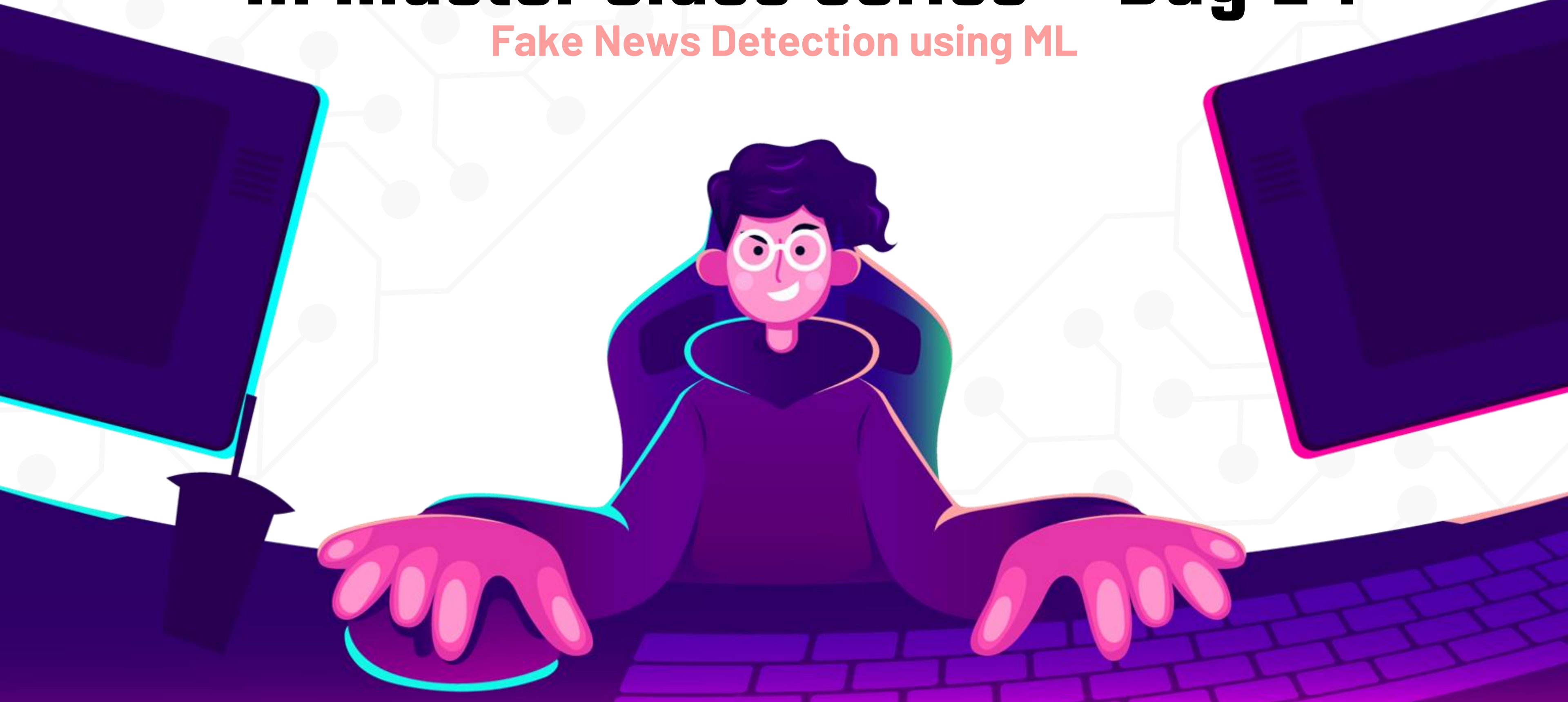
PANTECH SOLUTIONS®
Technology Beyond the Dreams

AI Master Class series – Day 24

Fake News Detection using ML



Batch 1 & 2

Don't Forget to attend Today's Event Technical Hackathon – Ad.On Watermarking 6:00 PM

Batch 3 Started

Day-24 Agenda.

01.

Confusion Matrix

Confusion Matrix & Accuracy Calculation

02.

Text Feature Extraction

Various Text Feature Extraction techniques & its Syntax

03.

ML Algorithm

New ML algorithms & Basic Syntax

04.

Deploying ML application

Fake News detection using Ml algorithm

05.

Q & A

SEND VIDEO TESTIMONIAL

Participants who sends Video Testimonial, Gonna have Live Video call Interaction session with Me.

Send ur Video Testimonial to the +91 9003113840

Link in Description

Video Testimonial – Video Feedback

Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Ex:
COVID – 19 - Binary prediction (Yes / No)
Total no. of Patient : 165
Real Data:
Covid Yes – 105 Patient
No Covid – 60 Patient

Our ML Predicted Data:
Covid Yes – 110 Patient
No Covid – 55 Patient

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Confusion Matrix

True positives (TP): These are cases in which we predicted yes (they have the covid), and they have the covid.

True negatives (TN): We predicted no, and they don't have the covid.

False positives (FP): We predicted yes, but they don't actually have the covid. (Also known as a "Type I error.")

False negatives (FN): We predicted no, but they actually have the covid. (Also known as a "Type II error.")

n=165	Predicted: NO	Predicted: YES
Actual: NO	TN = 50	FP = 10
Actual: YES	FN = 5	TP = 100

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Accuracy & Loss Calculation from Confusion Matrix

Accuracy: Overall correct.

(TP+TN)/total = (100+50)/165 = 0.91

Misclassification Rate: Overall wrong | Error Rate

(FP+FN)/total = (10+5)/165 = 0.09

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Other rates from Confusion Matrix

True Positive Rate: Sensitivity or Recall

TP/actual yes = 100/105 = 0.95

False Positive Rate:

FP/actual no = 10/60 = 0.17

True Negative Rate: Specificity

TN/actual no = 50/60 = 0.83

Precision:

TP/predicted yes = 100/110 = 0.91

Prevalence :

actual yes/total = 105/165 = 0.64

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Feature extraction in Text – Methods

build_analyzer() – Return a callable that handles preprocessing, tokenization and n-grams generation.

build_preprocessor() – Return a function to preprocess the text before tokenization.

build_tokenizer() – Return a function that splits a string into a sequence of tokens.

decode(doc) – Decode the input into a string of unicode symbols.

fit(raw_documents[, y]) – Learn a vocabulary dictionary of all tokens in the raw documents.

fit_transform(raw_documents[, y]) – Learn the vocabulary dictionary and return document-term matrix.

get_feature_names() – Array mapping from feature integer indices to feature name.

get_params([deep]) – Get parameters for this estimator.

get_stop_words() – Build or fetch the effective stop words list.

inverse_transform(X) – Return terms per document with nonzero entries in X.

set_params(**params) – Set the parameters of this estimator.

transform(raw_documents) – Transform documents to document-term matrix.

CountVectorizer – Types of feature extraction in Text

Convert a collection of text documents to a matrix of token counts

HashingVectorizer – Types of feature extraction in Text

Convert a collection of text documents to a matrix of token occurrences

TfidfVectorizer – Types of Feature extraction in Text

- Term-frequency times inverse document-frequency.
- Convert a collection of raw documents to a matrix of TF-IDF features.
- Equivalent to CountVectorizer followed by TfidfTransformer.

TfidfVectorizer. get_feature_names

```
DATA= [
    'This is the first document.',
    'This document is the second document.',
    'And this is the third one.',
    'Is this the first document?']
X = vectorizer.fit_transform(DATA)
TfidfVectorizer. get_feature_names
```

['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third', 'this']

naive_bayes – MultinomialNB – ML Algorithm

- Naive Bayes are mostly used in natural language processing (NLP) problems. Naive Bayes predict the tag of a text.

- They calculate the probability of each tag for a given text and then output the tag with the highest one.

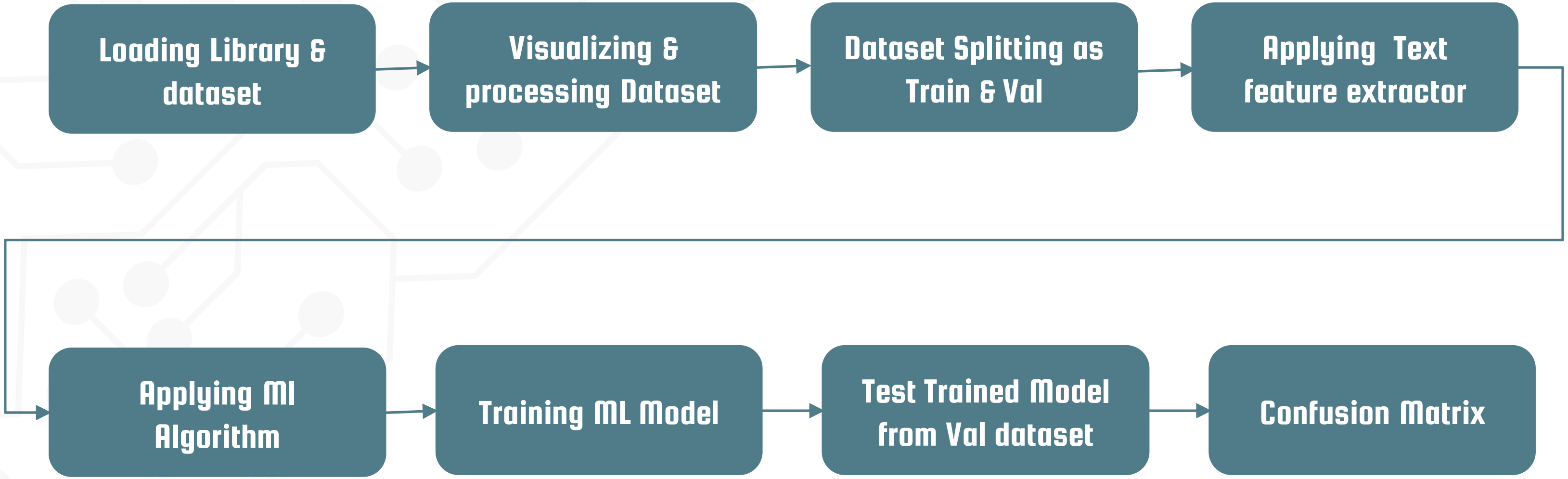
- Bayes theorem calculates probability P(c|x) where c is the class of the possible outcomes and x is the given instance which has to be classified

Removing Stopwords: These are common words that don't really add anything to the classification, such as an able, either, else, ever and so on.

Stemming: Stemming to take out the root of the word.

"ilikedthemovi"	positive
"itsagoodmovienicestori"	positive
"nicesongsbutsadlyboringend"	negative

Workflow of Fake News Detection.



ML Syntax

DataFrame

A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

```
df = pd.DataFrame({'month': [1, 4, 7, 10],
                   'year': [2012, 2014, 2013, 2014],
                   'sale': [55, 40, 84, 31]})
```

	month	year	sale
0	1	2012	55
1	4	2014	40
2	7	2013	84
3	10	2014	31

ML Syntax

set_index

Set the DataFrame index using existing columns.

df.set_index('month')

```
      year  sale
month
1     2012    55
4     2014    40
7     2013    84
10    2014    31
```

ML Syntax

drop('label', axis=1)

Drop specified labels from rows or columns.

df.drop('month‘,axis=1)   or df.drop(columns='month’)

year	sale
2012	55
2014	40
2013	84
2014	31

ML Syntax

Difference – Set

```
A = {10, 20, 30, 40, 80}
B = {100, 30, 80, 40, 60}
print (A.difference(B))
print (B.difference(A))
```

```
{10, 20}
{100, 60}
```

! PRACTICAL SESSION !

Fake News Detection using ML Algorithm

SEND VIDEO TESTIMONIAL

Participants who sends Video Testimonial, Gonna have Live Video call Interaction session with Me.

Send ur Video Testimonial to the +91 9003113840

Link in Description

Video Testimonial – Video Feedback

3 Combo Course

BUY 3 COURSE at Rs.999 Only

Data Analytics + Artificial Intelligence + Machine Learning

Batch-3 Last Batch started

- ✓ Access to all 90 Days Video Lectures (Value ₹12,000)
- ✓ 50+ Projects - From Scratch (Value ₹75,000)
- ✓ 100+ Source Code (Value ₹10,000)
- ✓ Technical Materials (PPT & Mindmap) (Value ₹10,000)
- ✓ Bonus Task, Assignment & Mindset Lectures (Value ₹5,000)
- ✓ Forum Telegram discussion & Support (Value ₹2,000)
- ✓ 3 - Internship E-Certificate on Artificial Intelligence, Machine Learning and Data Analytics

~~Total Value Rs 1,14,000~~

Today Just Rs. 999

Q & A session

Thanks!

Connect with me on **LinkedIn:**
link in Description

Product & Project:
www.pantechsolutions.net

Course:
Learn.pantechsolutions.net

Tomorrow session

AI Snake Game