A mini project report submitted on

FAKE NEWS DETECTION

For the partial fulfillment for the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

Submitted by

G.Shyamala	319129510019
M.Gopi	319129510027
G.Surya Vamsi	319129510017
T.Praneeth	319129510061
B.Prem Kumar	319129510004

Under the Esteemed Guidance of

Mrs.T.ANUSHA M.tech

Assistant Professor

Department of Computer Science & Engineering

WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT

(Approved by AICTE , New Delhi and affiliated to Andhra University)Pinagadi,Pendurthi,
Visakhapatnam-531173,Andhra Pradesh,India.

WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT

(Approved by AICTE , New Delhi and affiliated to Andhra University)Pinagadi,Pendurthi,
Visakhapatnam-531173,Andhra Pradesh,India.

CERTIFICATE

This is to certify that the mini project report entitled **“Fake News Detection”** being submitted by

G.Shyamala	**319129510019**
M.Gopi	**319129510027**
G.Surya Vamsi	**319129510017**
T.Praneeth	**319129510061**
B.Prem Kumar	**319129510004**

In partial fulfillments for the award of the degree Bachelor of Technology in Computer Science and Engineering to the Andhra Pradesh is a record of bonafide project work carried out under my guidance and supervision. This results embodied in this project report have not been submitted to any University for the award of any degree.

INTERNAL GUIDE

Mrs.T ANUSHA M.tech
Assistant Professor ,
Department of CSE ,
WISTM Engg.College

HEAD OF THE DEPARTMENT

Mrs. K.V.LAKSHMI M.Tech
Assistant Professor,
Department of CSE,
WISTM Engg.College

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crown all the effort with success. I would like to thank Dr.M.James Stephen, Principal, Wellfare Institute of Science Technology and Management, for his kind cooperation. It gives me boundless pleasure to avail this opportunity to express my deep sense of gratitude and wholehearted thanks to Head of the Department **Mrs. K.V.LAKSHMI, Assistant Professor**, Department of CSE and all the faculty of computer science department for their valuable suggestions and co-operation during the development of this project.

G.Shyamala	**319129510019**
M.Gopi	**319129510027**
G.Surya Vamsi	**319129510017**
T.Praneeth	**319129510061**
B.Prem Kumar	**319129510004**

DECLARATION

I here declare that the project entitled “Fake News Detection” has been done under the guidance of **Mrs.T.Anusha,Assistant Professor, Department of CSE** and is dissertation of my own work except where specifically ask to the contrary and is submitted to the department of computer science and Engineering, wellfare Insistute of science technology and management for the partial fulfilment of the requirement for the award of B.Tech degree.

G.Shyamala	**319129510019**
M.Gopi	**319129510027**
G.Surya Vamsi	**319129510017**
T.Praneeth	**319129510061**
B.Prem Kumar	**319129510004**

TABLE OF CONTENTS

1. ABSTRACT

In our modern era where the internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in the use of social media platforms like Facebook, Twitter, etc. news spread rapidly among millions of users within a very short span of time. The spread of fake news has far-reaching consequences like the creation of biased opinions to swaying election outcomes for the benefit of certain candidates. Moreover, spammers use appealing news headlines to generate revenue using advertisements via click-baits. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

2. INTRODUCTION

As an increasing amount of our lives is spent interacting online through social media platforms, more and more people tend to hunt out and consume news from social media instead of traditional news organizations.The explanations for this alteration in consumption behaviours are inherent within the nature of those social media platforms: (i) it's often more timely and fewer expensive to consume news on social media compared with traditional journalism , like newspapers or television; and (ii) it's easier to further share, discuss , and discuss the news with friends or other readers on social media. For instance, 62 percent of U.S. adults get news on social media in 2016, while in 2012; only 49 percent reported seeing news on social media . It had been also found that social media now outperforms television because the major news source. Despite the benefits provided by social media, the standard of stories on social media is less than traditional news organisations. However, because it's inexpensive to supply news online and far faster and easier to propagate through social media, large volumes of faux news, i.e., those news articles with intentionally false information, are produced online for a spread of purposes, like financial and political gain. ithad been estimated that over 1 million tweets are associated with fake news “Pizzagate" by the top of the presidential election. Given the prevalence of this new phenomenon, “Fake news" was even named the word of the year by the Macquarie dictionary in 2016 . The extensive spread of faux news can have a significant negative impact on individuals and society. First, fake news can shatter the authenticity equilibrium of the news ecosystem for instance; it's evident that the most popular fake news was even more outspread on Facebook than the most accepted genuine mainstream news during the U.S. 2016 presidential election. Second, fake news intentionally persuades consumers to simply accept biased or false beliefs. Fake news is typically manipulated by propagandists to convey political messages or influence for instance, some report shows that Russia has created fake accounts and social bots to spread false stories. Third, fake news changes the way people interpret and answer real news, for instance, some fake news was just created to trigger people's distrust and make them confused; impeding their abilities to differentiate what's true from what's not. To assist mitigate the negative effects caused by fake news (both to profit the general public and therefore the news ecosystem). It's crucial that we build up methods to automatically detect fake news broadcast on social media.

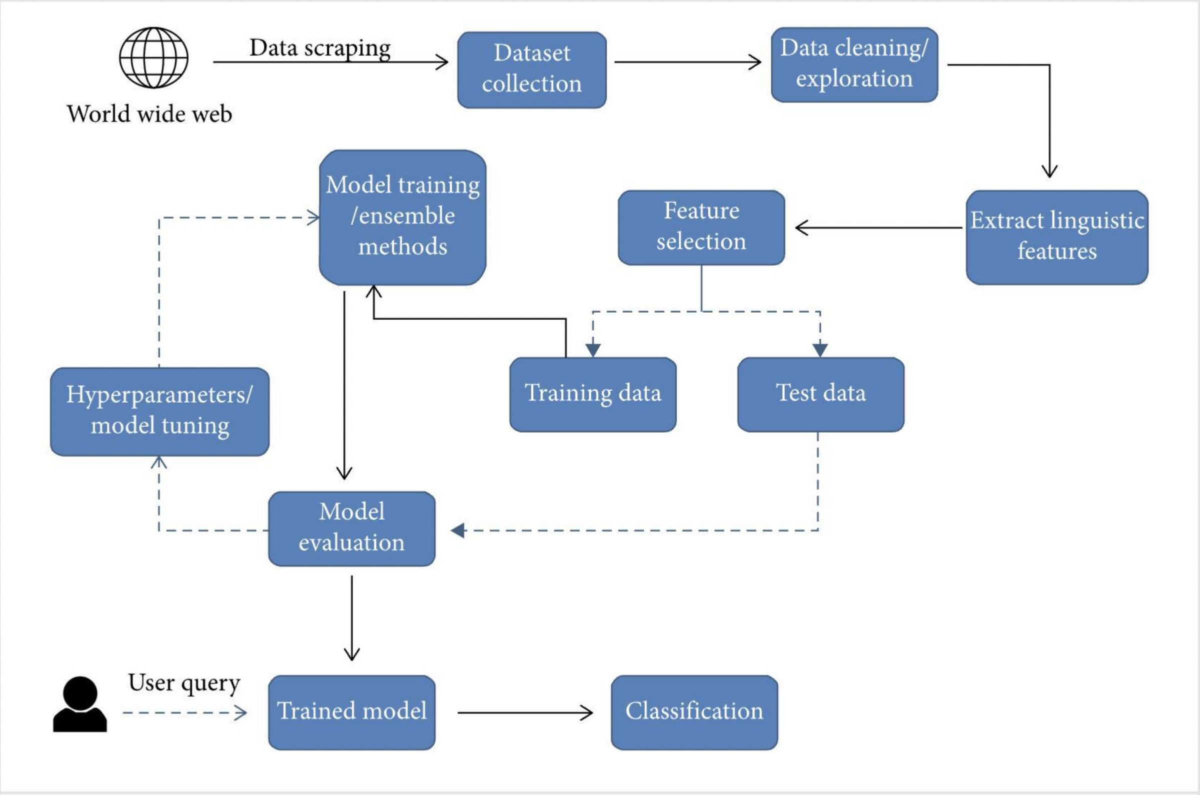
3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEMS:

In the current fake news corpus, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text. However, most of the literature focuses on specific datasets or domains, most prominently the politics domain. Therefore, the algorithm trained works best on a particular type of article's domain and does not achieve optimal results when exposed to articles from other domains. Since articles from different domains have a unique textual structure, it is difficult to train a generic algorithm that works best on all particular news domains.

3.2 PROPOSED SYSTEM:

In our proposed system, as illustrated in Figure 1, we are expanding on the current literature by introducing ensemble techniques with various linguistic feature sets to classify news articles from multiple domains as true or fake. The ensemble techniques along with Linguistic Inquiry and Word Count (LIWC) feature set used in this research are the novelty of our proposed approach.



4. SYSTEM REQUIREMENTS

SOFTWARE

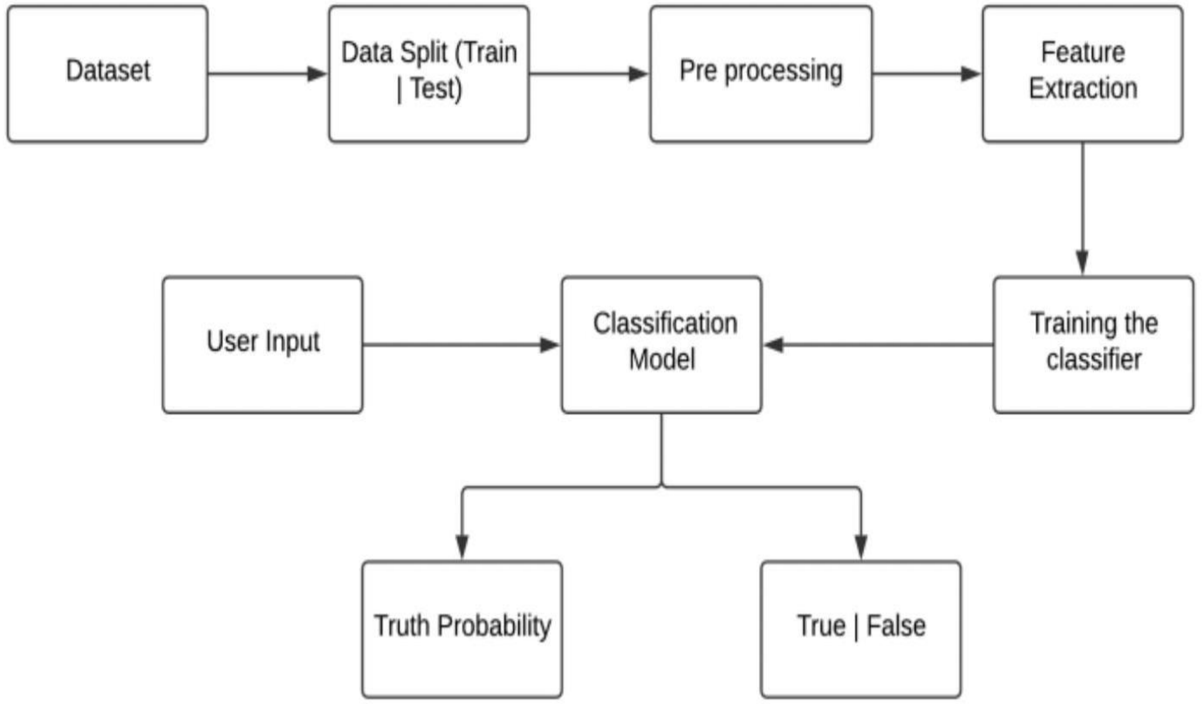
- PYTHON (>3.7)
- ANACONDA NAVIGATOR
- JUPITER NOTEBOOK

HARDWARE

- Windows 10
- Intel i3 processor
- RAM 4GB
- External 500GB

5. METHODOLOGY

5.1 System Architecture :



Logistic Regression

It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected). Mathematically, the log odds of the outcome are modelled as a linear combination of the predictor variables.

Odds = p/(1-p) = probability of event occurrence / probability of not event occurrence

ln(odds) = ln(p/(1-p))

logit(p)=ln(p/(1-p))= b0+b1X1+b2X2+b3X3....+bkXk

Passive Aggressive Classifier

The Passive Aggressive Algorithm is an online algorithm; ideal for classifying massive streams of data (e.g. twitter). It is easy to implement and very fast. It works by taking an example, learning from it and then throwing it away [24]. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

```
Classification Report :

              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       1.00      0.75      0.86         4
           2       0.88      1.00      0.93         7

    accuracy                           0.93        15
   macro avg       0.96      0.92      0.93        15
weighted avg       0.94      0.93      0.93        15
```

TF-IDF Vectorizer

TF (Term Frequency): The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

IDF (Inverse Document Frequency): Words that occur many times a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

$$idf(w) = log(\frac{N}{df_t})$$

Finally, Tfidf vectorizer

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Static Search Implementation

In static part, we have trained and used 3 out of 4 algorithms for classification. They are Naïve Bayes, Random Forest and Logistic Regression.

Step 1: In first step, we have extracted features from the already pre-processed dataset. These features are; Bag-of-words, Tf-Idf Features and N-grams.

Step 2: Here, we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, and Random forest classifiers from sklearn. Each of the extracted features was used in all of the classifiers.

Step 3: Once fitting the model, we compared the f1 score and checked the confusion matrix.

Step 4: After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification.

Step 5: We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.

Step 6: Finally selected model was used for fake news detection with the probability of truth.

Step 7: Our finally selected and best performing classifier was Logistic Regression which was then saved on disk. It will be used to classify the fake news.

It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth.

Dynamic Search Implementation

Our dynamic implementation contains 3 search fields which are

1) Search by article content.

2) Search using key terms.

3) Search for website in database.

In the first search field we have used Natural Language Processing for the first search field to come up with a proper solution for the problem, and hence we have attempted to create a model which can classify fake news according to the terms used in the newspaper articles. Our application uses NLP techniques like CountVectorization and TF-IDF Vectorization before passing it through a Passive Aggressive Classifier to output the authenticity as a percentage probability of an article.

The second search field of the site asks for specific keywords to be searched on the net upon which it provides a suitable output for the percentage probability of that term actually being present in an article or a similar article with those keyword references in it. The third search field of the site accepts a specific website domain name upon which the implementation looks for the site in our true sites database or the blacklisted sites database. The true sites database holds the domain names which regularly provide proper and authentic news and vice versa. If the site isn't found in either of the databases then the implementation doesn't classify the domain it simply states that the news aggregator does not exist.

Working—

The problem can be broken down into 3 statements

1) Use NLP to check the authenticity of a news article
.
2) If the user has a query about the authenticity of a search query then we he/she can directly search on our platform and using our custom algorithm we output a confidence score.

3) Check the authenticity of a news source.

These sections have been produced as search fields to take inputs in 3 different forms in our implementation of the problem statement.

Evaluation Matrices

Evaluate the performance of algorithms for fake news detection problem; various evaluation metrics have been used. In this subsection, we review the most widely used metrics for fake news detection. Most existing approaches consider the fake news problem as a classification problem that predicts whether a news article is fake or not:

True Positive (TP): when predicted fake news pieces are actually classified as fake news.

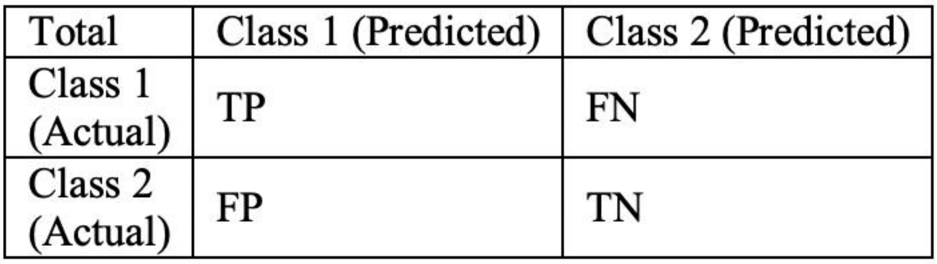
True Negative (TN): when predicted true news pieces are actually classified as true news.

False Negative (FN): when predicted true news pieces are actually classified as fake news.

False Positive (FP): when predicted fake news pieces are actually classified as true news.

Confusion Matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made .



Total	Class 1 (Predicted)	Class 2 (Predicted)
Class 1 (Actual)	TP	FN
Class 2 (Actual)	FP	TN

By formulating this as a classification problem, we can define following metrics-

6.TECHNOLOGIES STACK

PYTHON :

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

IN BUILT FUNCTIONS IN PYTHON:

abs(x): Return the absolute value of a number. The argument may be an integer, a floating point number, or an object implementing __abs__(). If the argument is a complex number, its magnitude is returned.

Aiter(async_iterable**) :** Return an asynchronous iterator for an asynchronous iterable. Equivalent to calling x.__aiter__().

all(*iterable*) **:** Return True if all elements of the *iterable* are true (or if the iterable is empty).

any(*iterable*) **:** Return True if any element of the *iterable* is true. If the iterable is empty, return False.

bin(*x*) : Convert an integer number to a binary string prefixed with “0b”. The result is a valid Python expression. If *x* is not a Python int object, it has to define an __index__() method that returns an integer.

breakpoint(**args*, ***kws*) : This function drops you into the debugger at the call site.

callable(*object*) : Return True if the *object* argument appears callable, False if not. If this returns True, it is still possible that a call fails, but if it is False, calling *object* will never succeed.

chr(*i*) : Return the string representing a character whose Unicode code point is the integer *i*.

dir([*object*]) : Without arguments, return the list of names in the current local scope. With an argument, attempt to return a list of valid attributes for that object.

divmod(*a*, *b*) : Take two (non-complex) numbers as arguments and return a pair of numbers consisting of their quotient and remainder when using integer division.

enumerate(*iterable*, *start=0*) : Return an enumerate object. *iterable* must be a sequence, an iterator, or some other object which supports iteration.

eval(*expression*[, *globals*[, *locals*]]) : The arguments are a string and optional globals and locals. If provided, *globals* must be a dictionary. If provided, *locals* can be any mapping object.

exec(*object*[, *globals*[, *locals*]]) : This function supports dynamic execution of Python code. *object* must be either a string or a code object.

filter(*function*, *iterable*) : Construct an iterator from those elements of *iterable* for which *function* returns true.

float([*x*]) : Return a floating point number constructed from a number or string *x*.

format(*value*[, *format_spec*]) : Convert a *value* to a "formatted" representation, as controlled by *format_spec*. The interpretation of *format_spec* will depend on the type of the *value* argument.

getattr(*object*, *name*[, *default*]) : Return the value of the named attribute of *object*. *name* must be a string. If the string is the name of one of the object's attributes, the result is the value of that attribute.

globals() : Return the dictionary implementing the current module namespace. For code within functions, this is set when the function is defined and remains the same regardless of where the function is called.

hasattr(*object*, *name*) : The arguments are an object and a string. The result is True if the string is the name of one of the object's attributes, False if not.

hash(*object*) : Return the hash value of the object (if it has one). Hash values are integers.

hex(*x*) : Convert an integer number to a lowercase hexadecimal string prefixed with "0x".

id(*object*) : Return the “identity” of an object. This is an integer which is guaranteed to be unique and constant for this object during its lifetime.

input([*prompt*]) : If the *prompt* argument is present, it is written to standard output without a trailing newline.

isinstance(*object*, *classinfo*) : Return True if the *object* argument is an instance of the *classinfo* argument.

issubclass(*class, classinfo*) : Return True if *class* is a subclass (direct, indirect, or virtual) of *classinfo*. A class is considered a subclass of itself.

iter(*object*[, *sentinel*]) : Return an iterator object. The first argument is interpreted very differently depending on the presence of the second argument.

len(*s*) : Return the length.

locals() : Update and return a dictionary representing the current local symbol table.

map(*function*, *iterable*, ...) : Return an iterator that applies *function* to every item of *iterable*, yielding the results.

max(*iterable*, *[, *key*, *default*]) : Return the largest item in an iterable or the largest of two or more arguments.

min(*iterable*, *[, *key*, *default*]) : Return the smallest item in an iterable or the smallest of two or more arguments.

next(*iterator*[, *default*]) : Retrieve the next item from the iterator by calling its **__**next**__**() method.

oct(*x*) : Convert an integer number to an octal string prefixed with “0o”.

open(*file, mode='r', buffering=1, encoding=None, errors=None, newline=None, closefd=True, opener=None*) : Open *file* and return a corresponding file object.

ord(*c*) : Given a string representing one Unicode character, return an integer representing the Unicode code point of that character.

pow(*base*, *exp*[, *mod*]) : Return *base* to the power *exp.*

print(**objects, sep=' ', end='\n', file=sys.stdout, flush=False***) :** Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

range(*start*, *stop*[, *step*]) : Rather than being a function, range is actually an immutable sequence type.

repr(*object*) : Return a string containing a printable representation of an object.

reversed(*seq*) : Return a reverse iterator.

round(*number*[, *ndigits*]) : Return *number* rounded to *ndigits* precision after the decimal point.

setattr(*object*, *name*, *value*) : This is the counterpart of getattr().

sorted(*iterable, /, *, key=None, reverse=False*) : Return a new sorted list from the items in *iterable*.

str(*object=''*) : Return a str version of *object*.

sum(*iterable, /, start=0*) : Sums *start* and the items of an *iterable* from left to right and returns the total. The *iterable*'s items are normally numbers, and the start value is not allowed to be a string.

super([*type*[, *object-or-type*]]) : Return a proxy object that delegates method calls to a parent or sibling class of *type*.

tuple([*iterable*]) : Rather than being a function, tuple is actually an immutable sequence type, as documented in Tuples .

type(*object*) : With one argument, return the type of an *object*. The return value is a type object and generally the same object.

vars([*object*]) : Return the __dict__ attribute for a module.

zip(**iterables, strict=False*) : Iterate over several iterables in parallel, producing tuples with an item from each one.

PYTHON INSTALLATION :

Python is a widely used high-level programming language. To write and execute code in python, we first need to install Python on our system.

Installing Python on Windows takes a series of few easy steps.

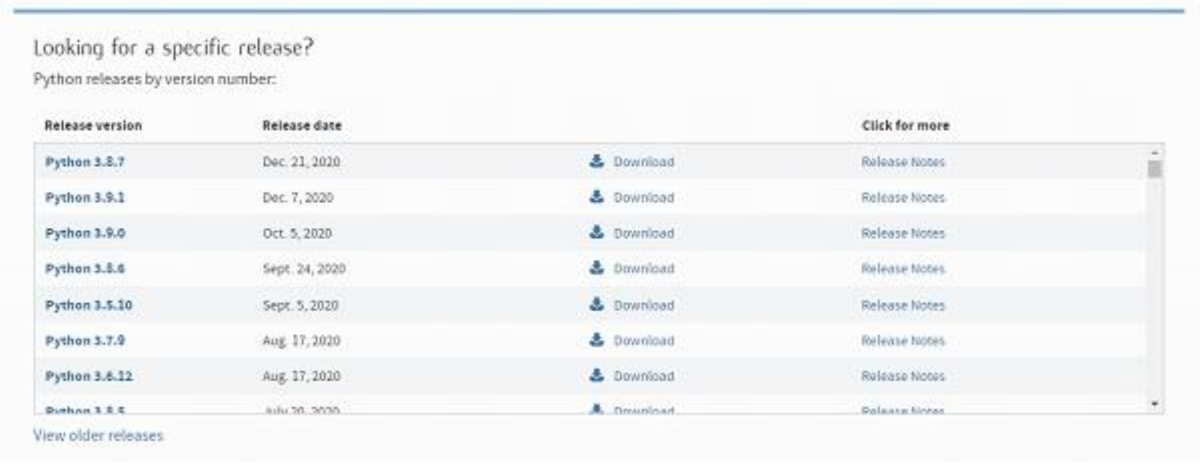
Step 1 − Select Version of Python to Install

Python has various versions available with differences between the syntax and working of different versions of the language. We need to choose the version which we want to use or need. There are different versions of Python 2 and Python 3 available.

Step 2 − Download Python Executable Installer

On the web browser, in the official site of python (www.python.org), move to the Download for Windows section.

All the available versions of Python will be listed. Select the version required by you and click on Download. Let suppose, we chose the Python 3.9.1 version.



Looking for a specific release?

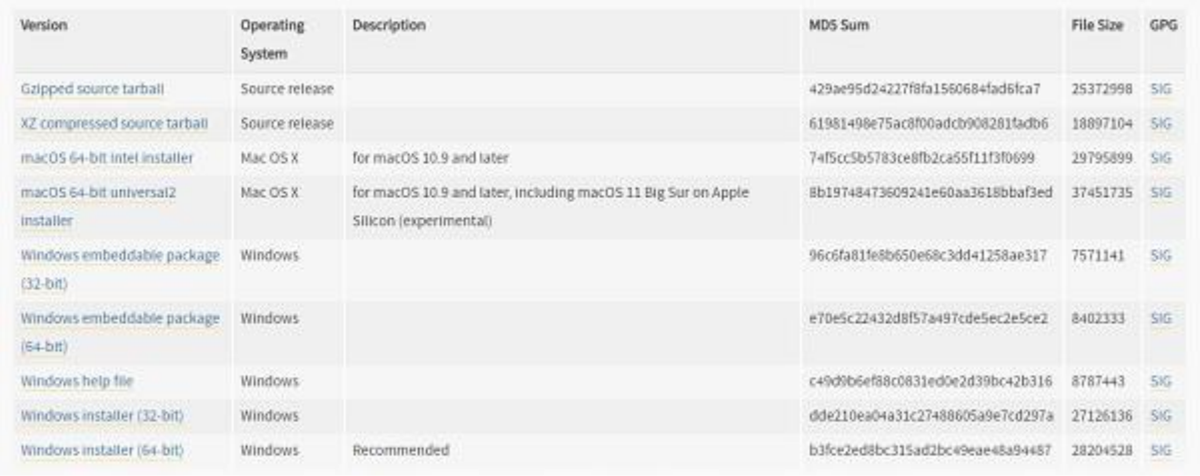
Python releases by version number:

Release version	Release date		Click for more
Python 3.8.7	Dec. 21, 2020	Download	Release Notes
Python 3.9.1	Dec. 7, 2020	Download	Release Notes
Python 3.9.0	Oct. 5, 2020	Download	Release Notes
Python 3.8.6	Sept. 24, 2020	Download	Release Notes
Python 3.5.10	Sept. 5, 2020	Download	Release Notes
Python 3.7.9	Aug. 17, 2020	Download	Release Notes
Python 3.6.12	Aug. 17, 2020	Download	Release Notes
Python 3.8.5	July 20, 2020	Download	Release Notes

View older releases

On clicking download, various available executable installers shall be visible with different operating system specifications. Choose the installer which suits your system operating system and download the instlaller. Let suppose, we select the Windows installer(64 bits).

The download size is less than 30MB.

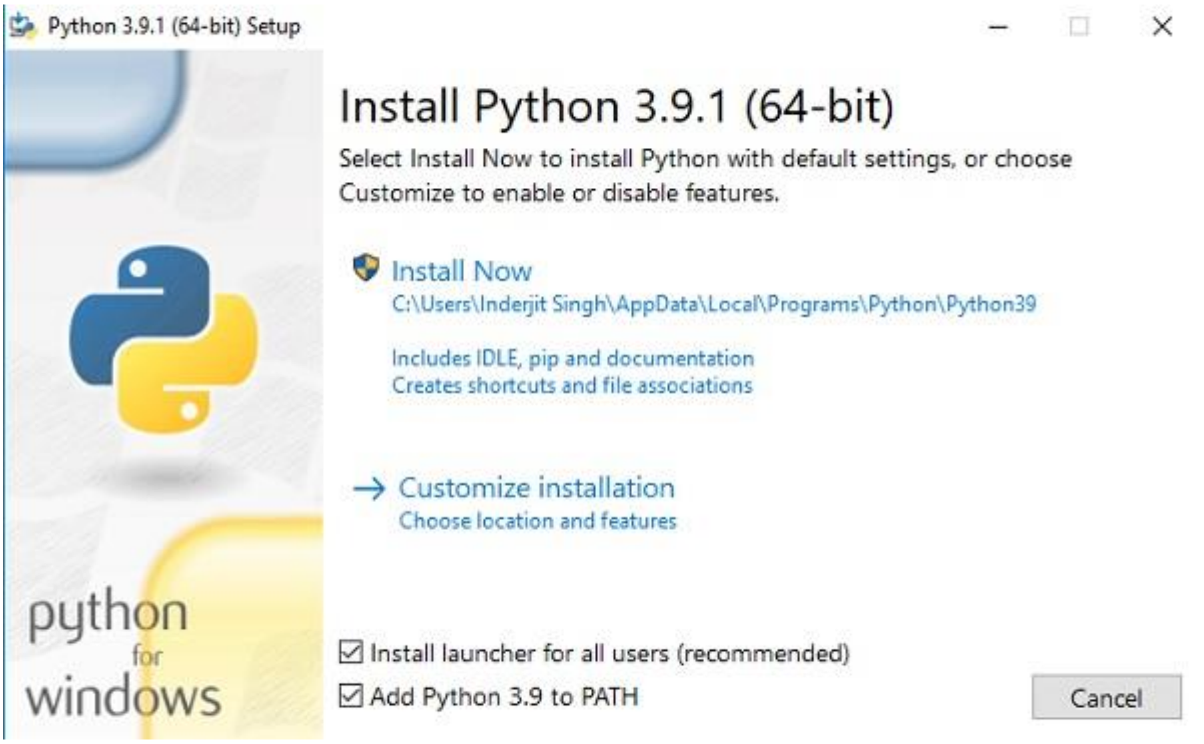


Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		429ae95d24227f8fa1560684fad6fca7	25372998	SIG
XZ compressed source tarball	Source release		61981498e75ac8f00adcb908281fadb6	18897104	SIG
macOS 64-bit intel installer	Mac OS X	for macOS 10.9 and later	74f5cc5b5783ce8fb2ca55f11f3f0699	29795899	SIG
macOS 64-bit universal2 installer	Mac OS X	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	8b19748473609241e60aa3618bbaf3ed	37451735	SIG
Windows embeddable package (32-bit)	Windows		96c6fa81fe8b650e68c3dd41258ae317	7571141	SIG
Windows embeddable package (64-bit)	Windows		e70e5c22432d8f57a497cde5ec2e5ce2	8402333	SIG
Windows help file	Windows		c49d9b6ef88c0831ed0e2d39bc42b316	8787443	SIG
Windows installer (32-bit)	Windows		dde210ea04a31c27488605a9e7cd297a	27126136	SIG
Windows installer (64-bit)	Windows	Recommended	b3fce2ed8bc315ad2bc49eae48a94487	28204528	SIG

Step 3 − Run Executable Installer

We downloaded the Python 3.9.1 Windows 64 bit installer.

Run the installer. Make sure to select both the checkboxes at the bottom and then click Install New.



Python 3.9.1 (64-bit) Setup

Install Python 3.9.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Inderjit Singh\AppData\Local\Programs\Python\Python39

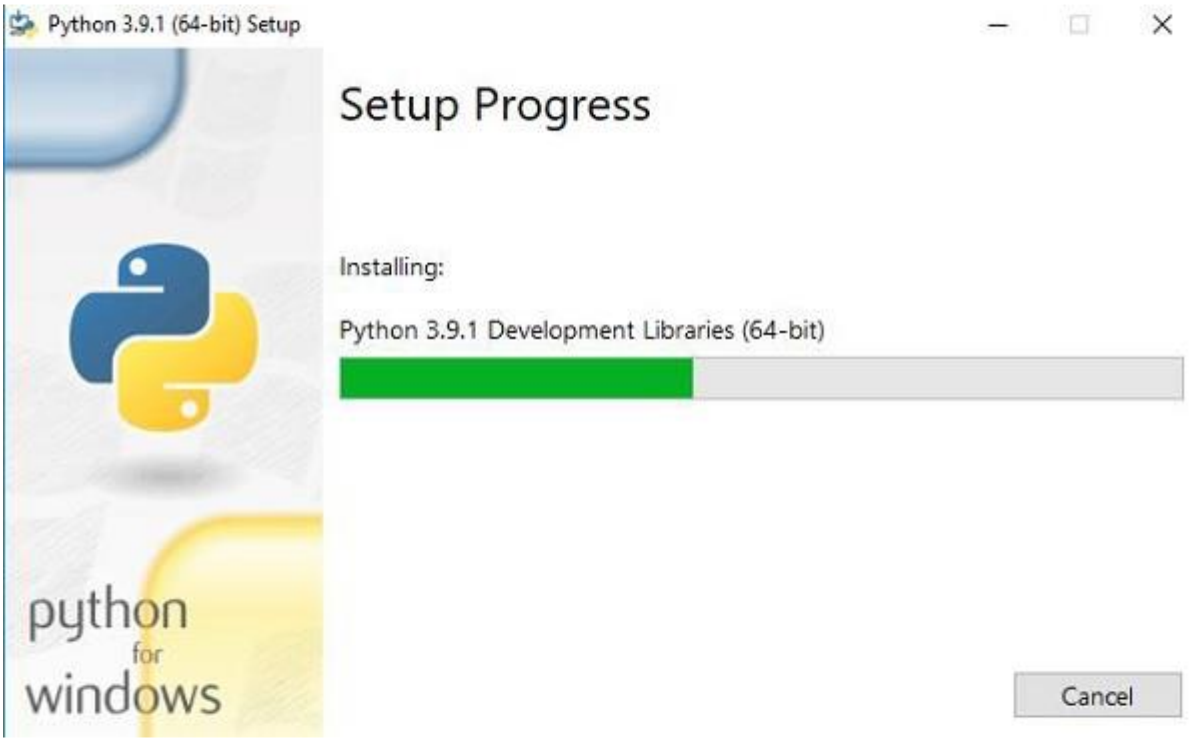
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.9 to PATH

On clicking the Install Now, The installation process starts.

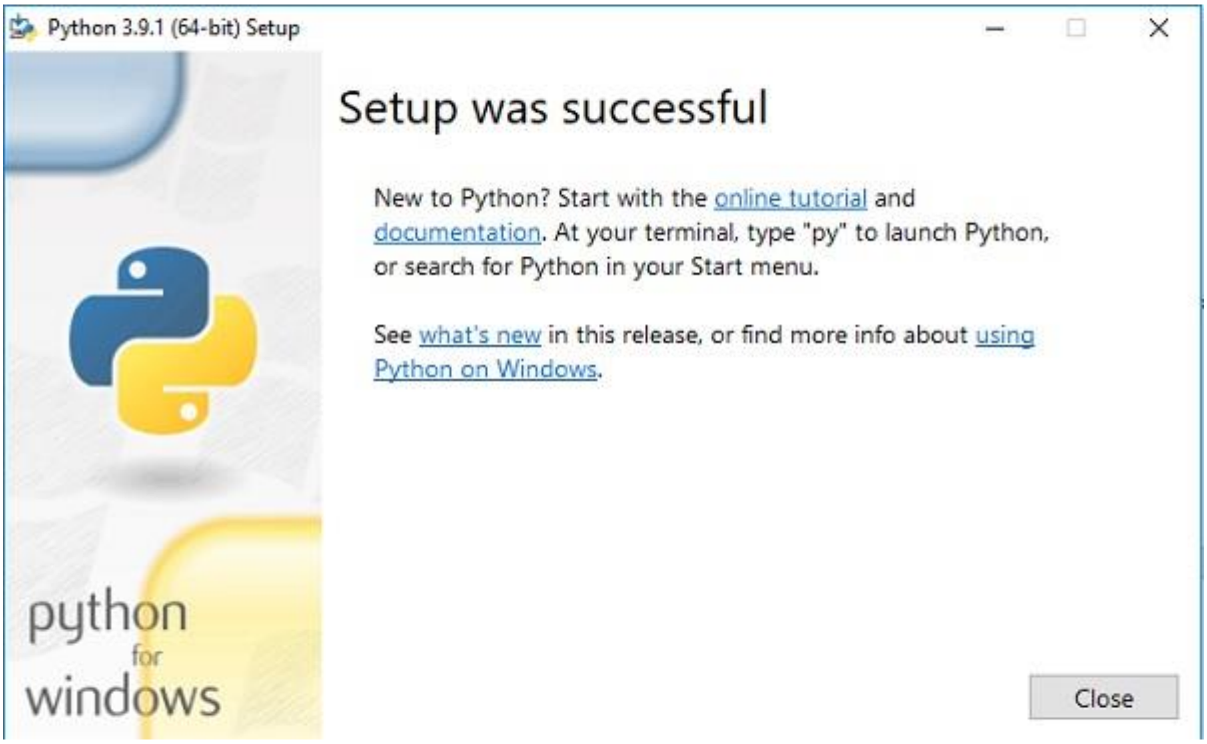


Setup Progress

Installing:

Python 3.9.1 Development Libraries (64-bit)

The installation process will take few minutes to complete and once the installation is successful, the following screen is displayed.



Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

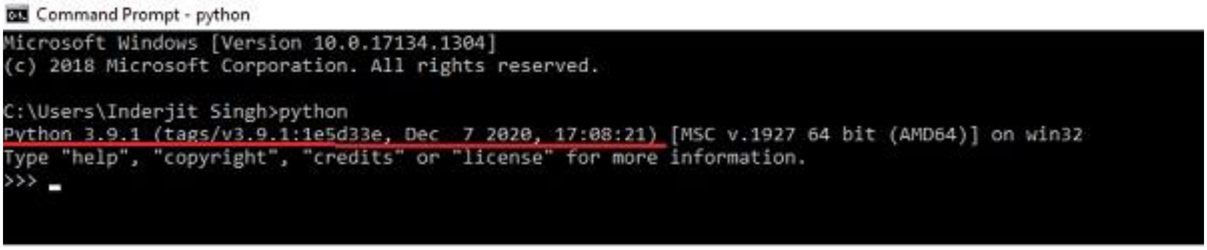
See what's new in this release, or find more info about using Python on Windows.

Step 4 − Verify Python is installed on Windows

To ensure if Python is succesfully installed on your system. Follow the given steps −

- Open the command prompt.
- Type ‘python’ and press enter.

- The version of the python which you have installed will be displayed if the python is successfully installed on your windows.



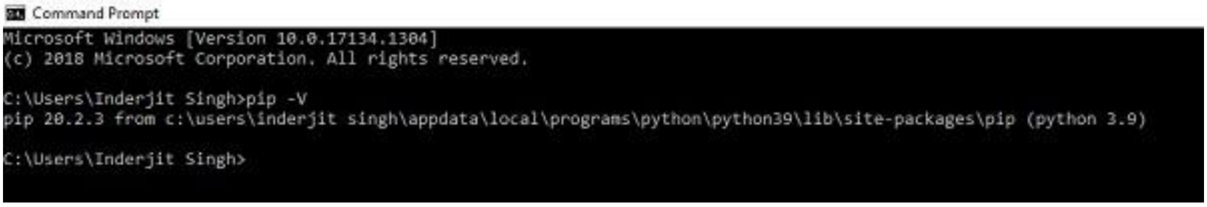
```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Inderjit Singh>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Step 5 − Verify Pip was installed

Pip is a powerful package management system for Python software packages. Thus, make sure that you have it installed.

To verify if pip was installed, follow the given steps −

- Open the command prompt.
- Enter pip –V to check if pip was installed.
- The following output appears if pip is installed successfully.



```
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Inderjit Singh>pip -V
pip 20.2.3 from c:\users\inderjit singh\appdata\local\programs\python\python39\lib\site-packages\pip (python 3.9)

C:\Users\Inderjit Singh>
```

We have successfully installed python and pip on our Windows system.

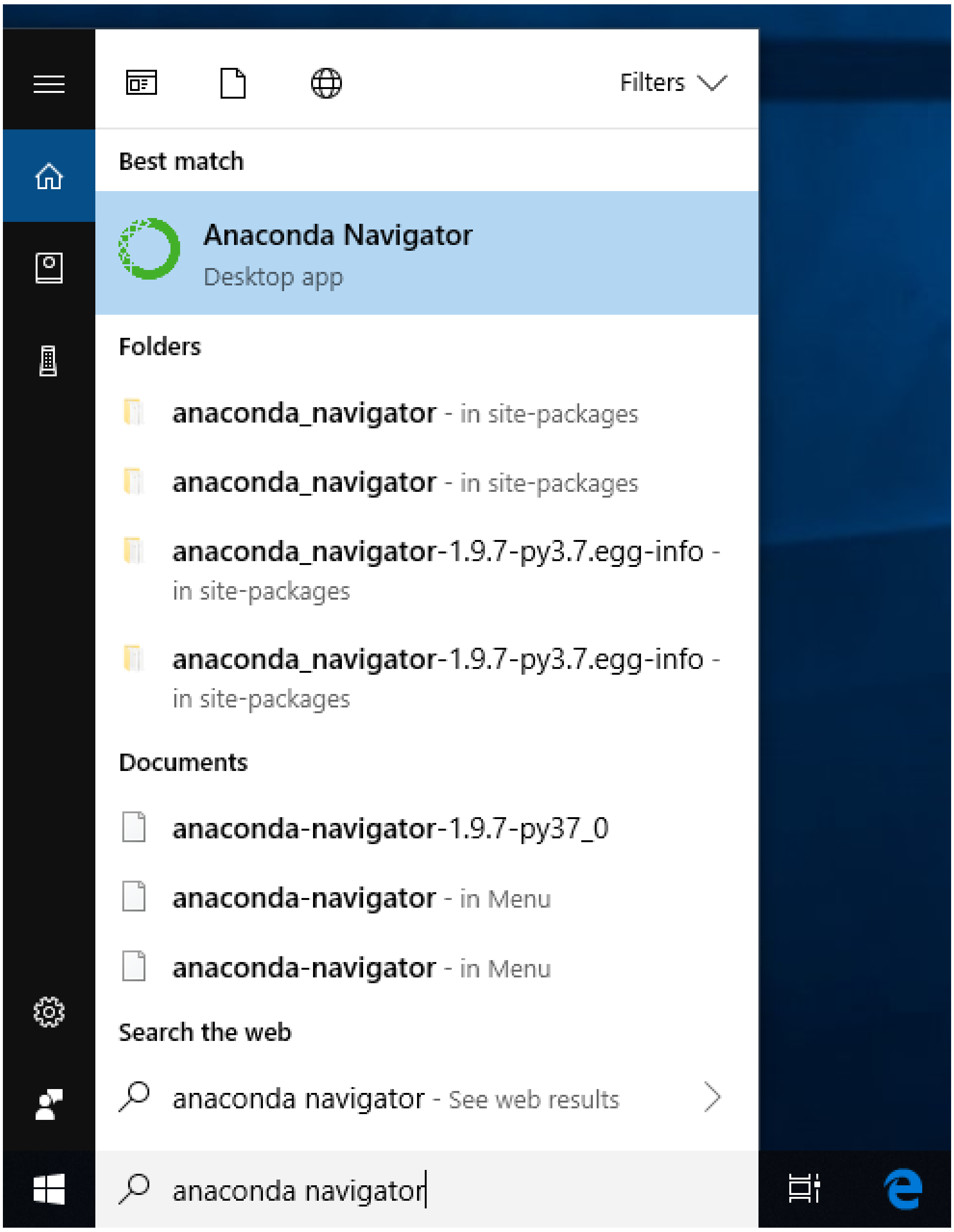
Anaconda Navigator Installation :

Anaconda Navigator is a graphical user interface to the conda package and environment manager.

Starting Navigator

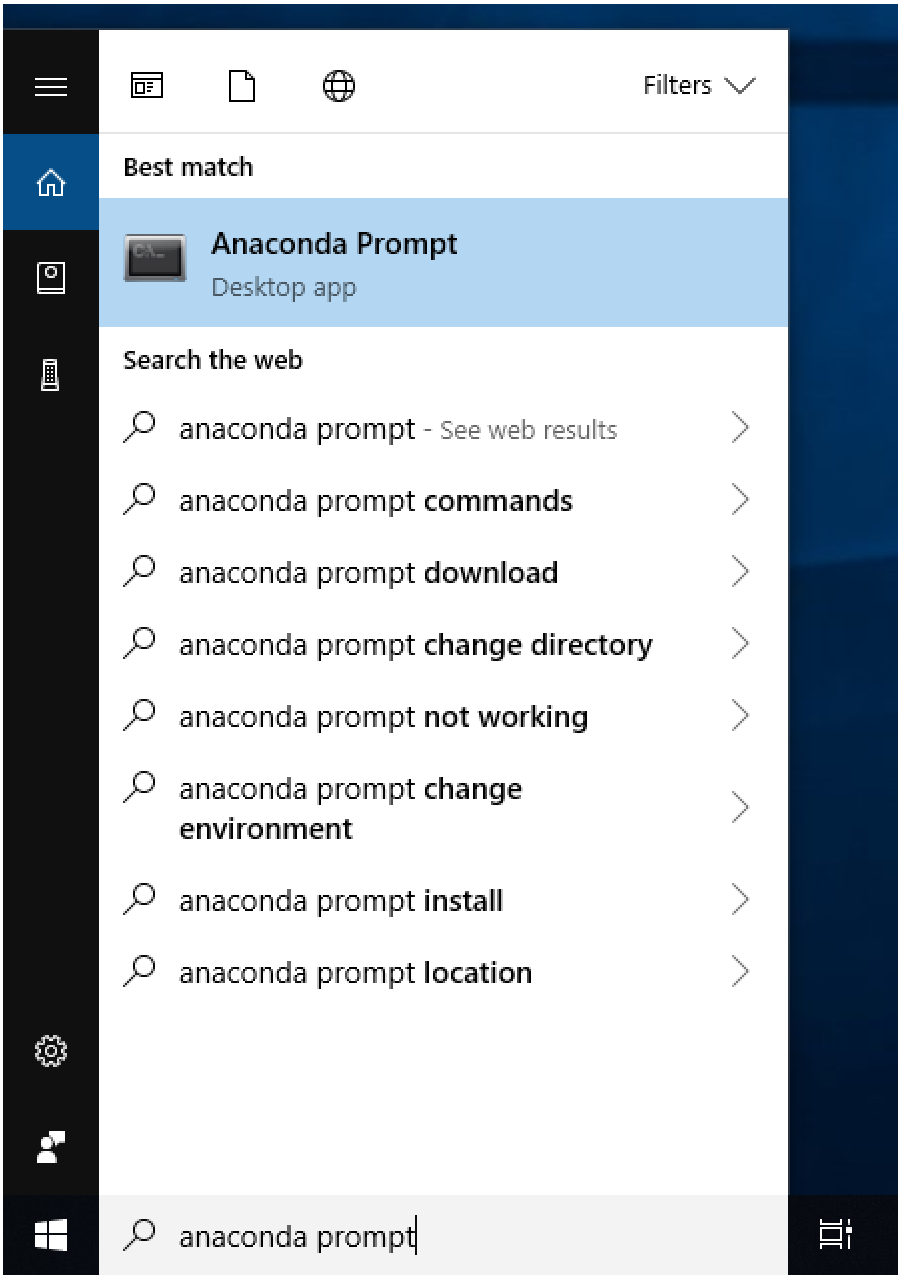
Windows

- From the Start menu, click the Anaconda Navigator desktop app.



Filters
Best match
Anaconda Navigator
Desktop app
Folders
anaconda_navigator - in site-packages
anaconda_navigator - in site-packages
anaconda_navigator-1.9.7-py3.7.egg-info - in site-packages
anaconda_navigator-1.9.7-py3.7.egg-info - in site-packages
Documents
anaconda-navigator-1.9.7-py37_0
anaconda-navigator - in Menu
anaconda-navigator - in Menu
Search the web
anaconda navigator - See web results
anaconda navigator

Or from the Start menu, search for and open “Anaconda Prompt” and type the command `anaconda-navigator`.



anaconda prompt

MacOS

- Open Launchpad, then click the Anaconda-Navigator icon.
- Or open Launchpad and click the terminal icon. Then in terminal, type `anaconda-navigator`.

Linux

- Open a terminal window and type `anaconda-navigator`.

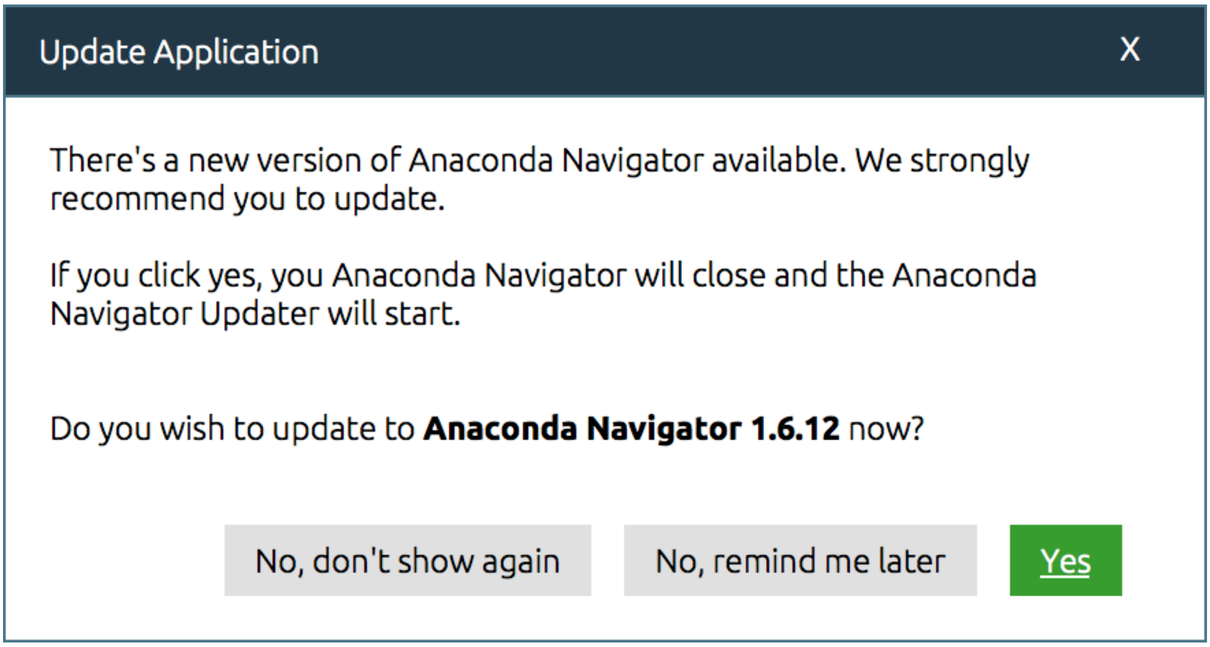
Managing Navigator

Verify that Anaconda is installed and running on your system.

- *When Navigator starts up, it verifies that Anaconda is installed.*
- *If Navigator does not start up, go back to Anaconda installation and make sure you followed all the steps.*

Check that Navigator is updated to the current version.

- *When you start Navigator, it automatically checks for a new version. If Navigator finds a new version, you will see a dialog box like this:*



Update Application

There's a new version of Anaconda Navigator available. We strongly recommend you to update.

If you click yes, you Anaconda Navigator will close and the Anaconda Navigator Updater will start.

Do you wish to update to **Anaconda Navigator 1.6.12** now?

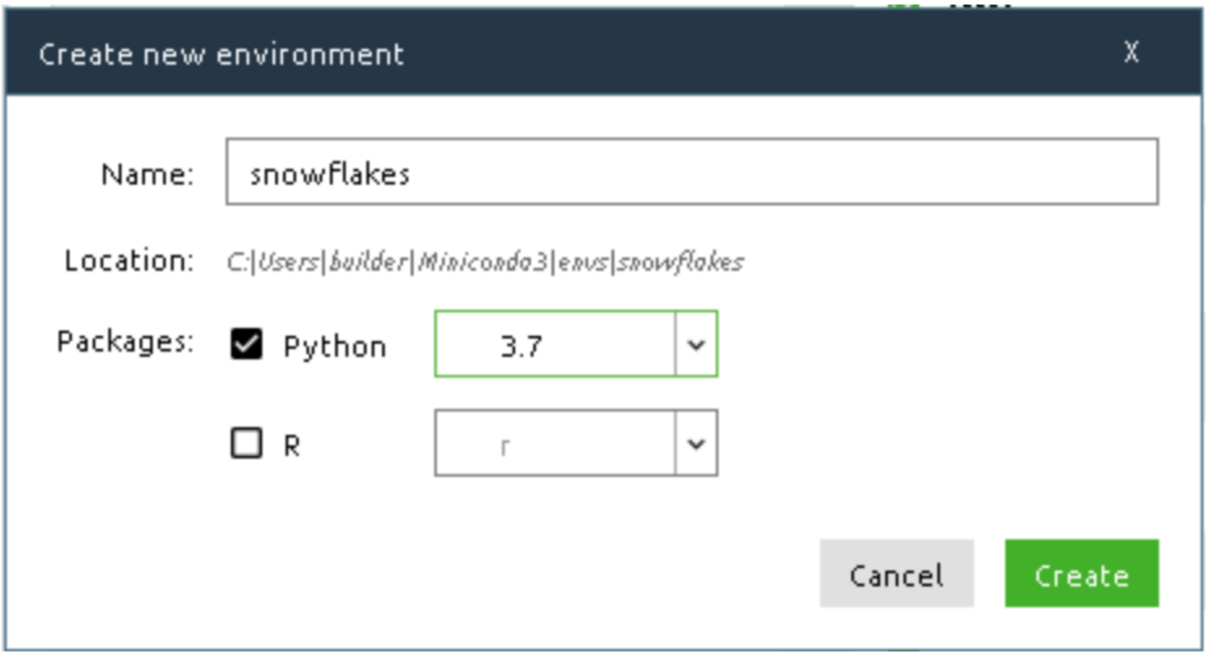
Click the "Yes" button to update Navigator to the current version.

Managing Environments

Navigator uses conda to create separate environments containing files, packages, and their dependencies that will not interact with other environments.

Create a new environment named snowflakes and install a package in it:

1. In Navigator, click the **Environments** tab, then click the Create button. The **Create new environment** dialog box appears.
2. In the **Environment** name field, type a descriptive name for your environment.



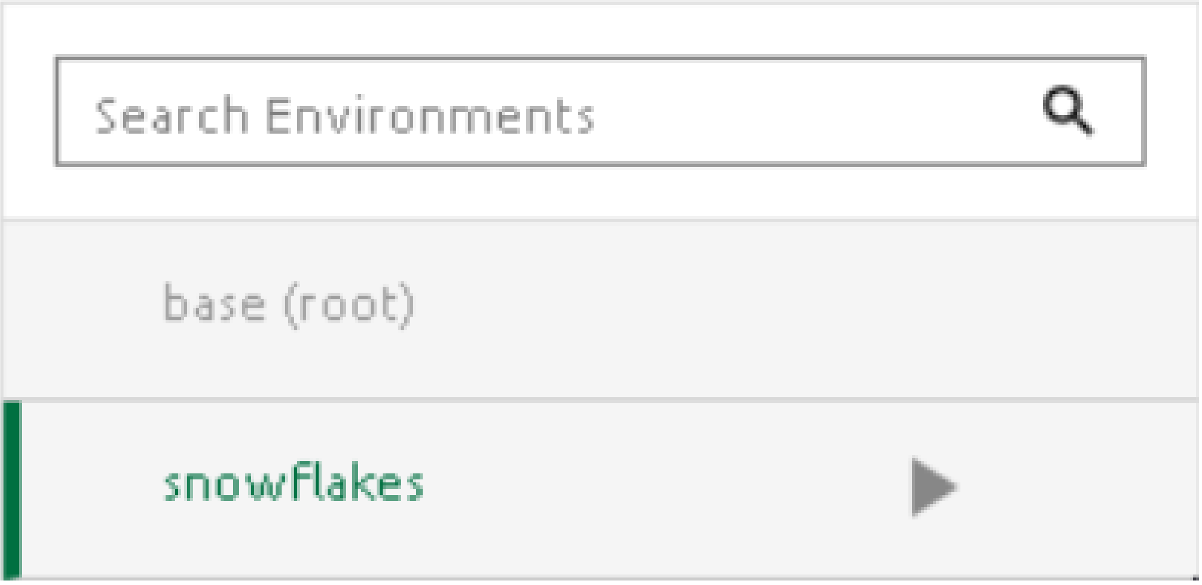
Name: snowflakes

Location: C:|Users|builder|Miniconda3|envs|snowflakes

Packages: ☑ Python 3.7

□ R r

1. Click **Create**. Navigator creates the new environment and activates it.



Now you have two environments, the default environment base (root), and snowflakes.

2. Switch between them (activate and deactivate environments) by clicking the name of the environment you want to use.
3. Return to the other environment by clicking its name.

Managing Python

When you create a new environment, Navigator installs the same Python version you used when you downloaded and installed Anaconda. If you want to use a different version of Python, for example Python 3.5, simply create a new environment and specify the version of Python that you want in that environment.

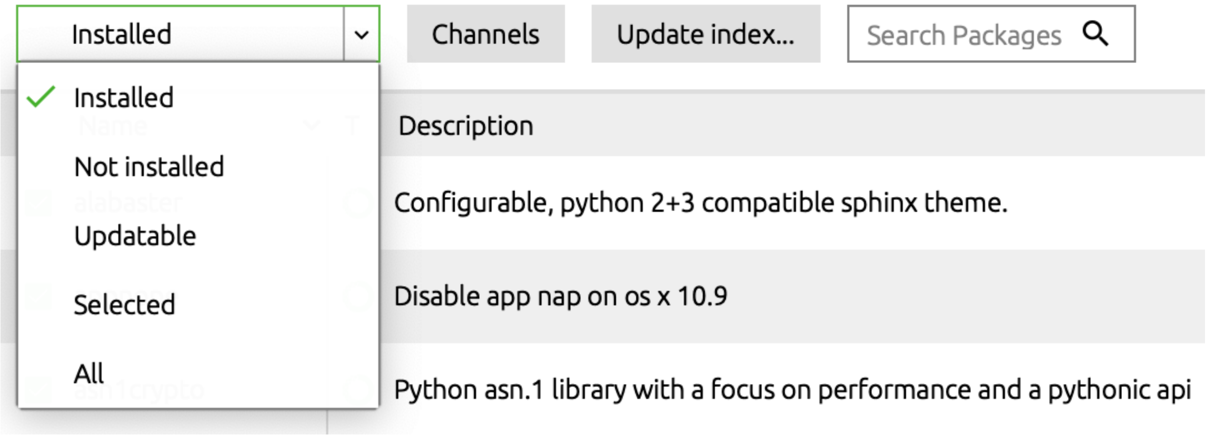
Create a new environment named “snakes” that contains Python 3.5:

1. In Navigator, click the **Environments** tab, then click the Create button. The Create new environment dialog box appears.
2. In the Environment name field, type the descriptive name “snakes” and select the version of Python you want to use from the Python Packages box (3.8, 3.7, 3.6, 3.5, or 2.7). Select a different version of Python than is in your other environments, base or snowflakes.
3. Click the Create button.
4. Activate the version of Python you want to use by clicking the name of that environment.

Managing packages

In this section, you check which packages you have installed, check which are available, and look for a specific package and install it.

1. To find a package you have already installed, click the name of the environment you want to search. The installed packages are displayed in the right pane.
2. You can change the selection of packages displayed in the right pane at any time by clicking the drop-down box above it and selecting Installed, Not Installed, Updatable, Selected, or All.

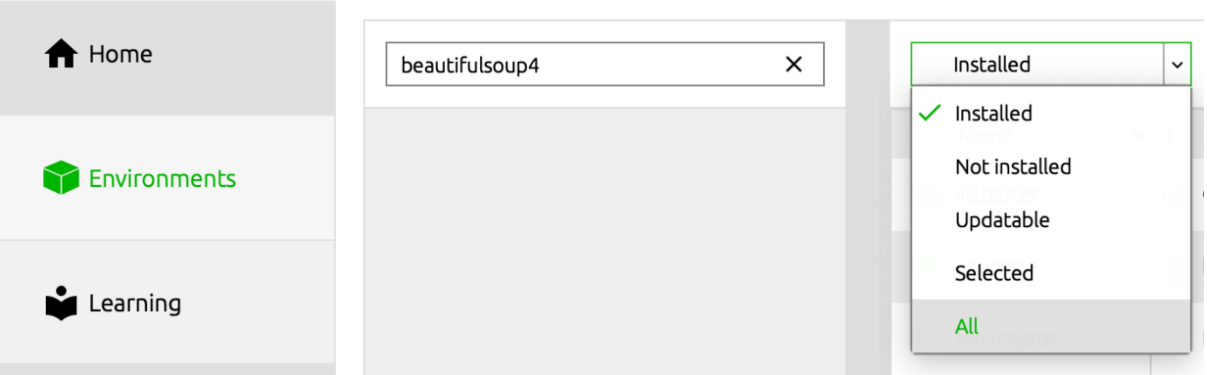


Configurable, python 2+3 compatible sphinx theme.

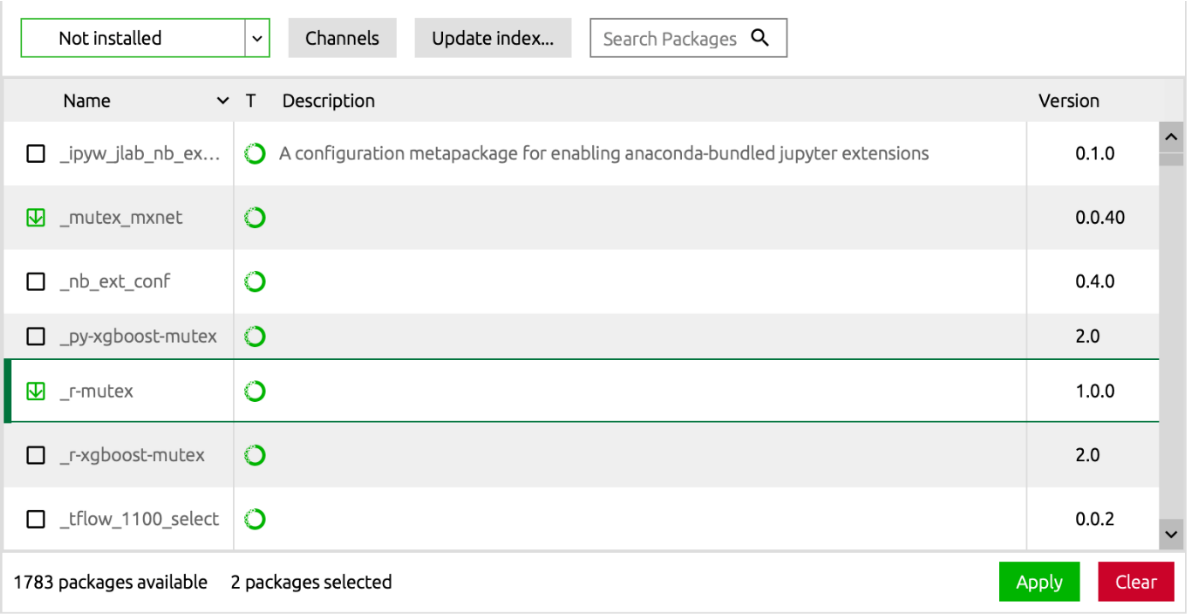
Disable app nap on os x 10.9

Python asn.1 library with a focus on performance and a pythonic api

3. Check to see if a package you have not installed named “beautifulsoup4” is available from the Anaconda repository (must be connected to the Internet). On the Environments tab, in the Search Packages box, type beautifulsoup4, and from the Search Subset box select All or Not Installed.



4. To install the package into the current environment, check the checkbox next to the package name, then click the bottom Apply button.



	Name	T	Description	Version
□	_ipyw_jlab_nb_ex…	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑	_mutex_mxnet	○		0.0.40
□	_nb_ext_conf	○		0.4.0
□	_py-xgboost-mutex	○		2.0
☑	_r-mutex	○		1.0.0
□	_r-xgboost-mutex	○		2.0
□	_tflow_1100_select	○		0.0.2

1783 packages available 2 packages selected

The newly installed program is displayed in your list of installed programs.

Installing the classic Jupyter Notebook interface

This section includes instructions on how to get started with **Jupyter Notebook**. But there are multiple Jupyter user interfaces one can use, based on their needs. Please checkout the list and links below for additional information and instructions about how to get started with each of them.

This information explains how to install the Jupyter Notebook and the IPython kernel.

Prerequisite: Python

While Jupyter runs code in many programming languages, **Python** is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook.

We recommend using the Anaconda distribution to install Python and Jupyter.

Installing Jupyter using Anaconda and conda

For new users, we **highly recommend** installing Anaconda. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

Use the following installation steps:

1. Download Anaconda. We recommend downloading Anaconda's latest Python 3 version (currently Python 3.9).
2. Install the version of Anaconda which you downloaded, following the instructions on the download page.
3. Congratulations, you have installed Jupyter Notebook. To run the notebook:
4. jupyter notebook

   See Running the Notebook for more details.

Alternative for Installing Jupyter with pip

Important

Jupyter installation requires Python 3.3 or greater, or Python 2.7. IPython 1.x, which included the parts that later became Jupyter, was the last version to support Python 3.2 and 2.6.

As an existing Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

First, ensure that you have the latest pip; older versions may have trouble with some dependencies:

```
pip3 install --upgrade pip
```

Then install the Jupyter Notebook using:

```
pip3 install jupyter
```

(Use pip if using legacy Python 2.)

Congratulations. You have installed Jupyter Notebook.

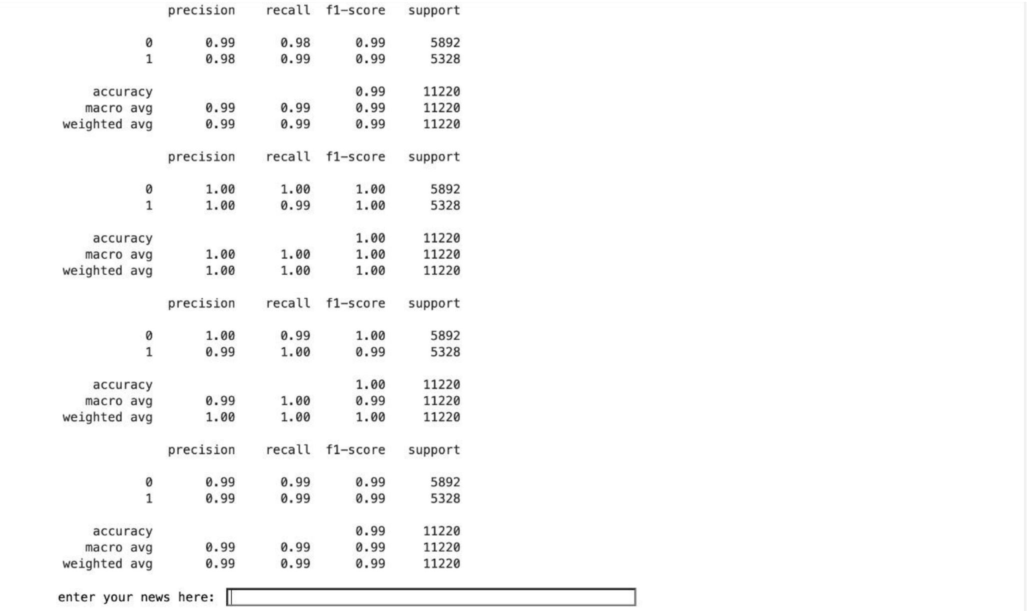
7.CODING

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string
df_fake = pd.read_csv("/Users/shyamala/Desktop/Fake_news/ Fake.csv")
df_true = pd.read_csv("/Users/shyamala/Desktop/Fake_news/ True.csv")
df_fake.head(5)
df_fake["class"] = 0
df_true["class"] = 1
df_fake.shape, df_true.shape
df_fake_manual_testing = df_fake.tail(10)
for i in range(23480,23470,-1):
df_fake.drop([i], axis = 0, inplace = True) df_true_manual_testing = df_true.tail(10) for i in range(21416,21406,-1):
df_true.drop([i], axis = 0, inplace = True)
df_fake.shape, df_true.shape
df_fake_manual_testing["class"] = 0
df_true_manual_testing["class"] = 1
df_fake_manual_testing.head(10)
df_true_manual_testing.head(10)
df_manual_testing = pd.concat([df_fake_manual_testing,df_true_manual_testing], axis = 0)
df_manual_testing.to_csv("manual_testing.csv")
df_marge = pd.concat([df_fake, df_true], axis =0 )
df_marge.head(10)
df_marge.columns
df = df_marge.drop(["title", "subject","date"], axis = 1)
df.isnull().sum()
```

```
df = df.sample(frac = 1)

df.head()

def wordopt(text):

text = text.lower()

text = re.sub('\[.*?\]', ", text)

text = re.sub("\\W"," ",text)

text = re.sub('https?://\S+|www\.\S+', ", text)

text = re.sub('<.*?>+', ", text)

text = re.sub('[%s]' % re.escape(string.punctuation), ", text)

text = re.sub('\n', ", text)

text = re.sub('\w*\d\w*', ", text)

return text

df["text"] = df["text"].apply(wordopt)

x = df["text"]

y = df["class"]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25) from
sklearn.feature_extraction.text import TfidfVectorizer vectorization = TfidfVectorizer()xv_train =
vectorization.fit_transform(x_train)

xv_test = vectorization.transform(x_test)

from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()

LR.fit(xv_train,y_train)

pred_lr=LR.predict(xv_test)

LR.score(xv_test, y_test)

print(classification_report(y_test, pred_lr))

from sklearn.tree import DecisionTreeClassifier

DT = DecisionTreeClassifier()

DT.fit(xv_train, y_train)

pred_dt = DT.predict(xv_test)

DT.score(xv_test, y_test)

print(classification_report(y_test, pred_dt))

from sklearn.ensemble import GradientBoostingClassifier GBC =
GradientBoostingClassifier(random_state=0) GBC.fit(xv_train, y_train)

pred_gbc = GBC.predict(xv_test)
```

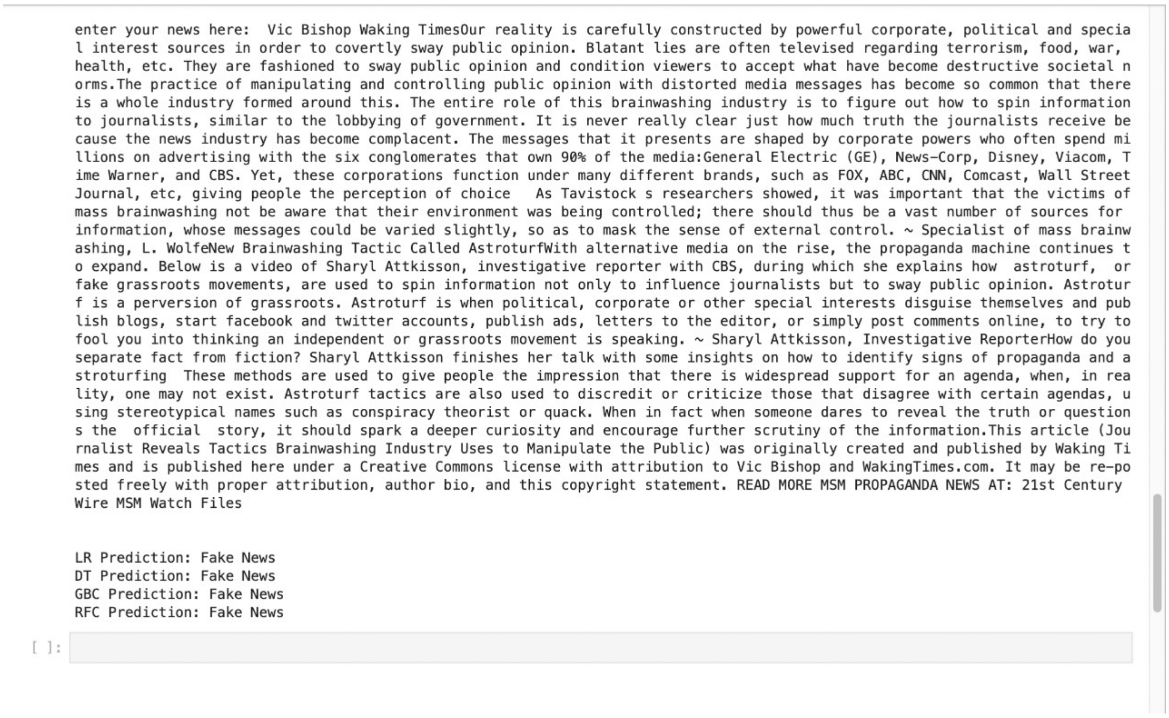
```
GBC.score(xv_test, y_test)

print(classification_report(y_test, pred_gbc))

from sklearn.ensemble import RandomForestClassifier RFC =
RandomForestClassifier(random_state=0) RFC.fit(xv_train, y_train)

pred_rfc = RFC.predict(xv_test)

RFC.score(xv_test, y_test)

print(classification_report(y_test, pred_rfc))

def output_lable(n):

if n == 0:

return "Fake News"

elif n == 1:

return "Not A Fake News"

def manual_testing(news):

testing_news = {"text":[news]}

new_def_test = pd.DataFrame(testing_news)

new_def_test["text"] = new_def_test["text"].apply(wordopt)

new_x_test = new_def_test["text"]

new_xv_test = vectorization.transform(new_x_test)

pred_LR = LR.predict(new_xv_test)

pred_DT = DT.predict(new_xv_test)

pred_GBC = GBC.predict(new_xv_test)

pred_RFC = RFC.predict(new_xv_test)

return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {}

\nRFC Prediction: {}".format(output_lable(pred_LR[0]),

output_lable(pred_DT[0]),

output_lable(pred_GBC[0]),

output_lable(pred_RFC[0])))

news = str(input("enter your news here: "))

manual_testing(news)
```

8.SCREENS:



```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      5892
           1       0.98      0.99      0.99      5328

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5892
           1       1.00      0.99      1.00      5328

    accuracy                           1.00     11220
   macro avg       1.00      1.00      1.00     11220
weighted avg       1.00      1.00      1.00     11220

              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5892
           1       0.99      1.00      0.99      5328

    accuracy                           1.00     11220
   macro avg       0.99      1.00      0.99     11220
weighted avg       1.00      1.00      1.00     11220

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5892
           1       0.99      0.99      0.99      5328

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220

enter your news here:
```

Fig : screen 1 (enter input..)



```
enter your news here:  Vic Bishop Waking TimesOur reality is carefully constructed by powerful corporate, political and specia
l interest sources in order to covertly sway public opinion. Blatant lies are often televised regarding terrorism, food, war,
health, etc. They are fashioned to sway public opinion and condition viewers to accept what have become destructive societal n
orms.The practice of manipulating and controlling public opinion with distorted media messages has become so common that there
is a whole industry formed around this. The entire role of this brainwashing industry is to figure out how to spin information
to journalists, similar to the lobbying of government. It is never really clear just how much truth the journalists receive be
cause the news industry has become complacent. The messages that it presents are shaped by corporate powers who often spend mi
llions on advertising with the six conglomerates that own 90% of the media:General Electric (GE), News-Corp, Disney, Viacom, T
ime Warner, and CBS. Yet, these corporations function under many different brands, such as FOX, ABC, CNN, Comcast, Wall Street
Journal, etc, giving people the perception of choice   As Tavistock s researchers showed, it was important that the victims of
mass brainwashing not be aware that their environment was being controlled; there should thus be a vast number of sources for
information, whose messages could be varied slightly, so as to mask the sense of external control. ~ Specialist of mass brainw
ashing, L. WolfeNew Brainwashing Tactic Called AstroturfWith alternative media on the rise, the propaganda machine continues t
o expand. Below is a video of Sharyl Attkisson, investigative reporter with CBS, during which she explains how  astroturf,  or
fake grassroots movements, are used to spin information not only to influence journalists but to sway public opinion. Astrotur
f is a perversion of grassroots. Astroturf is when political, corporate or other special interests disguise themselves and pub
lish blogs, start facebook and twitter accounts, publish ads, letters to the editor, or simply post comments online, to try to
fool you into thinking an independent or grassroots movement is speaking. ~ Sharyl Attkisson, Investigative ReporterHow do you
separate fact from fiction? Sharyl Attkisson finishes her talk with some insights on how to identify signs of propaganda and a
stroturfing  These methods are used to give people the impression that there is widespread support for an agenda, when, in rea
lity, one may not exist. Astroturf tactics are also used to discredit or criticize those that disagree with certain agendas, u
sing stereotypical names such as conspiracy theorist or quack. When in fact when someone dares to reveal the truth or question
s the  official  story, it should spark a deeper curiosity and encourage further scrutiny of the information.This article (Jou
rnalist Reveals Tactics Brainwashing Industry Uses to Manipulate the Public) was originally created and published by Waking Ti
mes and is published here under a Creative Commons license with attribution to Vic Bishop and WakingTimes.com. It may be re-po
sted freely with proper attribution, author bio, and this copyright statement. READ MORE MSM PROPAGANDA NEWS AT: 21st Century
Wire MSM Watch Files


LR Prediction: Fake News
DT Prediction: Fake News
GBC Prediction: Fake News
RFC Prediction: Fake News
```

Fig : screen 2 (output for input)

9.CONCLUSION:

In the 21st century, the majority of the tasks are done online. Newspapers that were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. Whatsapp’s forwards are also a major source. The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology. When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed. Thus, in order to curb the phenomenon, we have developed our Fake news Detection system that takes input from the user and classify it to be true or fake. To implement this, various NLP and Machine Learning Techniques have to be used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. The best model, i.e. the model with highest accuracy is used to classify the news headlines or articles. As evident above for static search, our best model came out to be Logistic Regression with an accuracy of 65%. Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 75%. Hence we can say that if a user feed a particular news article or its headline in our model, there are 75% chances that it will be classified to its true nature. The user can check the news article or keywords online; he can also check the authenticity of the website. The accuracy for dynamic system is 93% and it increases with every iteration.

We intend to build our own dataset which will be kept up to date according to the latest news. All the live news and latest data will be kept in a database using Web Crawler and online database.

10.REFERENCES

[1] S. B. Parikh and P. K. Atrey, "Media-Rich Fake News Detection: A Survey", 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 436-441, 2018, April.

[2] N. J. Conroy, V. L. Rubin and Y. Chen, "Automatic deception detection: Methods for finding fake news", Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, pp. 82, 2015, November.

[3] S. Helmstetter and H. Paulheim, "Weakly supervised learning for fake news detection on Twitter", 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 274-277, 2018, August.

[4] W. Y. Wang, "liar liar pants on fire", A new benchmark dataset for fake news detection., 2017.

[5] K. Stahl, Fake News Detection in Social Media, 2018.

[6] M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro and L. de Alfaro, "Automatic Online Fake News Detection Combining Content and Social Signals", 2018 22nd Conference of Open Innovations Association (FRUCT), pp. 272-279, 2018, May.

[7] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret and L. de Alfaro, Some like it hoax: Automated fake news detection in social networks, 2017.