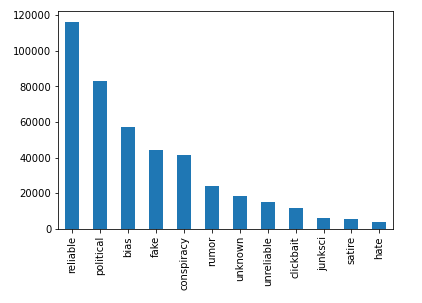
# FAKE NEWS PREDICTION

## Problem Statement:

Fake News represent fabricated information communicated through different media like newspapers, television, emails, tweets, WhatsApp forwards, memes, images, videos etc. In most cases, the purpose of faking the information is to mislead the readers or to damage the reputations of government, firms, communities or of individuals. In this project, we aim to make use of machine learning, deep learning and NLP algorithms to identify whether the information shared is fake or not.

## Exploratory Data Analysis:

Before performing modeling on the dataset, we got familiar with the data by performing basic analysis. We found that there are (426550, 17) rows and columns; out of which 3 columns were of numeric type and rest were of object type. Our strategy to handle missing values was to replace numeric values with mean and object values by mode. In the dataset,”Type” columns is the output, so we were excited to know its values.



Since our goal is to identify whether the content is fake or not; we marked all columns except fake as not-fake. But this approach results in class imbalance problem. The ratio of fake remains 1:9. In order to handle this problem, we plan to use SMOTE in the future for better sampling.

# Basic Modeling:

For now, we have used TfidfVectorizer model for prediction which results in Accuracy: 94.02%. But,we will explore more techniques for better results. The confusion matrix of the result is:

array([[ 6124, 2779], [ 2326, 74081]])

We too will evaluate based on other metrics like f1\_Score, precision and recall.

SMOTE to handle class imbalance:

In order to handle class imbalance, we created a pipeline:

newsClassifier =Pipeline([

('vect', CountVectorizer()),

('tfidf', TfidfTransformer()),

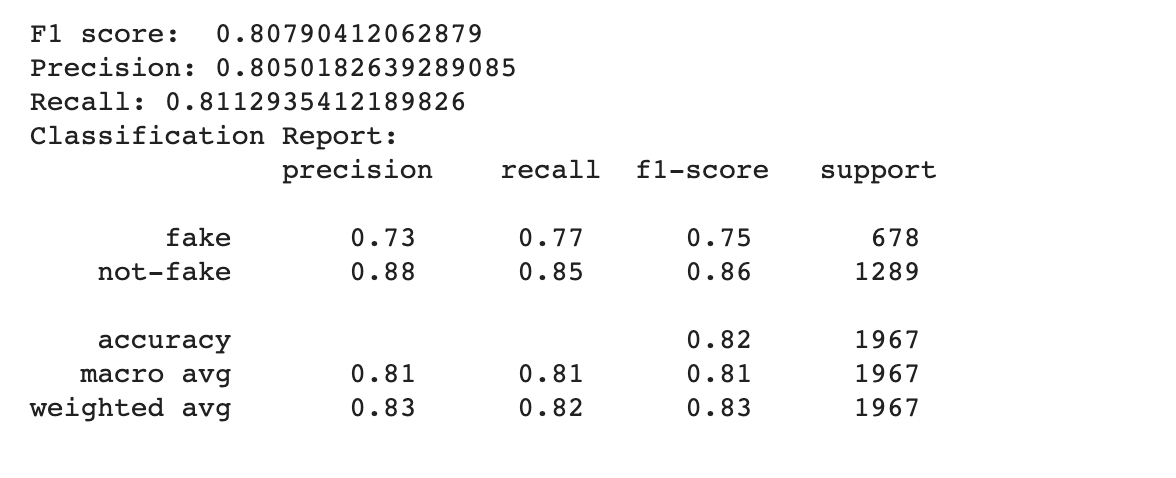
('smote', SMOTE(random\_state=12)),

('mnb', MultinomialNB(alpha =0.1))

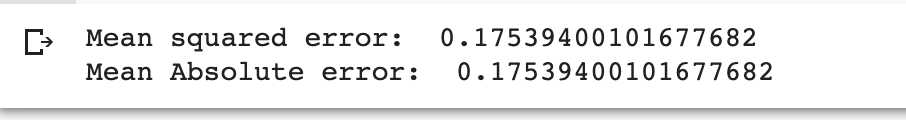
])

SMOTE handled imbalance pretty nicely and accuracy improved to Accuracy: 82.46%.

## Evaluation on TfidfVectorizer:

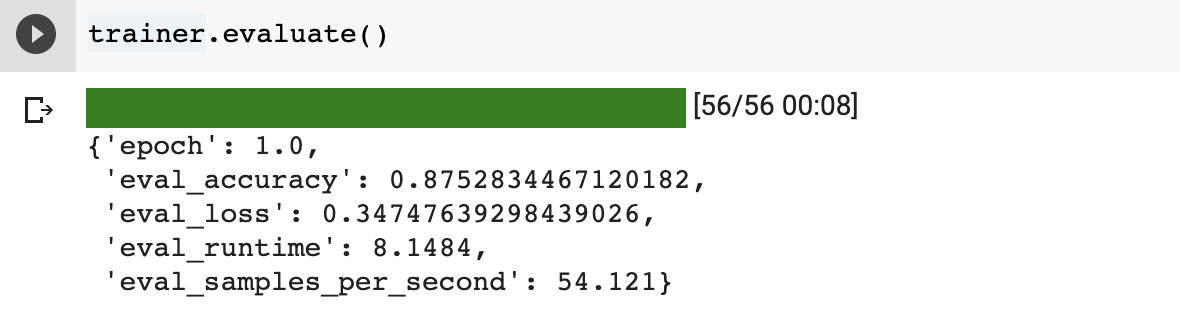


## Basis Error Analysis



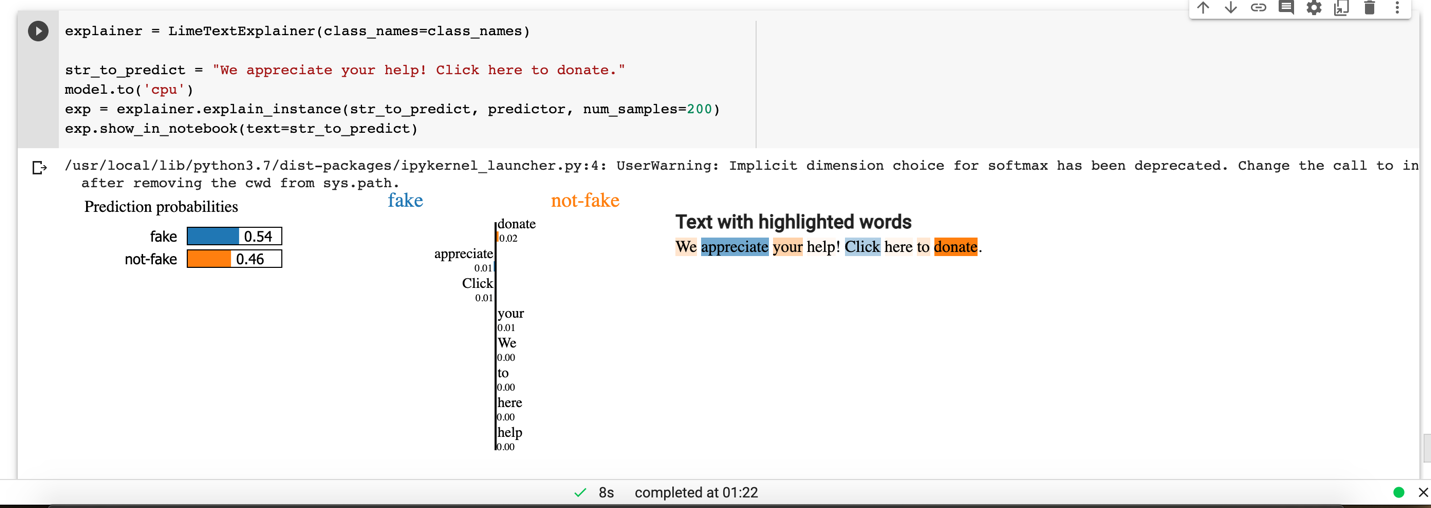
# Multilingual Transformer

We used **xlm-roberta-base** multilingual transformer that results in,



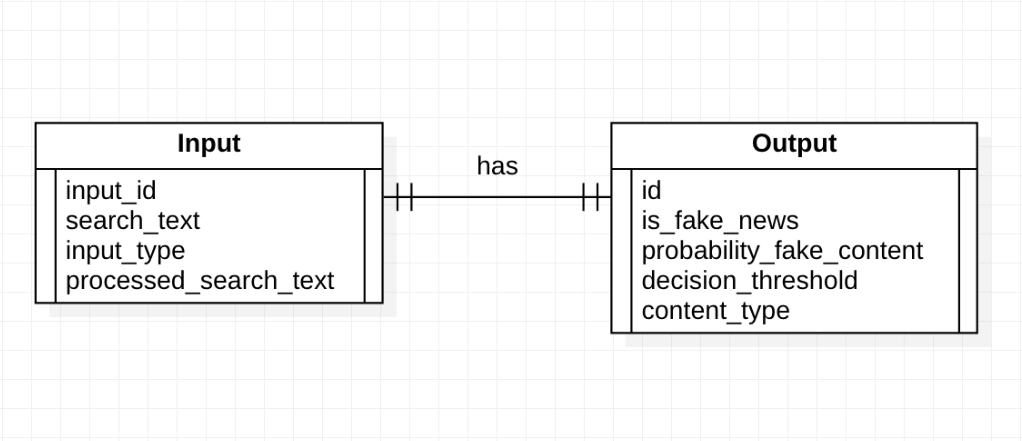
## Interpretability

We used LIME to understand the internal predictions of the model.   
For Input String: “We appreciate your help! Click here to donate.”; **appreciate** and **Click** seem to be words that results in **fake** predictions. **We**, **your**, **to** and **donate** results in **not-fake** predictions.



## Database Schema:

For the data given, we have designed two entities:



Metadata:

Input Entity:

input\_id: unique identifier for each input

search\_text: content provided by user

input\_type: domain, url, content etc

processed\_input: processed input suitable for our model

Output Entity:

id: unique identifier for each output

is\_fake\_news: whether or not the the content is classified as fake

probability\_fake\_content: Probability of content being fake

decision\_threshold: threshold on the basis of which validity of fake content is decided

content\_type: type of the content; reliable, political, hate etc