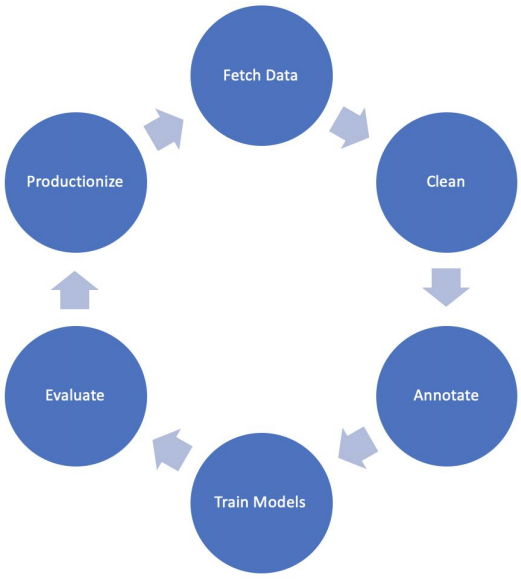
Versus Sentiment Methodology

Market & Language Coverage

Versus currently has primary coverage in Nigeria while we also primarily fetch tweets happening across Africa as well as a list of ~200 curated authentic African based sites.

Language use and nuances, English is key –

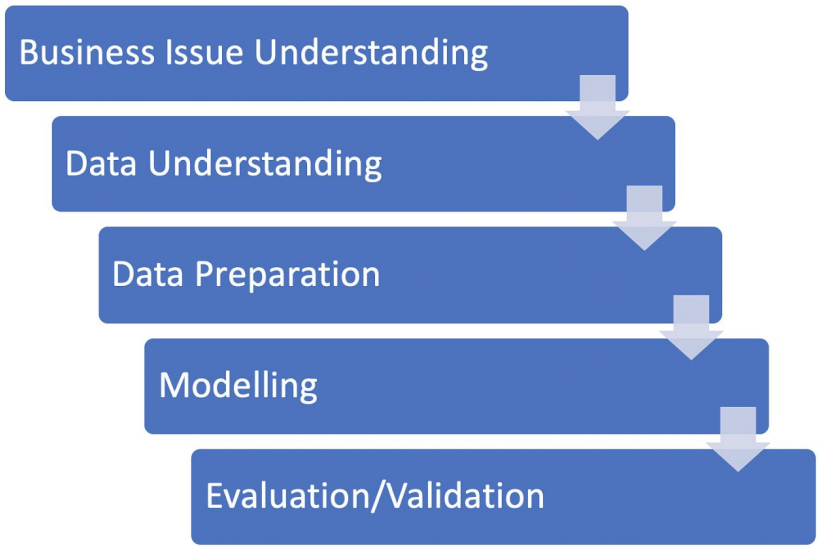
Versus works primarily with in-house linguists with an initial focus on covering key African languages spoken by over 200 million consumers. These include Pidgin English (Unique English spoken across most of West Africa), Igbo, Yoruba, Hausa and French. These linguists work closely with our data analysts to tag data that is subsequently used to train our models. With this continuous cycle we aim to scale across other different African countries expanding to more languages like Swahili, Amharik, and even Portuguese - leveraging our partnerships in these regions as well to scale learning patterns faster.



NLP approach and Coding/Verifying Sentiment

Versus uses the Cross-Industry Standard Process for Data Mining (CRISP-DM) to train multiple models which we eventually evaluate and the best performing models are put into production. We make use of context-specific models so that the models we utilize for analyzing tweets is different from that used for analyzing news.

The CRISP-DM methodology is codified in the following process diagram.



The process is detailed below:

Business Issue Understanding

The problem that we set out to solve was that of predicting or inferring sentiment from a body of text. What this means is we would like to read a piece of text and then tell you with a certain degree of certainty whether the sentiment behind it is positive, neutral, or negative. To do this, we need to collect data and use the data to train our model.

Data Understanding

This is one of the more important phases of the process. At this point, we ask ourselves whether we have the data we need to train our model. If not, we continue to collect further data. The preferred source of data will be from the same sources where we will be gathering the data that we need to continue to predict sentiment further down the line. This is also the point at which we analyze the data we have collected in order to extract insights.

Data Preparation

In this phase of our operations we manually label our training data. This is the most important step because any errors introduced here will be carried over to our model. We also pre-process the data by removing any unnecessary tokens in our data. We also carry out any feature engineering that is necessary to make working with our data possible. Examples of engineering here are tokenization, encoding, and embedding.

In the tokenization phase, we split our large body of text into a list of word tokens. In the encoding phase, we convert our word tokens into numbers, with each unique token represented by a number, up to a predetermined number which we call our dictionary size, n. The top n most frequent tokens are retained, with all other tokens represented as UNKNOWN.

Dictionaries, and by extension encodings, are large sparse matrices, typically 10,000 to 20,000 tokens in size. Working with this size will lead to computational difficulties. To simplify this, we train word embeddings. These are smaller multi-dimensional vectors normally having a size up to 128. Various papers have been published about this technique and can be found here and here.

Modeling

This is the point at which we train our model using the data that we have engineered. Training is an iterative forward and backward process. We begin by formulating a hypothesis, which is a mathematical equation with randomly initialized values called weights. In the forward pass, we compute the value of the equation using the weights. We then compare the quality of our prediction against our ground truth and compute a loss. In the backward step, we make adjustments to our weights so that in the next forward pass our computations produce a lower loss. This is called back propagation. The error function used is called the Logistic Loss.

We continue to do this until the loss is either acceptable or can't be reduced any further. This is called training to convergence. Because of the size of the training data, we make use of gradient descent for this. Various approaches to gradient descent have been published in research papers and a good summary is available here.

At the end of training, we have a vector of weights which we call a model. Prediction is simply the process of multiplying our weights with the numerical representation of our inputs, passing the results through an activation function, and thresholding the resulting output.

Our model is put into production and used to make predictions.

Validation

In this phase, we evaluate our model quality against human experts. We periodically fetch some of our predictions and compare them to what the human experts labelled. This evaluation lets us know how to improve our model. This is important because models get outdated. Improving the model might require adjusting the threshold of our predictions.

The model is updated periodically by adding new training data and repeating the process.

Summary

All of the math and statistics that we apply is published in various research papers. We're also engineering the features of our training data to work with our nuanced data. And that is the key, it's the data: how it is collected, labeled, and engineered.