Fake News Classification with Natural Language Processing (NLP) and Long Short-Term Memory (LSTM)

SUBMITTING TO:

JASPREET KAUR MA'AM

NAME: GUDLADHANA HARSHITH

REG.NO: 11903248

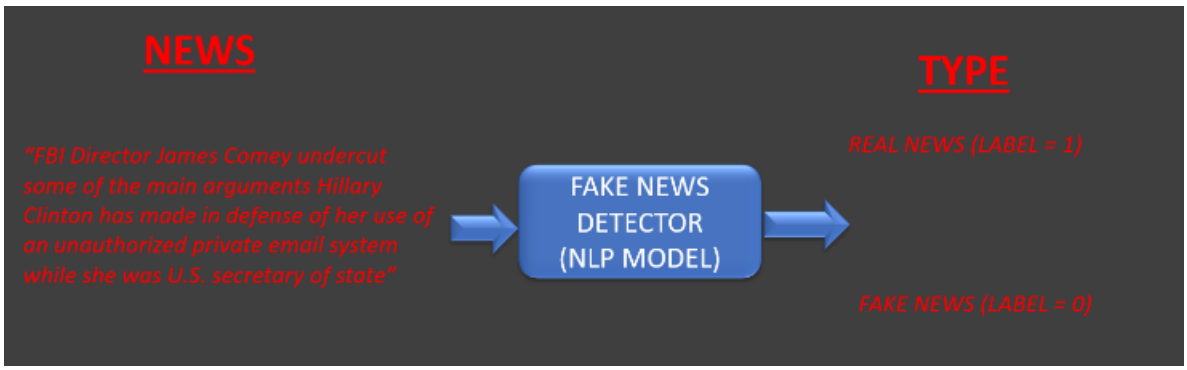
SECTION: KM053

ROLL.NO: A06

GITHUB LINK: https://github.com/GudladhanaHarshith/FakeNewsClassification

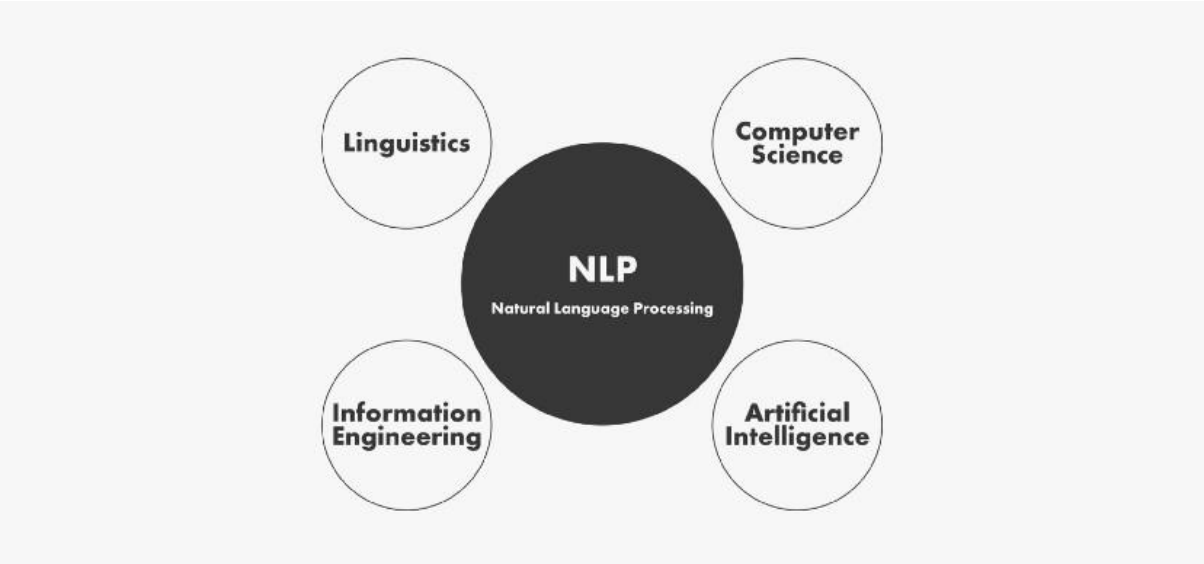
INTRODUCTION:

1. Right now, we are living in a world of mis-information and fake news. The goal of this project is to detect the fake news based on Recurrent Neural Network which is Long Short Term Memory (LSTM) and Natural Language Processing (NLP).
2. Natural Language Processors (NLP) work by converting words(text) into numbers.
3. These numbers are then used to train a Deep Learning Model to make predictions.
4. This model is very important to the companies and media to automatically predict whether the circulating news is fake or not.
5. In this project we will analyse thousands of news text to detect if it is fake or not
6. We will label the news as 1 if it was fake and label it as 0 if it was not a fake news
7. When we have a pandemic (COVID-19) there are more fake news which are widely spread by unknown authority and make people panic about the pandemic, so I got an idea to create a fake news detection model to get rid of all unnecessary news.
8. In this project, I use lot of modules which are imported at the starting of the project. I performed data cleaning, data visualization, exploratory data analysis, prepare the data by tokenization and padding, prepared model, trained model, come out with predictions.



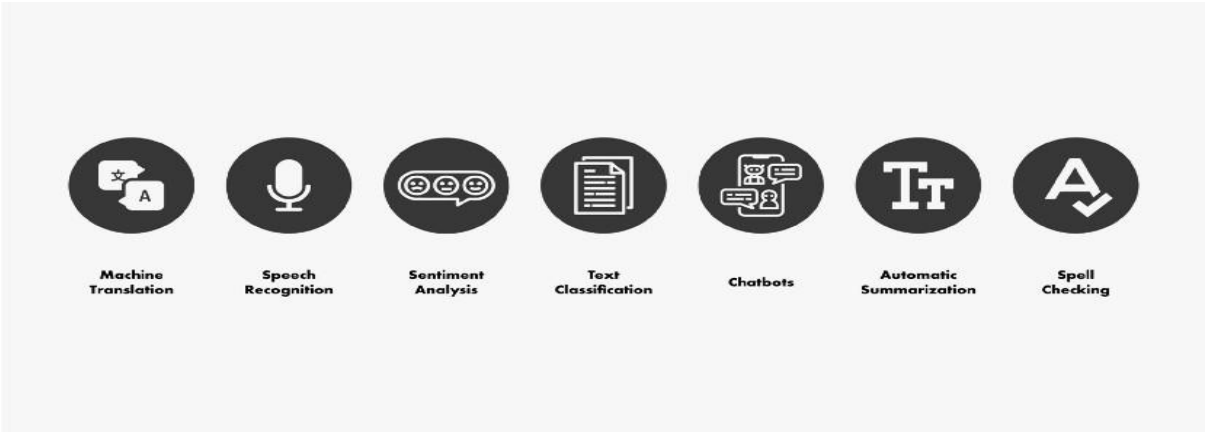
NATURAL LANGUAGE PROCESSING:

Natural language processing is a subfield of linguistics computer science and artificial intelligence concerned with the interactions between computers and human language in particular how to program computers to process and analyse large amounts of natural language data. The goal is for a computer to be able to comprehend the contents of documents. The technology can extract information and insights from the documents, as well as organize them. Speech recognition, natural language understanding and natural language generation are some of the challenges in natural language processing.



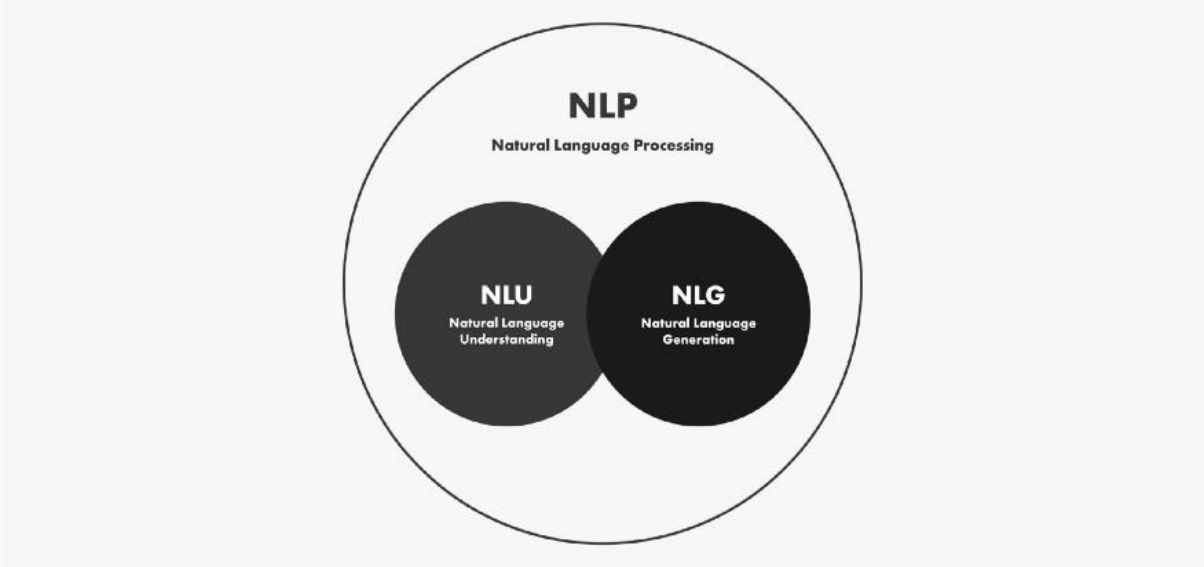
Applications of NLP:

1. Machine Translation
2. Speech Recognition
3. Sentimental Analysis
4. Text Classifications
5. Chatbots
6. Spell Checking
7. Automatic Summarization



Components of NLP:

Generally Natural Language Processing can be split into two parts the Understanding part (NLU) and the Generation part (NLG) :



The Components of Natural Language Processing: NLP = NLU + NLG

1. Natural Language Understanding (NLU):

NLU or **Natural Language Understanding** is a subset of NLP that is responsible for the semantic parsing and analysis, entities extracting…etc. NLU tries to structure the input data to understand the meaning behind written text.

For example, after having the speech recognition software convert speech into text, The NLU system comes into the picture to decipher its meaning. It is quite possible that the same text has various meanings, or different words have the

same meaning, or that the meaning changes with the context. Knowing the rules and structure of the language, understanding the text without ambiguity are some of the challenges faced by NLU systems.

2. Natural Language Generation (NLG):

NLG or **Natural Language Generation** is the step when the machine tries to transform the structured (from NLU) data into human-readable language, so NLG does exactly the opposite of NLU.

Given the data, it analyses it and generates narratives in conversational language. It goes way beyond template-based systems, having been configured with the domain knowledge and experience of a human expert to produce well-researched, accurate output within seconds. Narratives can be generated for people across all hierarchical levels in an organization, in multiple languages.

Natural Language Generation has made its presence felt in various domains ranging from banking and financial services to media, healthcare, and education. It is here to stay to help businesses improvise their processes and tread faster on the path of advancement!

DATASET DESCRIPTION:

1. In This Project, I have used two dataset which represents the both fake news and true news.
2. The True news data set is having all the true news data.
3. The Fake News dataset is having all the fake news data.

```
import warnings
warnings.simplefilter("ignore")
!pip install plotly
!pip install --upgrade nbformat
!pip install nltk
!pip install spacy # spaCy is an open-source software library for advanced natural language processing
!pip install WordCloud
!pip install gensim # Gensim is an open-source library for unsupervised topic modeling and natural language processing
import nltk
nltk.download('punkt')

import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS
import nltk
import re
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
#import keras
from tensorflow.keras.preprocessing.text import one_hot, Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Embedding, Input, LSTM, Conv1D, MaxPool1D, Bidirectional
from tensorflow.keras.models import Model
import warnings
warnings.simplefilter("ignore")
```

I started with importing all the necessary modules that are required for the Visualization, Data Cleaning, Exploratory data analysis, Training of the Model, Testing of the Model.

Plotly:

Plotly.py is an open source and browser based graphing library. Plotly.py is a high-level charting library built on top of plotly.js. Plotly.js ships with over 30 chart types including scientific charts, 3D graphs, statistical charts and more.

Nltk (Natural Language Tool Kit):

NLTK is a popular Python framework for dealing with data of human language. It includes a set of text processing libraries for classification and semantic reasoning, as well as wrappers for industrial-strength NLP libraries and an active discussion forum. NLTK is ideal for linguists, engineers, students, and

industry users. Nltk is very useful in a python programming language. Nltk in python we can learn new thing, nltk is very useful to learn new technology.

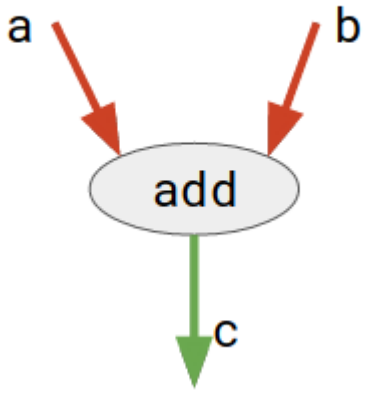
Spacy:

spaCy is an advanced Natural Language Processing library. It was designed from the first day to be used in real products. spaCy supports tokenization and training in 60 languages. It has state-of-the-art speed, neural network models, pretrained transformers, a production ready training system and easy model packaging for deployment.

TensorFlow:

TensorFlow is an open-source software library. **TensorFlow** was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well!

1. nodes in the graph represent mathematical operations.
2. edges in the graph represent the multidimensional data arrays (called **tensors**) communicated between them.



Stop Words:

Stop Words are commonly used words excluded from searches to help index and parse faster. While most Internet search engines and NLP (natural language processing) utilize stop words, they do not prevent users from using them. Instead, the words are only ignored when the search results are displayed.

Loading of Data:

```
:  1 # load the data
   2 df_true = pd.read_csv("True.csv") # reading the true news data
   3 df_fake = pd.read_csv("Fake.csv") # reading the fake news data
```

We have to import both fake and true data with the help of pandas data frames.

OBSERVATION OF DATASET:

```
1 df_true
```

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017
...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017

21417 rows × 4 columns

As we observe that the True News dataset has 21417 Rows * 4 Columns.

```
1 df_fake
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017
...	...	...	...	...
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016

23481 rows × 4 columns

As we can observe that the fake news dataset has 23481 Rows * 4 Columns.

INFORMATION OF BOTH FAKE AND TRUE NEWS DATASET:

```
1 df_true.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
```

```
1 df_fake.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

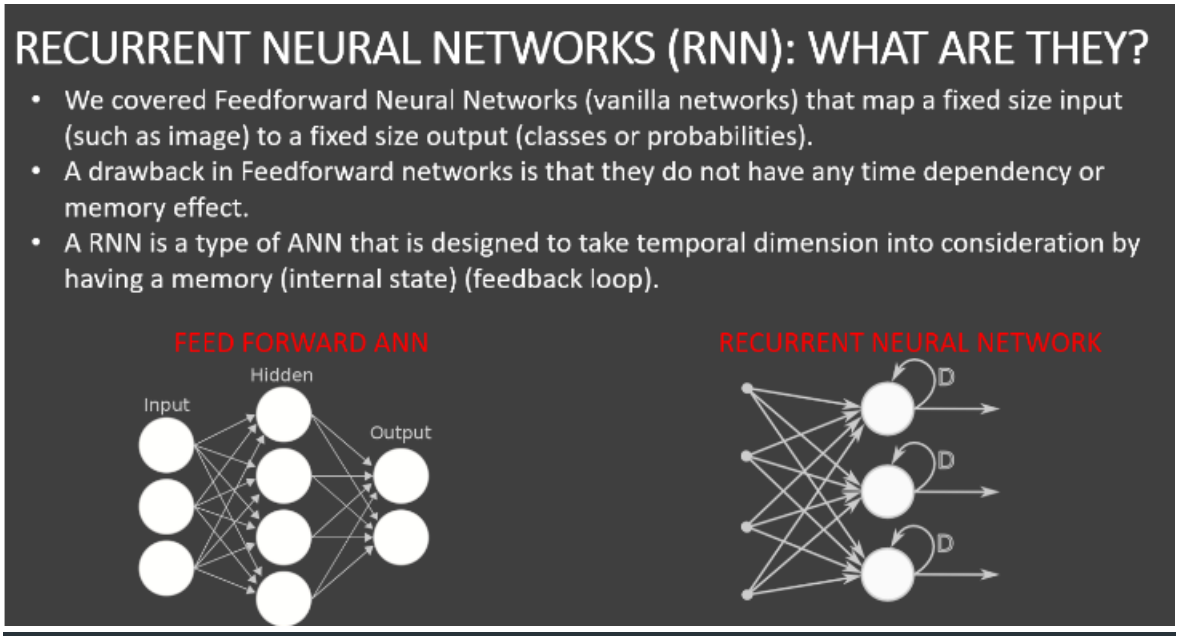
As we observe that the there is no null values and the memory usage of the dataset is quiet low (in KB).

PROJECT METHODOLOGY:

In my project I have used the Sequential Model. And I had added the Embedding of 128 dimensions, I have added the LSTM of 128 dimensions, I used the dense layers of 128 dimensions with activation functions of relu and one more layer of dense with activation function of sigmoid
I compiled the project with Adam Optimizer with loss function of binary_crossentropy and metrics of accuracy.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 128)         13914112

 bidirectional (Bidirectiona  (None, 256)              263168
 l)

 dense (Dense)               (None, 128)               32896

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 14,210,305
Trainable params: 14,210,305
Non-trainable params: 0
_________________________________________________________________
```

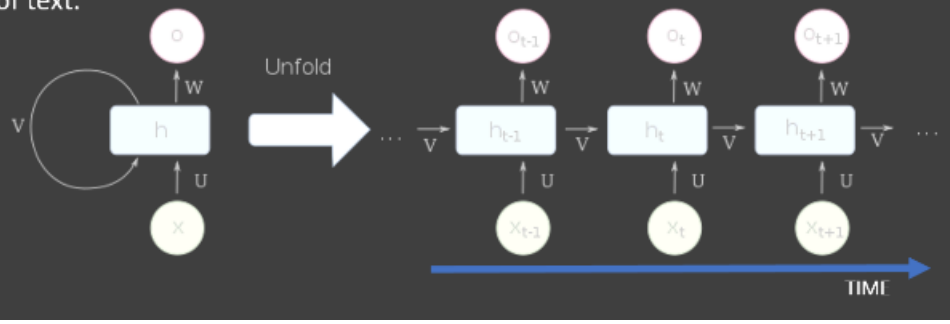
RECURRENT NEURAL NETWORKS:



RNN ARCHITECTURE:

RNN ARCHITECTURE

- A RNN contains a temporal loop in which the hidden layer not only gives an output but it feeds itself as well.
- An extra dimension is added which is time!
- RNN can recall what happened in the previous time stamp so it works great with sequence of text.

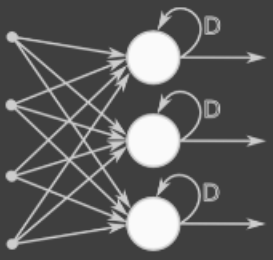


WHY I USE THE RNN IN MY PROJECT?

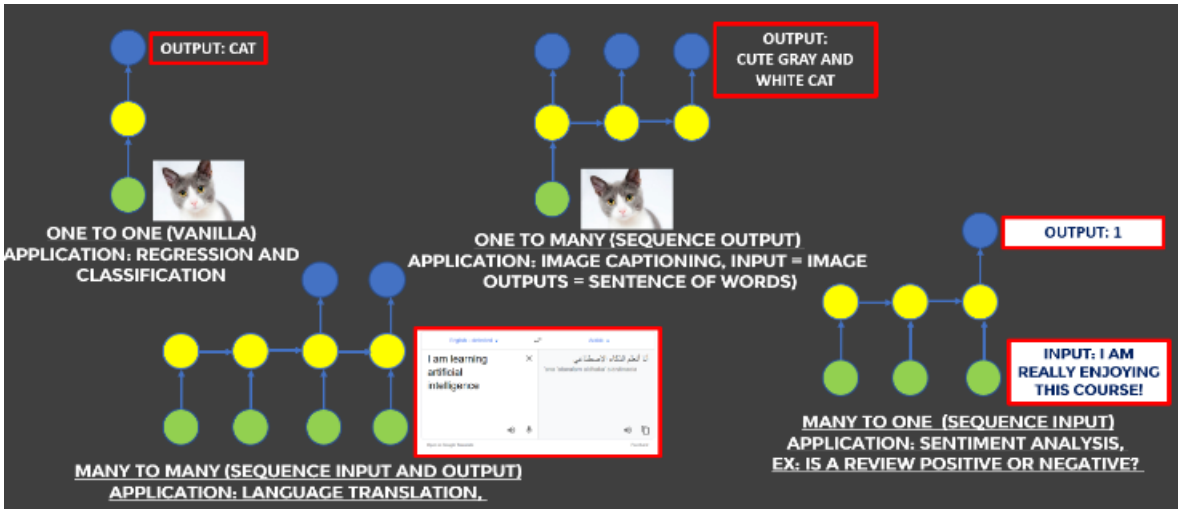
WHAT MAKE RNNs SO SPECIAL?

- Feedforward ANNs are so constrained with their fixed number of input and outputs.
- For example, a CNN will have fixed size image (28x28) and generates a fixed output (class or probabilities).
- Feedforward ANN have a fixed configuration, i.e.: same number of hidden layers and weights.
- Recurrent Neural Networks offer huge advantage over feedforward ANN and they are much more fun!
- RNN allow us to work with a sequence of vectors:
  - Sequence in inputs
  - Sequence in outputs
  - Sequence in both!

RECURRENT NEURAL NETWORK



DIFFERENCE BETWEEN THE ANN AND RNN:



VANISHING GRADIENT DESECENT PROBLEM:

- LSTM networks work much better compared to vanilla RNN since they overcome the vanishing gradient problem.
- The error has to propagate through all the previous layers resulting in a vanishing gradient.
- As the gradient goes smaller, the network weights are no longer updated.
- As more layers are added, the gradients of the loss function approaches zero, making the network hard to train.

EACH LAYER DEPENDS ON THE OUTPUT FROM THE PREVIOUS LAYERS, THE "V" IS MULTIPLIED SEVERAL TIMES RESULTING IN VANISHING GRADIENT

0.1 * 0.1 * 0.1 *.....*0.1 =1e-10

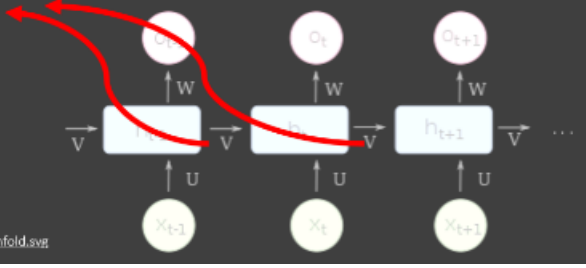
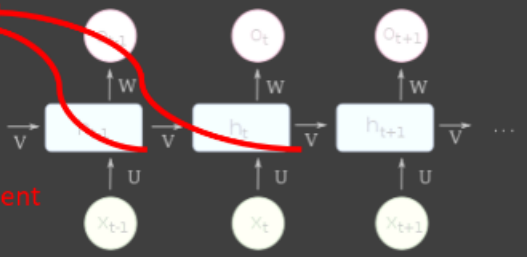


Photo Credit: https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg

- ANN gradients are calculated during backpropagation.
- In backpropagation, we calculate the derivatives of the network by moving from the outermost layer (close to output) back to the initial layers (close to inputs).
- The chain rule is used during this calculation in which the derivatives from the final layers are multiplied by the derivatives from early layers.
- The gradients keeps diminishing exponentially and therefore the weights and biases are r longer being updated.

EACH LAYER DEPENDS ON THE OUTPUT FROM THE PREVIOUS LAYERS, THE "V" IS MULTIPLIED SEVERAL TIMES RESULTING IN VANISHING GRADIENT, (ex: 0.1 * 0.1 * 0.1 *.....*0.1 =1e-10)



New Weight = Old Weight – Learning rate * gradient
9.09999 = 10.1 – 1 * 0.001

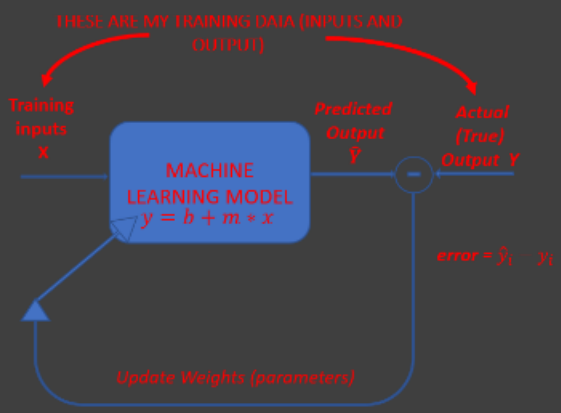
GRADIENT DESCENT WORKING:

GRADIENT DESCENT

- Let's assume that we want to obtain the optimal values for parameters 'm' and 'b'.

$$y = b + m * x$$

GOAL IS TO FIND BEST PARAMETERS



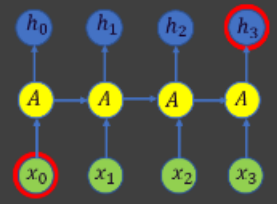
- We need to first formulate a loss function as follows:

$$Cost\ Function\ f(m,b) = \frac{1}{N}\sum_{i=1}^{n}(error)^2 = \frac{1}{N}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

LONG SHORT TERM MEMORY:

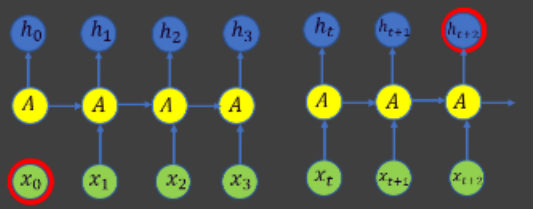
- LSTM networks work better compared to vanilla RNN since they overcome vanishing gradient problem.
- In practice, RNN fail to establish long term dependencies.
- Reference: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

The tree color is "green"



RNN PERFORMS WELL SINCE THE GAP BETWEEN THE PREDICTION "GREEN" AND THE NECESSARY CONTEXT INFORMATION "TREE" IS SMALL

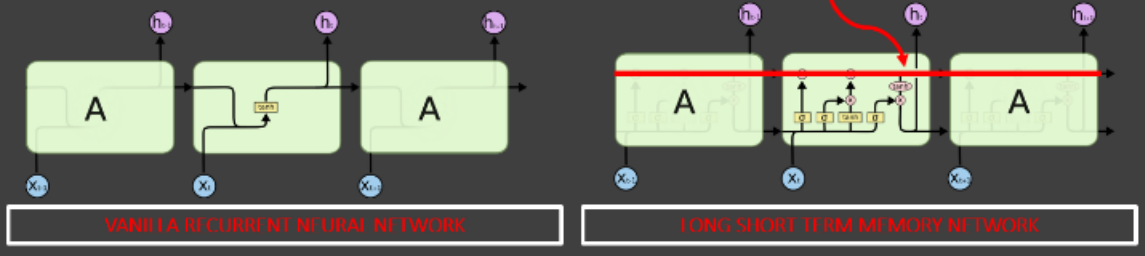
I live in Quebec in Northern Canada……where I live, the weather is generally "cold" most of the year



RNN PERFORMS POORLY WHEN THE GAP BETWEEN THE PREDICTION "COLD" AND THE NECESSARY CONTEXT INFORMATION "CANADA" IS LARGE

- LSTM networks are type of RNN that are designed to remember long term dependencies by default.
- LSTM can remember and recall information for a prolonged period of time.
- Recall that each line represents a full vector.

THIS HORIZONTAL LINE (MEMORY) OR CELL STATE ENABLES LSTM TO REMEMBER VERY OLD INFORMTION

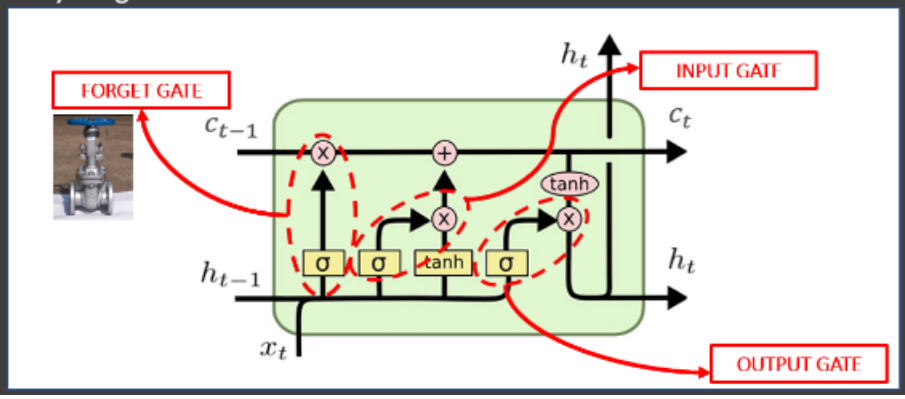


VANILLA RECURRENT NEURAL NETWORK

LONG SHORT TERM MEMORY NETWORK

GATES IN LSTM:

- LSTM contains gates that can allow or block information from passing by.
- Gates consist of a sigmoid neural net layer along with a pointwise multiplication operation.
- Sigmoid output ranges from 0 to 1:
  - 0 = Don't allow any data to flow
  - 1 = Allow everything to flow!



TRAINING OF THE DATA:

In [52]:
```
# train the model
model.fit(padded_train, y_train, batch_size = 64, validation_split = 0.1, epochs = 2)
```

```
Epoch 1/2
506/506 [==============================] - 185s 339ms/step - loss: 0.0420 - acc: 0.9820 - val_loss: 0.0049 - val_acc: 0.9981
Epoch 2/2
506/506 [==============================] - 166s 327ms/step - loss: 0.0023 - acc: 0.9995 - val_loss: 0.0055 - val_acc: 0.9986
```

Out[52]: <keras.callbacks.History at 0x13f0306c280>

I trained my data with the help of the 2 epochs with batch size of 64 and validation_split of 0.1, so that I get better predictions.

ACCESSING THE TRAINING DATA:

In [53]:
```
# make prediction
pred = model.predict(padded_test)
```

```
281/281 [==============================] - 12s 32ms/step
```

TOKENIZATION OF DATA:

- Tokenizer allows us to vectorize text corpus by turning each text into a sequence of integers.
- SENTENCE:

" budget fight looms republicans flip fiscal script Washington Reuters head conservative republican faction congress voted month ..."

- TOKENS:

[3138, 3581, 2895, 27, 5354, 22457, 3505, 9, 3138, 35, 2895, 208, 213, 3581, 29, 71, 5354, 22457, 1275, 335, 2, 619, 2903, 27, 10461, 43213, 4908, ...]

Train_Test_Split AND Tokenization OF DATA:

```
# split data into test and train
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df.clean_joined, df.isfake, test_size = 0.2)
#x=features=df.clean_joined
#y=target=df.isfake
```

```
from nltk import word_tokenize
```

```
# Create a tokenizer to tokenize the words and create sequences of tokenized words
tokenizer = Tokenizer(num_words = total_words)
tokenizer.fit_on_texts(x_train)
# we will feed the x_train data and fit to get tokenizer
train_sequences = tokenizer.texts_to_sequences(x_train)
test_sequences = tokenizer.texts_to_sequences(x_test)

"""
```

This is snippet of code in which I split the data into training data and testing data and tokenize the data into numerical form so that we can pass this data into model to run the predictions.

RESULTS:

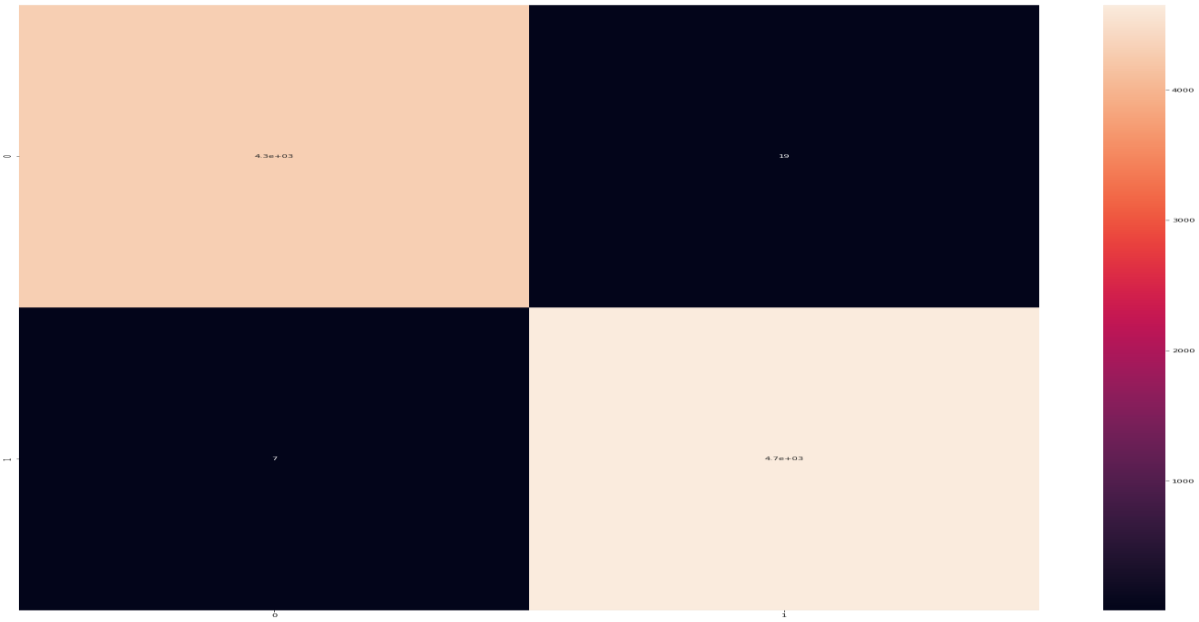
```
: 1 # getting the accuracy
  2 from sklearn.metrics import accuracy_score
  3
  4 accuracy = accuracy_score(list(y_test), prediction)
  5
  6 print("Model Accuracy : ", accuracy)
```

```
Model Accuracy :  0.9971046770601336
```

My model accuracy was 0.9971046770601336, which is quite good equivalent to 99%

CONFUSION MATRIX:

```
ı [56]: 1 # get the confusion matrix
        2 from sklearn.metrics import confusion_matrix
        3 cm = confusion_matrix(list(y_test), prediction)
        4 plt.figure(figsize = (25, 25))
        5 sns.heatmap(cm, annot = True)
```



This is the confusion matrix, which I drew with y_test and predictions which I made with the model.

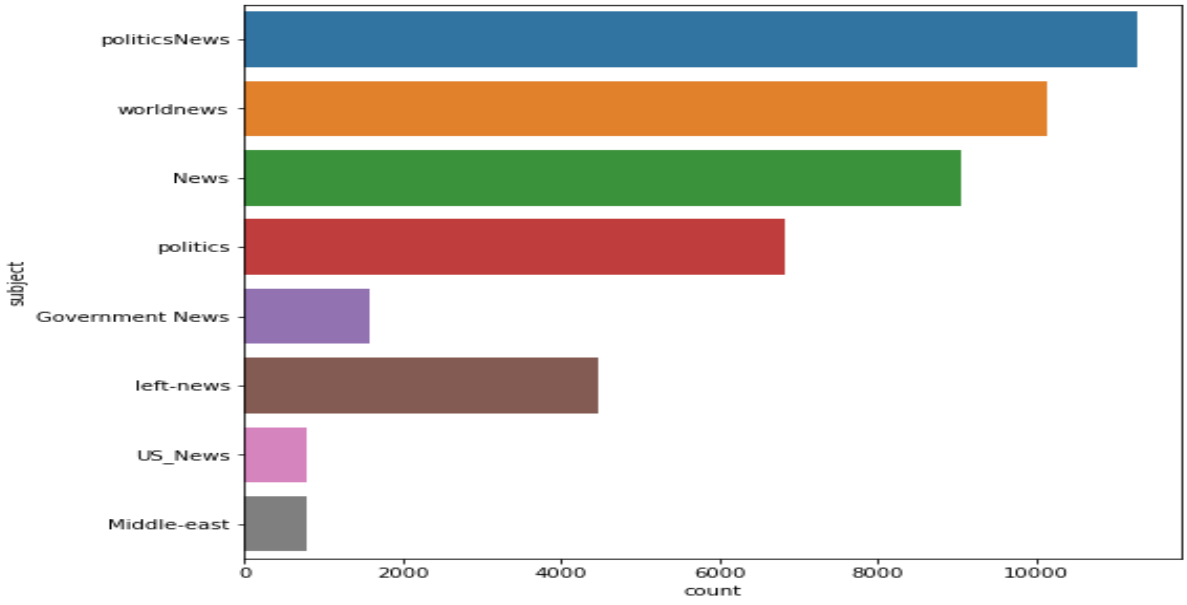
FEW MORE PREDICTION METRICS:

```
In [57]:
1 from sklearn.metrics import precision_score
2 precision_score(y_test, prediction, average='macro')

Out[57]: 0.9971536737928328

In [58]:
1 from sklearn.metrics import accuracy_score
2 accuracy_score(y_test,prediction)

Out[58]: 0.9971046770601336

In [59]:
1 from sklearn.metrics import recall_score
2 recall_score(y_test,prediction)

Out[59]: 0.9984972091026192

In [60]:
1 from sklearn.metrics import f1_score
2 f1_score(y_test,prediction)

Out[60]: 0.99721269296741
```

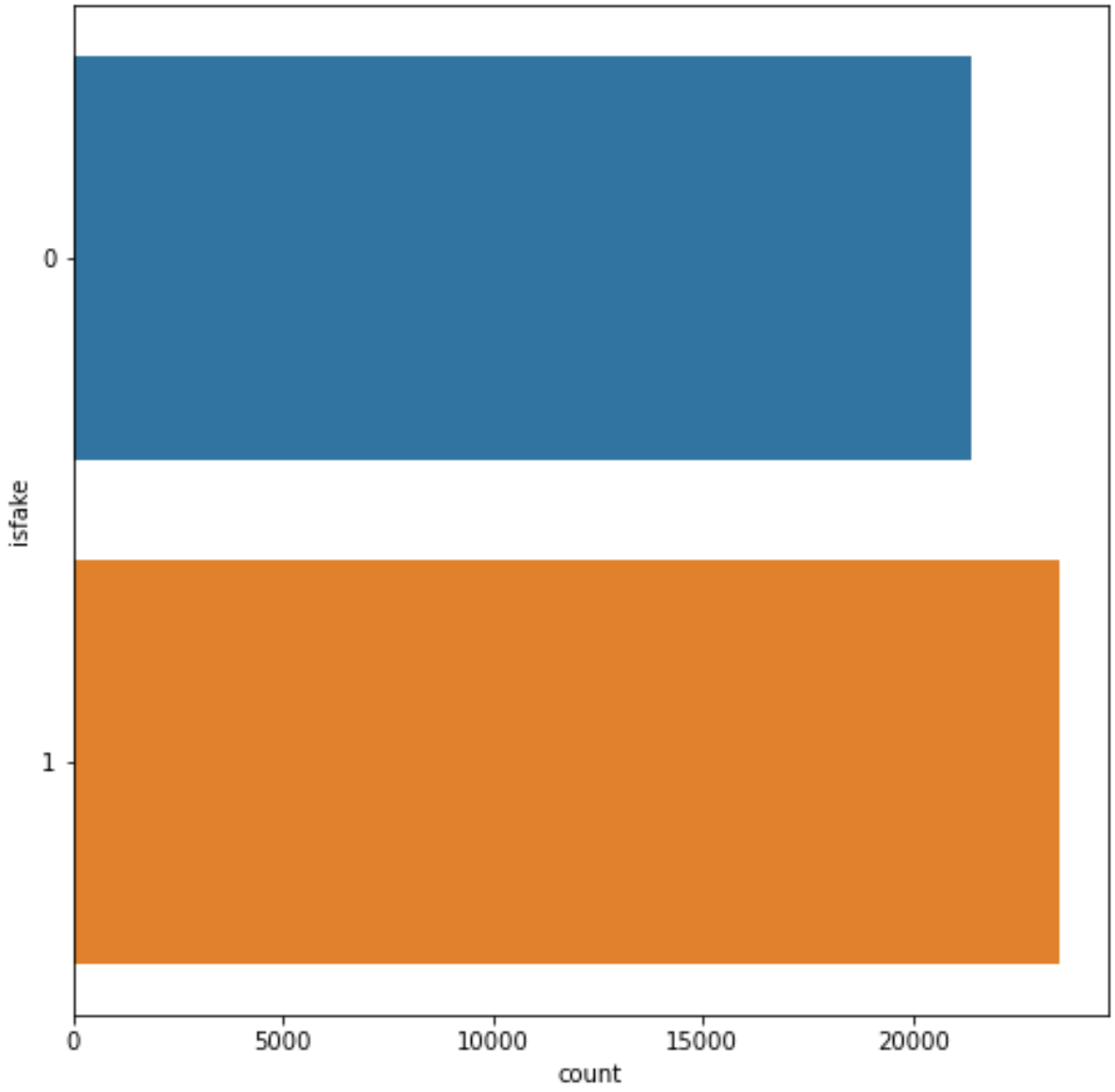
These were the values of few metrics, I imported from the sklearn module and tested then against the y_test and prediction, which gave good prediction scores**.**

QUANTITATIVE ANALYSIS:

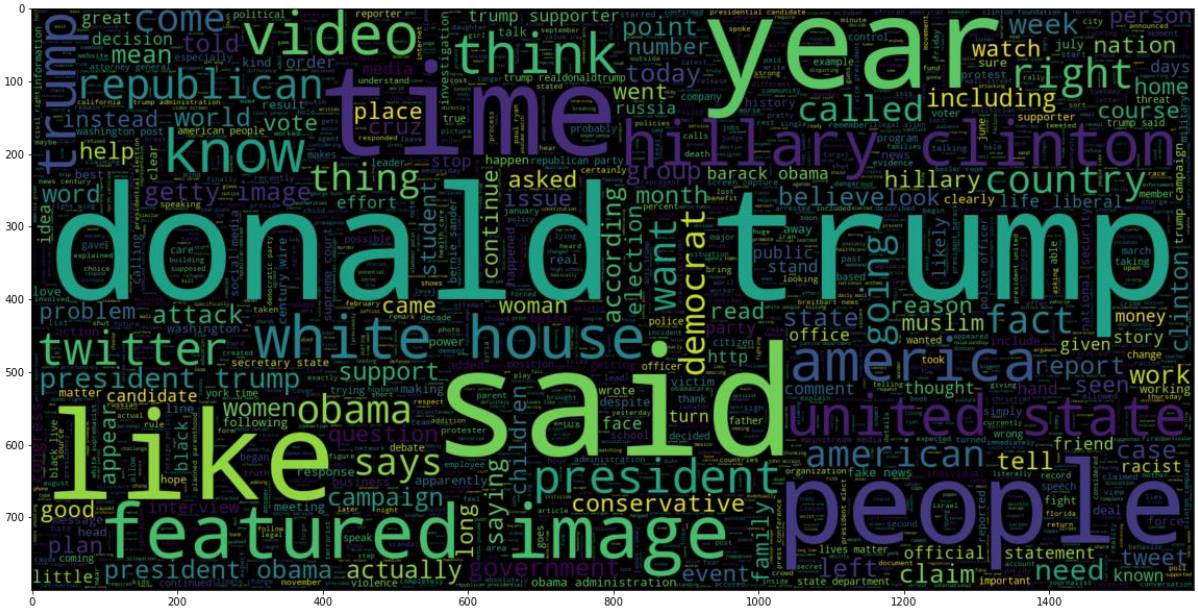
I have made few visualizations that helped me to understand the data in a proper way.



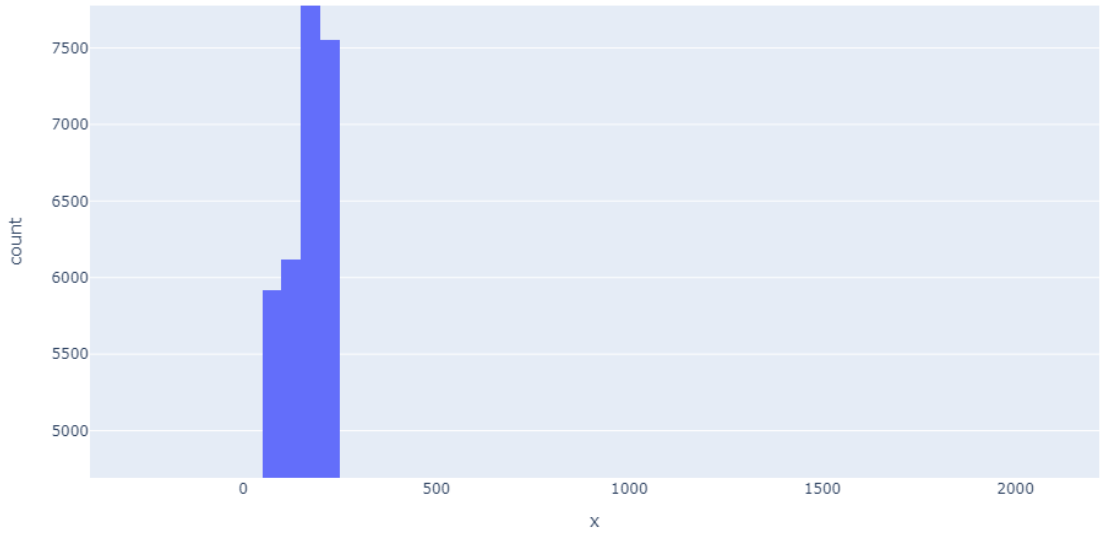
This is the visualization that help me to understand the kind of news that are there in datasets**.**



This is the visualization that help me to understand the no of fake and true news in the datasets**.**



This is the visualization that is made out of wordcloud that help me the no of words in the dataset, it helps me understand the frequency of the words**.**



This is visualization that help to understand count of words and the frequency of the words.

CONCLUSIONS:

1) From this project I learnt a lot about RNN and LSTM and Sequential model.
2) This model will predict the news whether fake or not with 99% Accuracy and classify then into two categories as true and fake.
3) I learnt a lot of new visualization tools like CountPlot, Plotly,Wordcloud,..etc,
4) I use lot of accuracy metrics like precision_score, accuracy_score, f1_score, recall_score in the model to get the correct values of accuracy.
5) Learnt about the importance of NLP and applications of NLP in real world.
6) Learnt the intuition behind the RNN, Gradient Descent, Vanishing Gradient Descent Problem, LSTM.

REFERENCES:

https://www.coursera.org/learn/nlp-fake-news-detector/supplement/DpHEA/project-based-course-overview

https://medium.com/analytics-vidhya/part-1-introduction-to-natural-language-processing-nlp-a66ad8773b3

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://analyticsindiamag.com/evaluation-metrics-in-ml-ai-for-classification-problems-wpython-code/

https://drive.google.com/file/d/1VEyWc7-gopiGKEDJw6FF_m81ac_0xyet/view?usp=sharing

https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

----------THANK YOU----------