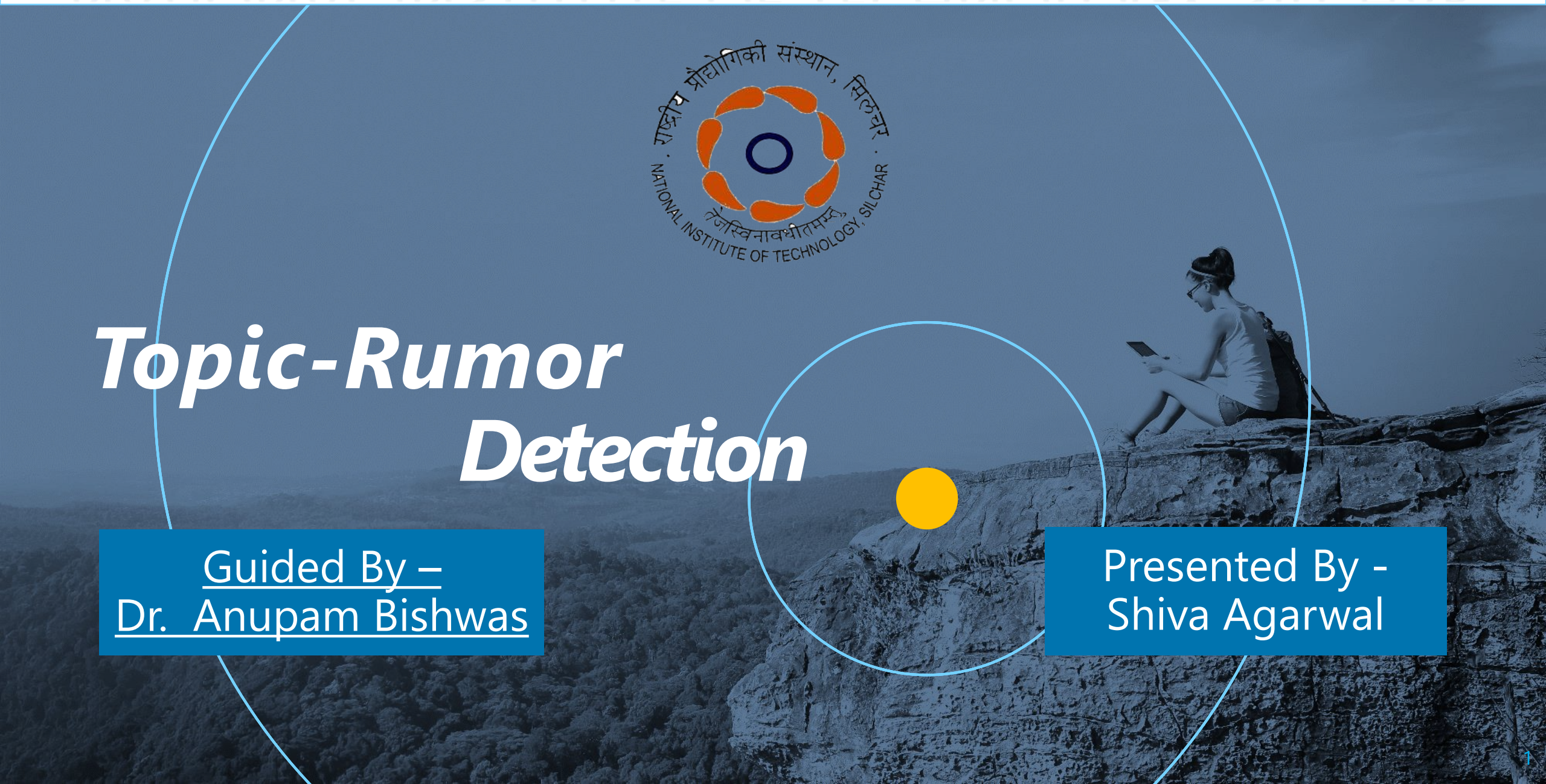
NATIONAL INSTITUTE OF TECHNOLOGY, SILCHAR



Topic-Rumor Detection

Guided By –
Dr. Anupam Bishwas

Presented By -
Shiva Agarwal

What are rumors ???

“ **Rumor** is a piece of news or information that many people are talking about but that is possibly not true”

“A tall tale of explanations of events circulating from person to person and pertaining to an object, event, or issue in public concern."

CONTENTS

Problem Overview

Automatic rumor detection is technically very challenging. In this work, we try to learn discriminative features from tweets content by following their non-sequential propagation structure and generate more powerful representations for identifying different type of rumors.

Different methods and prediction

Logistic Regression
Decision Tree Classification
Gradient Boosting Classifier
Random Forest Classifier

Drawback and improvement

LSTM- It is special kind of recurrent neural network that is capable of learning long term dependencies in data. This is achieved because the recurring module of the **model** has a combination of four layers interacting with each other.

What new in it?

Why Fake News Detection is Tough ?

HUMAN ARE ALSO CONFUED?

Fake News also becomes more challenging as social media continues to dominate our everyday lives and hence accelerating the speed of which Fake News travel.

In a recent study published by the journal Science, it analysed millions of tweets sent between 2006 and 2017 and it was found that: “Falsehood diffused significantly farther, faster, deeper, and more broadly than the truth in all categories of information.”

It also concluded that “it took the truth about six times as long as falsehood to reach 1,500 people.”

Different Type of Fake News

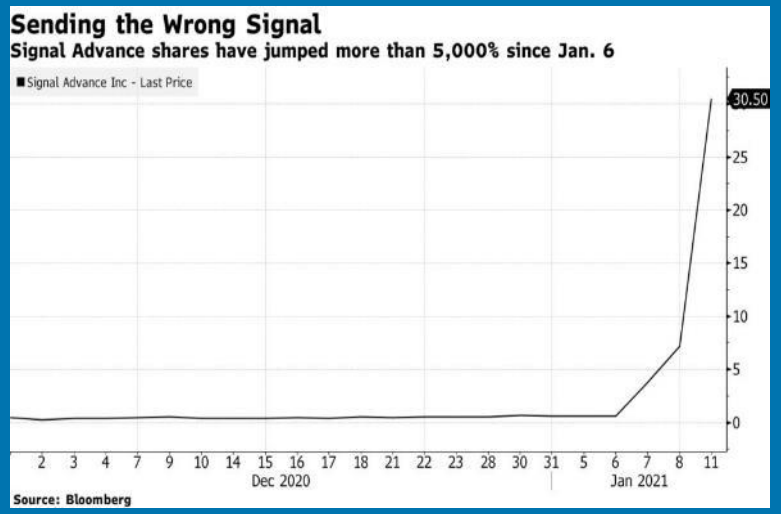
?

Claire Wardle of *First Draft News* identifies seven types of fake news:

1. satire or parody ("no intention to cause harm but has potential to fool")
2. false connection ("when headlines, visuals or captions don't support the content")
3. misleading content ("misleading use of information to frame an issue or an individual")
4. false context ("when genuine content is shared with false contextual information")
5. impostor content ("when genuine sources are impersonated" with false, made-up sources)
6. manipulated content ("when genuine information or imagery is manipulated to deceive", as with a "doctored" photo)
7. fabricated content ("new content is 100% false, designed to deceive and do harm")

Fake News Detection

?



The International Federation of Library Associations and Institutions (IFLA) published a summary in diagram form to assist people in recognizing fake news. Its main points are:

1. Consider the source (to understand its mission and purpose)
2. Read beyond the headline (to understand the whole story)
3. Check the authors (to see if they are real and credible)
4. Assess the supporting sources (to ensure they support the claims)
5. Check the date of publication (to see if the story is relevant and up to date)
6. Ask if it is a joke (to determine if it is meant to be satire)
7. Review your own biases (to see if they are affecting your judgment)
8. Ask experts (to get confirmation from independent people with knowledge)

International Federation of Library Associations and Institutions (IFLA) published a summary in diagram form to assist people in recognizing fake news.

IMPACT OF RUMOR

Financial impacts or impacts on my health

Bullying and violence against innocent people

Fear

Racist ideas

Democratic impacts

PREPARE DATA

Collect Training Dataset and Data Preprocessing

Dataset from Kaggle link :- https://www.kaggle.com/c/fake-news/data

DATA REDUCTION

Due to our machine capability is limited so we are working on less data and also selection required attributes of computation.

DATA TRANSFORMATION

We are performing Normalization to our data and check the attributed required for computation.

DATA CLEANING

We are checking missing values and verify that our dataset. Removing noisy data.

DATA PROCESSING

USING PYTHON FOR PROCESSING OF DATA

DROPING ROWS NOT IN USE

```
df = df_marge.drop(["title", "subject","date"], axis = 1)
```

```
df.isnull().sum()
```

```
text     0
class    0
dtype: int64
```

REMOVING SPECIAL CHARACTERS

```
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)
```

CONVERTING TEXT TO VECTORES

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

DIFFERENT MTEHODS WE ARE USING AND WORKING ON.

1

- Logistic Regression

2

- Decision Tree Classification

3

- Gradient Boosting Classifier

4

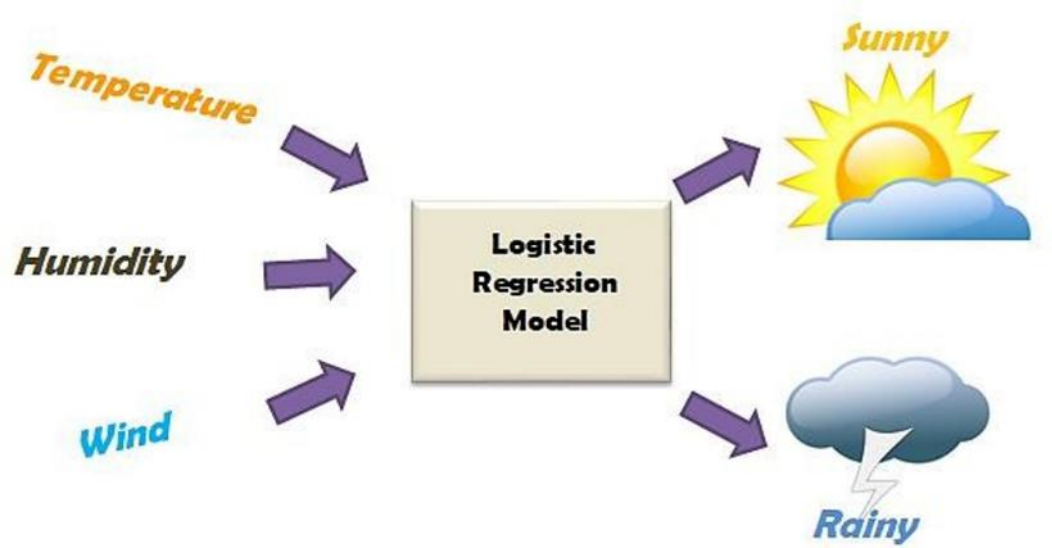
- Random Forest Classifier

5

- LSTM -Long Short-Term Memory

LOGISTIC REGRESSION

Logistic regression is a statistical **model** that in its basic form uses a **logistic** function to **model** a binary dependent variable, although many more complex extensions exist.




```
from sklearn.linear_model import LogisticRegression
```

```
LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

```
LogisticRegression()
```

```
pred_lr=LR.predict(xv_test)
```

```
LR.score(xv_test, y_test)
```

```
0.9859180035650624
```

```
print(classification_report(y_test, pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      5882
           1       0.98      0.99      0.99      5338

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


12

Decision tree

```
from sklearn.tree import DecisionTreeClassifier
```

```
DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
```

```
DecisionTreeClassifier()
```

```
pred_dt = DT.predict(xv_test)
```

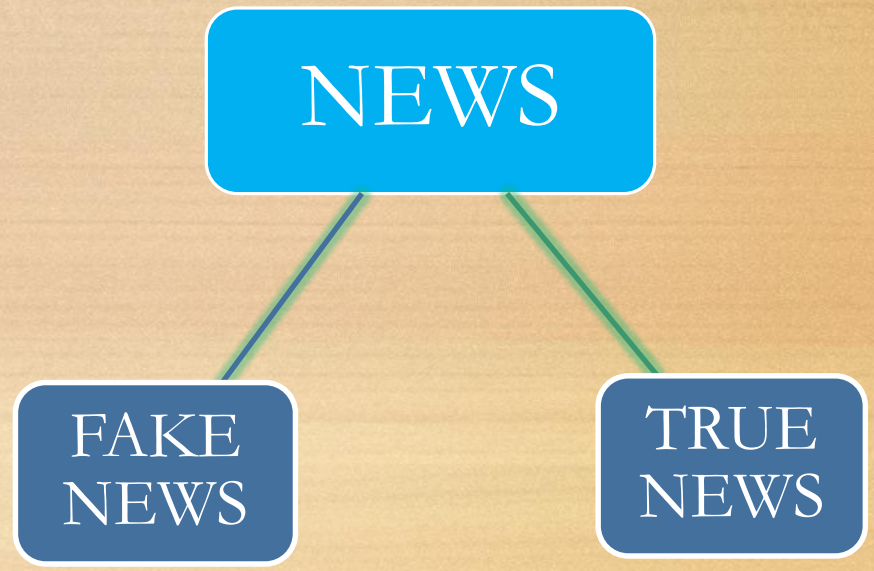
```
DT.score(xv_test, y_test)
```

```
0.9953654188948307
```

```
print(classification_report(y_test, pred_dt))
```

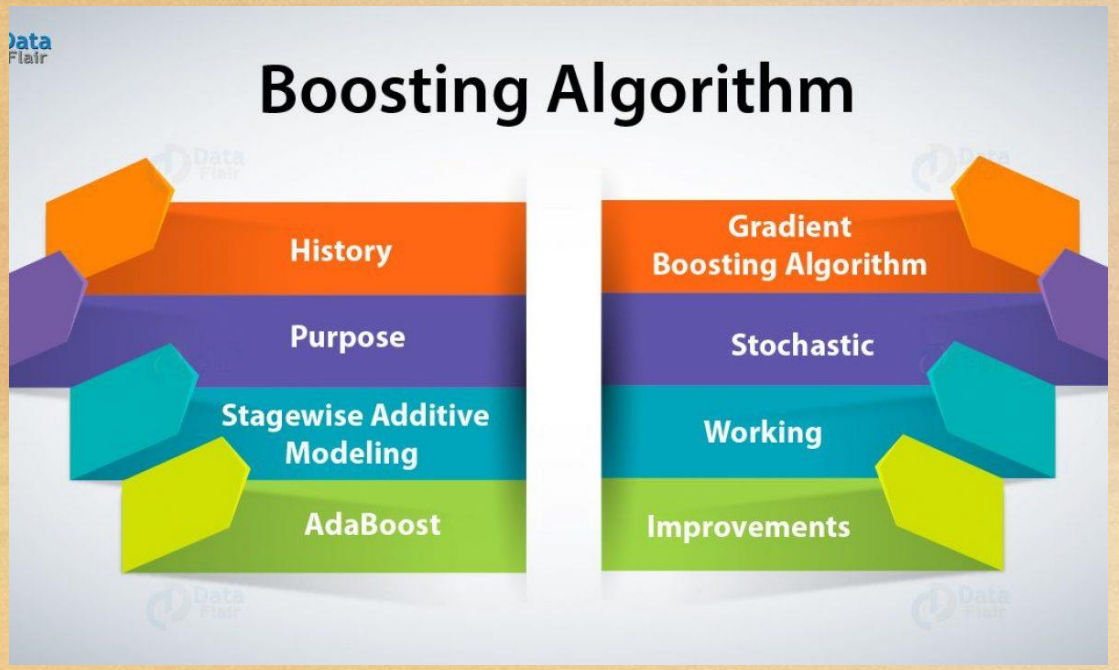
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5882
           1       1.00      0.99      1.00      5338

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

A **decision tree** is a **decision** support tool that uses a **tree**-like model of **decisions** and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.



Gradient Bosting technique

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.




```
from sklearn.ensemble import GradientBoostingClassifier
```

```
GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)
```

```
GradientBoostingClassifier(random_state=0)
```

```
pred_gbc = GBC.predict(xv_test)
```

```
GBC.score(xv_test, y_test)
```

```
0.9953654188948307
```

```
print(classification_report(y_test, pred_gbc))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5882
           1       0.99      1.00      1.00      5338

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

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
```

```
RandomForestClassifier(random_state=0)
```

```
pred_rfc = RFC.predict(xv_test)
```

```
RFC.score(xv_test, y_test)
```

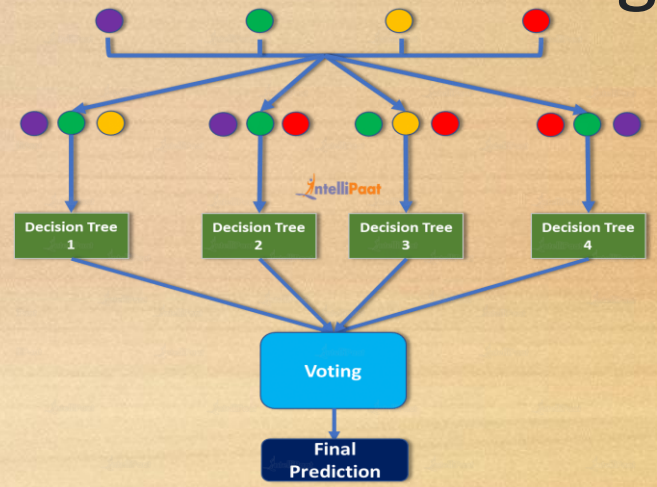
```
0.9890374331550802
```

```
print(classification_report(y_test, pred_rfc))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5882
           1       0.99      0.99      0.99      5338

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

Random forest classifier

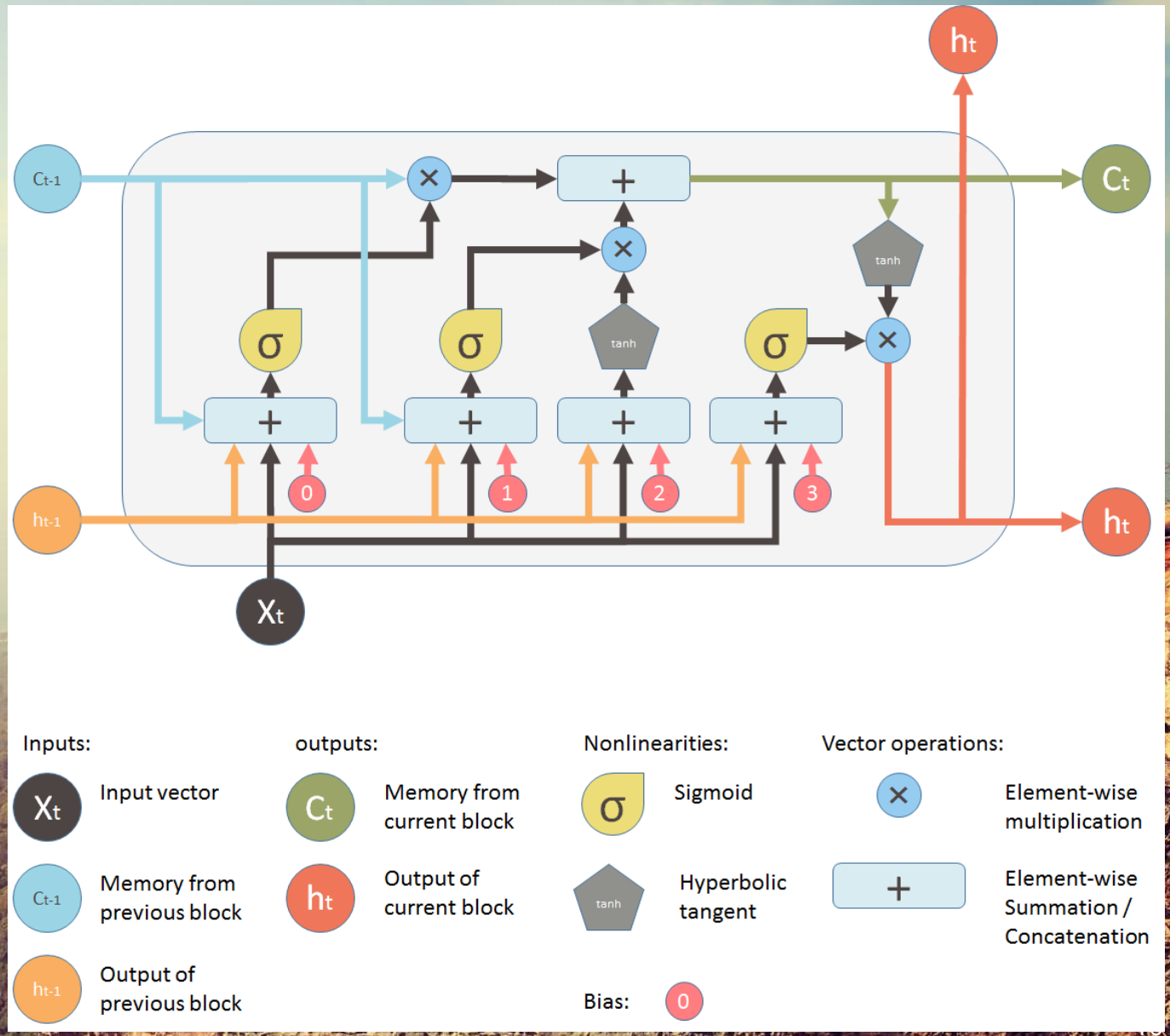
A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.



Random Forest Example

LSTM - Long Short-Term Memory

Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence. Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies.



Automatic Rumor Detection Advantages

Fake news has altered society in negative ways in politics and culture. It has adversely affected both online social network systems as well as offline communities and conversations. Using automatic machine learning classification models is an efficient way to combat the widespread dissemination of fake news.

A multimodal system with the aim to differentiate between fake and real posts. Our system is based on a neural network and combines textual, visual and semantic information. The textual information is extracted from the content of the post, the visual one from the image that is associated with the post and the semantic refers to the similarity between the image and the text of the post.

THANK YOU FOR YOUR ATTENTION!

ANY QUESTION?