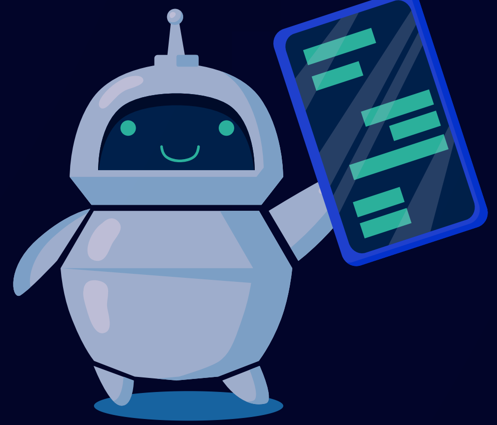


NLP CHALLENGE

PRESENTED BY : NEURAL CORE.

OUR TEAM

Raneem

Ranwah

Wejdan

INTRODUCTION

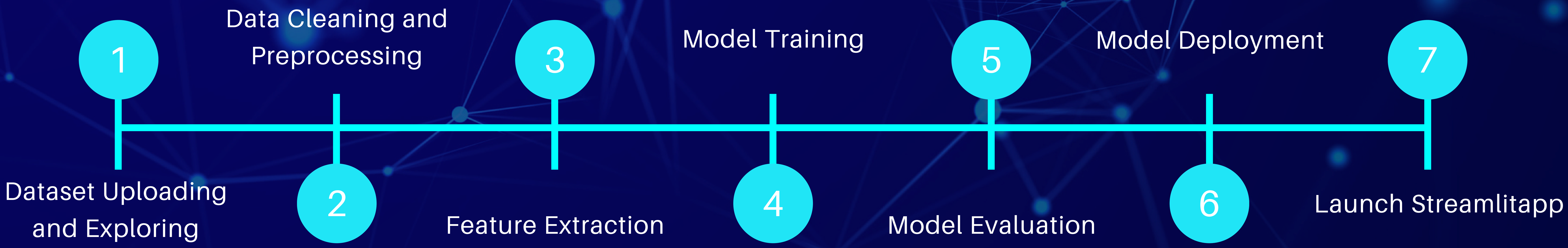
What is Fake News Classification?

- With the rise of digital news, fake news has become a major issue.
- The goal of this project is to build an AI model that differentiates between real and fake news.

Why Natural Language Processing (NLP)?

- NLP helps analyze text automatically using AI.
- Understanding linguistic patterns in fake vs. real news.

PROJECT WORKFLOW



1 Dataset Uploading and Exploring

2 Data Cleaning and Preprocessing

3 Feature Extraction

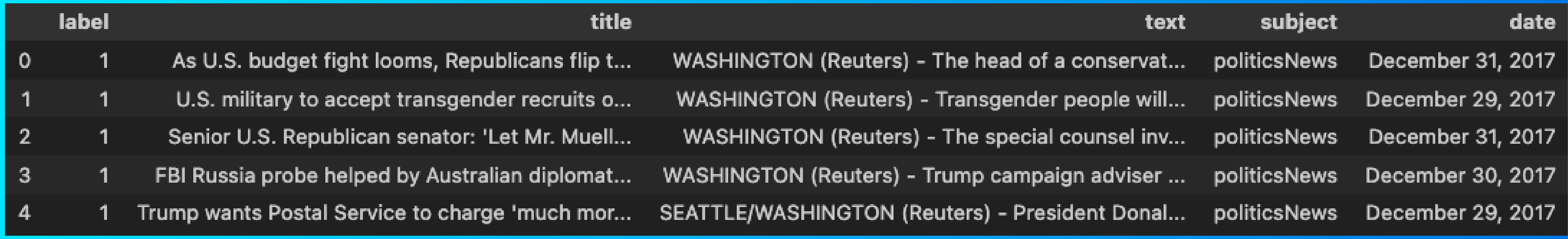
4 Model Training

5 Model Evaluation

6 Model Deployment

7 Launch Streamlitapp

DATASET OVERVIEW



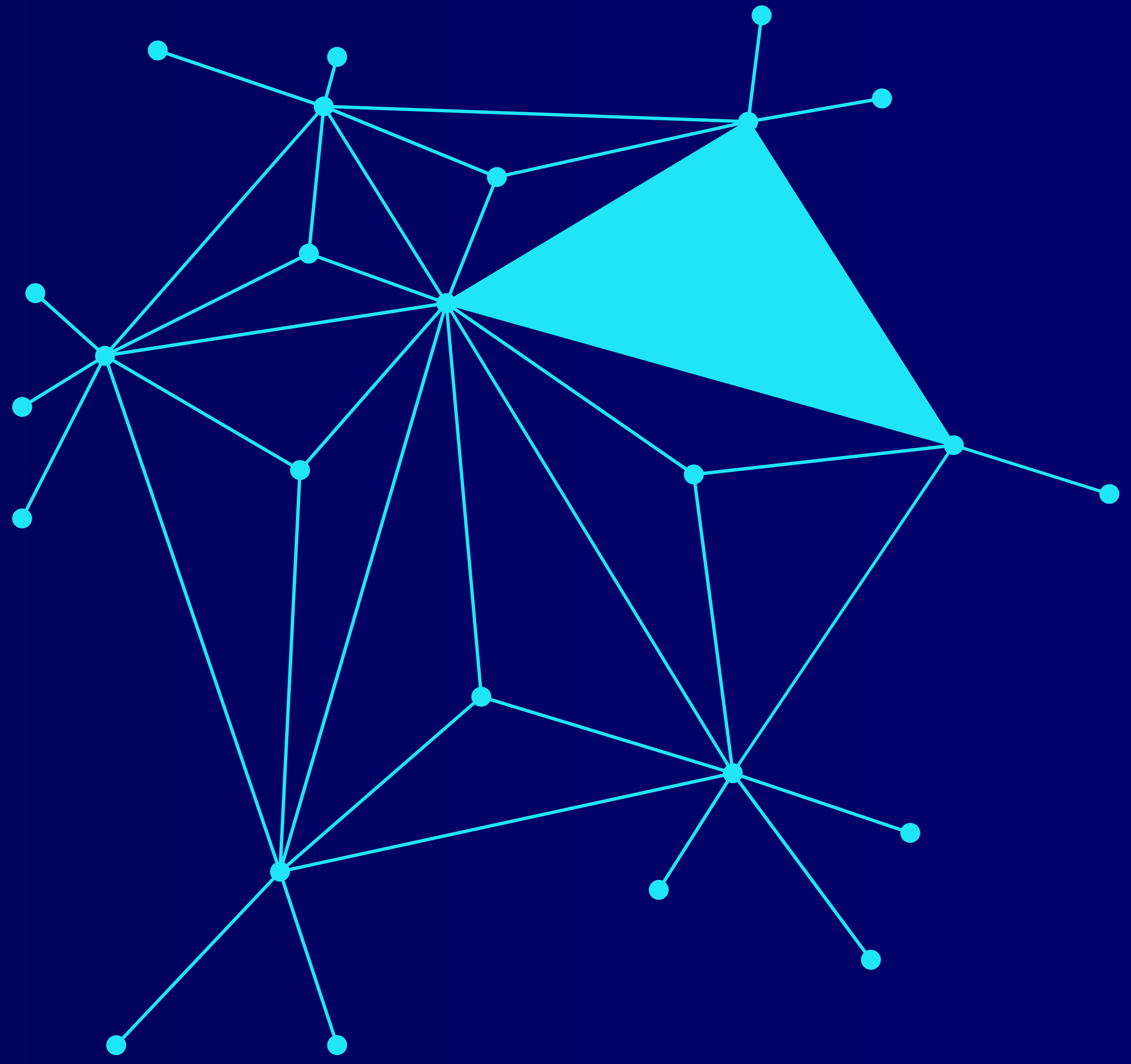
	label	title	text	subject	date
0	1	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	1	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	1	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	1	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Dataset Description:

- Contains columns: title , text, subject, data, label.
- Label:
- 0 = Fake News
- 1 = Real News
- Total number of samples: [40.000].

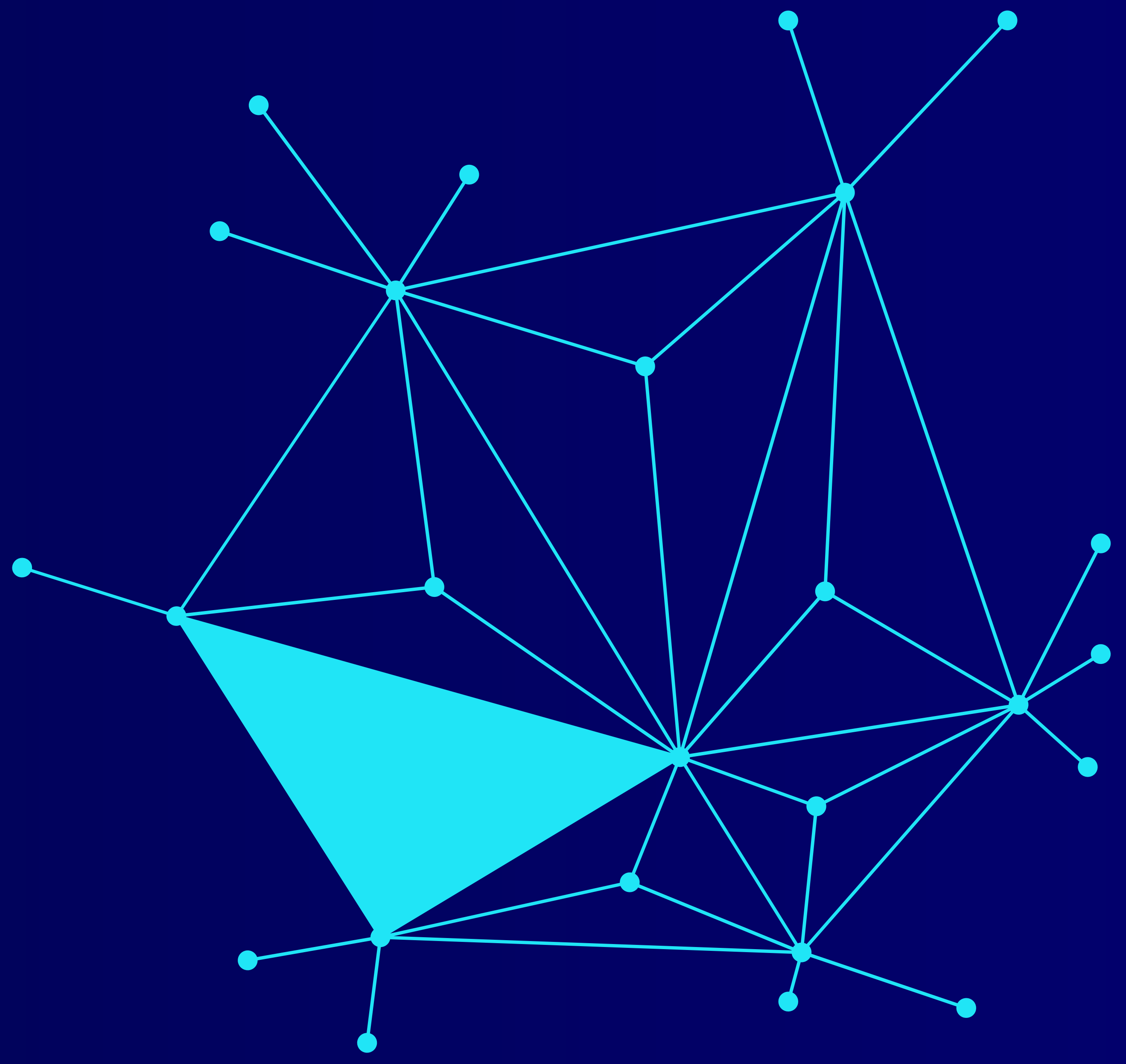
DATA PREPROCESSING STEPS

- **Preprocessing:** data was preprocessed by removing punctuation, converting to lowercase, expanding contractions, tokenization, stopword removal, combines the title and text, lemmatization, and generating n-grams.



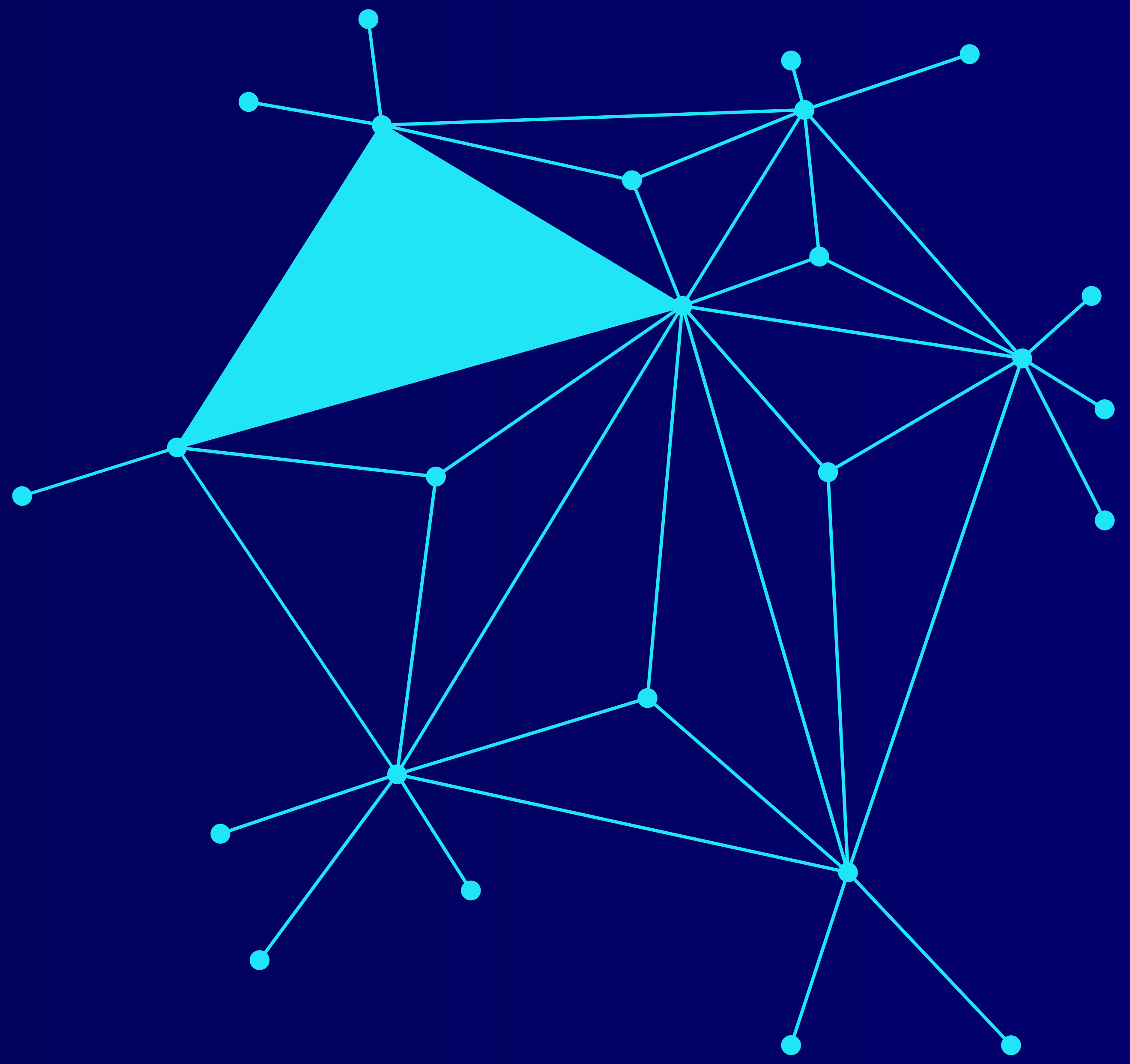
DATA PREPROCESSING STEPS

- **Feature Extraction:** We used TF-IDF to convert text into numerical features and Word2Vec embeddings to capture word semantics for richer analysis.

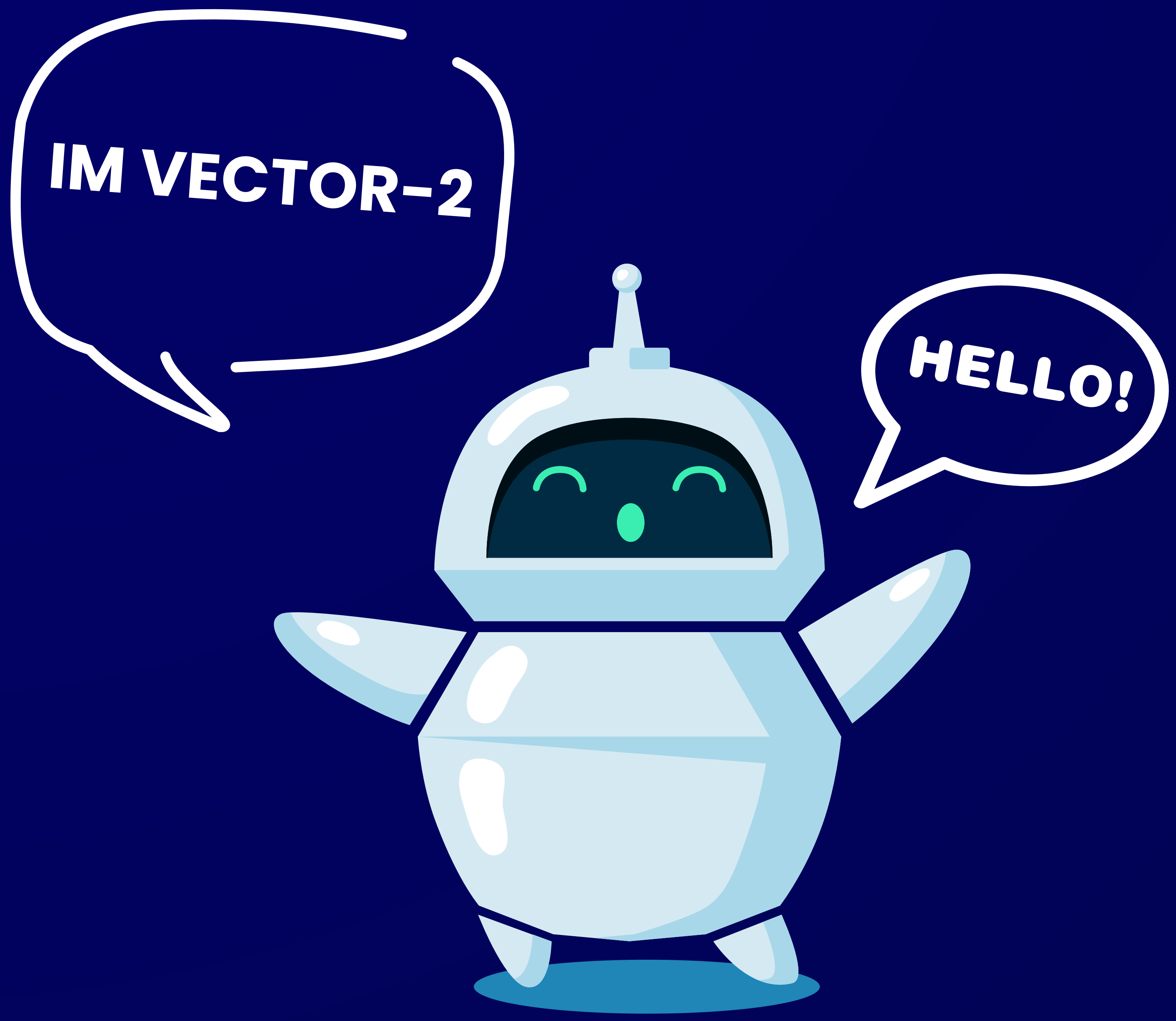


DATA PREPROCESSING STEPS

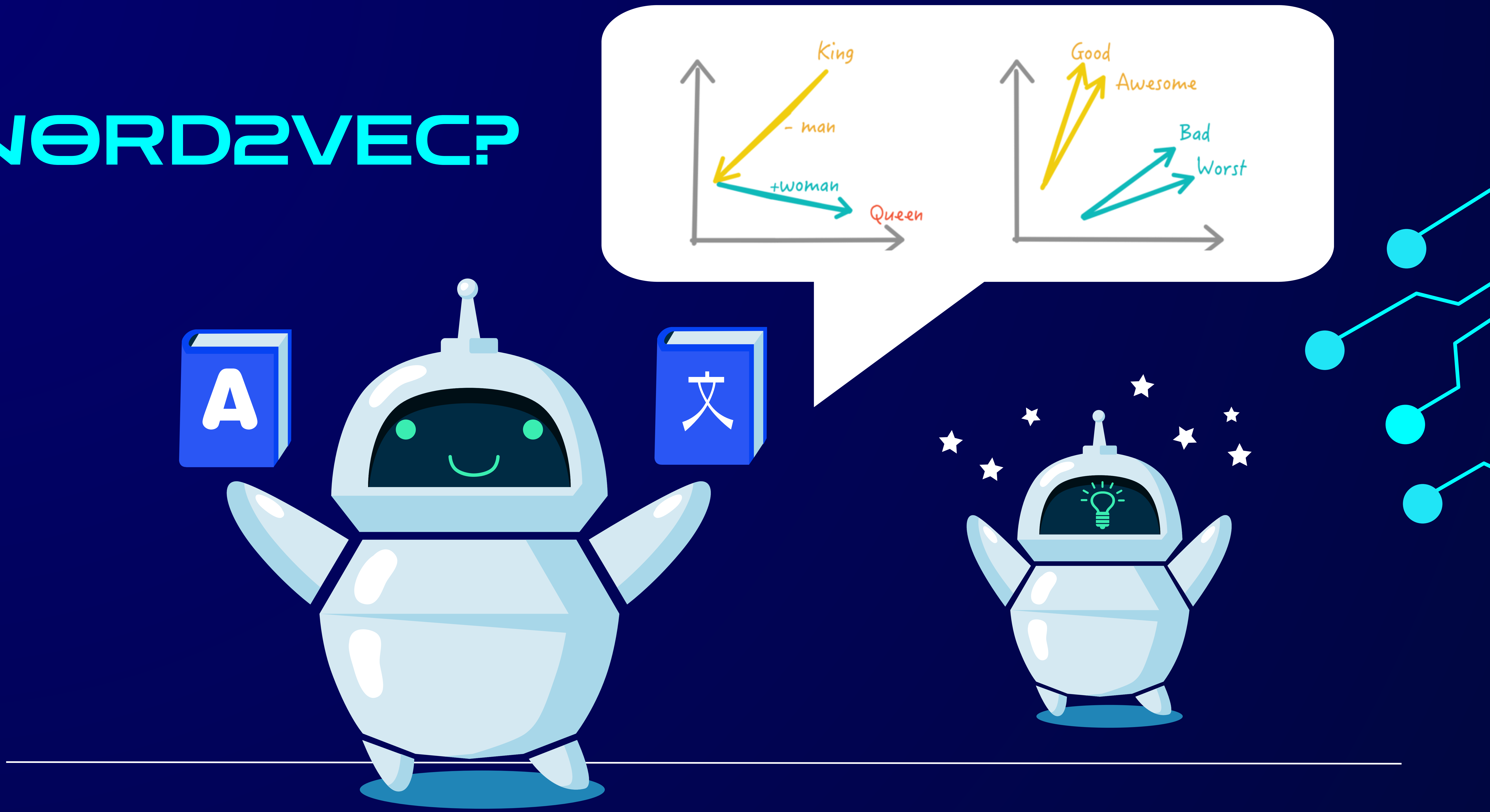
- **Modeling:** Multiple models were trained, including Naïve Bayes, Logistic Regression, Support Vector Machine (SVM).



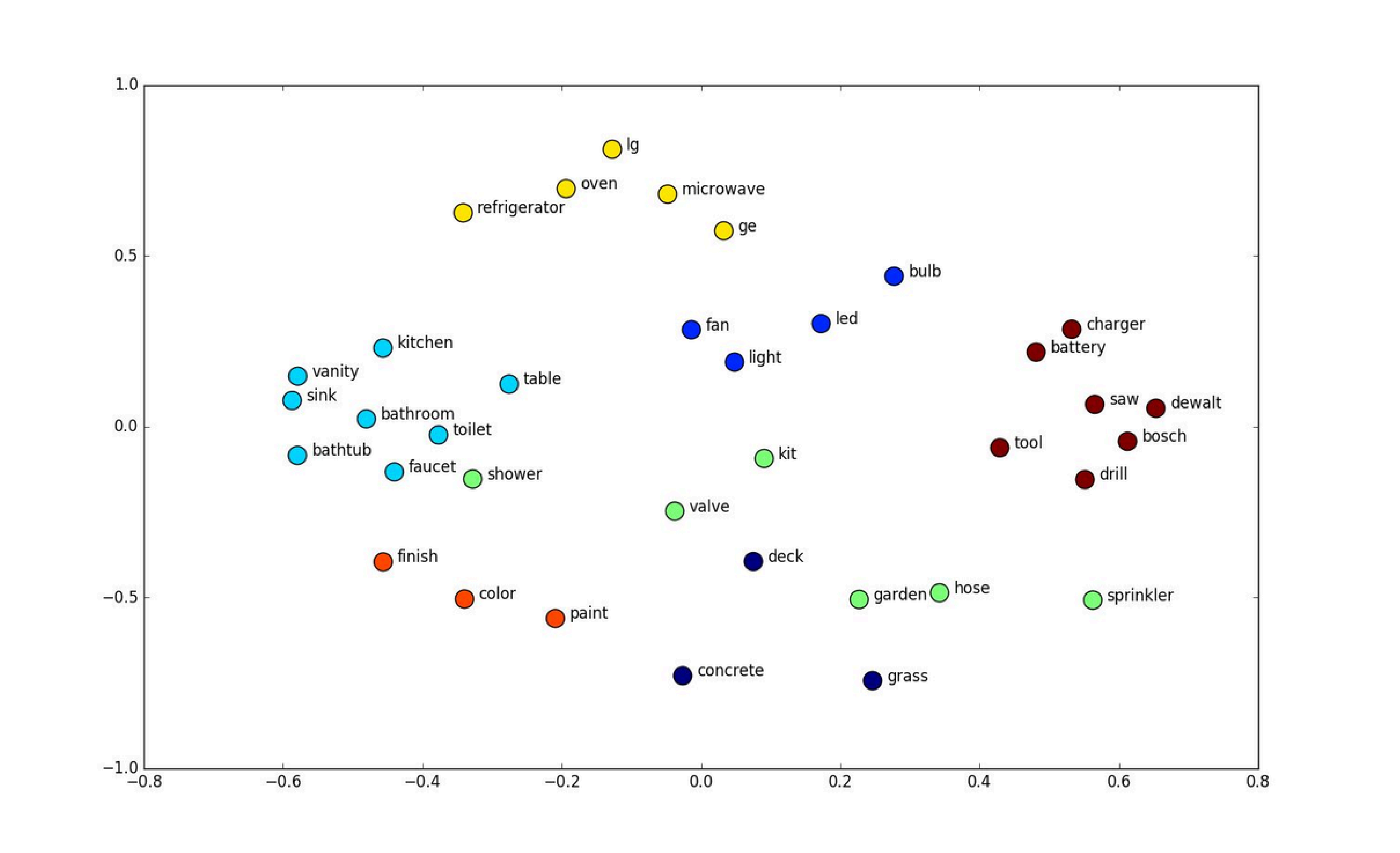
WORD2VEC?



WORD2VEC?

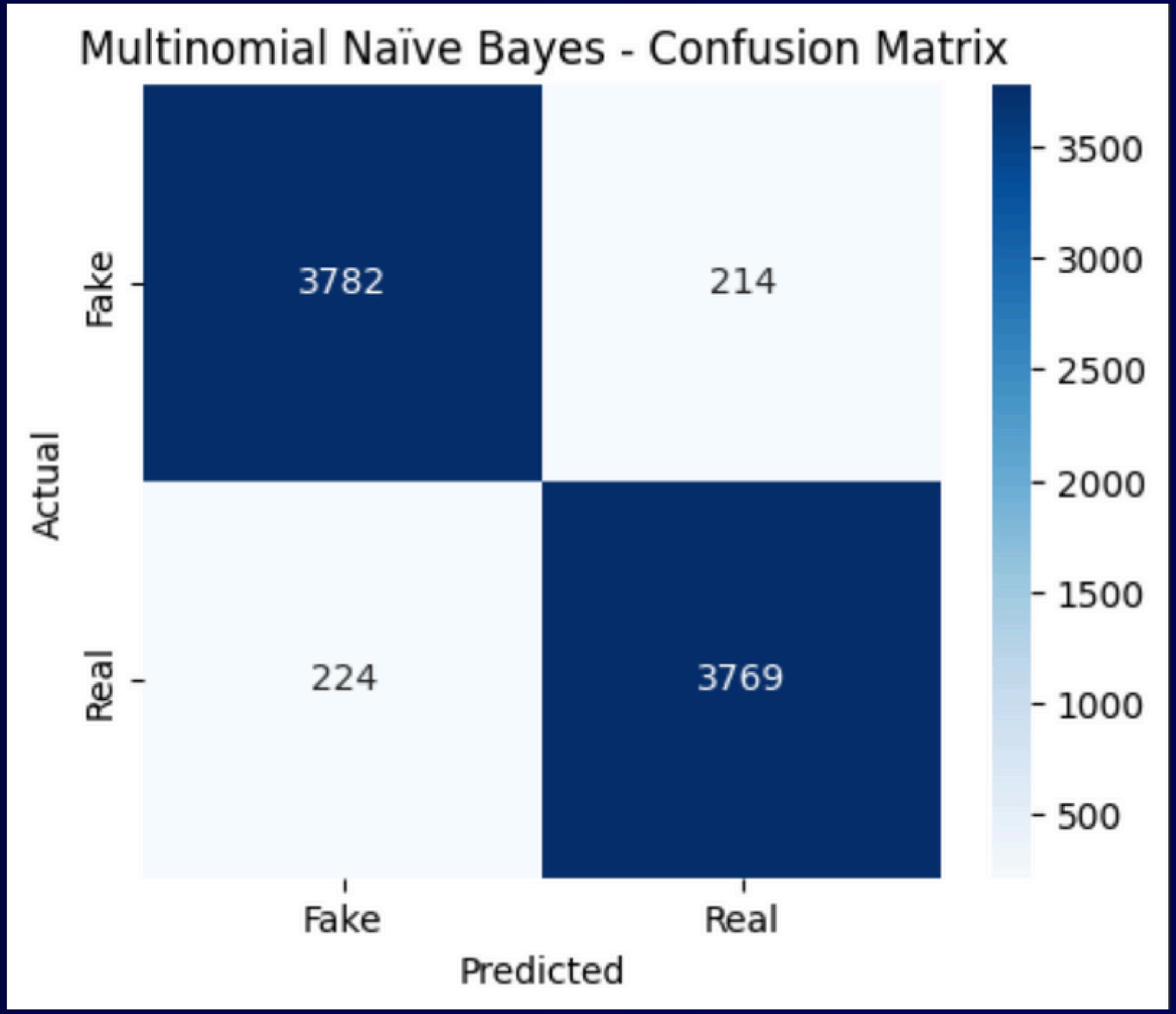


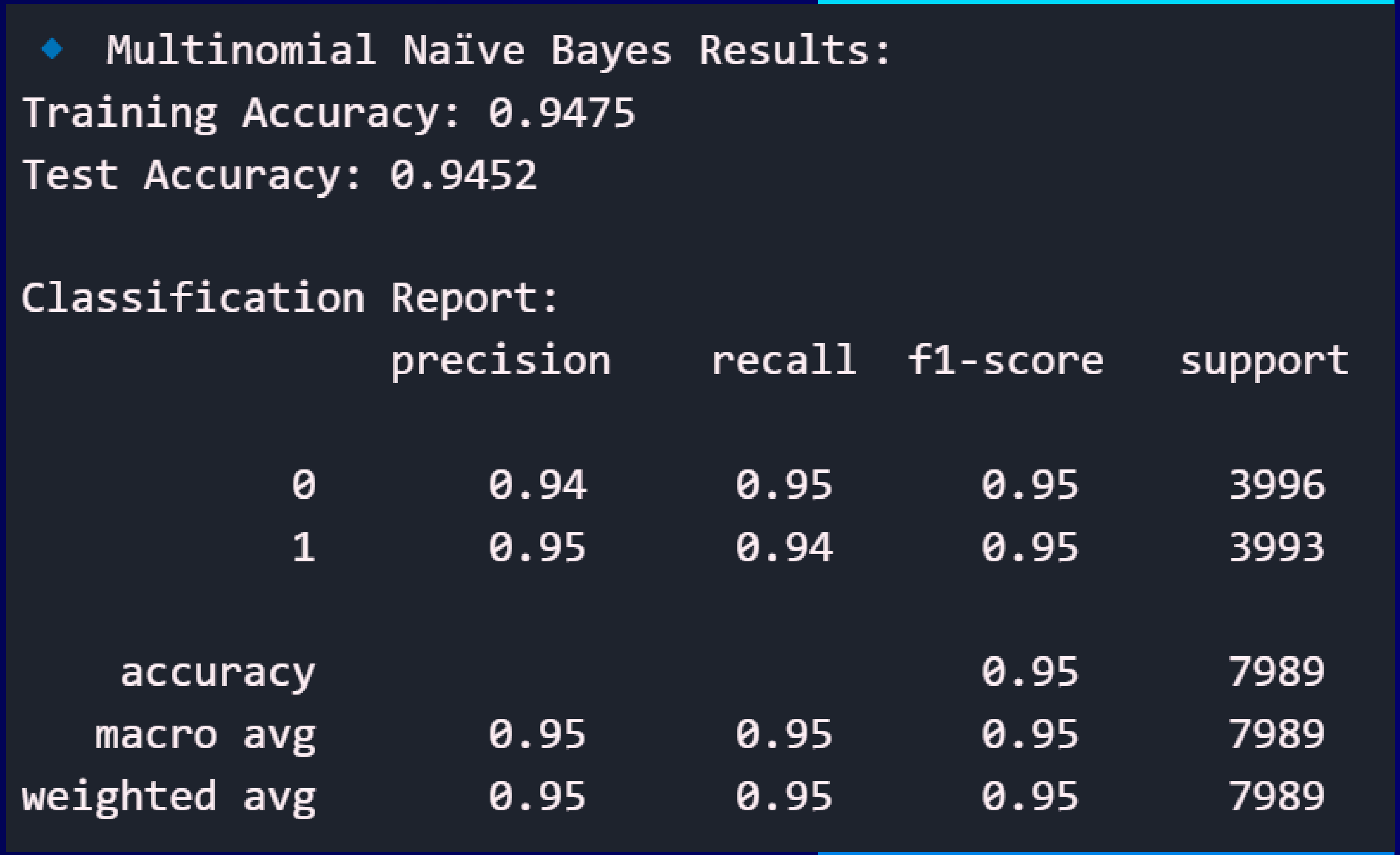
WORD2VEC?



A MAP?

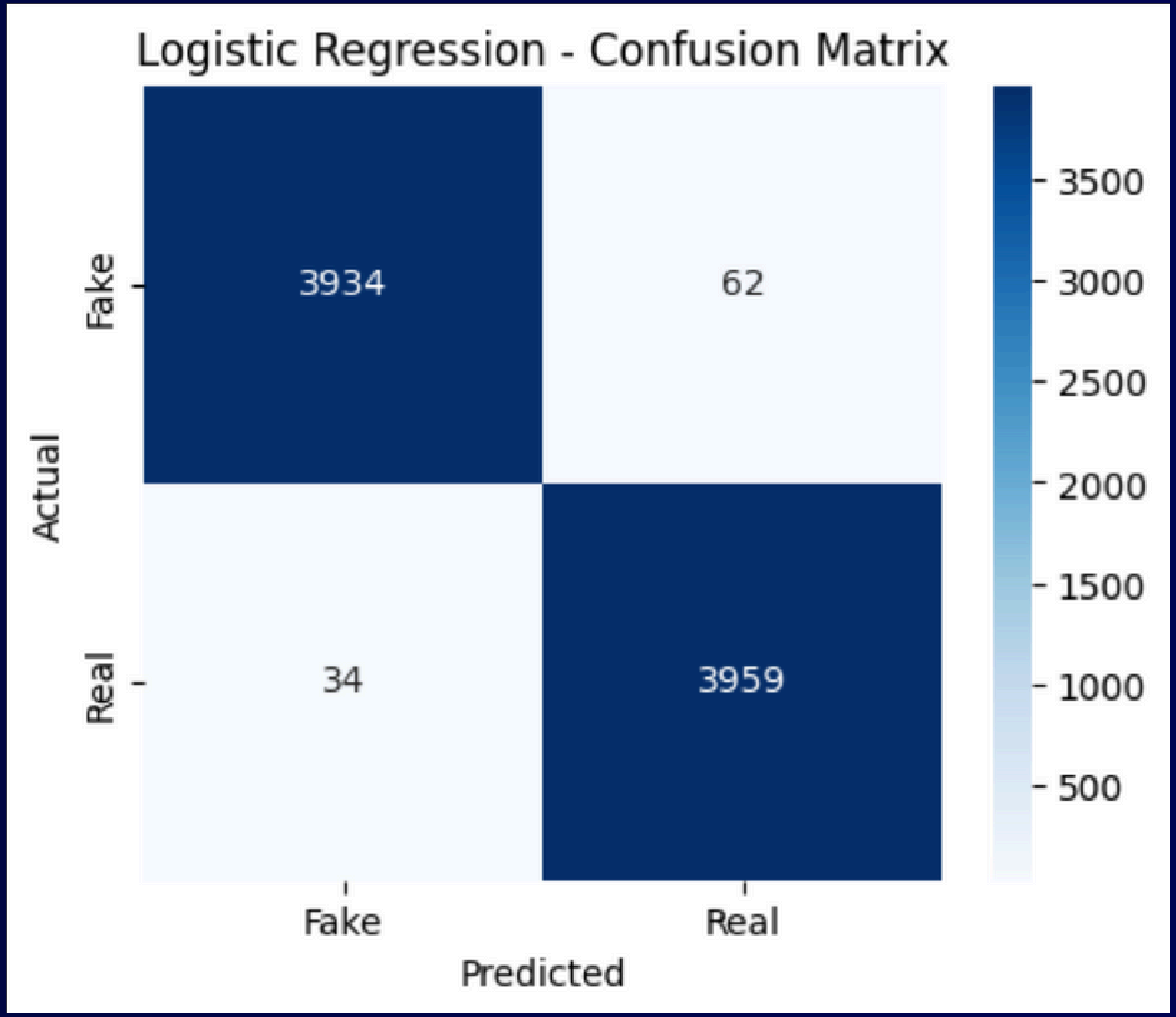
MULTINOMIAL NAÏVE BAYES MODEL

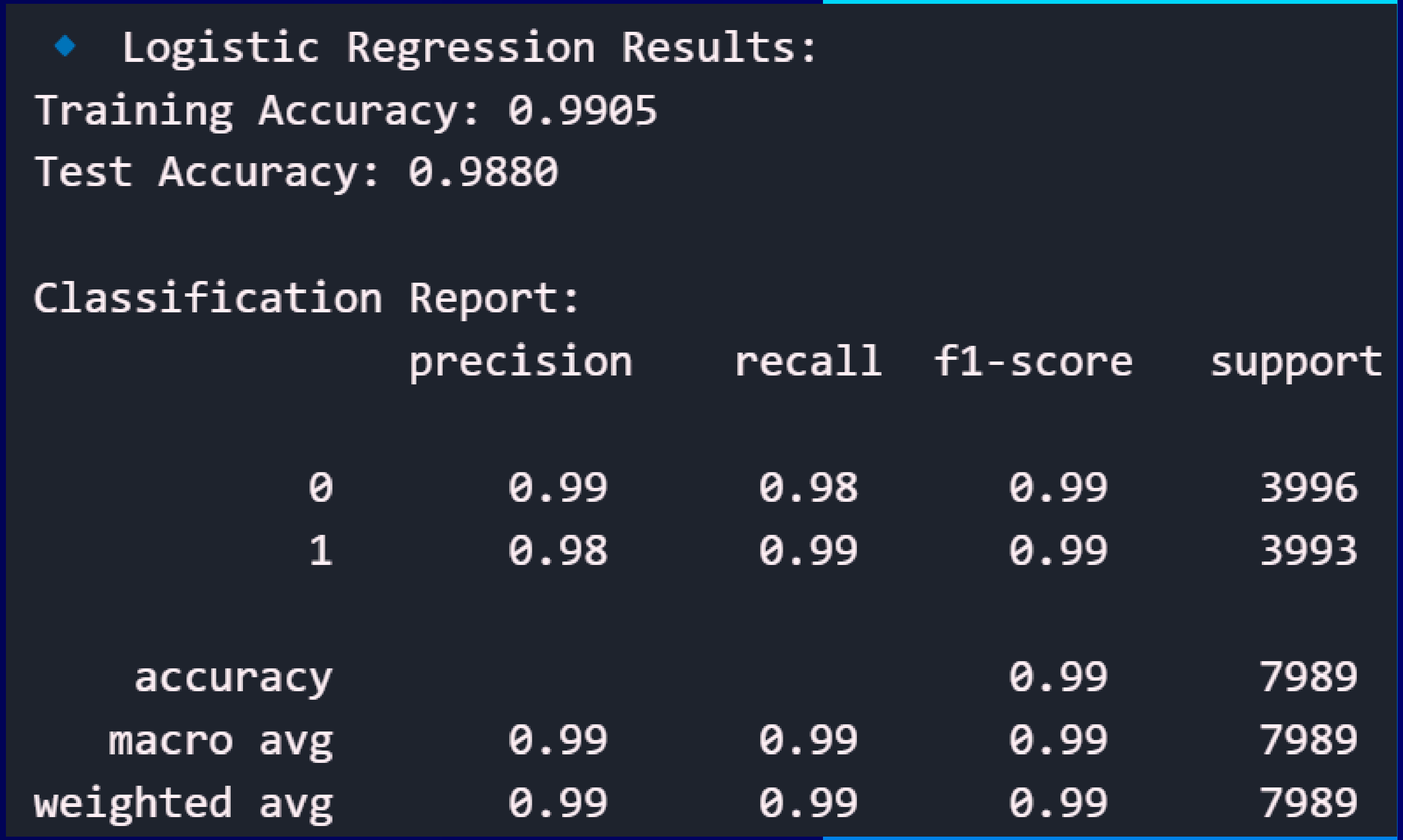




```
🔹 Multinomial Naïve Bayes Results:
Training Accuracy: 0.9475
Test Accuracy: 0.9452

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      3996
           1       0.95      0.94      0.95      3993

    accuracy                           0.95      7989
   macro avg       0.95      0.95      0.95      7989
weighted avg       0.95      0.95      0.95      7989
```

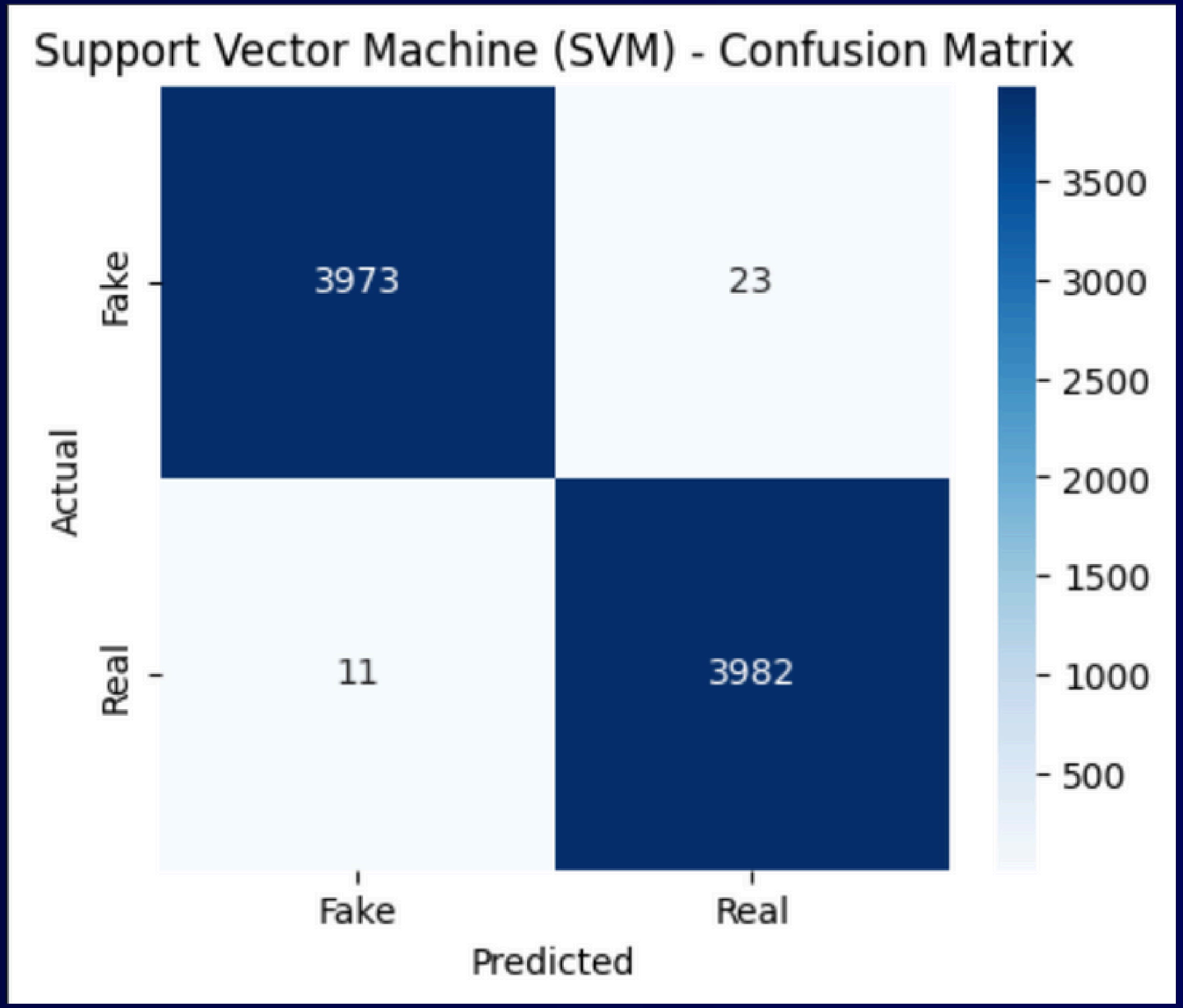
LOGISTIC REGRESSION

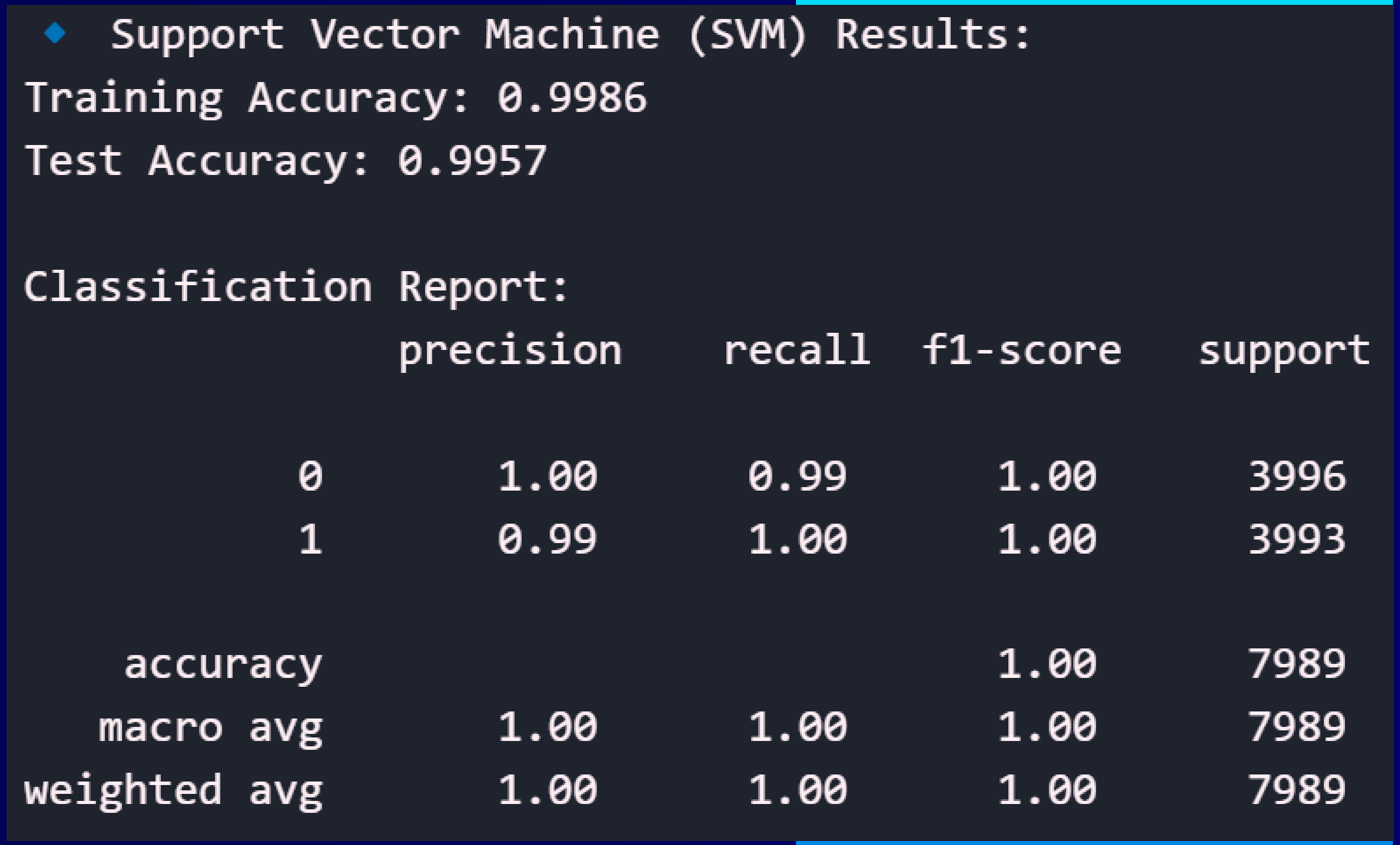




```
🔹 Logistic Regression Results:
Training Accuracy: 0.9905
Test Accuracy: 0.9880

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      3996
           1       0.98      0.99      0.99      3993

    accuracy                           0.99      7989
   macro avg       0.99      0.99      0.99      7989
weighted avg       0.99      0.99      0.99      7989
```

SUPPORT VECTOR MACHINE (SVM)





```
🔹 Support Vector Machine (SVM) Results:
Training Accuracy: 0.9986
Test Accuracy: 0.9957

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      3996
           1       0.99      1.00      1.00      3993

    accuracy                           1.00      7989
   macro avg       1.00      1.00      1.00      7989
weighted avg       1.00      1.00      1.00      7989
```

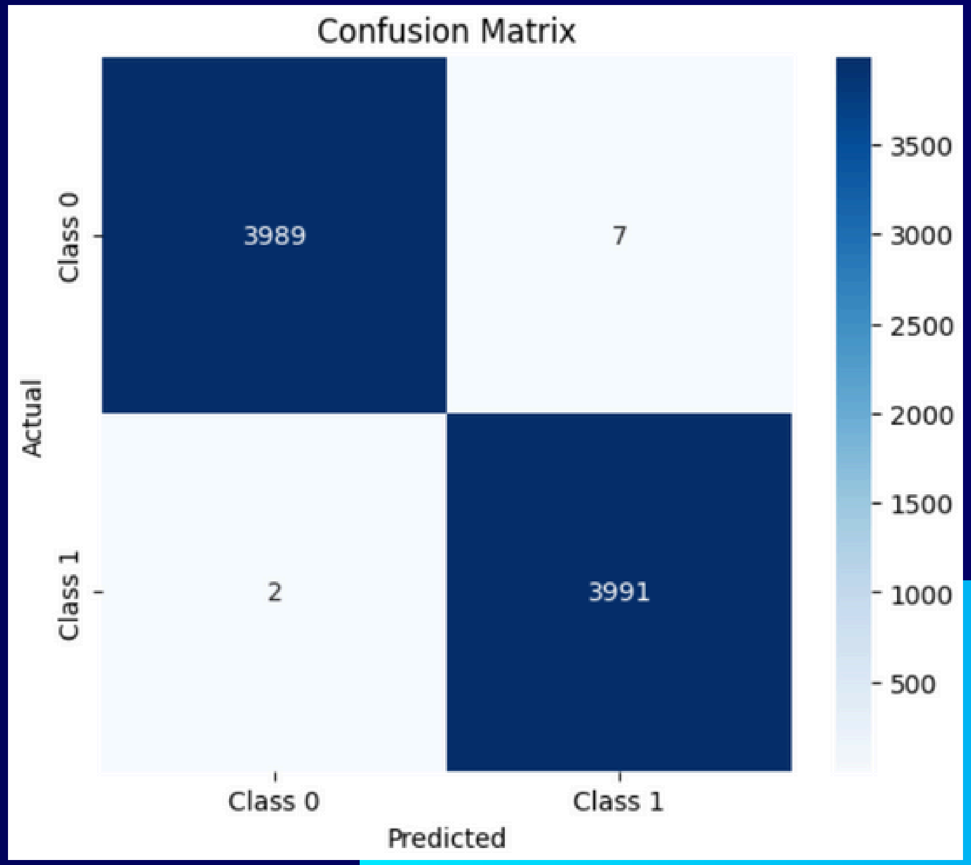
WORD2VEC-BASED WITH CONV1D

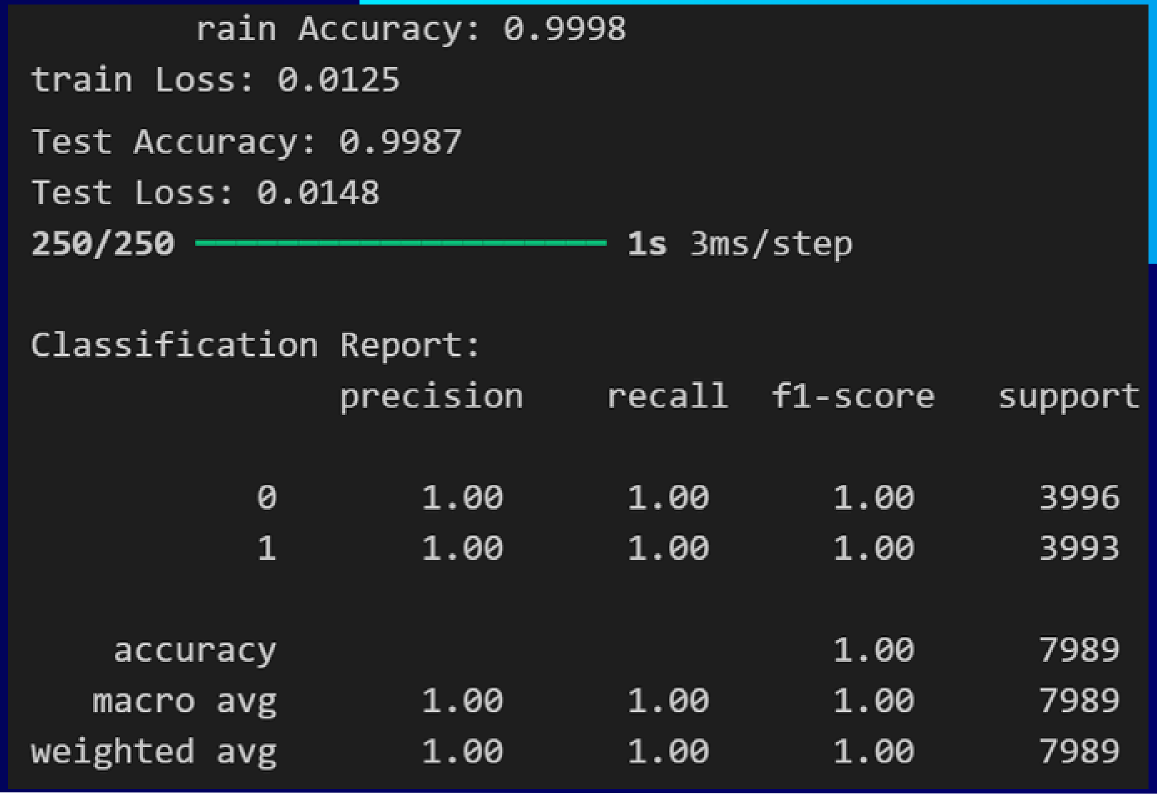
- Use Pre-trained Word2Vec: Loaded 300-dimensional embeddings from Google News.

- Tokenize & Pad Text: Converted text into sequences and applied padding.

- Create Embedding Matrix: Mapped words to their corresponding Word2Vec vectors.

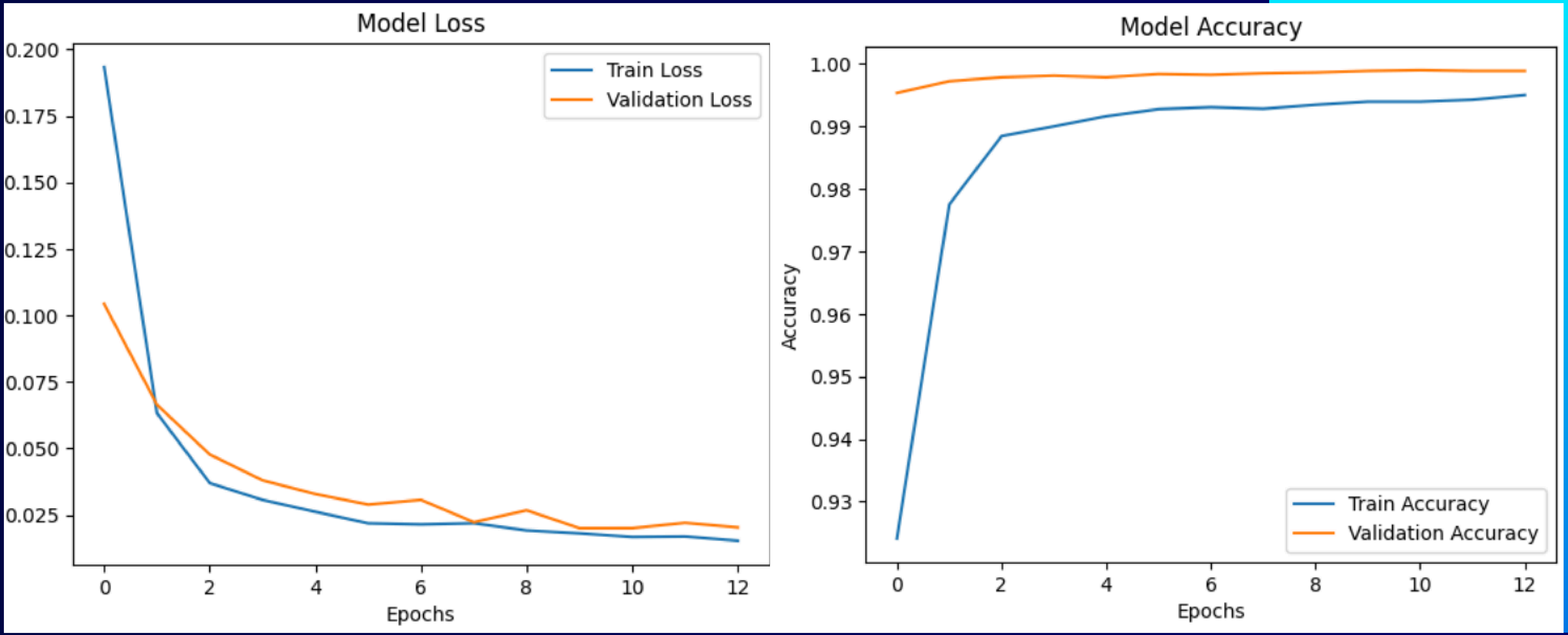
- Build Conv1D Model: Used convolutional layers with dropout and pooling for feature extraction.





```
        rain Accuracy: 0.9998
train Loss: 0.0125
Test Accuracy: 0.9987
Test Loss: 0.0148
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3996
           1       1.00      1.00      1.00      3993

    accuracy                           1.00      7989
   macro avg       1.00      1.00      1.00      7989
weighted avg       1.00      1.00      1.00      7989
```

WORD2VEC-BASED WITH CONV1D

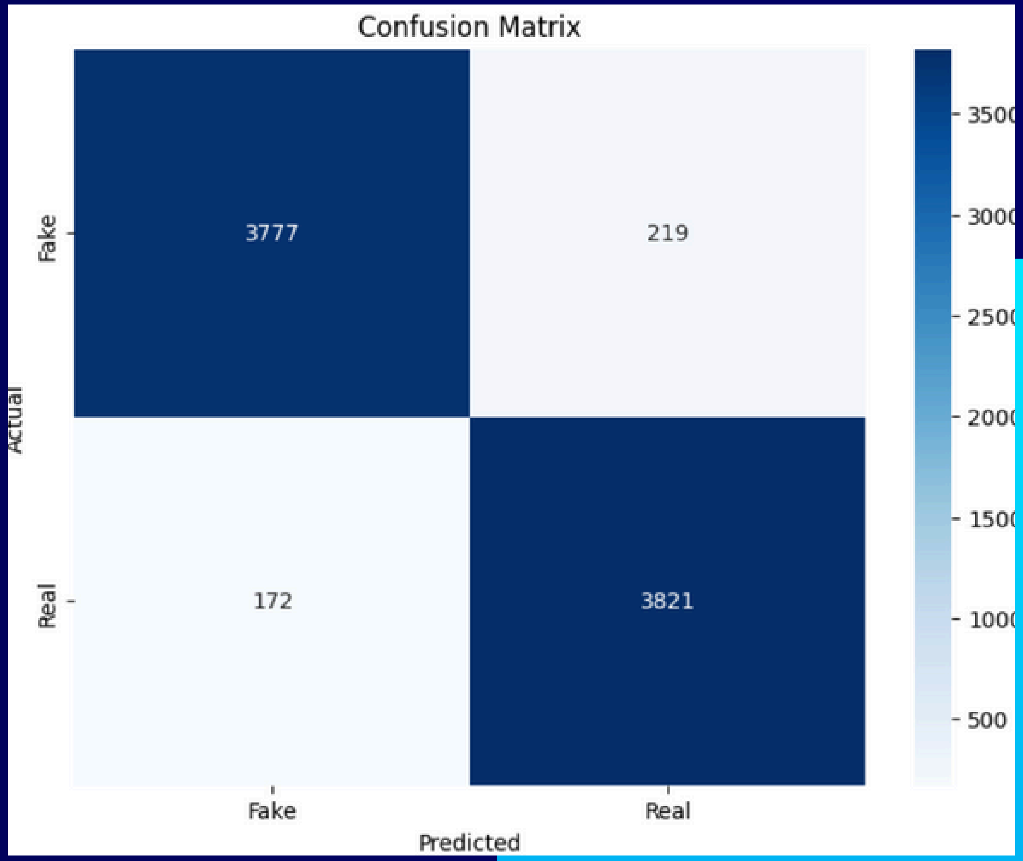


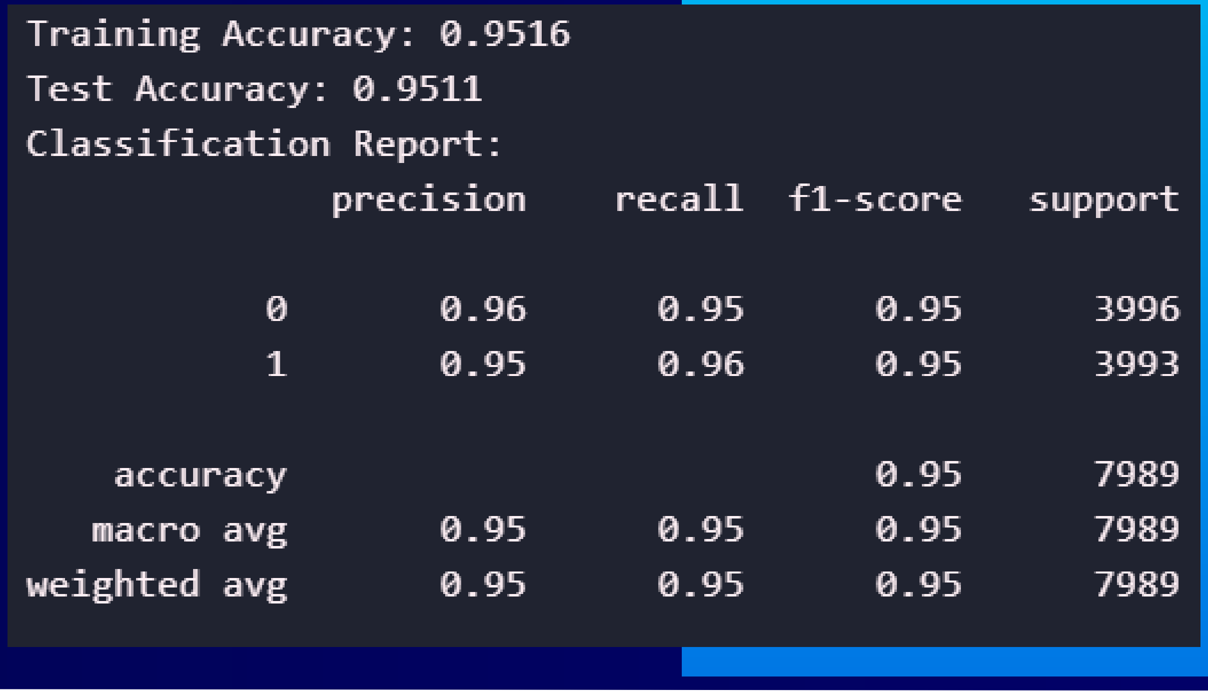
WORD2VEC WITH LOGISTIC REGRESSION

- Convert Text to Word Vectors: Used pre-trained Word2Vec to transform text into numerical vectors by averaging word embeddings.

- Preprocess the Data: Combined title and text, applied preprocessing, and converted text into Word2Vec vectors.

- Train the Model: Split data, converted it into NumPy arrays, and trained a Logistic Regression model.

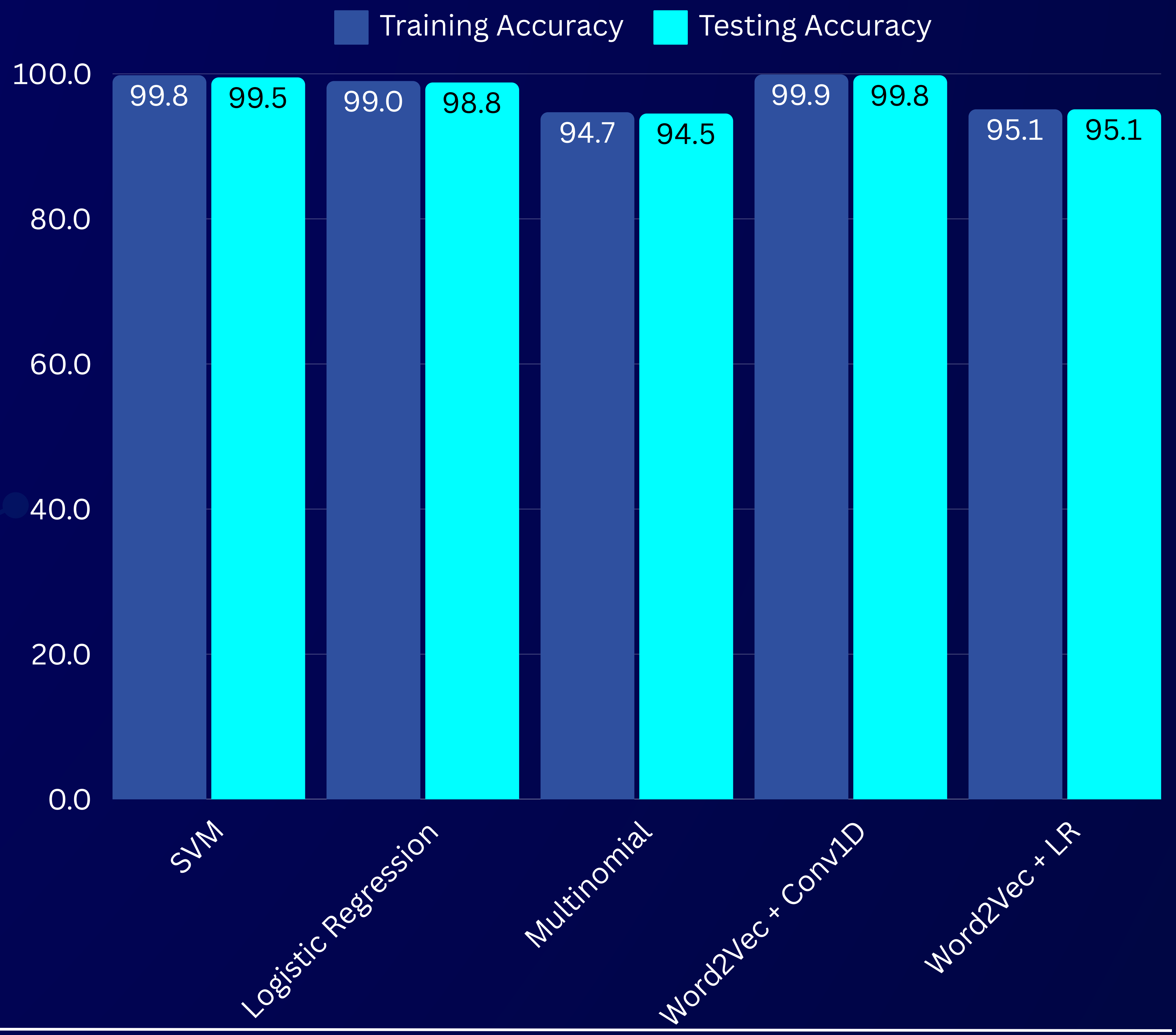




```
Training Accuracy: 0.9516
Test Accuracy: 0.9511
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.95      0.95      3996
           1       0.95      0.96      0.95      3993

    accuracy                           0.95      7989
   macro avg       0.95      0.95      0.95      7989
weighted avg       0.95      0.95      0.95      7989
```

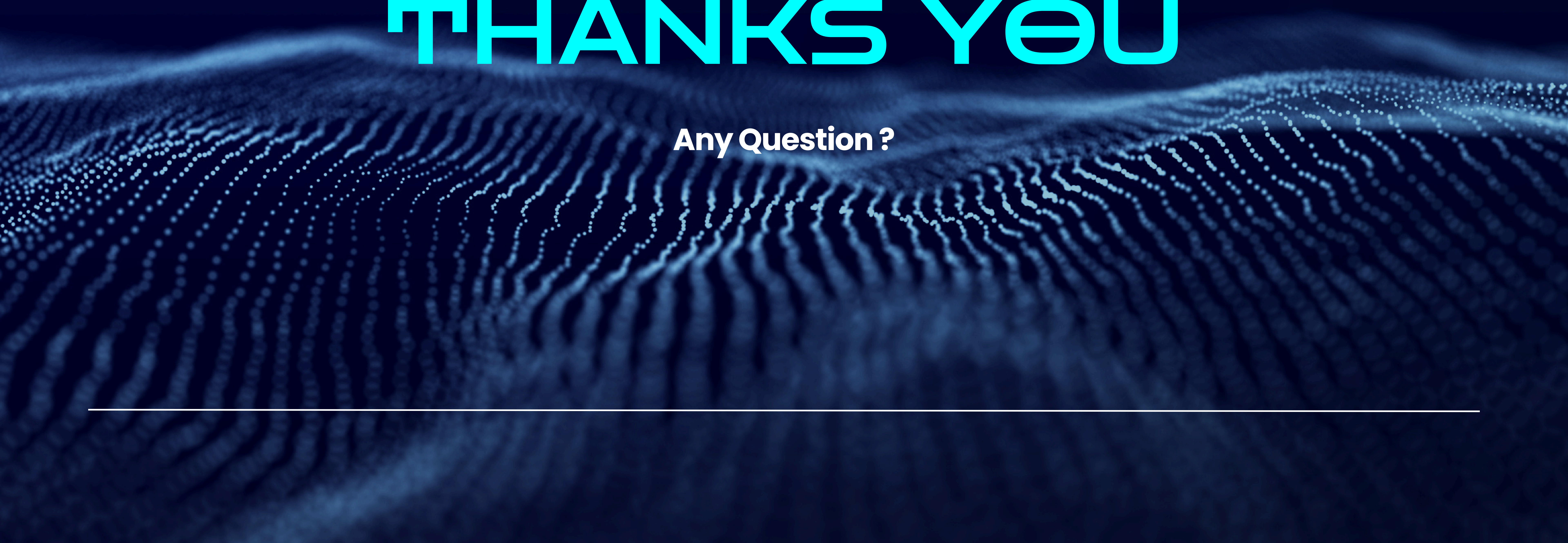
MODEL COMPARISON

- **Word2Vec + Conv1D**: Best overall performance, showing excellent generalization.
- **SVM**: Strong results, with a slight drop in testing accuracy.
- **Logistic Regression**: Reliable performance, slightly below SVM.
- **Word2Vec + Logistic Regression**: Balanced performance, with good generalization.
- **Multinomial Naïve Bayes**: Lower performance compared to the other models.



OUR STREAMLIT

🔗 fake news detection



THANKS YOU

Any Question ?