</i>	0.502156	0.933666	0.751797	0.676999	0.947914	0.787041
<i>recall</i>	1.000000	0.986144	0.852816	0.653652	0.979288	0.885943
<i>f1-score</i>	0.668580	0.959188	0.799127	0.665121	0.963346	0.833569

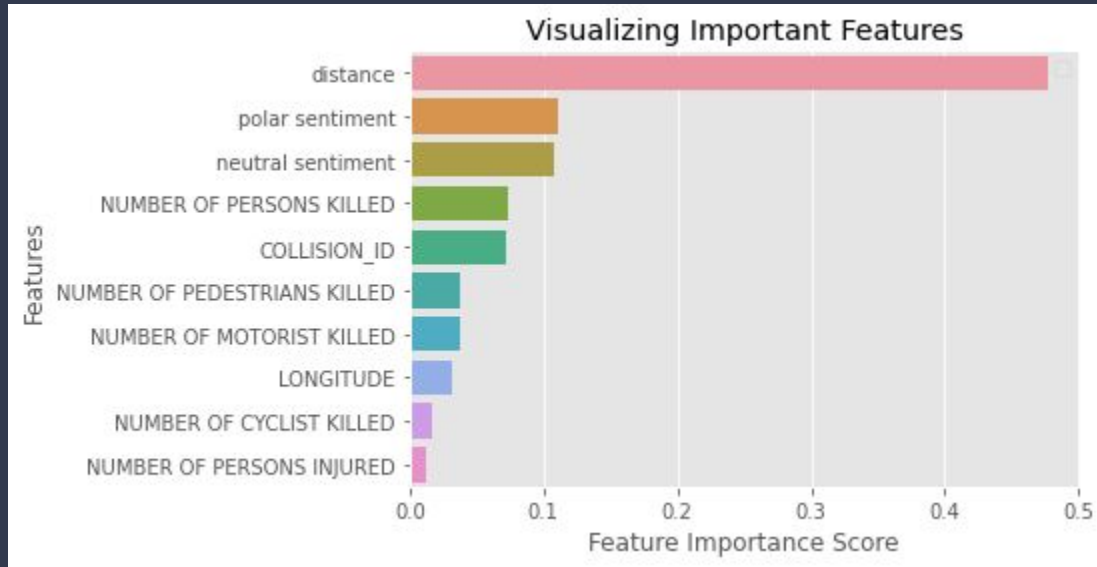
*Table 2. Evaluation of Logistic Regression model on different types of datasets, precision, recall, and f1-scores were measured*

# Experiments & Evaluation

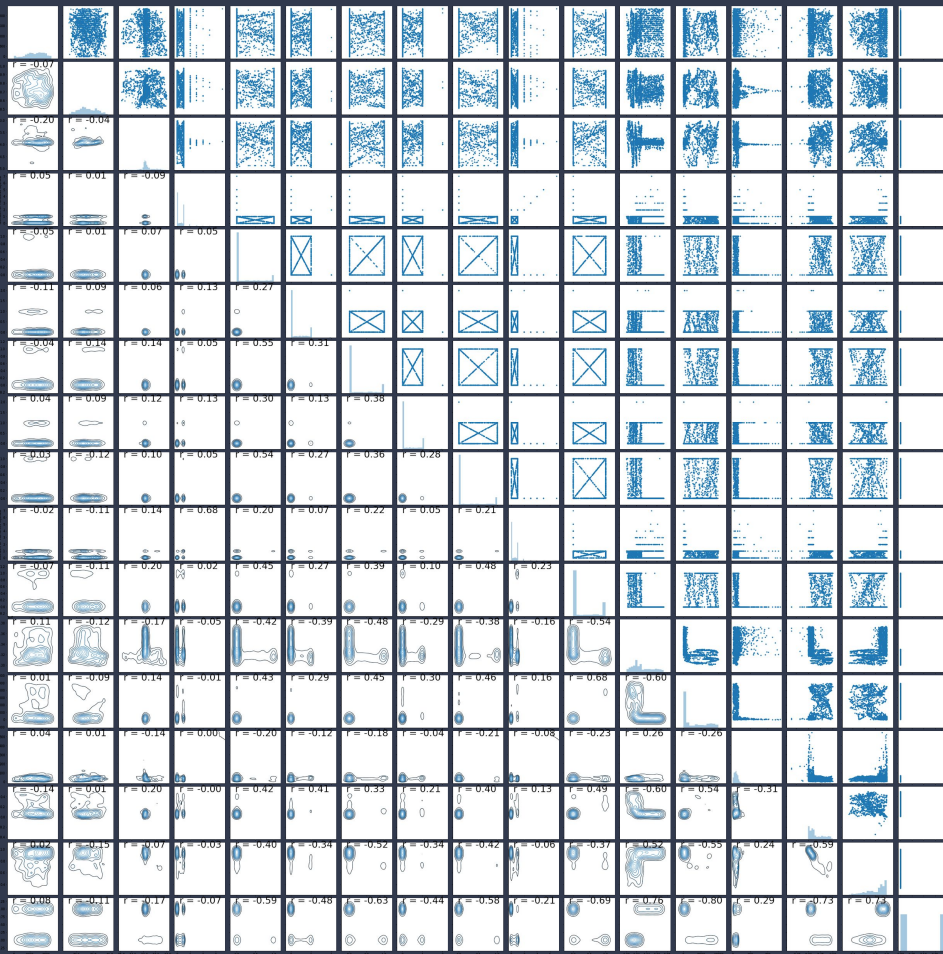


*ROC Curve and Precision-Recall Curves for according datasets :  
no twitter; smote, negative boosting / 6 months*

# Experiments & Evaluation



*Feature importance scores via Random Forest function  
negative boosting / 6 months*



## Experiments & Evaluation

*Seaborn Pairgrid with Pearson  $r$  scores :*  
*shows many correlation between the features - many*  
*of the  $r$  values are close to 0.5 or larger*

# Discussion

## 1. Dataset Collection - Missing data

Our initial choice of accident record was the Countrywide Traffic Accident (Moosavi et al., 2019) - however big portion of Queens data was missing. Switched to NYC Open Data, however it did not have weather or road infrastructure features - and there were many missing values in which we had to drop

Geolocation of 511 NY tweets distribution had slight disparity with the traffic record data - may need clearer text processing and understanding the context of the tweets

## 2. Class Imbalance - Adjustments

Having only one instance for binary classification, the negative instances were created randomly from a given range. SMOTE method was to amplify the negative instances to equal amount with the positive instances.

The proportion of the classes had to be adjusted due to the possibility that the created instances can be a noise to the dataset. We set the proportion to 1:4 temporarily, and it still showed significance in improving prediction scores compared to accident dataset without the Twitter features

# Lesson learned about ML topics

## 3. Difference in performance of machine learning models

Six types of ML models were run on different datasets - accident record data, Twitter feature added data with SMOTE method and random negative instance generation - the accuracy score of the models were measured

- a. Models with initial high accuracy scores - kNN, Random Forest, XGBoost  
Random Forest and XGBoost being tree-based learners are able to de-correlate the features to collect votes from different subsets or instances to improve prediction accuracy, kNN algorithms are similar in collecting votes from k nearest neighbors and branching out. These models had similarity in working efficiently for non-linear classification.
- b. Models with initial average accuracy scores - SVC, Gaussian NB  
Naive Bayes having strong assumption on datasets may have lowered the performance as many features in our dataset are dependent on each other, and as SVC depends on having enough feature sets or variable to create a linear equation, in which addition twitter features seem to boosted the performance.
- c. Model with greatest difference in accuracy scores - Logistic Regression  
Logistic Regression, having characteristic of creating a linear equation for classification seemed to require a set of core variables that would determine the performance. The initial score was quite low due to lack of features - 0.6, however adding the twitter increased the accuracy score up to around 0.9.



# Challenges faced

## **Bottlenecks that slowed down the process**

- Text conversion through Tweepy (Twitter API rate limit)
  - worked around by using multiple Twitter developer accounts
- Geocoding the text through ArcGIS
- Running ML models - especially SVC
  - utilized multithreading for faster performance

## **Barriers that can be possibly overcome in the future**

- Noise in the text that lowered the geocoding accuracy
  - geocoding of bus stops in the tweets (integration of NYC bus stop dataset)
- Balancing the dataset for realistic simulation of accident risks
  - additional features of weather & road infrastructure

**Thank you.**