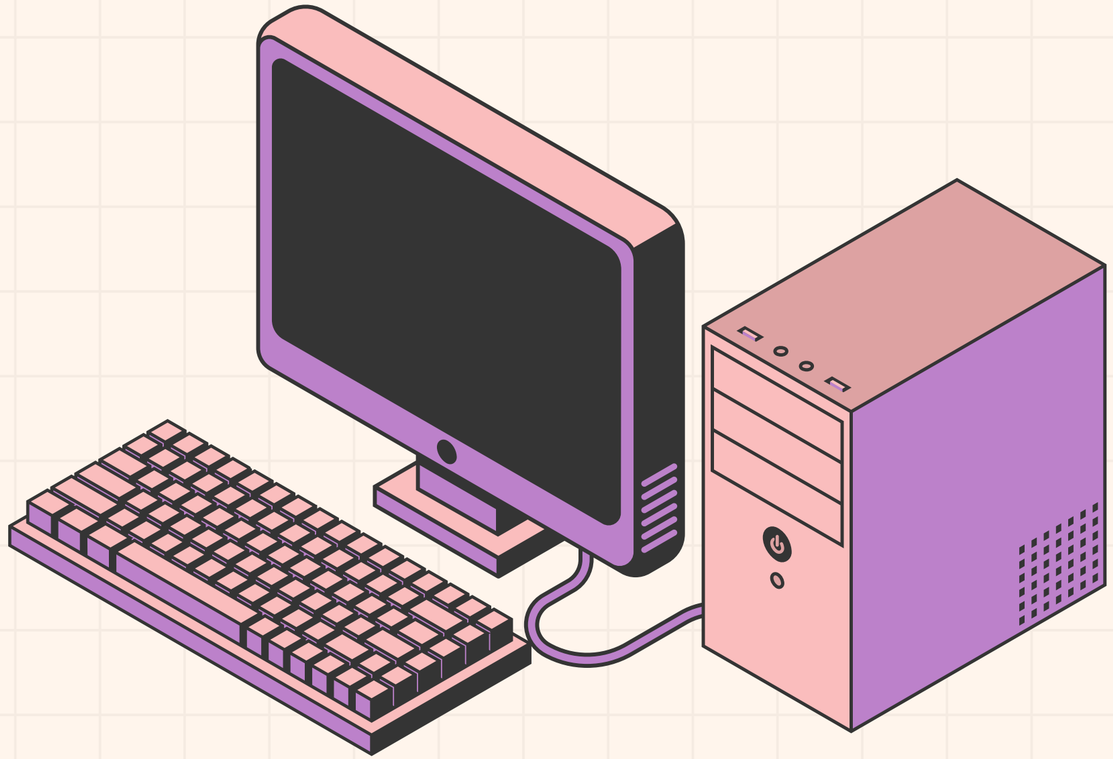
AMRITA VISHWA VIDYAPEETHAM

Mathematics For Computing II

Elements of Computing II

FAKE NEWS DETECTION USING LINEAR CLASSIFICATION



Group Members:

Diya Prakash-CB.SC.U4AIE24111

Dondluru Keerthana-CB.SC.U4AIE24112

V.R. Sridevi- CB.SC.U4AIE24166

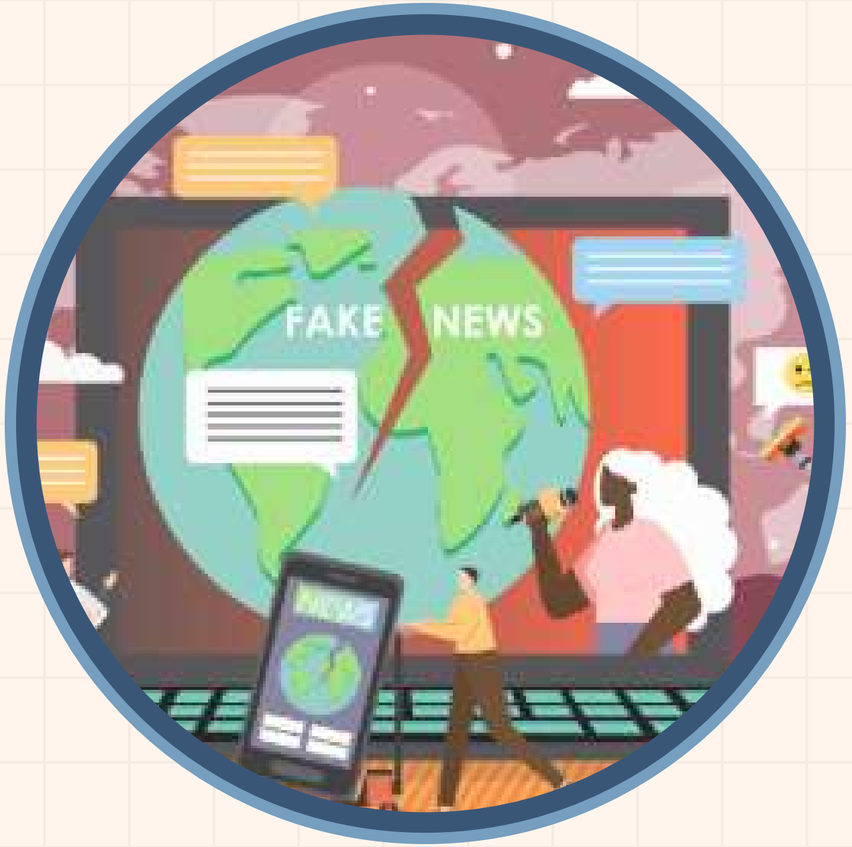
INTRODUCTION

1. Background of the Research Problem:

• The rise of social media has led to an increase in the spread of fake news, which can influence public opinion, elections, financial markets, and even health-related decisions.

• Fake news is often designed to evoke strong emotional reactions, making it more likely to be shared.

• The challenge lies in distinguishing real news from fabricated content.



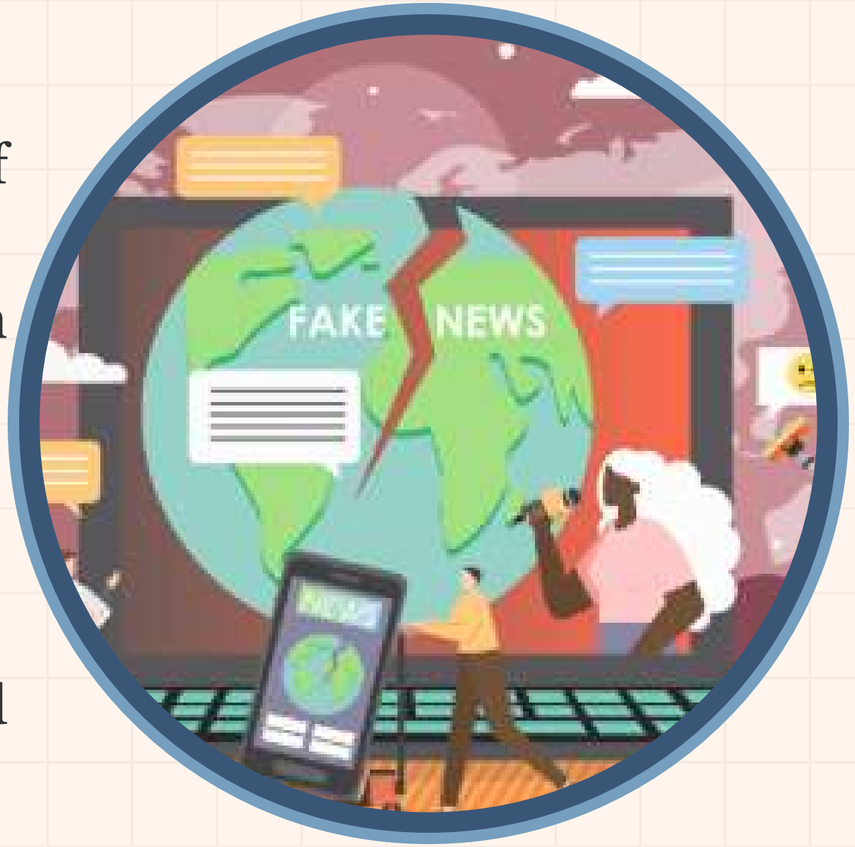
INTRODUCTION

2. Research Questions and Objectives:

• What are the most effective AI models for detecting fake news?

• How can feature extraction improve the accuracy of classification models?

• What role do metadata and user engagement metrics play in determining the credibility of news articles?



3. Importance of the Study:

• AI-powered fake news detection can help curb misinformation.

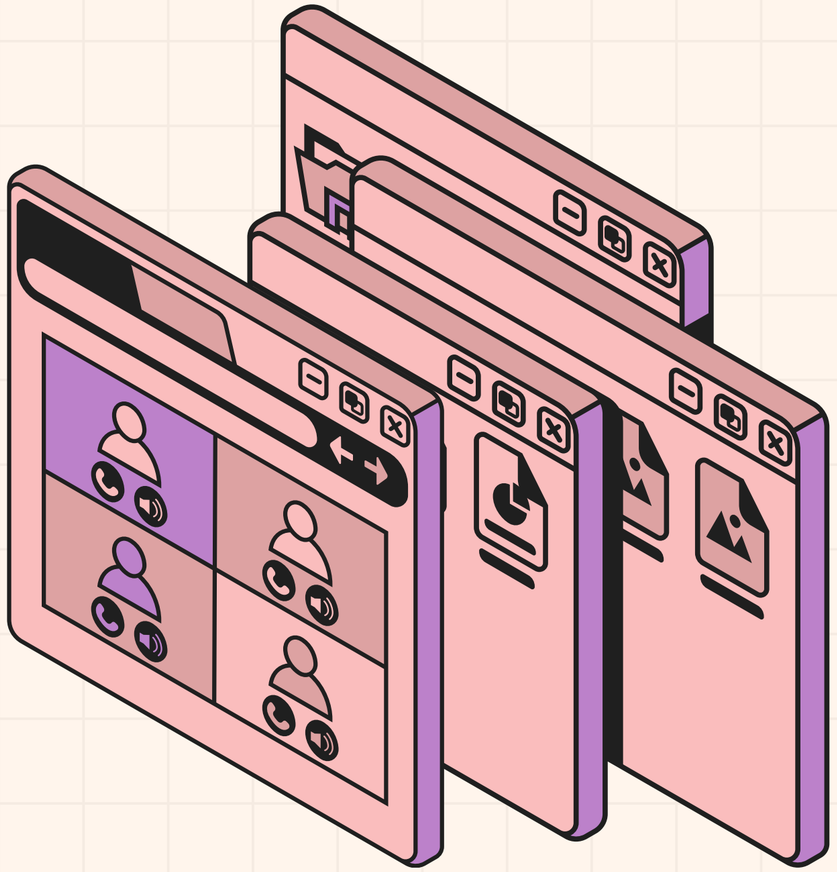
• It assists media organizations, social media platforms, and policymakers in regulating content.

• A better understanding of fake news detection can lead to stronger policies against misinformation.

OBJECTIVE

Primary Goal: Develop an efficient and scalable fake news detection model using linear classification and other techniques.

Key Objectives: Apply **Linear Classification, SVM, Logistic Regression, Forest Classification, XG Boost and Definition Tree** to classify news articles. Improve detection accuracy using feature selection and dimensionality reduction. Implement a mathematical classification system. Train and test the model using real-world datasets.



PROBLEM STATEMENT

The rapid spread of misinformation and fake news on online platforms, especially social media, has become a major societal issue Traditional fact-checking methods are slow and ineffective for handling large-scale information. Existing fake news detection methods rely on deep learning, which often requires extensive computational resources. This project proposes a mathematical approach using linear classification and other ML feature reduction techniques to detect fake news more efficiently, even with limited computational power.

PROPOSED TIMELINE AND OUR PROGRESS

Research & Data Collection (Weeks 1-2)

- Review existing fake news detection methods.
- Gather datasets (Kaggle, BuzzFeed News, FactCheck.org).

Preprocessing & Feature Extraction (Weeks 3-4)

- Clean and tokenize text, apply TF-IDF for feature extraction and processing of dataset

Model Implementation (Weeks 5-6)

- Train and optimize SVM, Logistic Regression and other similar mathematical models.

Evaluation & Comparison(Weeks 7-8)

- Measure accuracy, precision and recall.
- Compare results with pre-existing models.

Documentation & Final Presentation (Weeks 9-10)

- Compile findings, challenges, and future improvements.
- Prepare the final report and presentation.

We're here!

METHODOLOGY

1. **Dataset Collection and Preparation**
   - The dataset consists of multiple files containing news articles labeled as real or fake.
   - The files are loaded from the Combined_News_Datasets directory.
   - Labels are assigned: 1 for fake news and 0 for real news.
2. **Data Preprocessing**
   - The text data is vectorized using Term Frequency-Inverse Document Frequency (TF-IDF) representation to convert textual data into numerical features.
   - Stopwords are removed, and text normalization techniques such as lowercasing are applied.
3. **Dataset Splitting**
   - The dataset is split into training and testing sets using an 80-20 ratio.
   - The train_test_split function from sklearn.model_selection is used to ensure proper distribution of data.

METHODOLOGY

4. Model Testing and Training

- Linear Classification, SVM, Logistic Regression, Forest Classification, XG Boost and Definition Tree are tested as the classification model.
- The model is trained using the TF-IDF transformed dataset not only TF-IDF we can use N-grams, POS tagging, Metadata features.
- Hyperparameters are set to optimize model performance.

5. Model Evaluation

- The trained model is tested on the test dataset.
- Performance metrics such as accuracy, precision, recall, and F1-score are computed.
- The classification_report and accuracy_score from sklearn.metrics are used to evaluate the results.

6. Model Deployment and Saving

- The trained model is saved using pickle(to save and load objects) for future use.
- The saved model can be reloaded for real-world classification tasks without retraining.

CLASSIFICATION MODELS

1. Linear Classification

- Uses a straight line (or plane) to separate data into categories.
- Assigns weights to features and decides based on a threshold.

2. Support Vector Machine (SVM)

- Finds the best boundary that maximizes the gap between different classes.
- Can handle complex data using kernel tricks.

3. Logistic Regression

- Predicts probabilities for two categories (e.g., Yes/No).
- Uses a sigmoid function to convert outputs into values between 0 and 1.

4. Random Forest

- Uses multiple decision trees to improve accuracy.
- Combines results from many trees to make a final decision.

CLASSIFICATION MODELS

5. XGBoost

- A fast and optimized boosting technique that builds trees step by step.
- Corrects previous mistakes to improve predictions.

6. Decision Tree

- Breaks data into smaller parts by asking simple "Yes/No" questions.
- Reaches a final decision at the leaf nodes.

MATHEMATICAL BACKGROUND

LOGICAL REGRESSION

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- hθ(x) is the predicted probability that the output is 1 (true, positive class
- θ is the vector of weights/parameters.
- x is the vector of input features.
- θTx is the dot product of the parameters and input features.
- e is Euler's number, the base of the natural logarithm.

SVM

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2$$

- w is the weight vector (normal to the hyperplane).
- b is the bias term.
- xi are the input feature vectors.
- yi∈{-1,+1} are the class labels.
- The constraint ensures that all points are on the correct side of the margin.

LINEAR CLASSIFICATION

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

- x is the input feature vector.
- w is the weight vector (learned during training).
- b is the bias term.
- . denotes the dot product.

XGBoost

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k)$$

- l(yi^,yi) is the loss function (e.g., logistic loss for classification).
- fk is the prediction of tree k.
- Ω(fk) is the regularization term, which penalizes model complexity to prevent overfitting.

FOREST CLASSIFICATION

$$\hat{y} = \text{mode}\left(T_1(\mathbf{x}), T_2(\mathbf{x}), \ldots, T_N(\mathbf{x})\right)$$

- Ti(x) is the prediction from the iiith decision tree.
- mode takes the majority class across all tree

DECISION TREE

$$\text{Gini}(D) = 1 - \sum_{i=1}^{c} p_i^2$$

- D = dataset at a given node.
- c = number of classes.
- pi = proportion of samples in class iii.

OTHER FEATURE EXTRACTION METHODS

1. N-Grams

Definition: Groups of consecutive words or characters.

Example (for "fake news detection"):

- **Unigrams (n=1):** "fake", "news", "detection"
- **Bigrams (n=2):** "fake news", "news detection"
- **Trigrams (n=3):** "fake news detection"
- **Why use it?** Helps detect patterns in language, useful for spotting common fake news phrases.

2. POS Tagging (Part of Speech Tagging)

Definition: Identifies the role of each word (e.g., noun, verb, adjective).

Why use it? Fake news often overuses emotional or exaggerated words.

3. Metadata Features

Article length: Fake news is often shorter.

Punctuation & Caps: Excessive "!!!" or ALL CAPS may signal fake news.

Source reliability: Some websites are known for spreading false information.

These are the alternatives for TD-IDF which can also be implemented along with it for better results

ML MODELS AND THEIR ACCURACY RESULTS

1. Model: Logical Regression

```
Accuracy: 0.4015748031496063

Classification Report:
              precision    recall  f1-score   support

           0       0.36      0.41      0.39        58
           1       0.44      0.39      0.42        69

    accuracy                           0.40       127
   macro avg       0.40      0.40      0.40       127
weighted avg       0.41      0.40      0.40       127
```

2. Model: SVM

```
Accuracy: 0.3937007874015748

Classification Report:
              precision    recall  f1-score   support

           0       0.36      0.43      0.39        58
           1       0.43      0.36      0.39        69

    accuracy                           0.39       127
   macro avg       0.40      0.40      0.39       127
weighted avg       0.40      0.39      0.39       127
```

3. Model: Linear Classification

```
Accuracy: 0.3779527559055118

Classification Report:
              precision    recall  f1-score   support

           0       0.33      0.34      0.34        58
           1       0.42      0.41      0.41        69

    accuracy                           0.38       127
   macro avg       0.38      0.38      0.38       127
weighted avg       0.38      0.38      0.38       127
```

4. Model: Forest Classification

```
Accuracy: 0.4645669291338583

Classification Report:
              precision    recall  f1-score   support

           0       0.42      0.45      0.43        58
           1       0.51      0.48      0.49        69

    accuracy                           0.46       127
   macro avg       0.46      0.46      0.46       127
weighted avg       0.47      0.46      0.47       127
```

ML MODELS AND THEIR ACCURACY RESULTS

5. Model: XG Boost class

```
Accuracy: 0.36220472440944884

Classification Report:
              precision    recall  f1-score   support

           0       0.33      0.40      0.36        58
           1       0.40      0.33      0.36        69

    accuracy                           0.36       127
   macro avg       0.36      0.36      0.36       127
weighted avg       0.37      0.36      0.36       127
```

6. Model: Decision tree classifier

```
Accuracy: 0.31496062992125984

Classification Report:
              precision    recall  f1-score   support

           0       0.31      0.40      0.35        58
           1       0.33      0.25      0.28        69

    accuracy                           0.31       127
   macro avg       0.32      0.32      0.31       127
weighted avg       0.32      0.31      0.31       127
```

COMPARISION OF MODELS

Model	Accuracy	Best Precision	Best Recall	Best F1-Score
Logical Regression	0.40	0.44 (Label 1)	0.41 (Label 0)	0.42 (Label 1)
SVM	0.39	0.43 (Label 1)	0.41 (Label 0)	0.40 (Label 1)
Linear Classification	0.38	0.42 (Label 1)	0.41 (Label 1)	0.39 (Label 1)
Forest Classification	0.47	0.51 (Label 1)	0.48 (Label 1)	0.48 (Label 1)
XG Boost Classification	0.36	0.40 (Label 1)	0.48 (Label 0)	0.36 (Both)
Decision Tree Classification	0.32	0.33 (Label 1)	0.40 (Label 0)	0.35 (Label 0)

INFERENCE FROM THE RESULTS

Best Performing Method:

Among the methods analyzed, Forest Classification stands out as the best-performing method with:

- The highest accuracy (0.47).
- The best precision, recall, and F1-score, consistently outperforming other models (e.g., precision and recall of 0.51 and 0.48 for Label 1).

Significance of Findings:

- Model Selection: Your findings highlight the importance of evaluating multiple metrics, as the highest-performing method (Forest Classification) shows a balance between precision, recall, and F1-scores. This model likely generalizes better in the classification task compared to others.
- Context Dependency: Depending on the use case, the focus metric (e.g., recall vs. precision) might influence the model choice.
- Model Optimization: While Forest Classification performs the best, parameter tuning or feature engineering could enhance other models like SVM or XG Boost, depending on resource availability or computational constraints.

TAKEAWAYS, LIMITATIONS AND FUTURE IMPROVEMENTS

Key Takeaways:

- Linear Classification models can successfully detect fake news with high accuracy.
- Feature engineering plays a crucial role in classification performance.

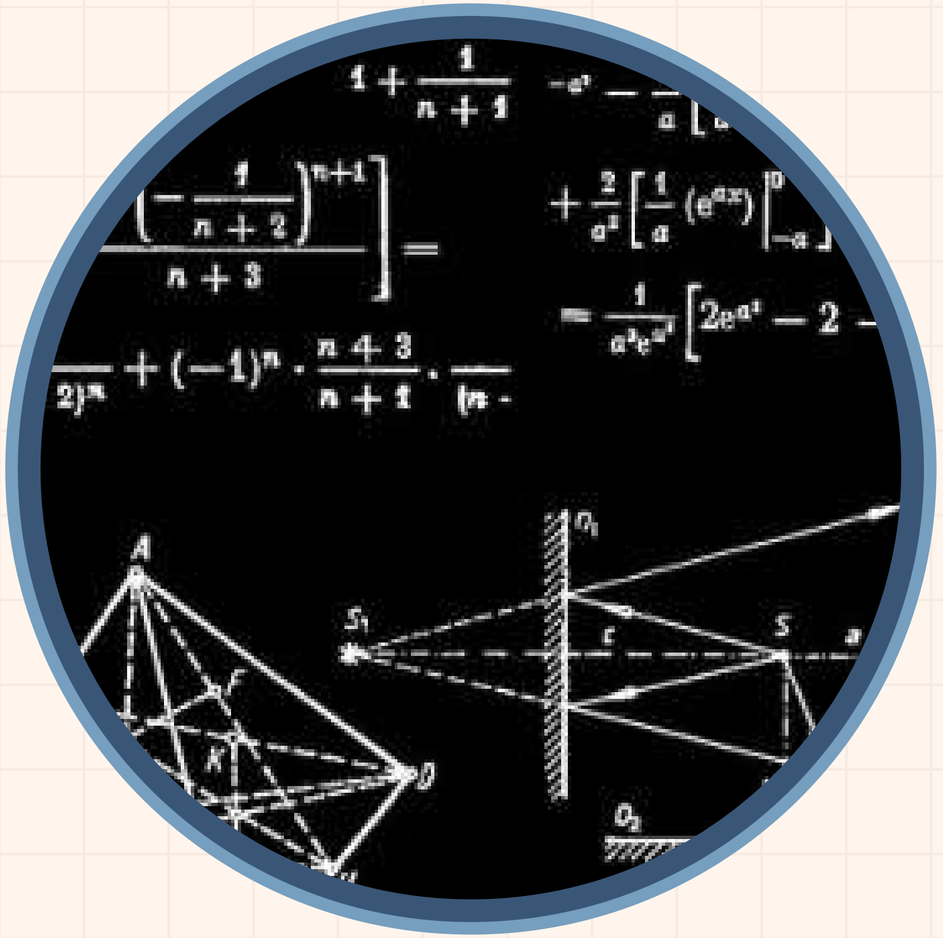
Limitations of the Study:

- The dataset used may not cover all forms of fake news.
- Fake news evolves over time, requiring continuous updates to the model.

Future Improvements possible:

- **Better Models:** Use advanced tools to understand news more accurately.
- **Real-Time Detection:** Spot fake news as soon as it appears online**.**
- **Multi-Language Support:** Find fake news in different languages and platforms**.**
- **Explainable AI:** Show clear reasons why news is marked as fake.
- **User Awareness Tools:** Teach people how to spot fake news themselves.

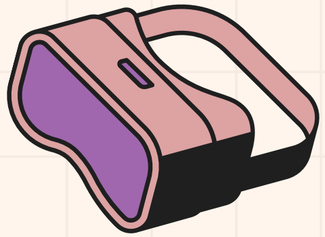
CONCLUSION

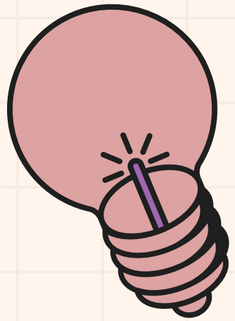


This project presents an efficient and scalable fake news detection system using linear classification techniques. By leveraging mathematical modeling and computational methods, we develop a solution that is:

- Efficient and lightweight, requiring minimal computational resources.
- Scalable, making it deployable for real-time fake news detection.
- Mathematically robust, utilizing concepts from Linear Algebra (MFC) and Elements of Computing (EOC).

This project will significantly contribute to the ongoing research on automated misinformation detection, making online information more reliable and trustworthy.





THANK YOU