Detection of Fake News Using Machine Learning Models

Tarp Project Review 1

Under the Guidance of Jacob Raglend Sir

Team Σ Members:

Swarup Tripathy - 19BEE0167

Sankalp Shukla - 19BEE0211

Aron Samuel Jacob - 19BEE0348

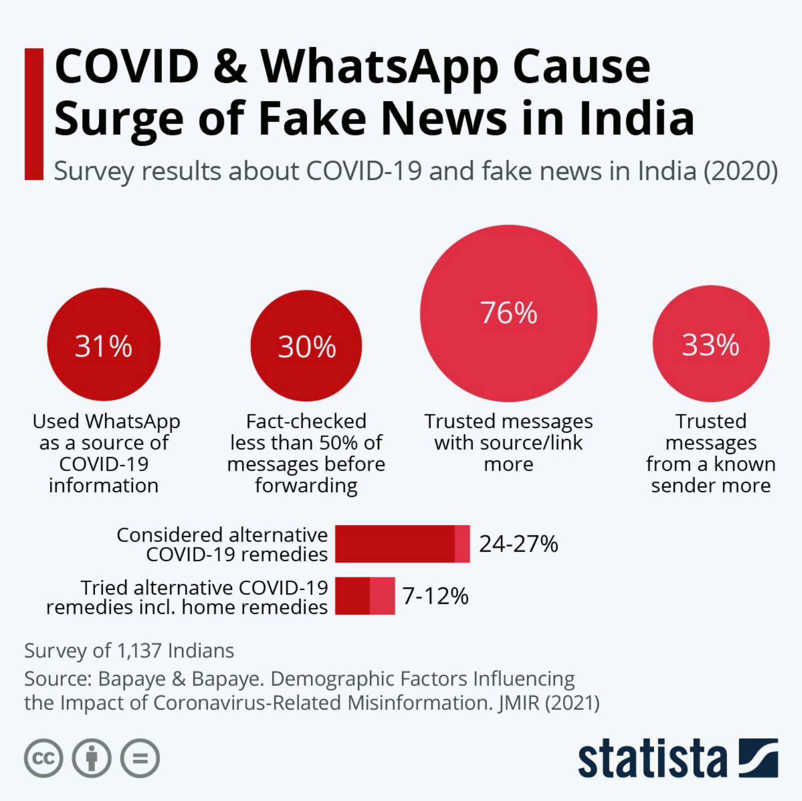
What is Fake News?

Fake news refers to information content that is false, misleading or whose source cannot be verified. This content may be generated to intentionally damage reputations, deceive, or to gain attention. The term rose to popularity during the 2016 US Presidential Elections. It was reported that fake news likely influenced the results of the elections.

The invasive nature of Fake news has led to a massive disruption of information in and around india for that past Decade.

Fake news is circulated through popular social media platforms for political, social and even religious propaganda.

Statistics revolving around Fake News:



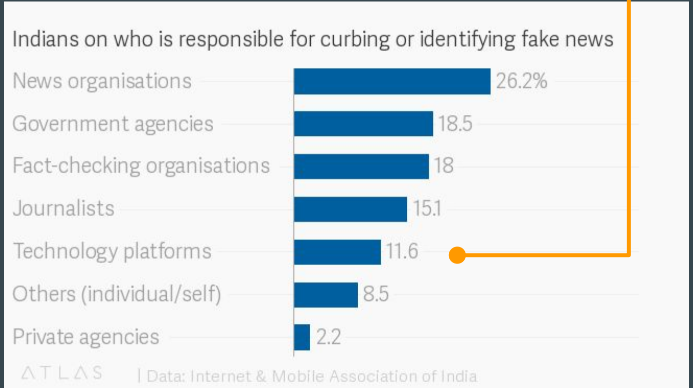
COVID & WhatsApp Cause Surge of Fake News in India

Survey results about COVID-19 and fake news in India (2020)

Survey of 1,137 Indians

Source: Bapaye & Bapaye. Demographic Factors Influencing the Impact of Coronavirus-Related Misinformation. JMIR (2021)

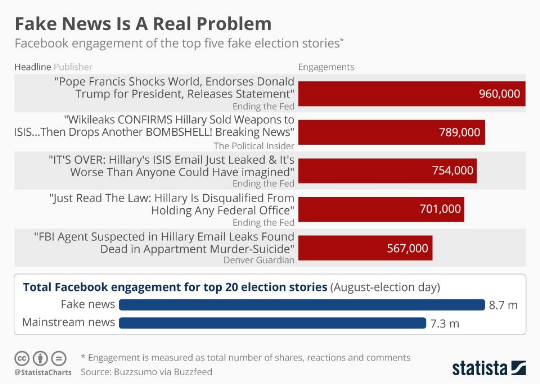
statista



Indians on who is responsible for curbing or identifying fake news

ATLAS | Data: Internet & Mobile Association of India

Tech Responsibility



Fake News Is A Real Problem

Facebook engagement of the top five fake election stories*

* Engagement is measured as total number of shares, reactions and comments

@StatistaCharts Source: Buzzsumo via Buzzfeed

statista

Problem Statement/Objective:

The aim of this project is to train a deep learning or machine learning model such that it can detect and remove fake news without the constant scrutiny of a supervisory personnel.

This will help improve the quality of news and information circulated in the public and prevent baseless information from causing major damage to the society.

The project aims at a methodology to create a model that will detect if a news is authentic or fake based on its words, phrases, sources and titles.

Existing Solutions/Flaws:

- ***Fact Checkers:*** fact checkers come from media organisations like the Washington Post and websites such as the urban legend debunking site Snopes.com.

- ***Fake Tag:*** Another warning appears if users try to share the story, although Facebook doesn't prevent such sharing or delete the fake news story. The "fake" tag will however negatively impact the story's score in Facebook's algorithm, meaning that fewer people will see it pop up in their news feeds.

The flaws in these approaches is that it is not humanly possible
to check the sheer amount of Data generated and circulated online in today's world.

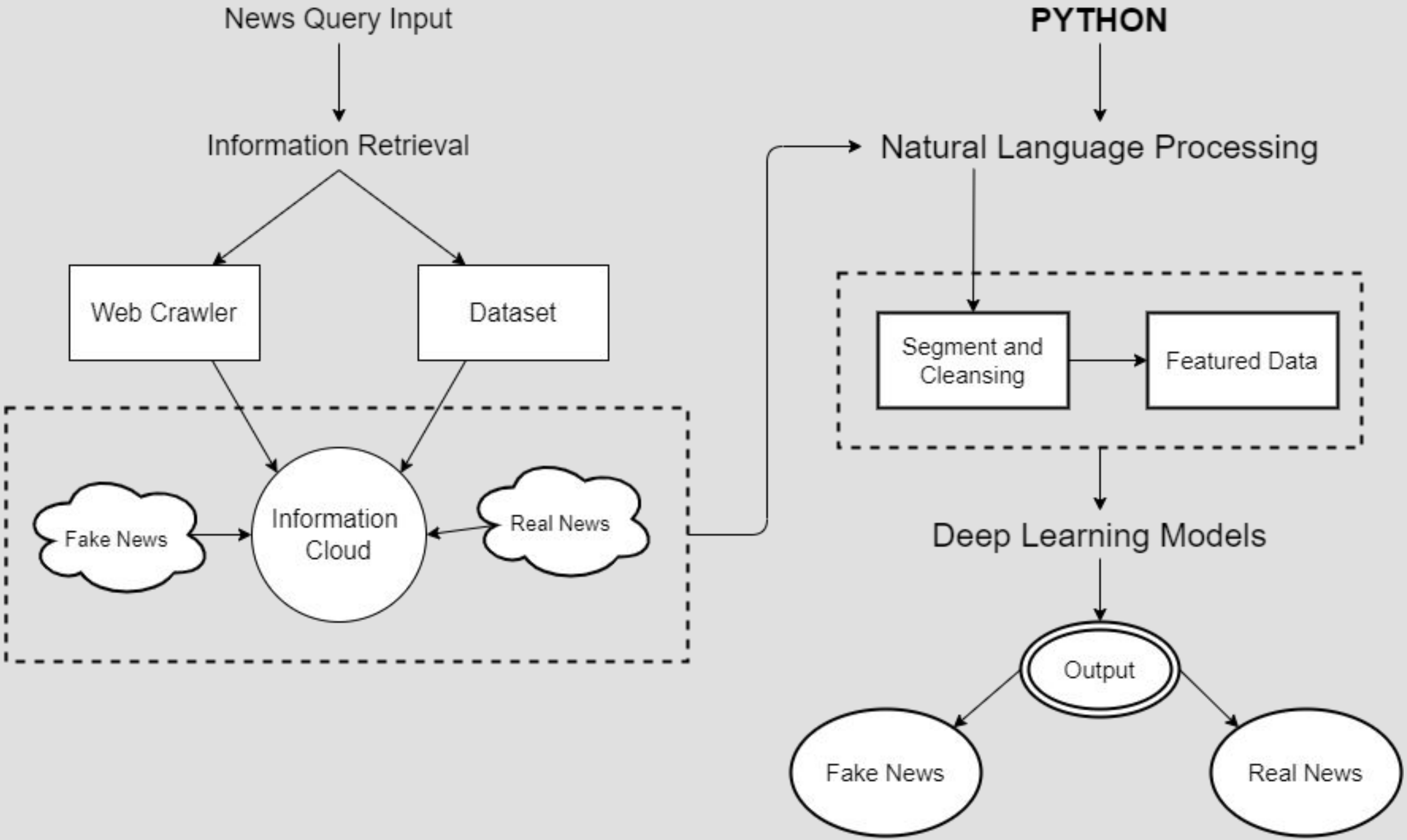
Information spreads fast online, making manual fact checking ineffective. Manual fact checking struggles to scale with the volume of data generated.

With the advent of machine learning it is possible to extend the range of surveillance beyond the human limit. It also reduces the mundane work to be done manually.

Methodology

- Python would be the primary language of use
- **Libraries:** spaCy, NLP Text Classification using PyCaret and many more subject to our use.

Datasets are used to refine the algorithms. The datasets is to be split as training data and test data.
This data is then ran through a classifier algorithm and the accuracy score for the model is obtained.


News Query Input
Information Retrieval
Web Crawler
Dataset
Fake News
Information Cloud
Real News
PYTHON
Natural Language Processing
Segment and Cleansing
Featured Data
Deep Learning Models
Output
Fake News
Real News

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular it deals with how to program computers to process and analyze large amounts of natural language data.

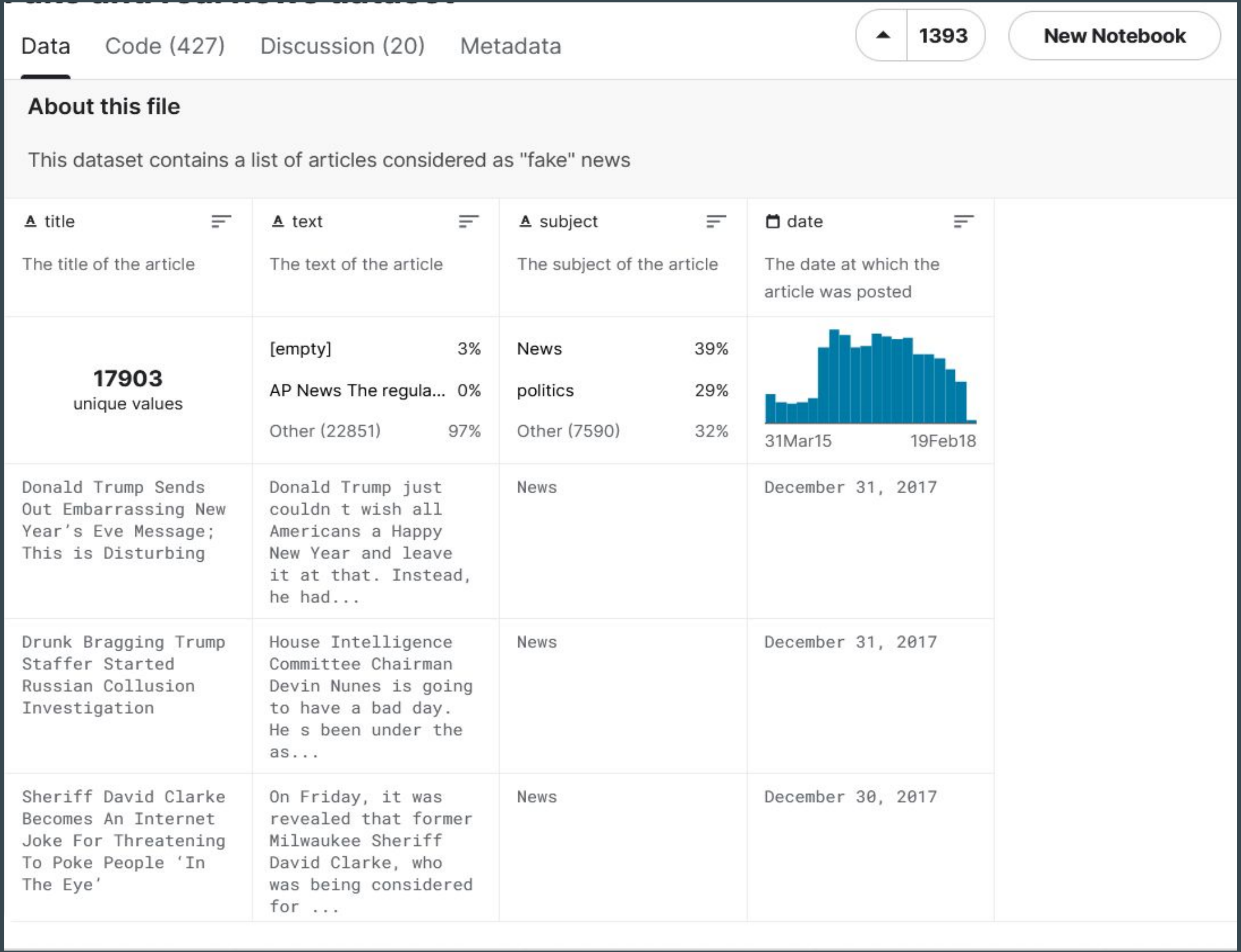
The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them.

The technology can then accurately extract information and content contained in the documents as well as categorize and organize the documents themselves.

Dataset:

Taken from Kaggle

This is temporary which is subject to change while working with dataset.



About this file

This dataset contains a list of articles considered as "fake" news

title	text	subject	date
The title of the article	The text of the article	The subject of the article	The date at which the article was posted
17903 unique values	[empty] 3% AP News The regula... 0% Other (22851) 97%	News 39% politics 29% Other (7590) 32%	31Mar15 19Feb18
Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had...	News	December 31, 2017
Drunk Bragging Trump Staffer Started Russian Collusion Investigation	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the as...	News	December 31, 2017
Sheriff David Clarke Becomes An Internet Joke For Threatening To Poke People 'In The Eye'	On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for ...	News	December 30, 2017

Analysis Of the Dataset

We can evaluate Machine Learning Algorithms using metrics like:

1. Accuracy
2. Precision
3. Recall
4. F1-score

Accuracy(Acc)%= $\dfrac{Tp+Tn}{Tp+Tn+Fp+Fn} \times 100$

Recall(Re)%= $\dfrac{Tp}{Tp+Fn} \times 100$

Precision(pre)%= $\dfrac{Tn}{Tn+Fp} \times 100$

F1-Score= $2 \times \dfrac{(\mathrm{Pr}\,ecision)(recall)}{\mathrm{Pr}\,ecision \div recall}$

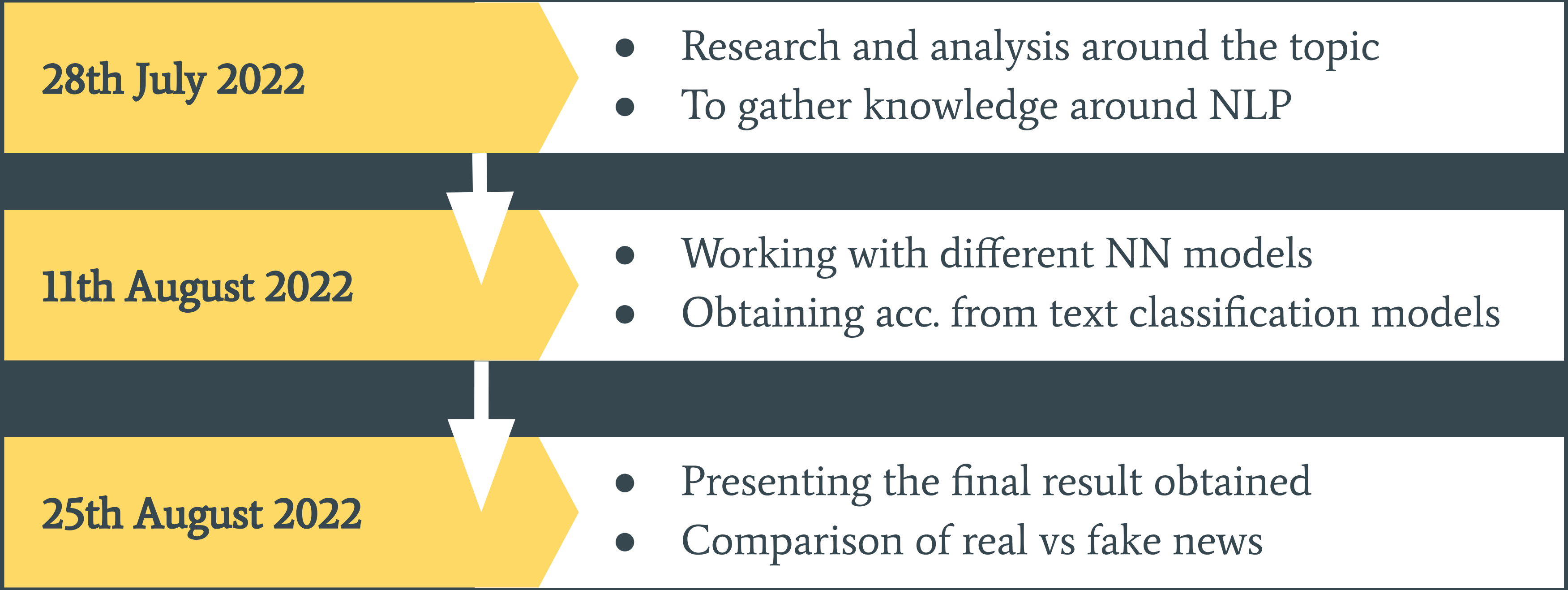
True positive (TP) = the number of cases correctly identified as fake news

False positive (FP) = the number of cases incorrectly identified as fake news

True negative (TN) = the number of cases correctly identified as factual news

False negative (FN) = the number of cases incorrectly identified as factual news.

Timeline



28th July 2022

- Research and analysis around the topic
- To gather knowledge around NLP

11th August 2022

- Working with different NN models
- Obtaining acc. from text classification models

25th August 2022

- Presenting the final result obtained
- Comparison of real vs fake news

References:

1. de Beer, Dylan & Matthee, Machdel. (2021). Approaches to Identify Fake News: A Systematic Literature Review. 10.1007/978-3-030-49264-9_2.

2. Poddar, Karishnu & Amali, Geraldine & S, Umadevi. (2019). Comparison of Various Machine Learning Models for Accurate Detection of Fake News. 10.1109/i-PACT44901.2019.8960044.

3. Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

4. Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).

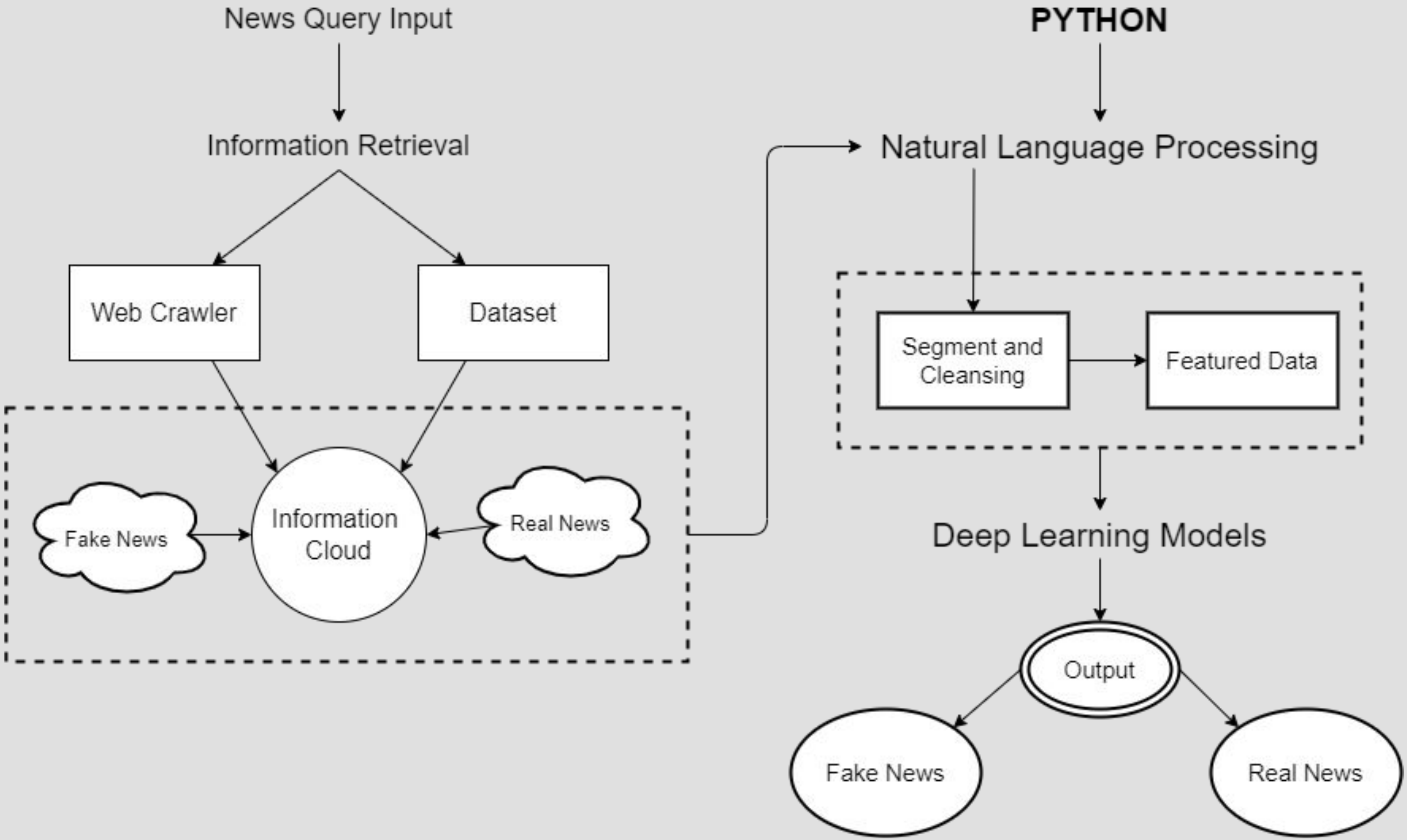
Thank You !

Project So Far........

Detection of Fake News Using Machine Learning Models

Tarp Project Review 3

Under the Guidance of Jacob Raglend Sir


News Query Input
Information Retrieval
Web Crawler
Dataset
Fake News
Information Cloud
Real News
PYTHON
Natural Language Processing
Segment and Cleansing
Featured Data
Deep Learning Models
Output
Fake News
Real News

Preparing the Dataset using Web Scraping

Web scraping is a simple technique that describes an automatic collection of a huge amount of data from websites. Data is of three types as structured, unstructured, and semi-structured. Websites hold all the types of data in an unstructured way web scraping is a technique that helps to collect this instructed data from websites and store it in a structured way.

Python Libraries used for web scraping:

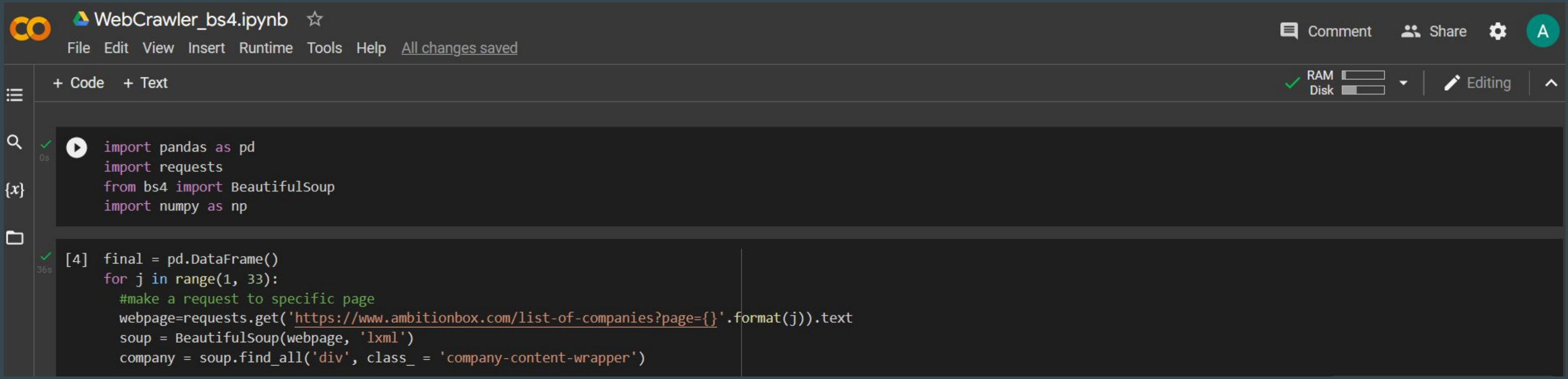
Beautiful Soup(bs4) – Beautiful Soup is a Python library used for web scraping. It sits at a top of an HTML or XML parser which provides python idioms for iterating, searching, and modifying a parse tree. It automatically converts incoming documents to Unicode and outgoing documents to UTF-8. Beautiful Soup is easy to learn, robust, beginner-friendly and, the most used web scraping library in recent times with request.

lxml – It is a high performance, fast HTML and XML parsing library. It is faster than a beautiful soup. It works well when we are aiming to scrape large datasets. It also allows you to extract data from HTML using XPath and CSS selectors.

Scrapy – a complete web scraping framework.

- helps you to scrape a large amount of dataset efficiently and effectively.
- It can be used for data mining to monitoring and automated testing.
- creates spiders that crawl across websites and retrieve the data. The best thing about scrapy is it is asynchronous, and with the help of spacy, you can make multiple HTTP requests simultaneously.

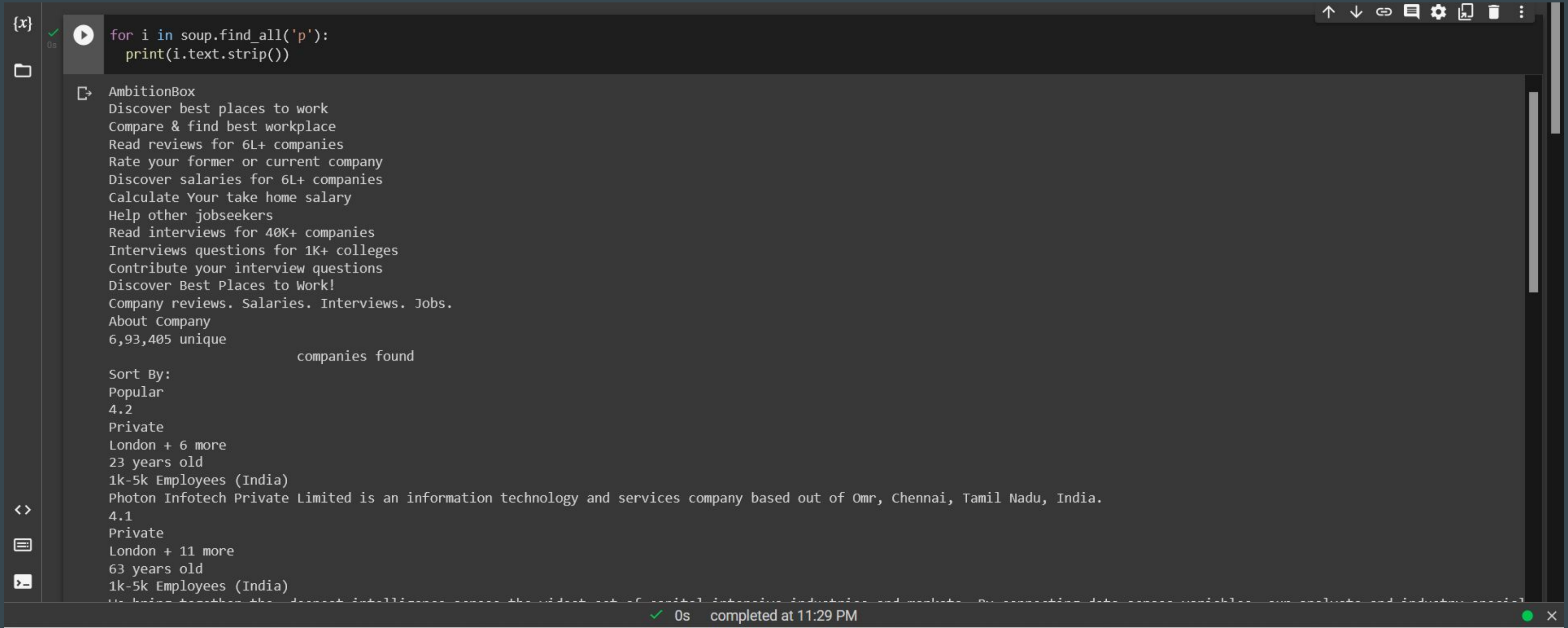
WebScraping using bs4 Python library



```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import numpy as np
```

```
[4] final = pd.DataFrame()
    for j in range(1, 33):
      #make a request to specific page
      webpage=requests.get('https://www.ambitionbox.com/list-of-companies?page={}'.format(j)).text
      soup = BeautifulSoup(webpage, 'lxml')
      company = soup.find_all('div', class_ = 'company-content-wrapper')
```

Importing library and accessing webpage

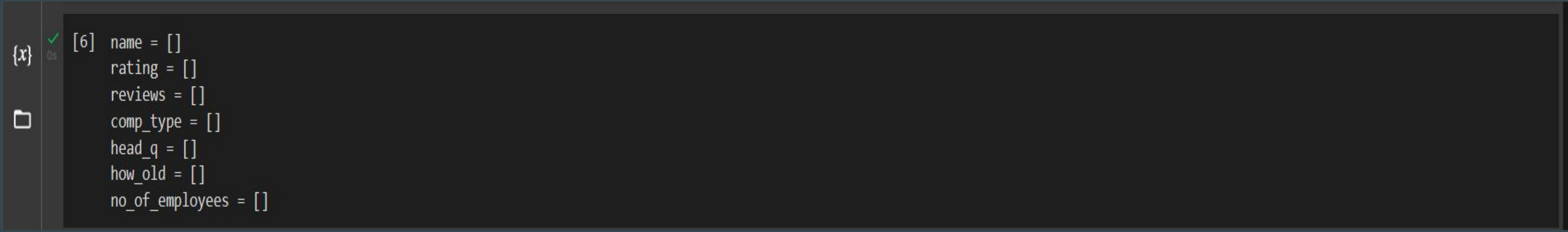


```python
for i in soup.find_all('p'):
  print(i.text.strip())
```

```
AmbitionBox
Discover best places to work
Compare & find best workplace
Read reviews for 6L+ companies
Rate your former or current company
Discover salaries for 6L+ companies
Calculate Your take home salary
Help other jobseekers
Read interviews for 40K+ companies
Interviews questions for 1K+ colleges
Contribute your interview questions
Discover Best Places to Work!
Company reviews. Salaries. Interviews. Jobs.
About Company
6,93,405 unique
                        companies found
Sort By:
Popular
4.2
Private
London + 6 more
23 years old
1k-5k Employees (India)
Photon Infotech Private Limited is an information technology and services company based out of Omr, Chennai, Tamil Nadu, India.
4.1
Private
London + 11 more
63 years old
1k-5k Employees (India)
We bring together the deepest intelligence across the widest set of capital-intensive industries and markets. By connecting data across variables, our analysts and industry special
```

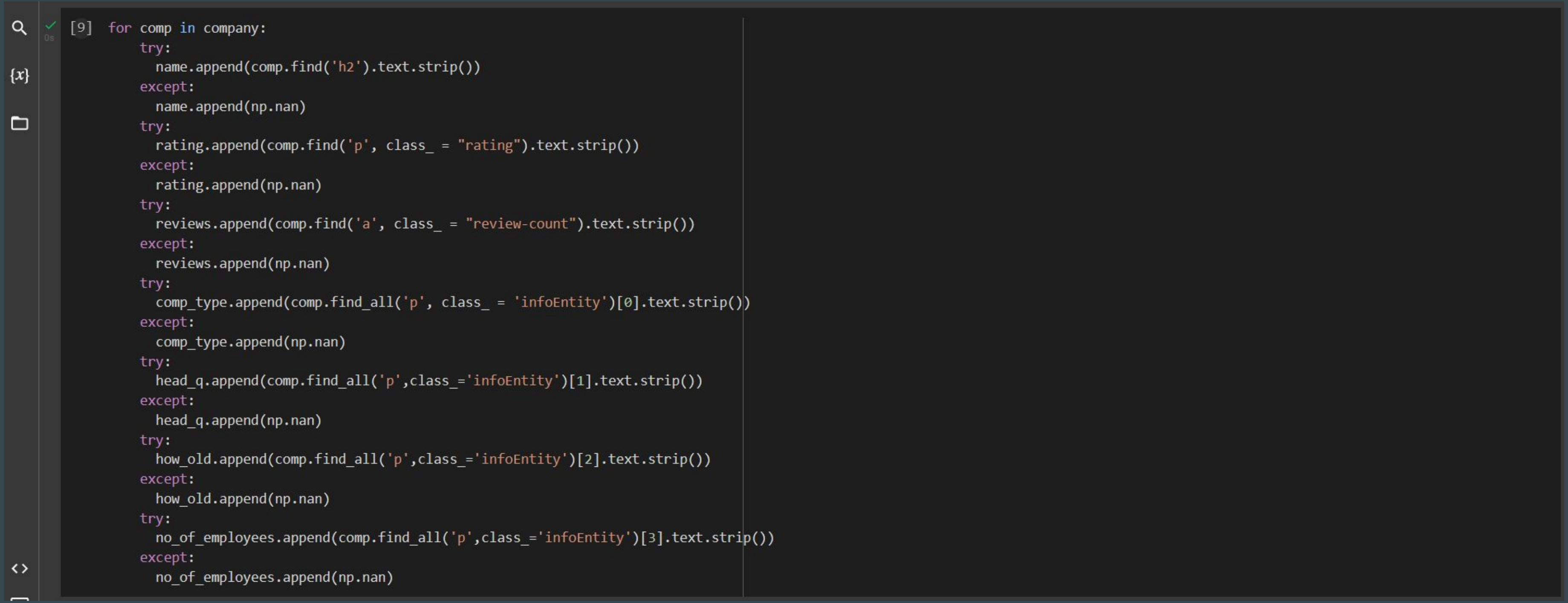
✓ 0s completed at 11:29 PM

Scraping data from webpage



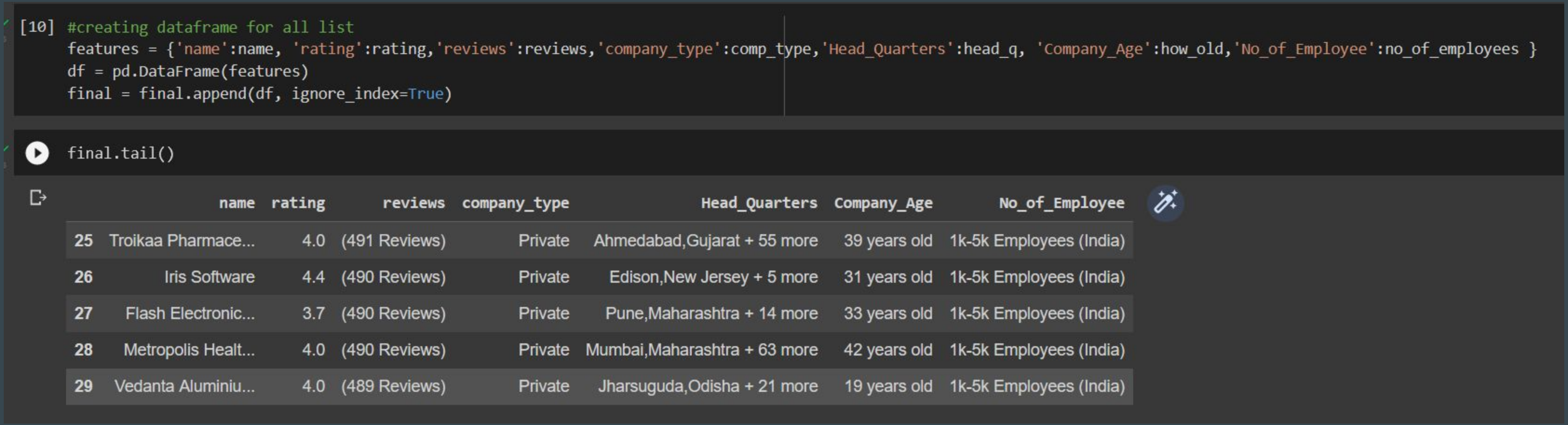
```
name = []
rating = []
reviews = []
comp_type = []
head_q = []
how_old = []
no_of_employees = []
```

Adding column headings to the dataset



```python
for comp in company:
    try:
      name.append(comp.find('h2').text.strip())
    except:
      name.append(np.nan)
    try:
      rating.append(comp.find('p', class_ = "rating").text.strip())
    except:
      rating.append(np.nan)
    try:
      reviews.append(comp.find('a', class_ = "review-count").text.strip())
    except:
      reviews.append(np.nan)
    try:
      comp_type.append(comp.find_all('p', class_ = 'infoEntity')[0].text.strip())
    except:
      comp_type.append(np.nan)
    try:
      head_q.append(comp.find_all('p',class_='infoEntity')[1].text.strip())
    except:
      head_q.append(np.nan)
    try:
      how_old.append(comp.find_all('p',class_='infoEntity')[2].text.strip())
    except:
      how_old.append(np.nan)
    try:
      no_of_employees.append(comp.find_all('p',class_='infoEntity')[3].text.strip())
    except:
      no_of_employees.append(np.nan)
```

Collection of data using HTML element tags



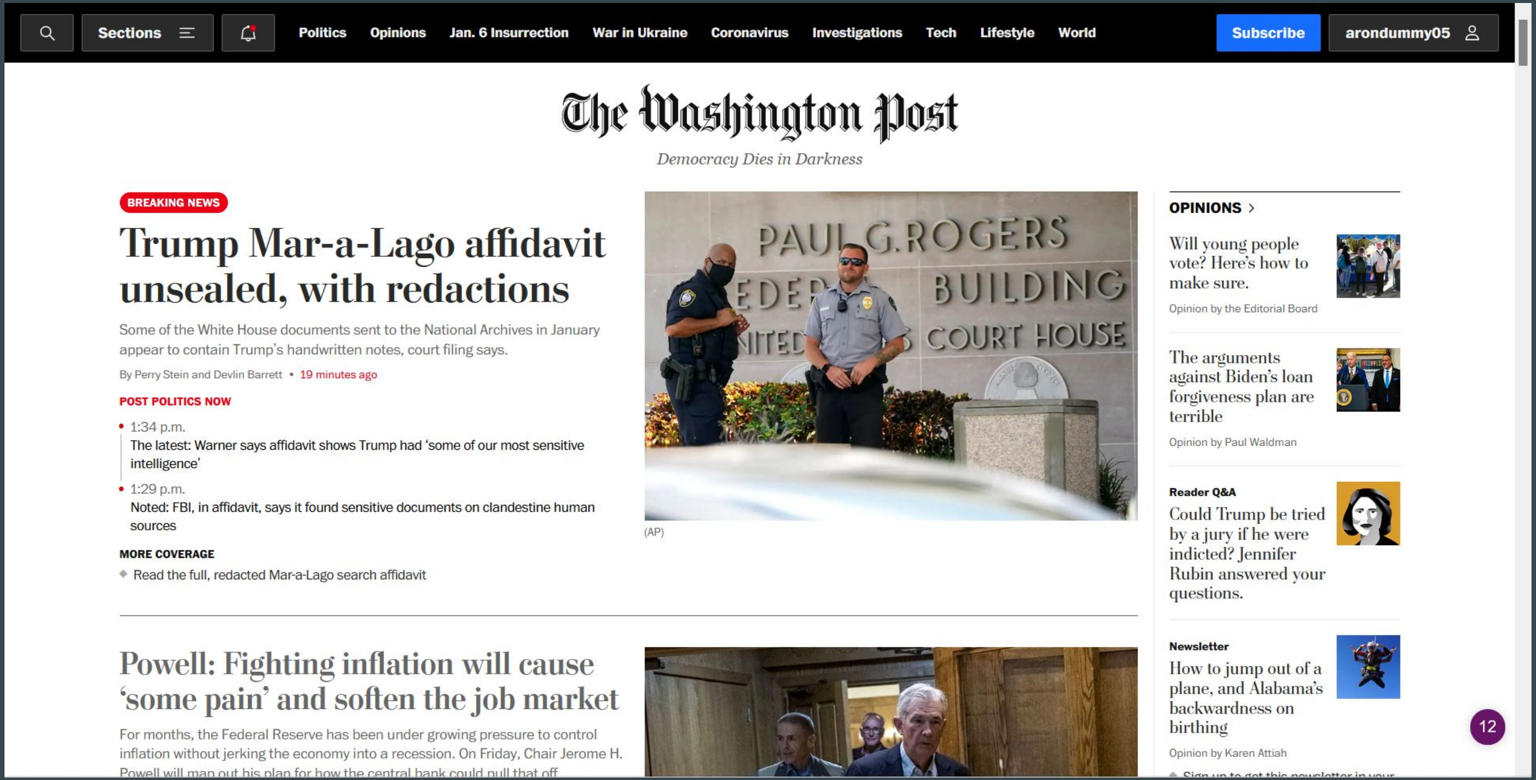
```
[10] #creating dataframe for all list
     features = {'name':name, 'rating':rating,'reviews':reviews,'company_type':comp_type,'Head_Quarters':head_q, 'Company_Age':how_old,'No_of_Employee':no_of_employees }
     df = pd.DataFrame(features)
     final = final.append(df, ignore_index=True)
```

```
final.tail()
```

	name	rating	reviews	company_type	Head_Quarters	Company_Age	No_of_Employee
25	Troikaa Pharmace...	4.0	(491 Reviews)	Private	Ahmedabad,Gujarat + 55 more	39 years old	1k-5k Employees (India)
26	Iris Software	4.4	(490 Reviews)	Private	Edison,New Jersey + 5 more	31 years old	1k-5k Employees (India)
27	Flash Electronic...	3.7	(490 Reviews)	Private	Pune,Maharashtra + 14 more	33 years old	1k-5k Employees (India)
28	Metropolis Healt...	4.0	(490 Reviews)	Private	Mumbai,Maharashtra + 63 more	42 years old	1k-5k Employees (India)
29	Vedanta Aluminiu...	4.0	(489 Reviews)	Private	Jharsuguda,Odisha + 21 more	19 years old	1k-5k Employees (India)

Creating Dataframe

Target website:



The Washington Post

Democracy Dies in Darkness

BREAKING NEWS

Trump Mar-a-Lago affidavit unsealed, with redactions

Some of the White House documents sent to the National Archives in January appear to contain Trump's handwritten notes, court filing says.

By Perry Stein and Devlin Barrett • 19 minutes ago

POST POLITICS NOW

- 1:34 p.m.
  The latest: Warner says affidavit shows Trump had 'some of our most sensitive intelligence'
- 1:29 p.m.
  Noted: FBI, in affidavit, says it found sensitive documents on clandestine human sources

MORE COVERAGE

- Read the full, redacted Mar-a-Lago search affidavit

(AP)

Powell: Fighting inflation will cause 'some pain' and soften the job market

For months, the Federal Reserve has been under growing pressure to control inflation without jerking the economy into a recession. On Friday, Chair Jerome H. Powell will map out his plan for how the central bank could pull that off

OPINIONS >

Will young people vote? Here's how to make sure.

Opinion by the Editorial Board

The arguments against Biden's loan forgiveness plan are terrible

Opinion by Paul Waldman

Reader Q&A

Could Trump be tried by a jury if he were indicted? Jennifer Rubin answered your questions.

Newsletter

How to jump out of a plane, and Alabama's backwardness on birthing

Opinion by Karen Attiah

- Sign up to get this newsletter in your

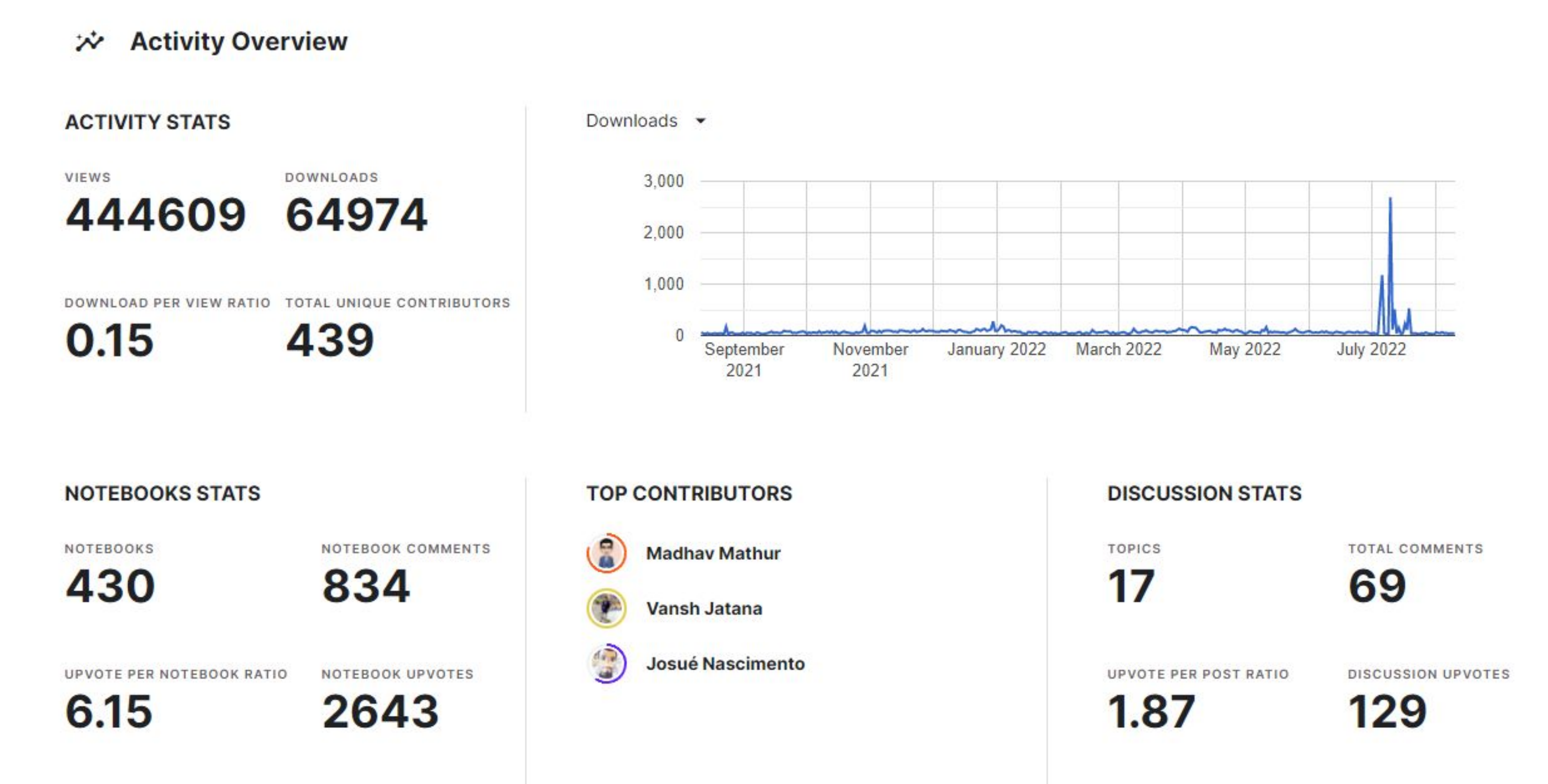
DATASETS

Fake.csv:

The dataset is formed by collecting absolute fake news curated manually to match with the factual dataset.There are (ostensibly) no genuine, reliable, or trustworthy news sources represented in this dataset (so far), so don't trust anything you read.

True.csv:

This dataset is formed by web scraping a centrist site “reuters.com” and verified manually to be true.



Activity Overview

ACTIVITY STATS

VIEWS	DOWNLOADS
444609	64974

DOWNLOAD PER VIEW RATIO	TOTAL UNIQUE CONTRIBUTORS
0.15	439

Downloads ▾

NOTEBOOKS STATS

NOTEBOOKS	NOTEBOOK COMMENTS
430	834

UPVOTE PER NOTEBOOK RATIO	NOTEBOOK UPVOTES
6.15	2643

TOP CONTRIBUTORS

- Madhav Mathur
- Vansh Jatana
- Josué Nascimento

DISCUSSION STATS

TOPICS	TOTAL COMMENTS
17	69

UPVOTE PER POST RATIO	DISCUSSION UPVOTES
1.87	129

The main objective was first to perform NLP(Natural Language Processing) Classification using PyCaret Library over fake and real news Dataset.

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular it deals with how to program computers to process and analyze large amounts of natural language data.

The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them.

The technology can then accurately extract information and content contained in the documents as well as categorize and organize the documents themselves.

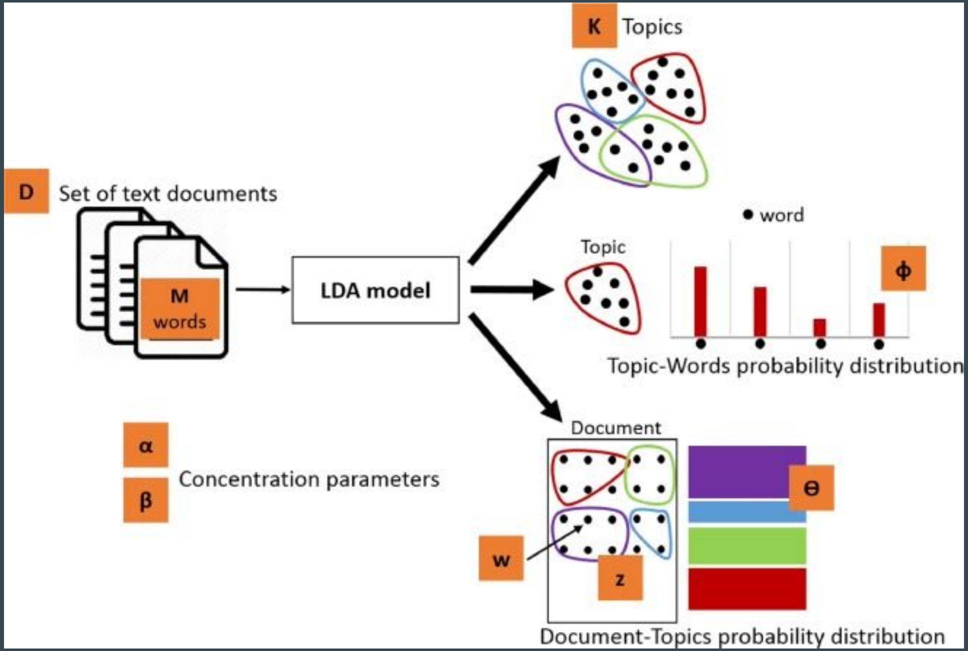
NLP Text Classification using PyCaret

PyCaret is a simple, easy to learn, low-code machine learning library in Python. With PyCaret, you spend less time coding and more time on analysis.

- Exploratory data analysis
- Data preprocessing
- Model Training
- Model Explainability
- MLOps

```
!pip install pycaret[full]
```

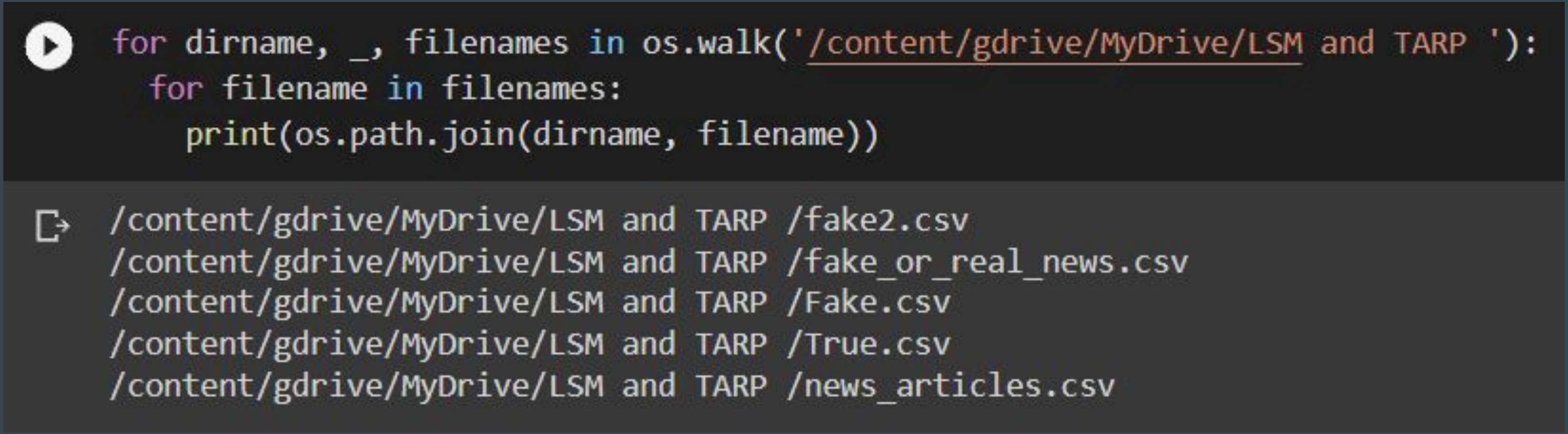
Latent Dirichlet Allocation



In natural language processing, the latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a topic model.

Accessing the Directory

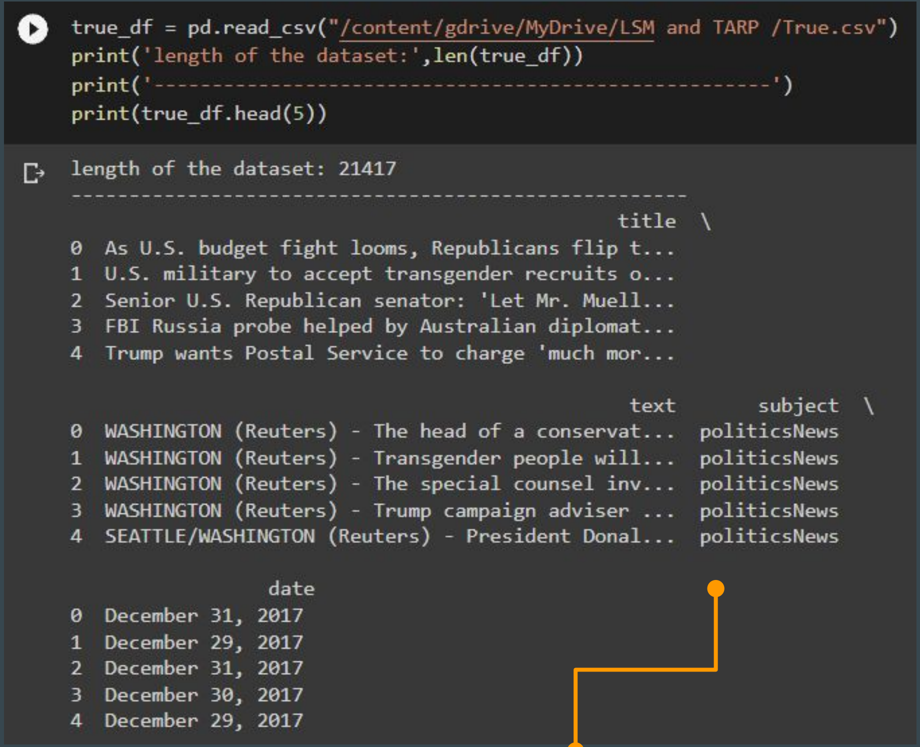


```
for dirname, _, filenames in os.walk('/content/gdrive/MyDrive/LSM and TARP '):
  for filename in filenames:
    print(os.path.join(dirname, filename))
```

```
/content/gdrive/MyDrive/LSM and TARP /fake2.csv
/content/gdrive/MyDrive/LSM and TARP /fake_or_real_news.csv
/content/gdrive/MyDrive/LSM and TARP /Fake.csv
/content/gdrive/MyDrive/LSM and TARP /True.csv
/content/gdrive/MyDrive/LSM and TARP /news_articles.csv
```

We have used Google Colab as a primary platform for executing our code

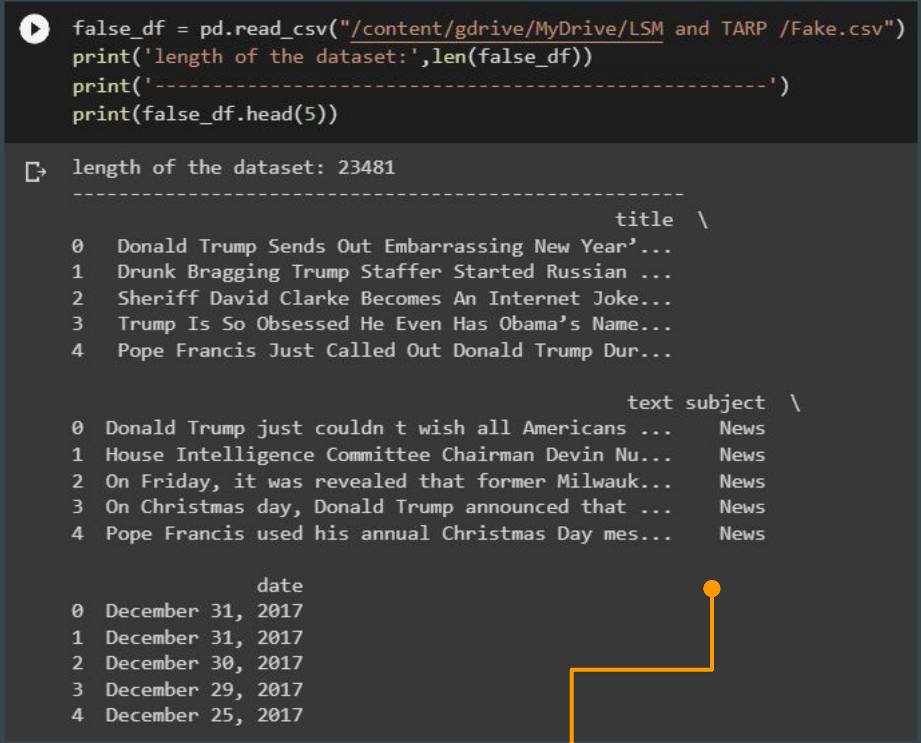
Analysing the Dataset



```
true_df = pd.read_csv("/content/gdrive/MyDrive/LSM and TARP /True.csv")
print('length of the dataset:',len(true_df))
print('-----------------------------------------------------')
print(true_df.head(5))
```

```
length of the dataset: 21417
-----------------------------------------------------
                                               title  \
0  As U.S. budget fight looms, Republicans flip t...
1  U.S. military to accept transgender recruits o...
2  Senior U.S. Republican senator: 'Let Mr. Muell...
3  FBI Russia probe helped by Australian diplomat...
4  Trump wants Postal Service to charge 'much mor...

                                                text       subject  \
0  WASHINGTON (Reuters) - The head of a conservat...  politicsNews
1  WASHINGTON (Reuters) - Transgender people will...  politicsNews
2  WASHINGTON (Reuters) - The special counsel inv...  politicsNews
3  WASHINGTON (Reuters) - Trump campaign adviser ...  politicsNews
4  SEATTLE/WASHINGTON (Reuters) - President Donal...  politicsNews

                 date
0  December 31, 2017
1  December 29, 2017
2  December 31, 2017
3  December 30, 2017
4  December 29, 2017
```

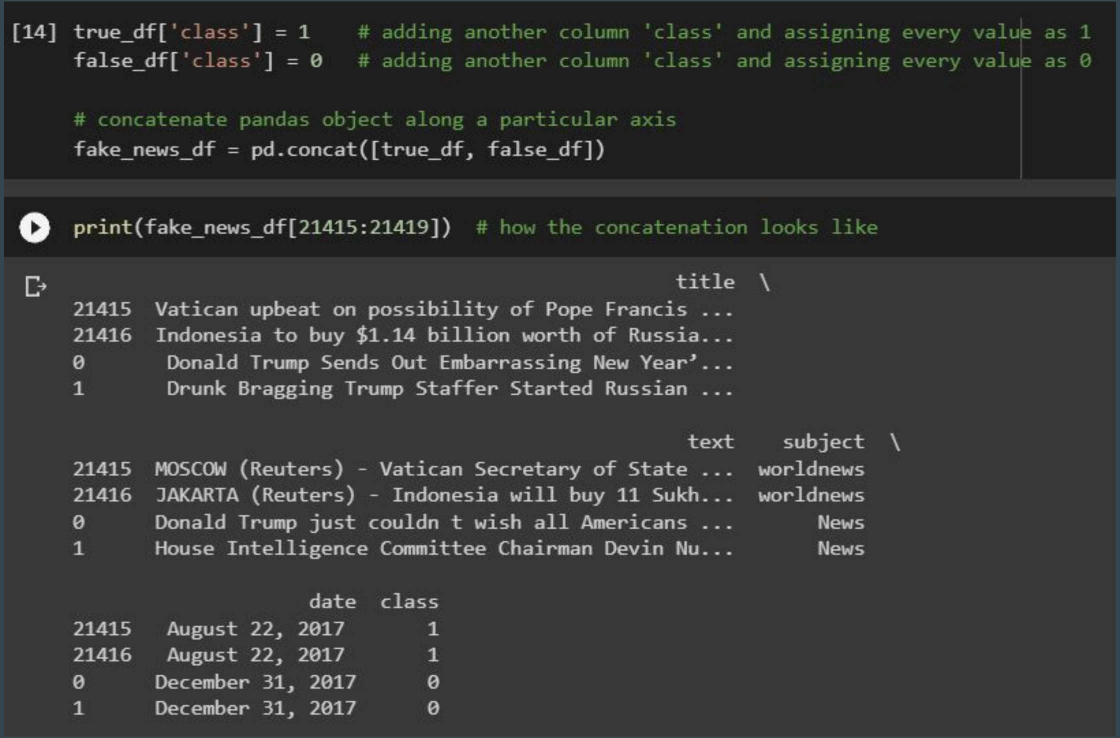
True.csv



```
false_df = pd.read_csv("/content/gdrive/MyDrive/LSM and TARP /Fake.csv")
print('length of the dataset:',len(false_df))
print('-----------------------------------------------------')
print(false_df.head(5))
```

```
length of the dataset: 23481
-----------------------------------------------------
                                               title  \
0   Donald Trump Sends Out Embarrassing New Year’...
1   Drunk Bragging Trump Staffer Started Russian ...
2   Sheriff David Clarke Becomes An Internet Joke...
3   Trump Is So Obsessed He Even Has Obama’s Name...
4   Pope Francis Just Called Out Donald Trump Dur...

                                                text subject  \
0  Donald Trump just couldn t wish all Americans ...    News
1  House Intelligence Committee Chairman Devin Nu...    News
2  On Friday, it was revealed that former Milwauk...    News
3  On Christmas day, Donald Trump announced that ...    News
4  Pope Francis used his annual Christmas Day mes...    News

                 date
0  December 31, 2017
1  December 31, 2017
2  December 30, 2017
3  December 29, 2017
4  December 25, 2017
```

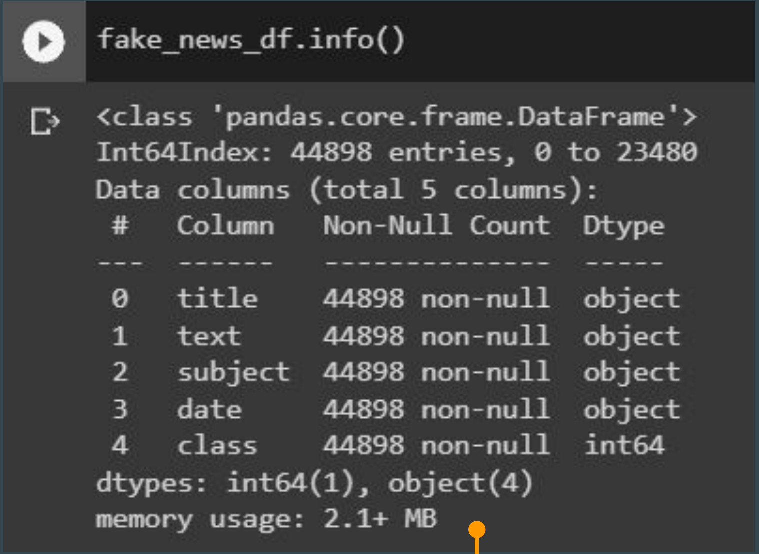
False.csv



```
[14] true_df['class'] = 1    # adding another column 'class' and assigning every value as 1
     false_df['class'] = 0   # adding another column 'class' and assigning every value as 0

     # concatenate pandas object along a particular axis
     fake_news_df = pd.concat([true_df, false_df])
```

```
print(fake_news_df[21415:21419])  # how the concatenation looks like
```

```
                                                   title  \
21415  Vatican upbeat on possibility of Pope Francis ...
21416  Indonesia to buy $1.14 billion worth of Russia...
0       Donald Trump Sends Out Embarrassing New Year’...
1       Drunk Bragging Trump Staffer Started Russian ...

                                                    text    subject  \
21415  MOSCOW (Reuters) - Vatican Secretary of State ...  worldnews
21416  JAKARTA (Reuters) - Indonesia will buy 11 Sukh...  worldnews
0      Donald Trump just couldn t wish all Americans ...       News
1      House Intelligence Committee Chairman Devin Nu...       News

                    date  class
21415   August 22, 2017      1
21416   August 22, 2017      1
0      December 31, 2017      0
1      December 31, 2017      0
```

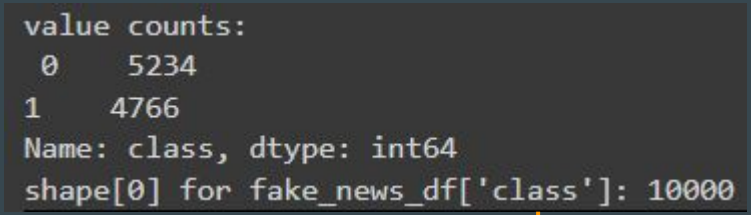


```
fake_news_df.info()
```

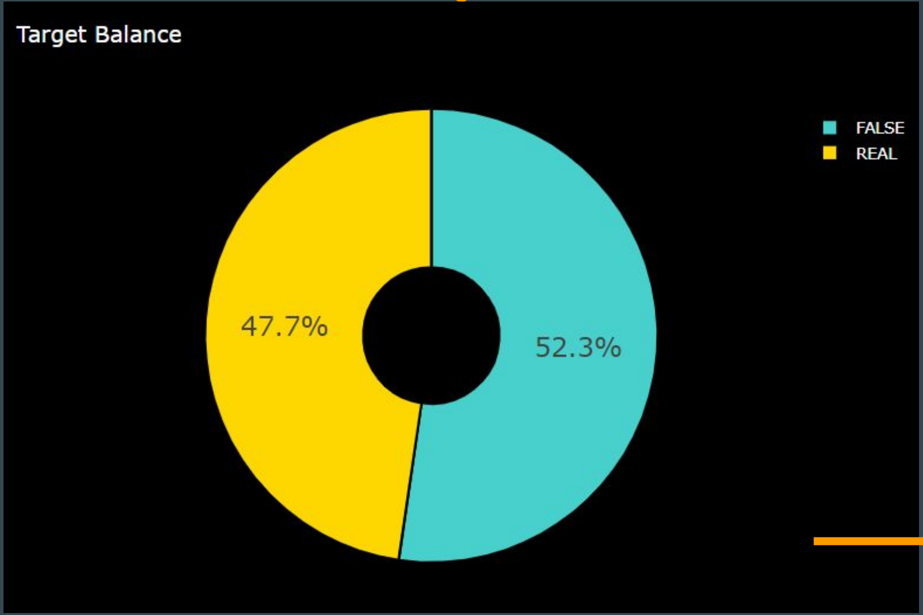
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   title    44898 non-null  object
 1   text     44898 non-null  object
 2   subject  44898 non-null  object
 3   date     44898 non-null  object
 4   class    44898 non-null  int64
dtypes: int64(1), object(4)
memory usage: 2.1+ MB
```

Information on the concatenation of
True and False News

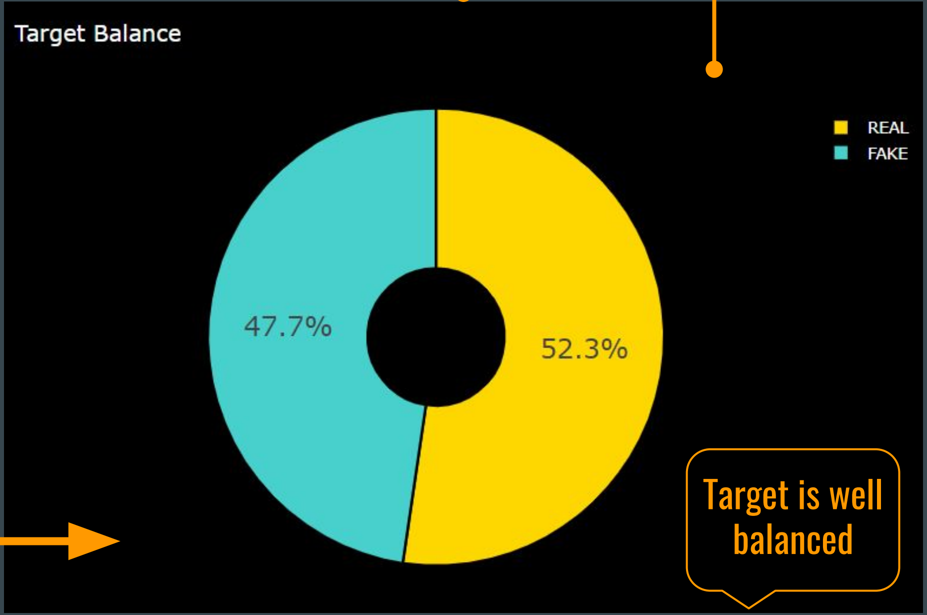
Using
import plotly.graph_objects as go
Helps in building figures



```
value counts:
 0    5234
1    4766
Name: class, dtype: int64
shape[0] for fake_news_df['class']: 10000
```



Initially



Target is well balanced

After Sampling is done

Detection of Fake News Using Machine Learning Models

Tarp Project Review 3

Under the Guidance of Jacob Raglend Sir

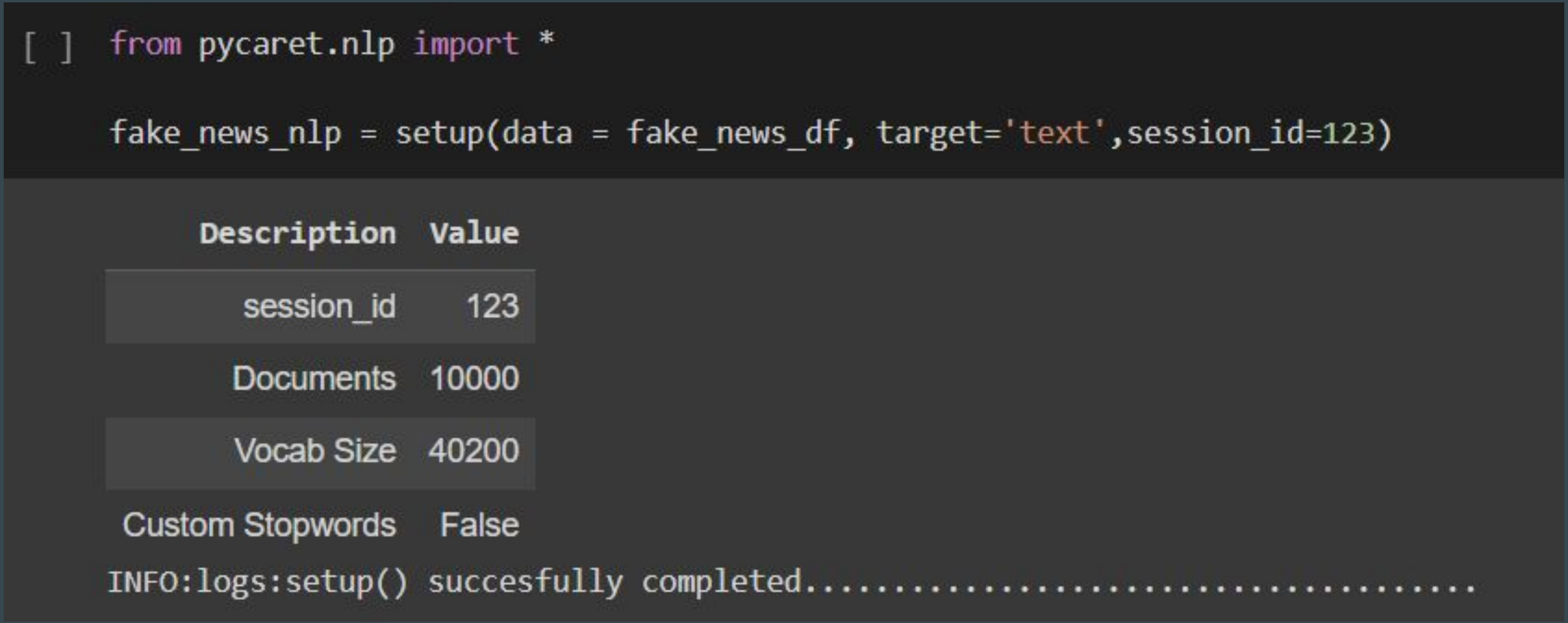
The main objective was first to perform NLP(Natural Language Processing) Classification using PyCaret Library over fake and real news Dataset.

NLP Text Classification using PyCaret

PyCaret is a simple, easy to learn, low-code machine learning library in Python. With PyCaret, you spend less time coding and more time on analysis.

- Exploratory data analysis
- Data preprocessing
- Model Training
- Model Explainability
- MLOps

```
!pip install pycaret[full]
```

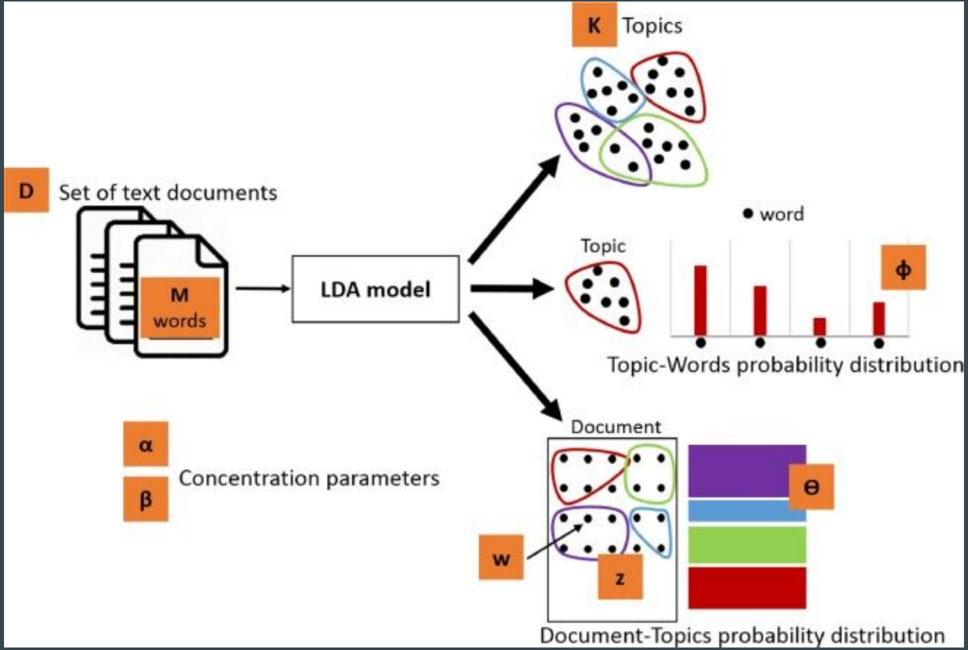


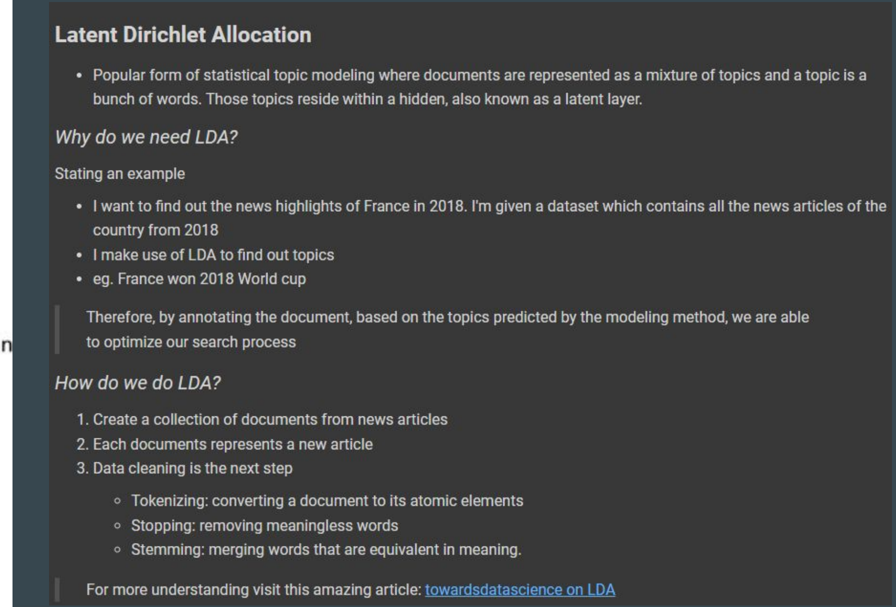
```
[ ] from pycaret.nlp import *

    fake_news_nlp = setup(data = fake_news_df, target='text',session_id=123)
```

Description	Value
session_id	123
Documents	10000
Vocab Size	40200
Custom Stopwords	False

```
INFO:logs:setup() succesfully completed......................................
```

Latent Dirichlet Allocation





Latent Dirichlet Allocation

- Popular form of statistical topic modeling where documents are represented as a mixture of topics and a topic is a bunch of words. Those topics reside within a hidden, also known as a latent layer.

Why do we need LDA?

Stating an example

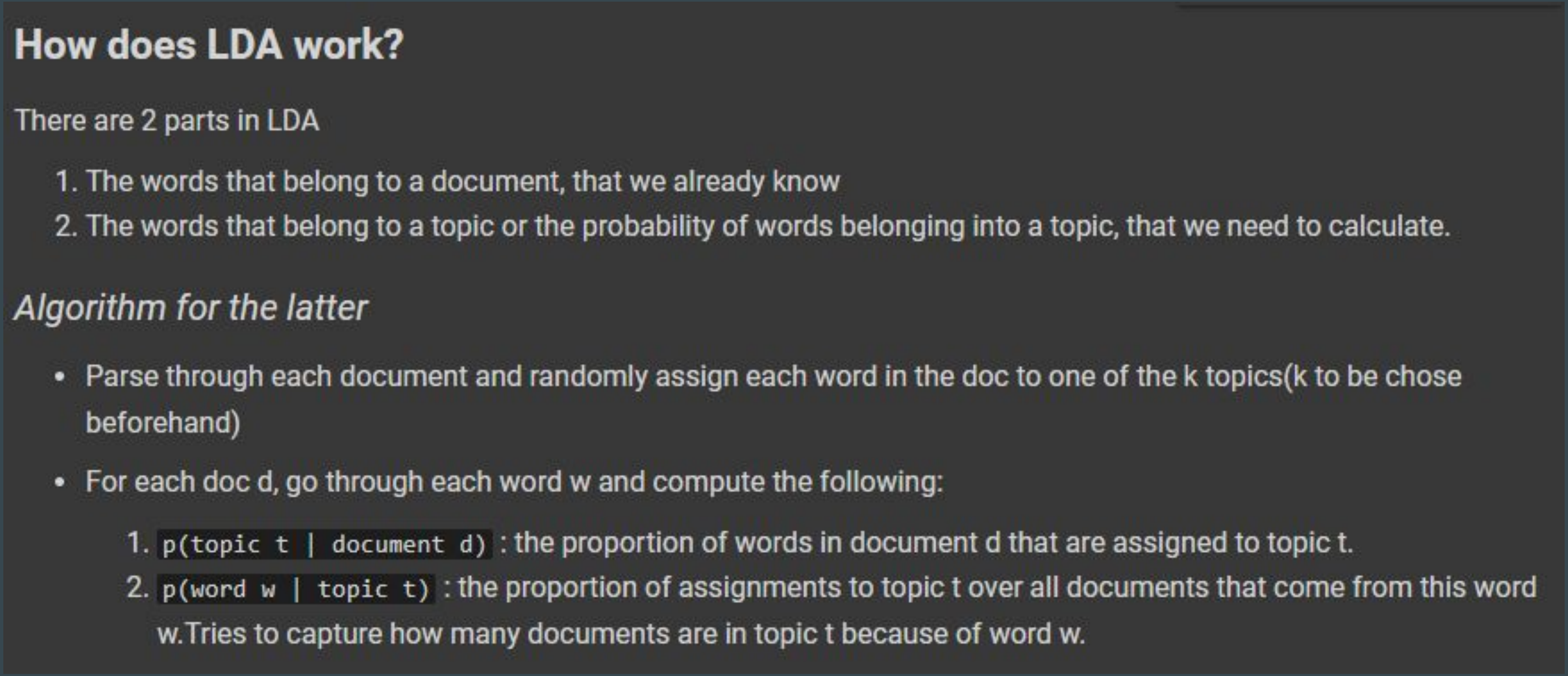
- I want to find out the news highlights of France in 2018. I'm given a dataset which contains all the news articles of the country from 2018
- I make use of LDA to find out topics
- eg. France won 2018 World cup

> Therefore, by annotating the document, based on the topics predicted by the modeling method, we are able to optimize our search process

How do we do LDA?

1. Create a collection of documents from news articles
2. Each documents represents a new article
3. Data cleaning is the next step
    - Tokenizing: converting a document to its atomic elements
    - Stopping: removing meaningless words
    - Stemming: merging words that are equivalent in meaning.

> For more understanding visit this amazing article: towardsdatascience on LDA



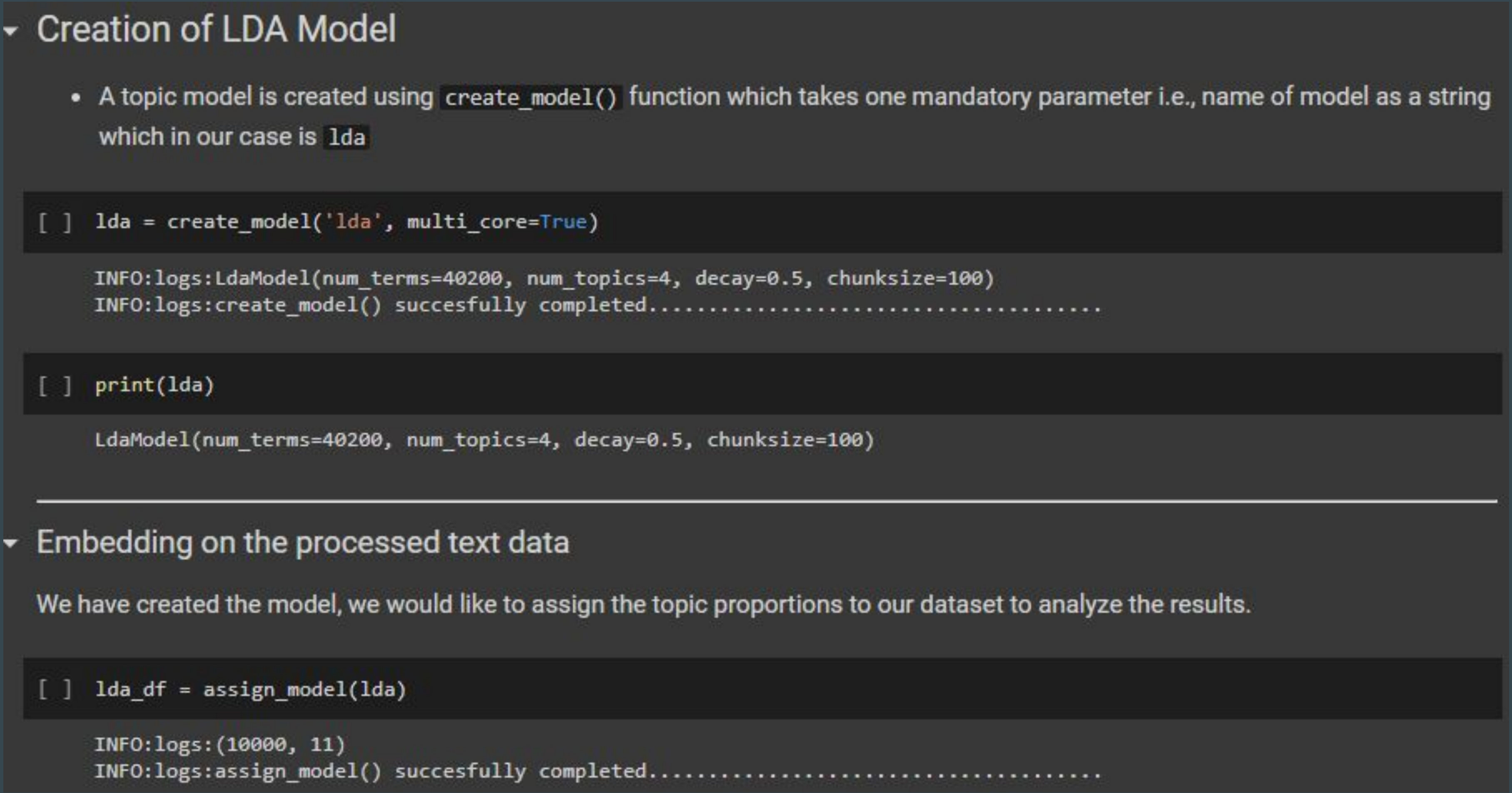
How does LDA work?

There are 2 parts in LDA

1. The words that belong to a document, that we already know
2. The words that belong to a topic or the probability of words belonging into a topic, that we need to calculate.

Algorithm for the latter

- Parse through each document and randomly assign each word in the doc to one of the k topics(k to be chose beforehand)
- For each doc d, go through each word w and compute the following:
    1. `p(topic t | document d)` : the proportion of words in document d that are assigned to topic t.
    2. `p(word w | topic t)` : the proportion of assignments to topic t over all documents that come from this word w.Tries to capture how many documents are in topic t because of word w.



Creation of LDA Model

- A topic model is created using `create_model()` function which takes one mandatory parameter i.e., name of model as a string which in our case is `lda`

```
[ ]  lda = create_model('lda', multi_core=True)
```

```
INFO:logs:LdaModel(num_terms=40200, num_topics=4, decay=0.5, chunksize=100)
INFO:logs:create_model() succesfully completed......................................
```

```
[ ]  print(lda)
```

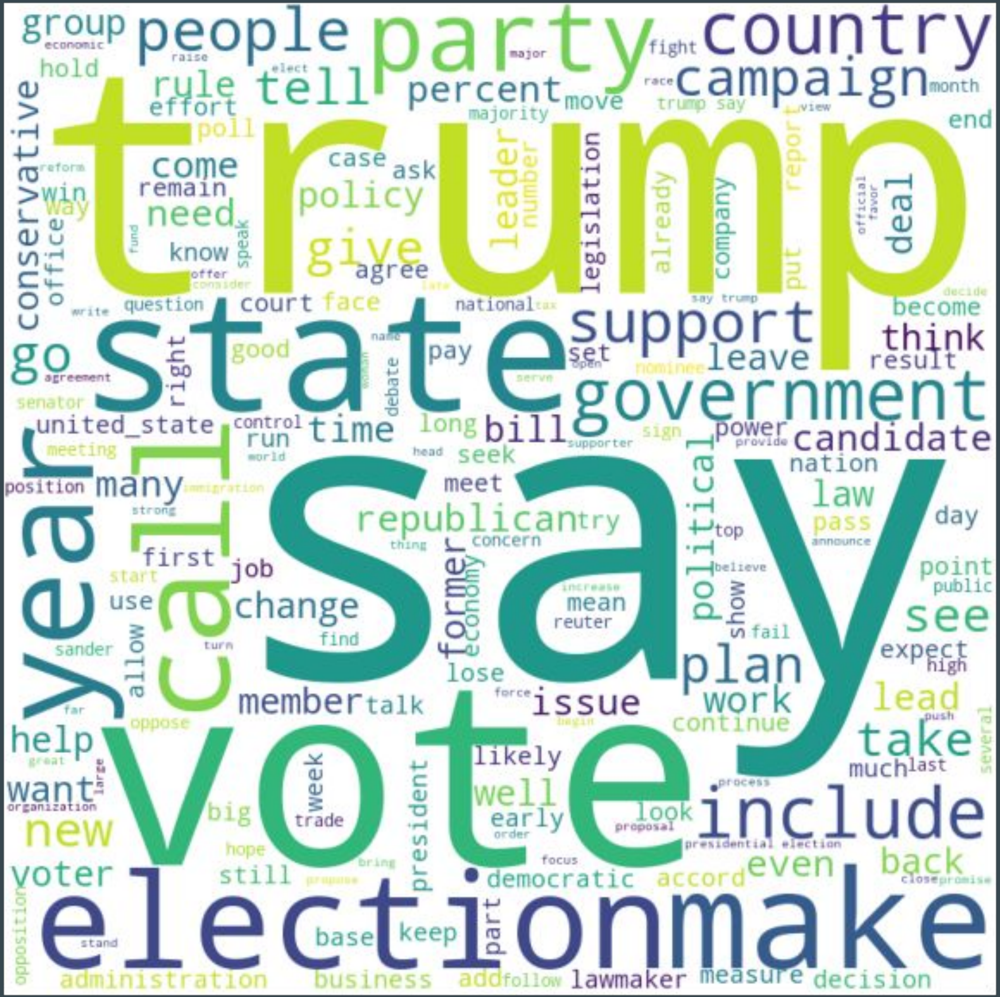
```
LdaModel(num_terms=40200, num_topics=4, decay=0.5, chunksize=100)
```

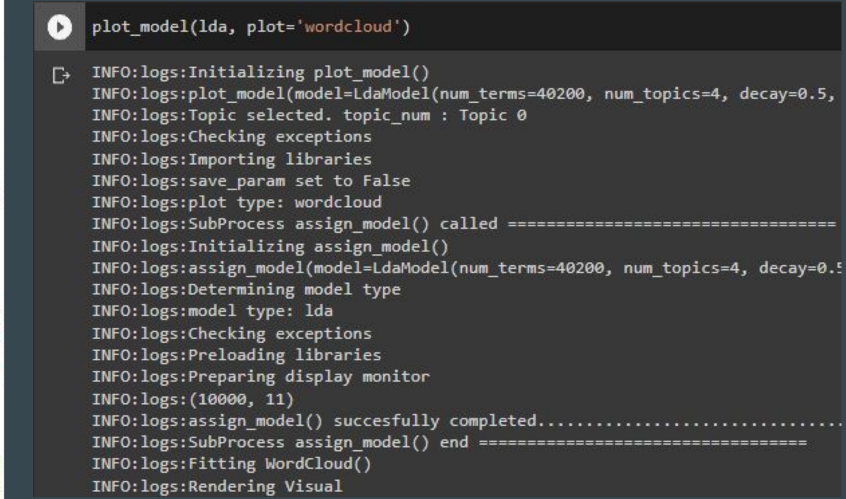
Embedding on the processed text data

We have created the model, we would like to assign the topic proportions to our dataset to analyze the results.

```
[ ]  lda_df = assign_model(lda)
```

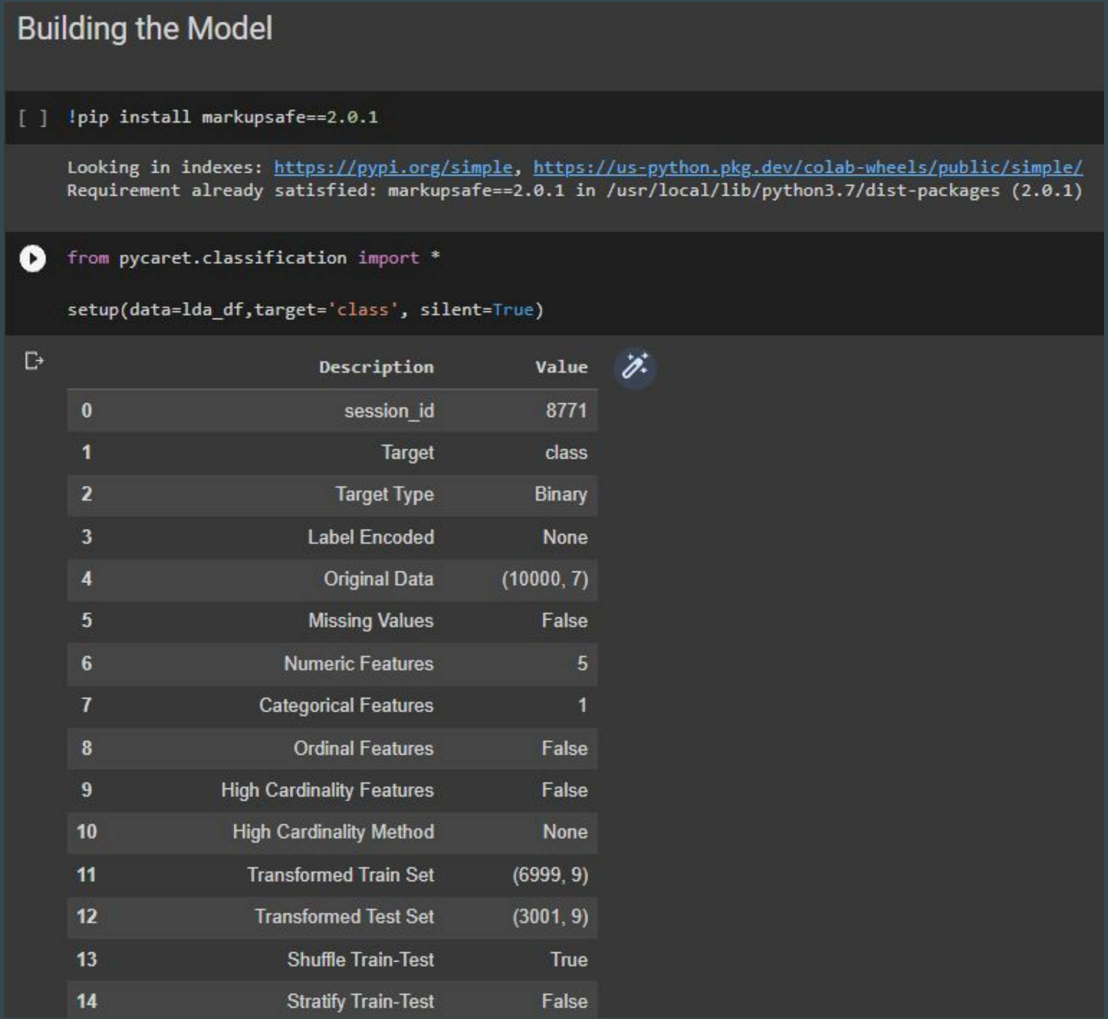
```
INFO:logs:(10000, 11)
INFO:logs:assign_model() succesfully completed......................................
```





```
plot_model(lda, plot='wordcloud')
```

```
INFO:logs:Initializing plot_model()
INFO:logs:plot_model(model=LdaModel(num_terms=40200, num_topics=4, decay=0.5,
INFO:logs:Topic selected. topic_num : Topic 0
INFO:logs:Checking exceptions
INFO:logs:Importing libraries
INFO:logs:save_param set to False
INFO:logs:plot type: wordcloud
INFO:logs:SubProcess assign_model() called ==================================
INFO:logs:Initializing assign_model()
INFO:logs:assign_model(model=LdaModel(num_terms=40200, num_topics=4, decay=0.5
INFO:logs:Determining model type
INFO:logs:model type: lda
INFO:logs:Checking exceptions
INFO:logs:Preloading libraries
INFO:logs:Preparing display monitor
INFO:logs:(10000, 11)
INFO:logs:assign_model() succesfully completed..............................
INFO:logs:SubProcess assign_model() end ==================================
INFO:logs:Fitting WordCloud()
INFO:logs:Rendering Visual
```



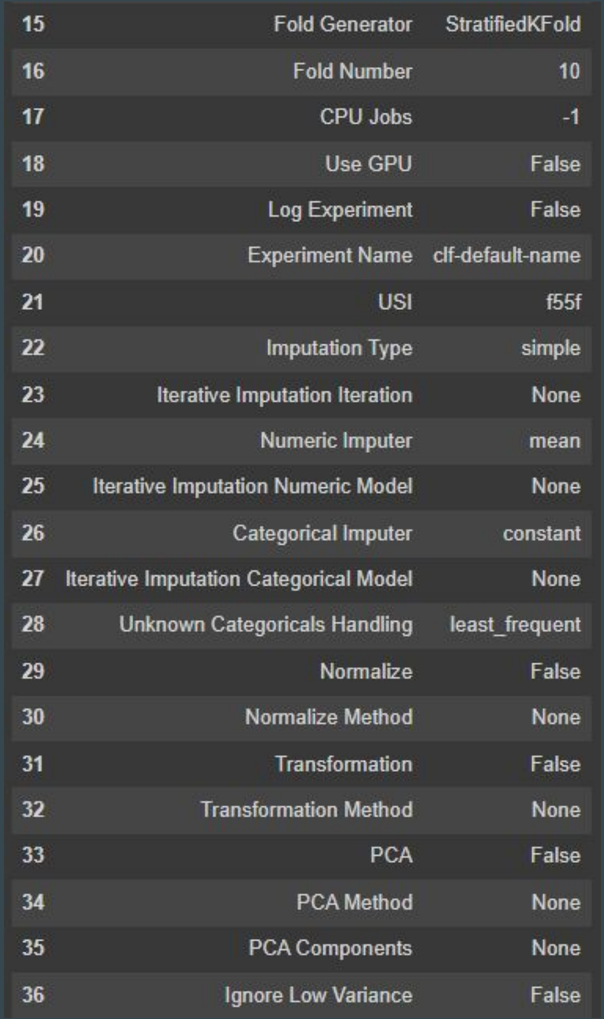
Building the Model

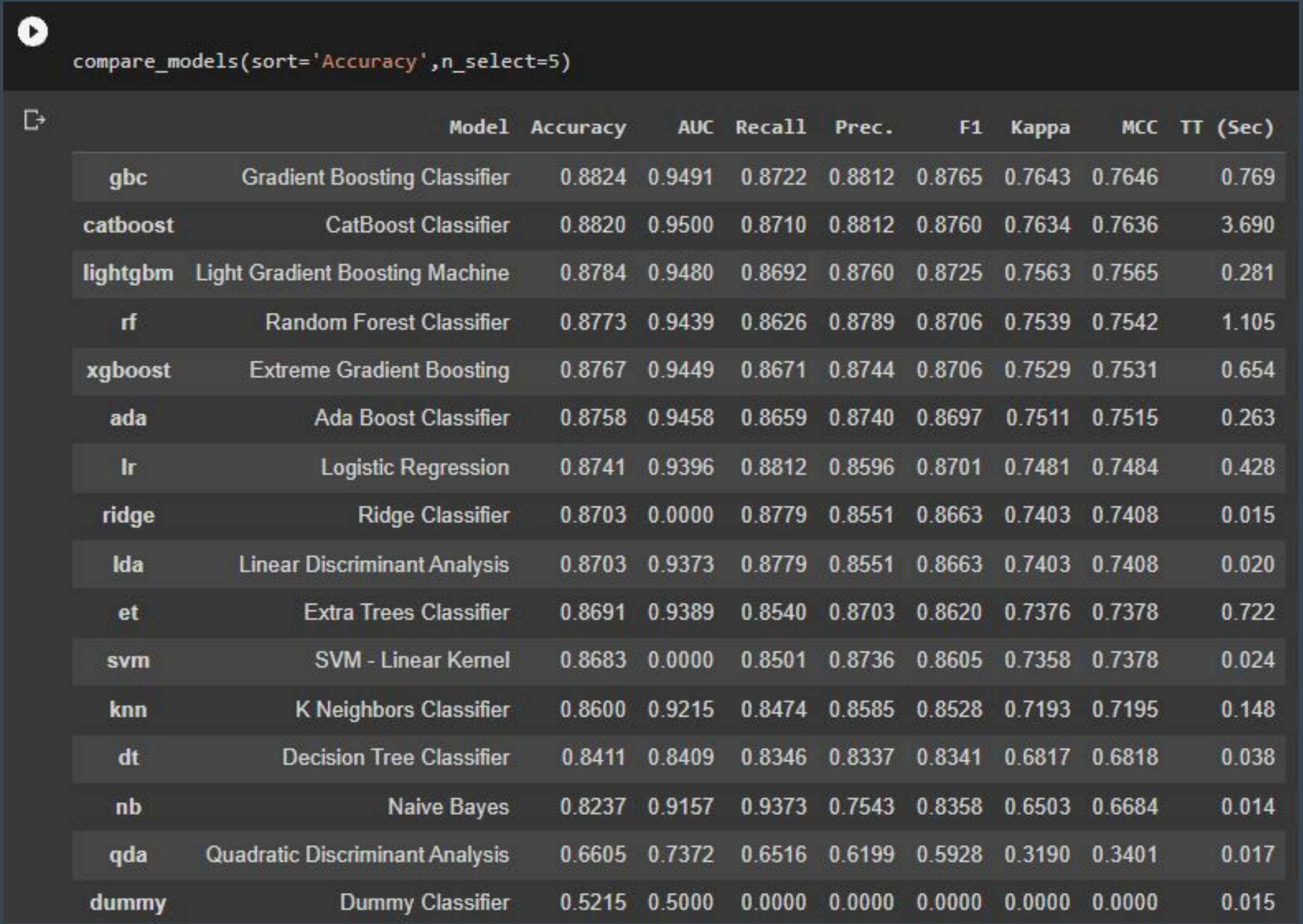
```
!pip install markupsafe==2.0.1
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: markupsafe==2.0.1 in /usr/local/lib/python3.7/dist-packages (2.0.1)
```

```
from pycaret.classification import *

setup(data=lda_df,target='class', silent=True)
```

	Description	Value
0	session_id	8771
1	Target	class
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(10000, 7)
5	Missing Values	False
6	Numeric Features	5
7	Categorical Features	1
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(6999, 9)
12	Transformed Test Set	(3001, 9)
13	Shuffle Train-Test	True
14	Stratify Train-Test	False
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	f55f
22	Imputation Type	simple
23	Iterative Imputation Iteration	None
24	Numeric Imputer	mean
25	Iterative Imputation Numeric Model	None
26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False
30	Normalize Method	None
31	Transformation	False
32	Transformation Method	None
33	PCA	False
34	PCA Method	None
35	PCA Components	None
36	Ignore Low Variance	False

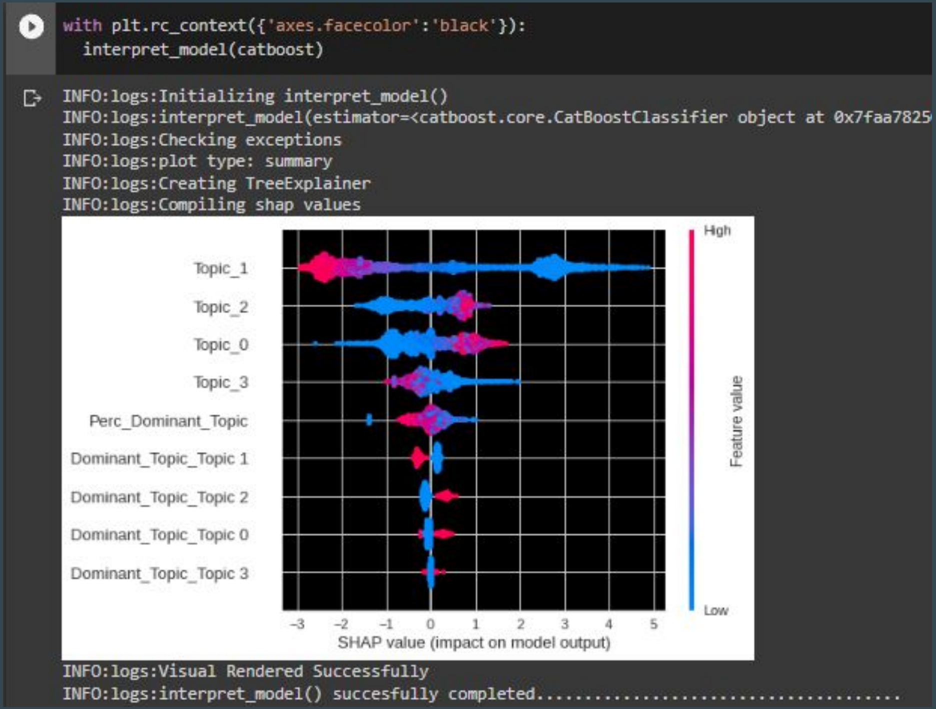




```
compare_models(sort='Accuracy',n_select=5)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.8824	0.9491	0.8722	0.8812	0.8765	0.7643	0.7646	0.769
catboost	CatBoost Classifier	0.8820	0.9500	0.8710	0.8812	0.8760	0.7634	0.7636	3.690
lightgbm	Light Gradient Boosting Machine	0.8784	0.9480	0.8692	0.8760	0.8725	0.7563	0.7565	0.281
rf	Random Forest Classifier	0.8773	0.9439	0.8626	0.8789	0.8706	0.7539	0.7542	1.105
xgboost	Extreme Gradient Boosting	0.8767	0.9449	0.8671	0.8744	0.8706	0.7529	0.7531	0.654
ada	Ada Boost Classifier	0.8758	0.9458	0.8659	0.8740	0.8697	0.7511	0.7515	0.263
lr	Logistic Regression	0.8741	0.9396	0.8812	0.8596	0.8701	0.7481	0.7484	0.428
ridge	Ridge Classifier	0.8703	0.0000	0.8779	0.8551	0.8663	0.7403	0.7408	0.015
lda	Linear Discriminant Analysis	0.8703	0.9373	0.8779	0.8551	0.8663	0.7403	0.7408	0.020
et	Extra Trees Classifier	0.8691	0.9389	0.8540	0.8703	0.8620	0.7376	0.7378	0.722
svm	SVM - Linear Kernel	0.8683	0.0000	0.8501	0.8736	0.8605	0.7358	0.7378	0.024
knn	K Neighbors Classifier	0.8600	0.9215	0.8474	0.8585	0.8528	0.7193	0.7195	0.148
dt	Decision Tree Classifier	0.8411	0.8409	0.8346	0.8337	0.8341	0.6817	0.6818	0.038
nb	Naive Bayes	0.8237	0.9157	0.9373	0.7543	0.8358	0.6503	0.6684	0.014
qda	Quadratic Discriminant Analysis	0.6605	0.7372	0.6516	0.6199	0.5928	0.3190	0.3401	0.017
dummy	Dummy Classifier	0.5215	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.015

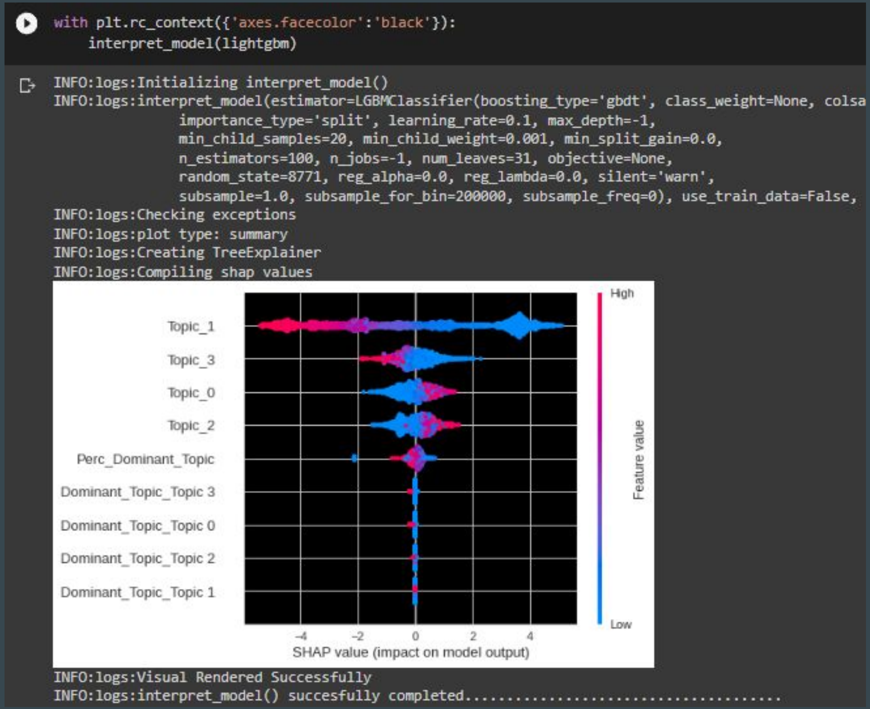
Interpreting the Model



```
with plt.rc_context({'axes.facecolor':'black'}):
  interpret_model(catboost)
```

```
INFO:logs:Initializing interpret_model()
INFO:logs:interpret_model(estimator=<catboost.core.CatBoostClassifier object at 0x7faa7825
INFO:logs:Checking exceptions
INFO:logs:plot type: summary
INFO:logs:Creating TreeExplainer
INFO:logs:Compiling shap values
```

```
INFO:logs:Visual Rendered Successfully
INFO:logs:interpret_model() succesfully completed......................................
```

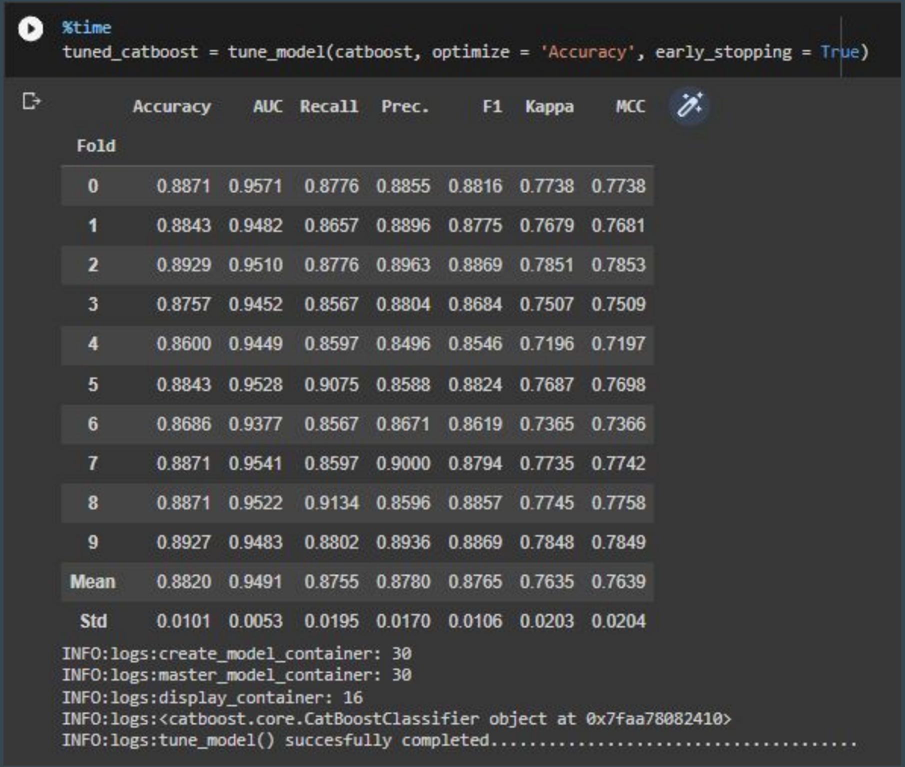


```
with plt.rc_context({'axes.facecolor':'black'}):
    interpret_model(lightgbm)
```

```
INFO:logs:Initializing interpret_model()
INFO:logs:interpret_model(estimator=LGBMClassifier(boosting_type='gbdt', class_weight=None, colsa
               importance_type='split', learning_rate=0.1, max_depth=-1,
               min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
               n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
               random_state=8771, reg_alpha=0.0, reg_lambda=0.0, silent='warn',
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0), use_train_data=False,
INFO:logs:Checking exceptions
INFO:logs:plot type: summary
INFO:logs:Creating TreeExplainer
INFO:logs:Compiling shap values
```

```
INFO:logs:Visual Rendered Successfully
INFO:logs:interpret_model() succesfully completed..................................
```

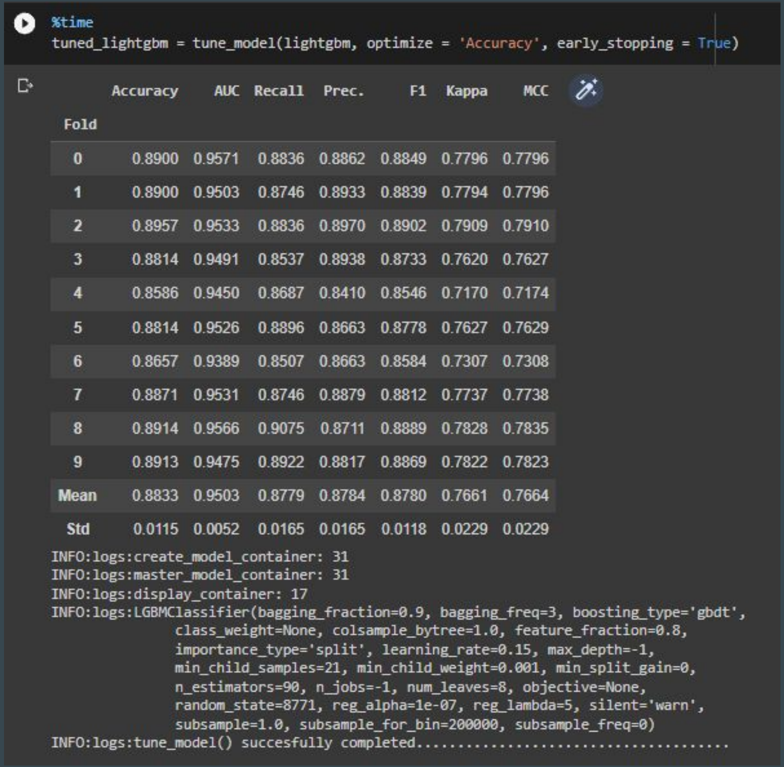
Tuning the Hyperparameters



```
%time
tuned_catboost = tune_model(catboost, optimize = 'Accuracy', early_stopping = True)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8871	0.9571	0.8776	0.8855	0.8816	0.7738	0.7738
1	0.8843	0.9482	0.8657	0.8896	0.8775	0.7679	0.7681
2	0.8929	0.9510	0.8776	0.8963	0.8869	0.7851	0.7853
3	0.8757	0.9452	0.8567	0.8804	0.8684	0.7507	0.7509
4	0.8600	0.9449	0.8597	0.8496	0.8546	0.7196	0.7197
5	0.8843	0.9528	0.9075	0.8588	0.8824	0.7687	0.7698
6	0.8686	0.9377	0.8567	0.8671	0.8619	0.7365	0.7366
7	0.8871	0.9541	0.8597	0.9000	0.8794	0.7735	0.7742
8	0.8871	0.9522	0.9134	0.8596	0.8857	0.7745	0.7758
9	0.8927	0.9483	0.8802	0.8936	0.8869	0.7848	0.7849
Mean	0.8820	0.9491	0.8755	0.8780	0.8765	0.7635	0.7639
Std	0.0101	0.0053	0.0195	0.0170	0.0106	0.0203	0.0204

```
INFO:logs:create_model_container: 30
INFO:logs:master_model_container: 30
INFO:logs:display_container: 16
INFO:logs:<catboost.core.CatBoostClassifier object at 0x7faa78082410>
INFO:logs:tune_model() succesfully completed......................................
```

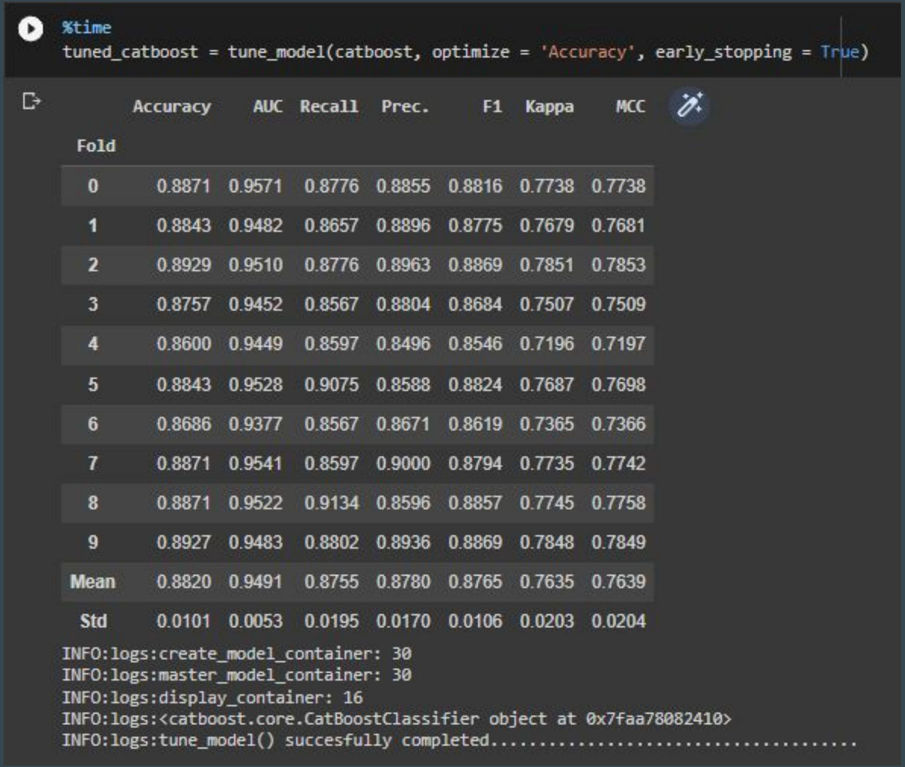


```
%time
tuned_lightgbm = tune_model(lightgbm, optimize = 'Accuracy', early_stopping = True)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8900	0.9571	0.8836	0.8862	0.8849	0.7796	0.7796
1	0.8900	0.9503	0.8746	0.8933	0.8839	0.7794	0.7796
2	0.8957	0.9533	0.8836	0.8970	0.8902	0.7909	0.7910
3	0.8814	0.9491	0.8537	0.8938	0.8733	0.7620	0.7627
4	0.8586	0.9450	0.8687	0.8410	0.8546	0.7170	0.7174
5	0.8814	0.9526	0.8896	0.8663	0.8778	0.7627	0.7629
6	0.8657	0.9389	0.8507	0.8663	0.8584	0.7307	0.7308
7	0.8871	0.9531	0.8746	0.8879	0.8812	0.7737	0.7738
8	0.8914	0.9566	0.9075	0.8711	0.8889	0.7828	0.7835
9	0.8913	0.9475	0.8922	0.8817	0.8869	0.7822	0.7823
Mean	0.8833	0.9503	0.8779	0.8784	0.8780	0.7661	0.7664
Std	0.0115	0.0052	0.0165	0.0165	0.0118	0.0229	0.0229

```
INFO:logs:create_model_container: 31
INFO:logs:master_model_container: 31
INFO:logs:display_container: 17
INFO:logs:LGBMClassifier(bagging_fraction=0.9, bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.8,
               importance_type='split', learning_rate=0.15, max_depth=-1,
               min_child_samples=21, min_child_weight=0.001, min_split_gain=0,
               n_estimators=90, n_jobs=-1, num_leaves=8, objective=None,
               random_state=8771, reg_alpha=1e-07, reg_lambda=5, silent='warn',
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
INFO:logs:tune_model() succesfully completed......................................
```

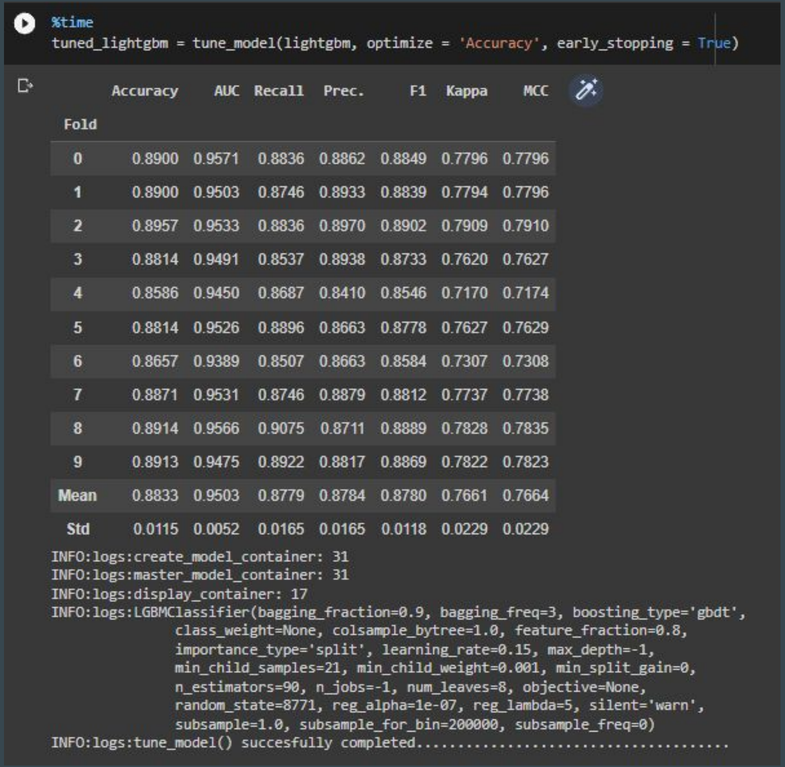
Voting Classifier



```
%time
tuned_catboost = tune_model(catboost, optimize = 'Accuracy', early_stopping = True)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8871	0.9571	0.8776	0.8855	0.8816	0.7738	0.7738
1	0.8843	0.9482	0.8657	0.8896	0.8775	0.7679	0.7681
2	0.8929	0.9510	0.8776	0.8963	0.8869	0.7851	0.7853
3	0.8757	0.9452	0.8567	0.8804	0.8684	0.7507	0.7509
4	0.8600	0.9449	0.8597	0.8496	0.8546	0.7196	0.7197
5	0.8843	0.9528	0.9075	0.8588	0.8824	0.7687	0.7698
6	0.8686	0.9377	0.8567	0.8671	0.8619	0.7365	0.7366
7	0.8871	0.9541	0.8597	0.9000	0.8794	0.7735	0.7742
8	0.8871	0.9522	0.9134	0.8596	0.8857	0.7745	0.7758
9	0.8927	0.9483	0.8802	0.8936	0.8869	0.7848	0.7849
Mean	0.8820	0.9491	0.8755	0.8780	0.8765	0.7635	0.7639
Std	0.0101	0.0053	0.0195	0.0170	0.0106	0.0203	0.0204

```
INFO:logs:create_model_container: 30
INFO:logs:master_model_container: 30
INFO:logs:display_container: 16
INFO:logs:<catboost.core.CatBoostClassifier object at 0x7faa78082410>
INFO:logs:tune_model() succesfully completed......................................
```

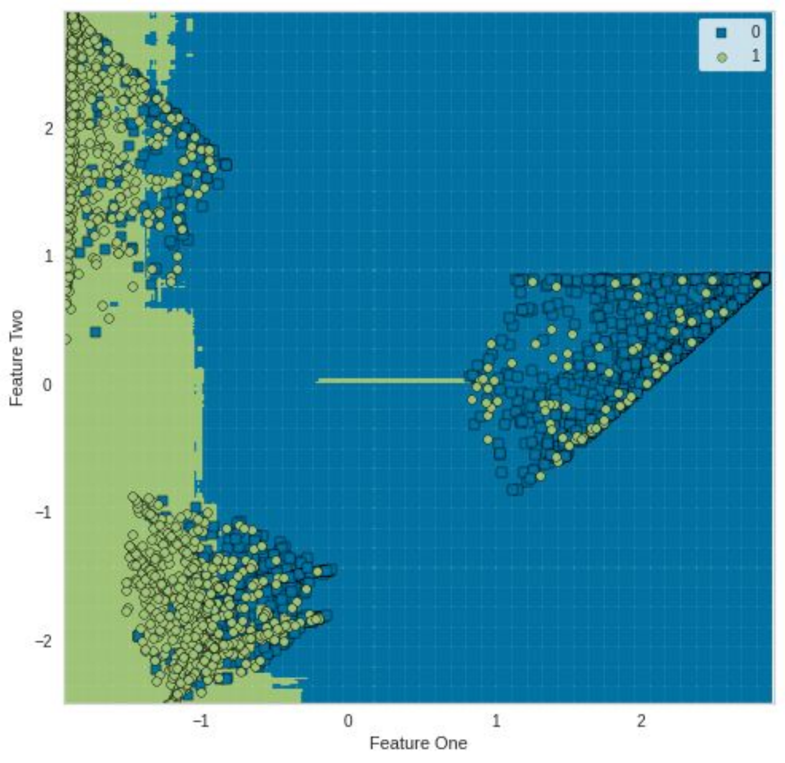


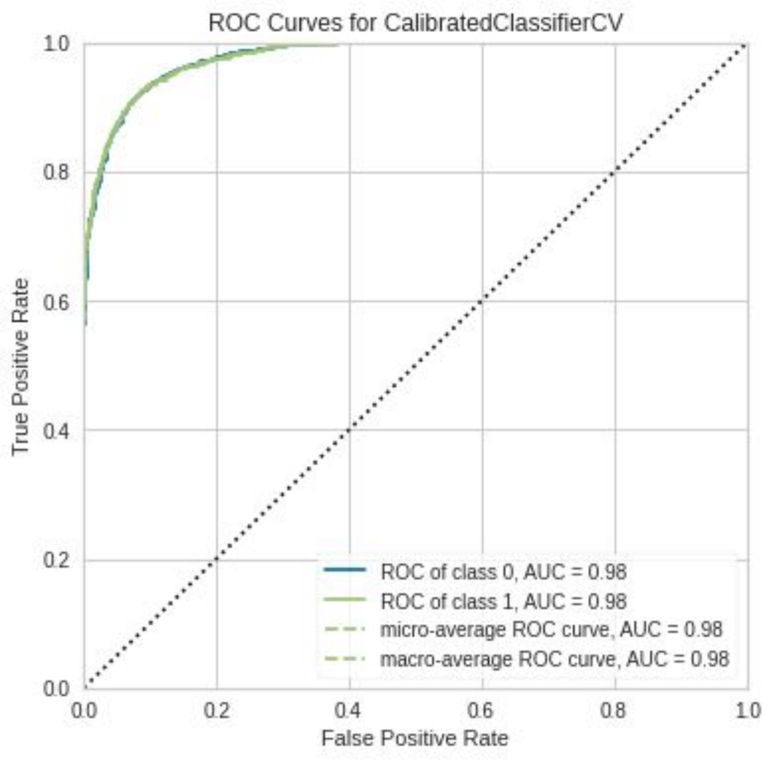
```
%time
tuned_lightgbm = tune_model(lightgbm, optimize = 'Accuracy', early_stopping = True)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8900	0.9571	0.8836	0.8862	0.8849	0.7796	0.7796
1	0.8900	0.9503	0.8746	0.8933	0.8839	0.7794	0.7796
2	0.8957	0.9533	0.8836	0.8970	0.8902	0.7909	0.7910
3	0.8814	0.9491	0.8537	0.8938	0.8733	0.7620	0.7627
4	0.8586	0.9450	0.8687	0.8410	0.8546	0.7170	0.7174
5	0.8814	0.9526	0.8896	0.8663	0.8778	0.7627	0.7629
6	0.8657	0.9389	0.8507	0.8663	0.8584	0.7307	0.7308
7	0.8871	0.9531	0.8746	0.8879	0.8812	0.7737	0.7738
8	0.8914	0.9566	0.9075	0.8711	0.8889	0.7828	0.7835
9	0.8913	0.9475	0.8922	0.8817	0.8869	0.7822	0.7823
Mean	0.8833	0.9503	0.8779	0.8784	0.8780	0.7661	0.7664
Std	0.0115	0.0052	0.0165	0.0165	0.0118	0.0229	0.0229

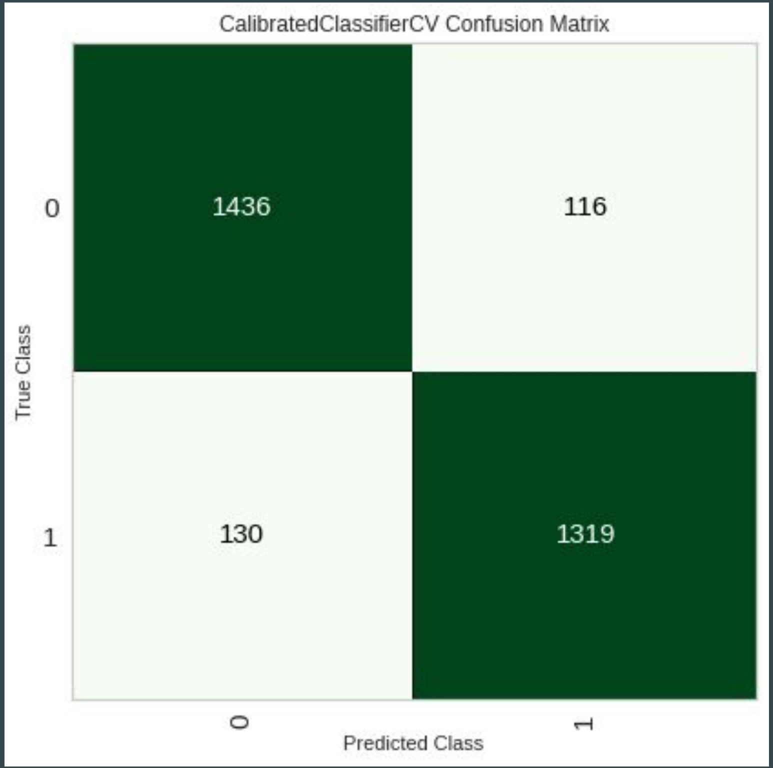
```
INFO:logs:create_model_container: 31
INFO:logs:master_model_container: 31
INFO:logs:display_container: 17
INFO:logs:LGBMClassifier(bagging_fraction=0.9, bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.8,
               importance_type='split', learning_rate=0.15, max_depth=-1,
               min_child_samples=21, min_child_weight=0.001, min_split_gain=0,
               n_estimators=90, n_jobs=-1, num_leaves=8, objective=None,
               random_state=8771, reg_alpha=1e-07, reg_lambda=5, silent='warn',
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
INFO:logs:tune_model() succesfully completed......................................
```

Plotting Results of Models

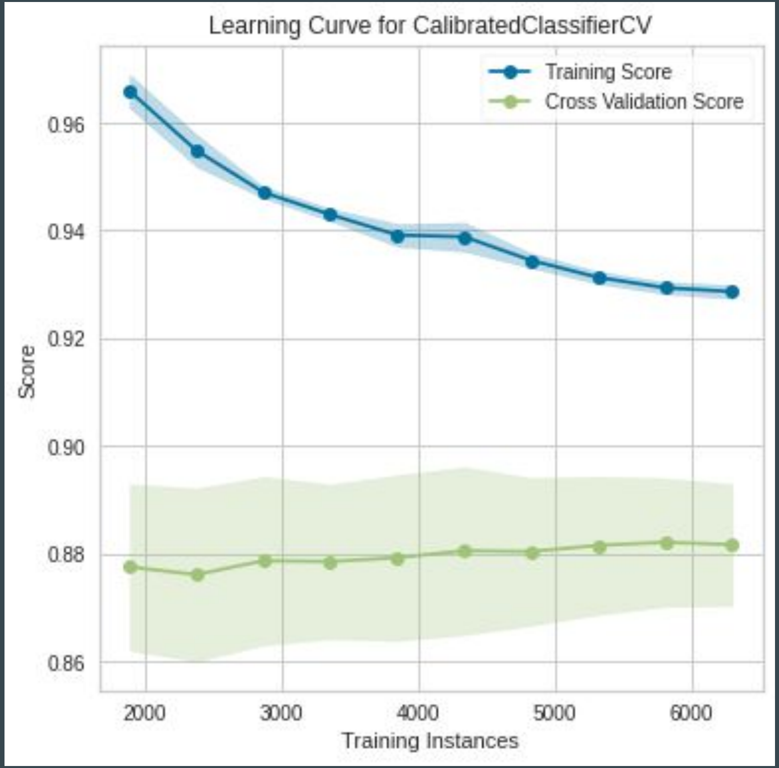




Plotting Results of Models



	0	1
0	1436	116
1	130	1319



Thank You !