Objective:

Analysis of tweets in terms of aptly identifying whether it is an emergency needing Police actions as first respondent or otherwise.

We followed a systematic sequential approach towards this design problem, as follows.

1. **Data Collection:**
   The social media used here is twitter. We readily get the twitter package available in R to extract the tweets relevant for our analysis. The positive examples in this context comprise of the tweets pertaining to bad weather conditions like snow, high winds and the subsequent traffic disruptions, alerts issued which can potentially be considered as "Emergency"-like situations needing police intervention as first respondents. The data has been sourced from various UB related twitter handles like @UBuffalo, @UBStudentExp, @UBAthletics and the positive examples have mainly been sourced from @UBuffaloPolice. For want of enough positive examples, we limited the ratio of negative to positive instances to around 10:1. The total instances count being around 200 tweets.

2. **Feature Generation:**
   This involved extracting the terms and forming the feature space using Bag of Words approach. We formed the corpus from the instances we formed as datasets. The data preprocessing involved various cleaning steps like removing punctuations, url, special characters, stemming (reducing the words to its generic root format) etc. This left us with a Document Term Frequency matrix which represents each instance (Document) and the corresponding frequency of the words occurred there in.

3. **Label Generation:**
   The corresponding labels were also appended to each document to make the algorithm learn the pattern of features pertaining to correct classification of that instance into positive (Emergency) and negative (No Emergency). This was quite an involved task in a sense, as a mentioned above, simply resorting to all the tweets from @UBuffaloPolice as potential "emergency" would have infused a great deal of impurity in the dataset and subsequent inefficient algorithm. For example, we get to see both types of tweets from this account like a couple of samples below.
   Tweet 1:
   `We have gotten more reports of these phone scams targeting International student.`
   Tweet 2:
   `Dr. Bruce McBride was the New York State University Police Commissioner for several years.`

   And so to ensure that only the relevant tweets get picked up and marked with utmost possible accuracy, we took out the csv file 'pos_tweets' and manually retained only the specific tweets pertaining to "emergency" like situations attributing to severe weather conditions like high wind, snow, traffic disruptions thereof.
   The filtered positive examples called 'filtered_positive_incidents.csv' were again brought into the program and then consolidated with the main dataset. The final data frame was simply a collection of text and label columns which then was parsed as Document Term Frequncy matrix using text mining functionalities offered by "tm" package.

4. Model Construction:

Once the dataset was in place, with a well-cleaned Document Term Matrix and the corresponding labels for each document as last column called 'label', we employed Gradient Descent Algorithm to train the model and subsequently validate on the validation (hold-out) subset. As demonstrated during the lab 4 session, the splitData function comes in quite handy as it creates train and test datasets on its own with just a single line code. The validation set can then subsequently be used for computing the accuracy metric.

From performance point of view we also tracked the 'time to run' for the three variants that were explored here. We used alpha as 0.01 (learning rate), max_Iterations as 1000 and momentum of 0.9 in case of MGD. As expected, we get to see SGD with fastest performance but slightly lower accuracy as compared to MGD. The summary of the results of the same are as follows:

Algorithm	Gradient Descent	Stochastic Gradient Descent	Momentum Gradient Descent
Mean Abs Error	1.6639	1.6485	1.6381
Run Time	2.5 sec	1.46 sec	2.56 sec

Summary:

This project can be viewed as a comprehensive package involving all the aspects like design, build, test and deployment of solution to a real world case. The design phase involved sourcing the data with due authentication, accessing through API. The raw data was duly cleaned up and made model-ready by applying well-established pre-processing techniques provided by R's package on Text Mining.

The Bag of words feature came handy in terms of identifying the occurrences of particular terms for a specific labeled class of our business problem. This helped in building the feature space with manually labelled classes. The model building involved application of three optimization algorithms studied over the course namely Gradient Descent, Stochastic Gradient Descent and the Momentum Gradient Descent (which has a flavor of Heavy ball variant). All the 3 models were duly validated over the hold-out set to gauge the performance in terms of both accuracy and speed. As expected, Stochastic Gradient Descent and Momentum Gradient Descent outperform the rest in terms of speed and accuracy respectively. The model thus built, can be deployed as an aid for first respondents to emergency like situation (in this context the Police), based on the social media messages (in this context the Twitter) as a real world use case.