**Language Detection**

**What is NLP?**

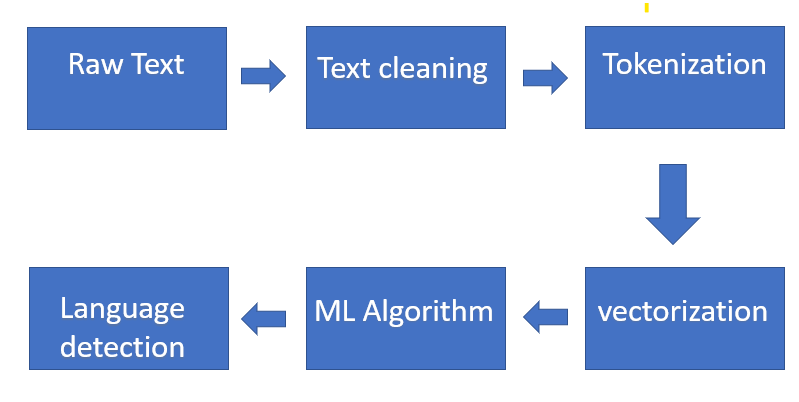
-Natural language processing is analysis and generation of natural language text.

-We use the nlp for text classification in our problem

**Required libraries in my project.?**

* Numpy
* Pandas
* Scikit-learn
* Nltk
* Seaborn
* Matplotlib

**The process of building language detection model.**



### First Feature Engineering / Preprocessing

1. Remove punctuations and special characters
   1. With the clean raw text sentences in hand.
2. Vectorization using (Tf-idf)

the natural languages are preprocessed using Scikit-Learn's TfidfVectorizer. Essentially, the raw text sentences are converted into a numerical feature matrix.

Tf-idf stands for Term frequency inverse document frequency, and the arguments I used are analyzer = 'char' and ngram\_range = (1,3). Analyzer = 'char' tells the vectorizer to look at characters rather than words.

**Second split the data.**

Split the data set into training and testing data.

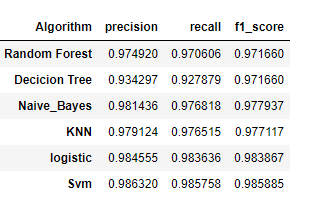
Train on 70% of data, test on remaining 30%.

**Third, Machine learning Model.**

1. Random Forest
2. Decision Tree
3. KNN
4. Naive Bayes
5. Logistic regression
6. Support vector machine

**Fourthly, Evaluate Machine learning algorithms.**

To find the best algorithm’s accuracy.

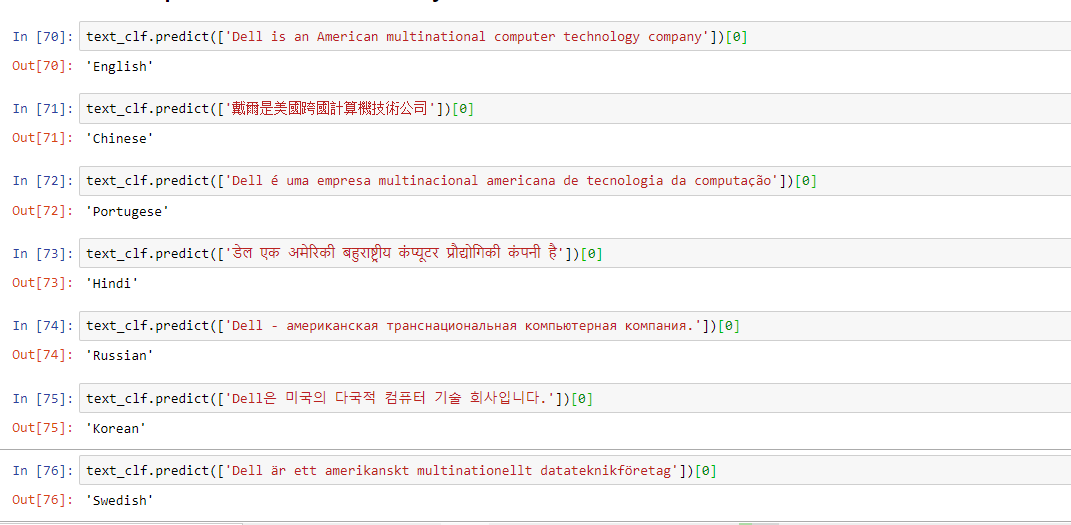


### Finally, we make a prediction with Naive Bayes Multinomial Model:

### In a Naive Bayes Multinomial Model, features are assumed to be generated from a multinomial distribution.

### The Multinomial Naive Bayes model is modeled with the best fit multinomial distribution. The model works by computing a large table of probabilities with the given data.

Using the Liklihood -- P(feature|Label), the Prior-- P(Label), and the Marginal Liklihood-- P(feature), the model takes advatage of Baye's Theorm to compute P(Label|feature). The P(Label|feature) with the highest probability gets predicted as the correct language.

****