Weather & Twitter Project Documentation

Originally we started to take forecasts for 3 locations: London, Manchester and Glasgow.

Had to settle for just London and Glasgow because there were no good observations the area near Manchester surprisingly. Only Liverpool and by this point we had already been taking the forecasts from Manchester.

We took the observations with the aim of comparing forecasts from 5days, 3days, 1day, xHrs ago to the actual forecast.

We started downloading tweets from London and Glasgow on the 8th of December and putting them in MongoDB. Chay had to use iJson because the JSON files couldn’t be loaded into Python because they were too large. The aim was to filter all the tweets by weather words and then run those tweets through the algorithm. Those that were left could then be analysed through Python and put into a pandas dataframe.

We put all the tweets into MongoDB, from there we ran a for loop to add a key called ‘W1’ which returned ‘Y’ when weather related and ‘N’ when not. We then ran the algorithm over this list which left us with tweets which should in theory be predominantly weather related.

**Data Processing Lifecycle**

**Ingestion  
Sourcing the Data:** This was performed by using API keys provided by the Met Office and the Twitter Feed. Both websites provided us the option to filter what we pulled. From the Met Office we pulled observation and forecast data from London and Glasgow, from the Met Office – Tweets from the relevant locations. Python scripts were ran at regular intervals via the command prompt for 4 weeks.

**Staging the Data:** The scripts pulled tweets and weather forecasts/observations onto our local machines and soon after imported to MongoDB. We pushed these daily to GitHub to avoid potential loss of data.

**Profiling the Data:** We decided on what analysis we wanted to make. Decisions were made on how to make the querying most efficient. (creating a time period field – nearest 3 hour). Getting rid of unwanted fields.

**Munging and Wrangling  
Cleaning/Processing the Data:** MetOffice Data: Removing unwanted fields and creating fields with data type *Date*

Tweets Data: The cleaning process largely intertwined with a Machine Learning toolkit we had developed. The toolkit relied on sensory analysis. We filtered through 2000 tweets containing weather related words, inputting whether they were actually weather related or not. This built up an algorithm which could accurately determine whether a tweet was weather related.

In order to run the algorithm accurately we first needed to filter out all tweets that did not contain any weather related words. This was achieved by updating fields in the MongoDB via regular expressions in a cursor in PyMongo. These tweets could then be accurately assessed by our toolkit by another cursor. From the tweets which had passed both tests, we finally determined whether they were *Rain, Cold, Warm* or *Windy.*

To make analysis easier, we included a Date-Time period field. This would allow all the tweets from a relevant period be linked to the corresponding weather observation.

**Modelling the Data:** Since we only had two collections (MongoDB tables) – Tweets and Weather Forecasts/Observations, modelling proved to be very easy. All that was required was a count of a weather type related tweets and then to join the observed data. We were able to easily move our data from MongoDB to a Pandas DataFrame by pickling a cursor list.

**Analysis  
Hypothesis and Test:** Our objective was largely open-ended. We set out to find trends between Tweets and weather as well as assessing how accurate weather forecasts were.

**Final Analysis**

**Assumptions:** That our machine learning was accurate.