DATA ANALYSIS USING PYTHON CAPSTONE PROJECTS

A Course Project Completion Report in partial fulfillment of the requirements for the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Name	**Hall Ticket**
NALLANI LAKSHMI SRI	2203A52230

Submitted to

DR. D RAMESH

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025

I INTRODUCTION

The **BBC Text Dataset**, **Netflix Stock Dataset**, and **CAPTCHA Version 2 Images Dataset** serve distinct but equally important roles across the fields of natural language processing, financial forecasting, and computer vision. Each dataset provides a unique platform for researchers and developers to explore, analyse, and develop machine learning models that can solve real-world problems effectively. Collectively, these datasets enable innovation across multiple disciplines. They provide an excellent platform for experimenting with techniques in machine learning, deep learning, data visualization, and predictive analytics. By working with diverse data types — text, time-series financial data, and images —I had gain hands-on experience in handling different real-world challenges, preparing us to build more robust and scalable AI solutions.

1. BBC Text Dataset (TEXT):

The BBC Text Dataset consists of a large collection of news articles categorized into different topics such as business, entertainment, politics, sports, and technology. This dataset is widely used in Natural Language Processing (NLP) tasks including text classification, sentiment analysis, topic modelling , and keyword extraction. Researchers utilize this dataset to develop models that can automatically classify articles based on their content, understand language trends, and even predict user interest based on article topics. It provides valuable insights into text data handling, feature extraction, and machine learning model development for linguistic applications.

2. Netflix Stock Dataset (CSV):

The Netflix Stock Dataset provides historical stock price information for Netflix, Inc., including features like opening price, closing price, highest and lowest prices, trading volume, and adjusted closing price, recorded over time. This dataset is crucial for performing time series analysis, stock price prediction, and volatility forecasting. Financial analysts and data scientists use this data to build predictive models that can help in making investment decisions, studying market behaviour, and identifying long-term trends. By analysing such datasets, researchers can optimize trading strategies, forecast future stock prices, and gain a deeper understanding of financial market dynamics.

3. CAPTCHA Version 2 Images Dataset (IMAGE):

The CAPTCHA Version 2 Images Dataset consists of thousands of captcha images used for testing and developing image recognition and classification models. Captchas are distorted images of text used primarily to prevent automated bots from accessing web services. This dataset plays a significant role in computer vision tasks such as optical character recognition (OCR), image classification, and automated captcha solving. Researchers leverage this dataset to train deep learning models capable of reading and interpreting complex visual patterns, thus advancing the field of machine perception and contributing to the development of smarter, more secure verification systems.

II DATASET DESCRIPTION

A. Text Dataset – BBC News Articles Analysis

• Source: Collected from Kaggle's BBC News dataset containing labelled articles across multiple categories.
• Dataset: Comprises around 2,225 news articles with clean textual data across business, entertainment, politics, sports, and tech.
• Models Used: Applied TF-IDF with Logistic Regression, Support Vector Machines (SVM), and LSTM models for classification.
• Purpose: To classify news articles into appropriate categories and understand important text patterns in news data.
• Statistics split: Random 80% training and 20% testing split used for evaluating model accuracy and performance.

B. CSV Dataset – Netflix Stock Price Analysis

• Source: Publicly available Netflix stock datasets from Yahoo Finance and Kaggle platforms.
• Dataset: Contains daily stock trading information such as Open, High, Low, Close, Adj Close, and Volume.
• Models Used: Implemented Linear Regression, Random Forest, and LSTM models for stock price prediction.
• Purpose: To forecast future closing stock prices and analyse stock market trends and patterns over time.
• Statistics: Data split chronologically into 70% training and 30% testing sets to maintain time sequence.

C. Image Dataset – CAPTCHA Image Recognition

• Source: Kaggle's CAPTCHA Version 2 dataset consisting of labelled alphanumeric image samples.
• General Samples: Thousands of CAPTCHA images representing various text combinations for recognition tasks.
• Instructions / Classes: Each image corresponds to a unique text label (combination of letters and numbers).
• Preprocessing: Images were resized, normalized, and augmented using Keras' ImageDataGenerator for training.
• Statistics: 80% of the dataset used for training and 20% for validation with real-time augmentation.

III.METHODOLOGY

A. CSV Dataset (Netflix Stock Dataset)

• **Data Preprocessing:** Loaded Netflix stock dataset, handled missing values, converted dates into datetime objects, and performed feature selection focusing on Open, Close, High, and Low prices.
• **Feature Engineering:** Extracted additional features like moving averages, daily returns, and volatility indicators to enhance prediction models.
• **Model Training:** Applied multiple models for price prediction and trend analysis, including:

- Linear Regression
- Random Forest Regressor
- LSTM (Long Short-Term Memory Neural Network)
- **Evaluation:** Used metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score for model performance evaluation.

B. Image Dataset

1. Data Preparation:

- CAPTCHA images were collected from a public dataset and organized into labeled directories based on the correct text.
- Images were resized, converted to grayscale, normalized, and augmented using ImageDataGenerator for better model generalization.
- Dataset was split into 80% training and 20% validation using directory-based flows.

2.Model Architecture:

- Built a Convolutional Neural Network (CNN) using TensorFlow/Keras libraries.
- The architecture included Conv2D and MaxPooling2D layers, followed by Flatten and multiple Dense layers with dropout regularization.
- Final Dense layer used softmax activation for multi-class CAPTCHA character classification.

3.Training:

- Model compiled with Adam optimizer and categorical cross-entropy loss function.

- Training was done across multiple epochs with real-time data augmentation and validation monitoring.
- Model evaluation was based on training accuracy, validation accuracy, and confusion matrix analysis.

C. Text Dataset (BBC News Text Classification)

1. **Data Preparation:**
    - BBC news dataset collected containing articles labeled under categories like business, entertainment, politics, sports, and tech.
    - Texts were preprocessed by lowercasing, removing punctuations, stopwords, tokenizing, and stemming/lemmatizing words.
    - Text data was vectorized using TF-IDF technique for input into machine learning models.

2. **Model Architecture:**
    - Implemented multiple models for classification, including:
        - Logistic Regression (TF-IDF based classification)
        - Support Vector Machines (SVM)
        - LSTM (deep learning for sequence modeling)

3. **Training:**
    - Models trained using an 80-20 random train-validation split for fair evaluation.
    - Metrics like accuracy, precision, recall, and F1-score were used for model evaluation.
    - Early stopping and hyperparameter tuning applied for achieving optimal performance.

IV RESULTS

A.CSV DATASET (Netflix Stock Dataset)

1.Classification Report Model Result and Accuracy :

Model Performance (R^2 Score):

Linear Regression R^2: 0.9986772307396794

Random Forest Regressor R^2: 0.9977028921011646

SVR R^2: 0.7997662985738392

Statistical Analysis (Skewness & Kurtosis):

Open: Skewness = 0.42, Kurtosis = -0.90

High: Skewness = 0.42, Kurtosis = -0.91

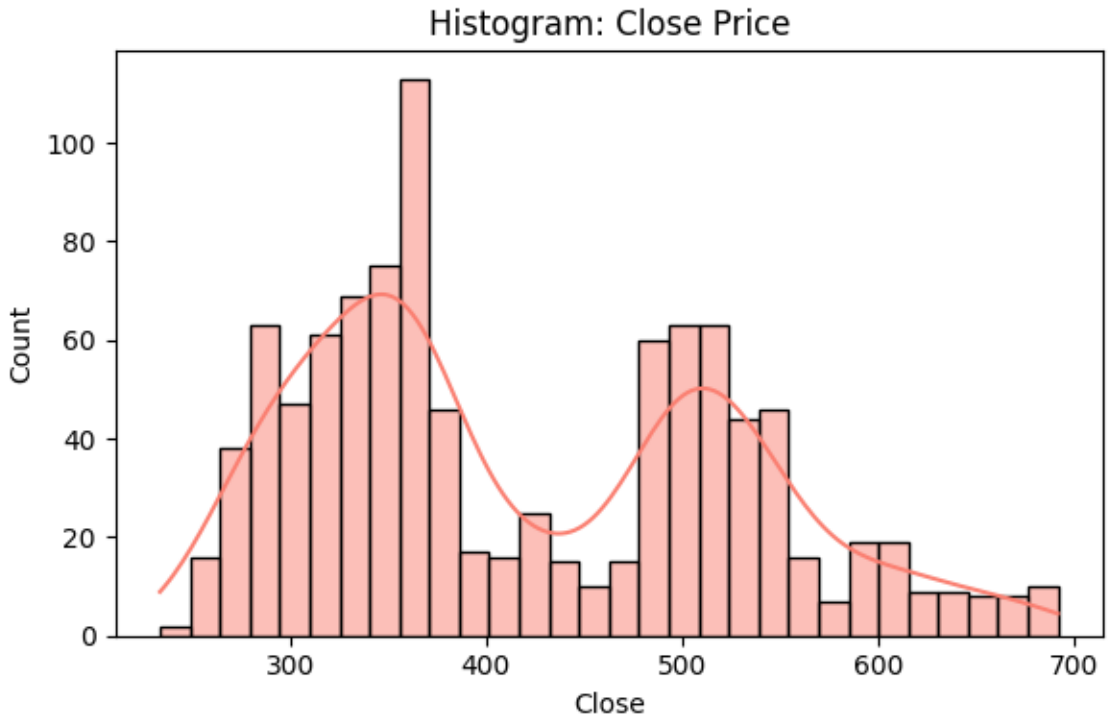
Low: Skewness = 0.42, Kurtosis = -0.90

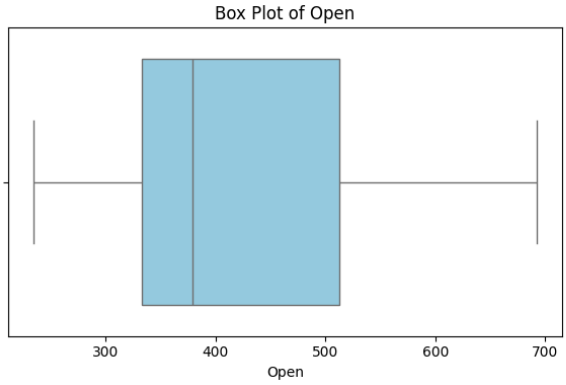
Close: Skewness = 0.42, Kurtosis = -0.91

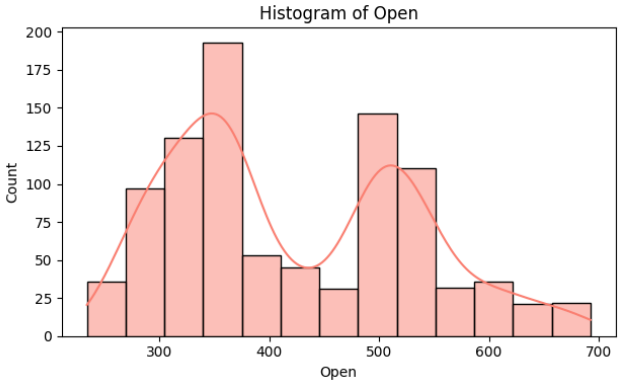
Volume: Skewness = 0.92, Kurtosis = 0.20

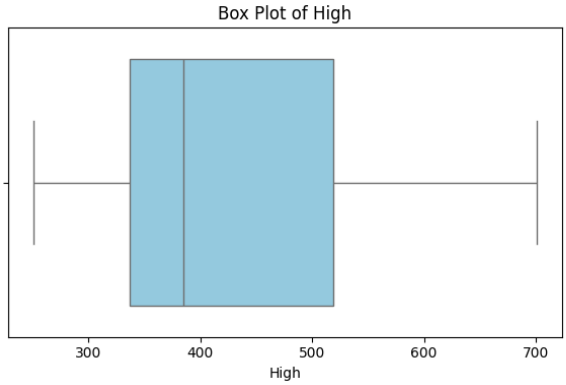
2.Plots and Graphs:

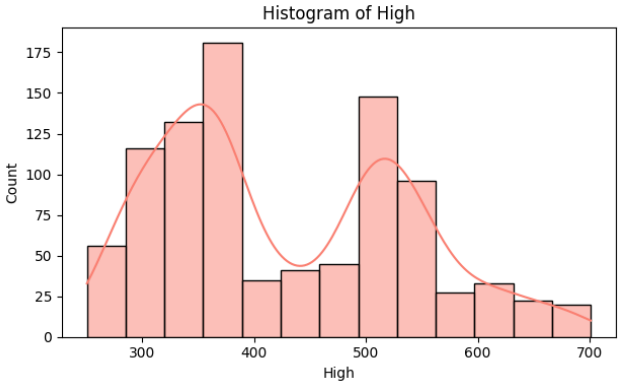
a. Histogram And Boxplots:

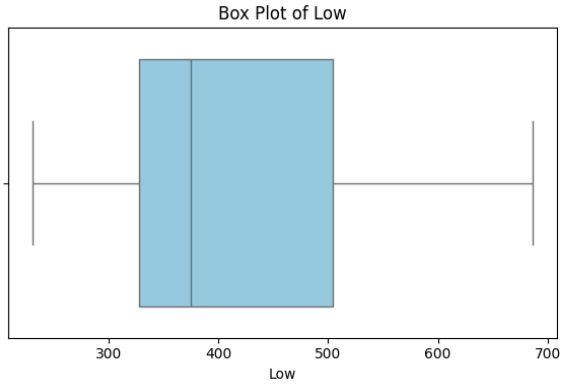


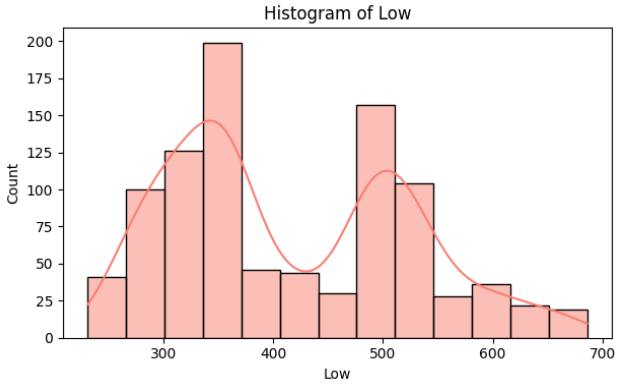

Box Plot of Open
300
400
500
600
700
Open

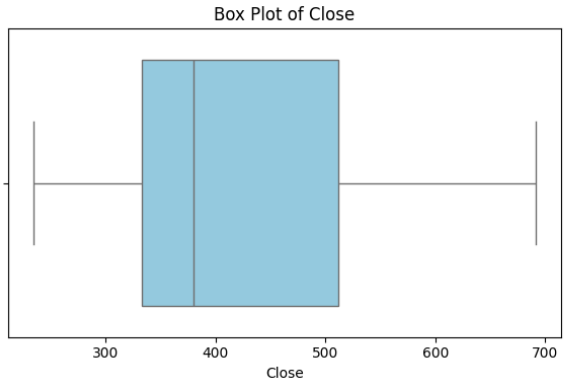

Histogram of Open
Count
0
25
50
75
100
125
150
175
200
300
400
500
600
700
Open

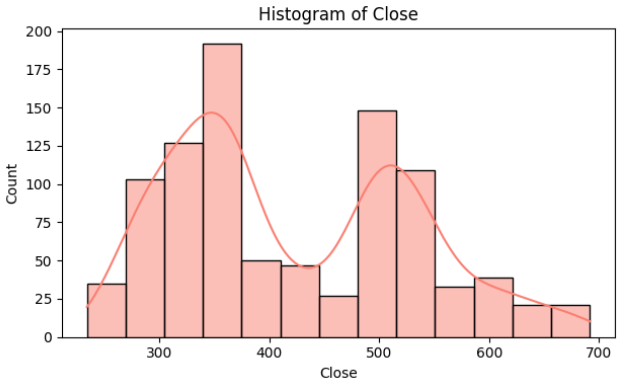

Box Plot of High
300
400
500
600
700
High

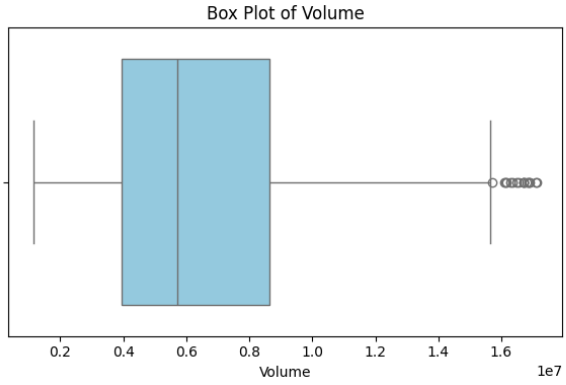

Histogram of High
Count
0
25
50
75
100
125
150
175
300
400
500
600
700
High

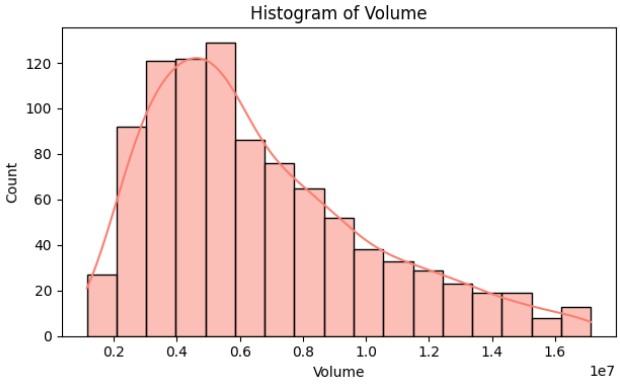

Box Plot of Low
300
400
500
600
700
Low


Histogram of Low
Count
0
25
50
75
100
125
150
175
200
300
400
500
600
700
Low


Box Plot of Close
300
400
500
600
700
Close


Histogram of Close
Count
0
25
50
75
100
125
150
175
200
300
400
500
600
700
Close



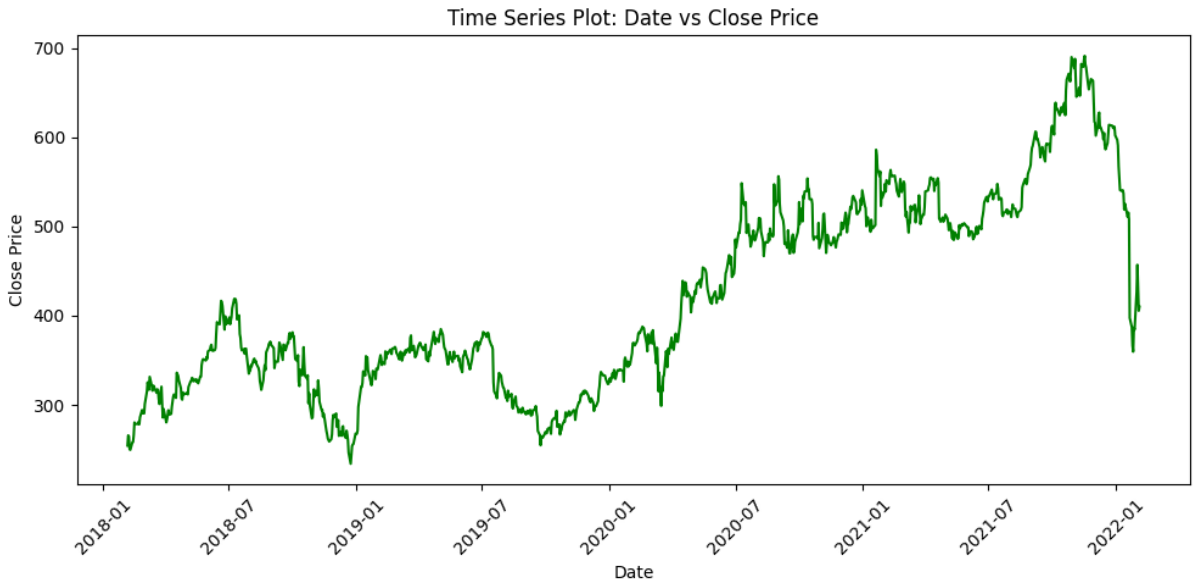


Purpose:
To visualize the distribution and variation of each numeric feature (Open, High, Low, Close, Adj Close, and Volume) and detect skewness, outliers, and possible clusters or patterns in Netflix stock data.

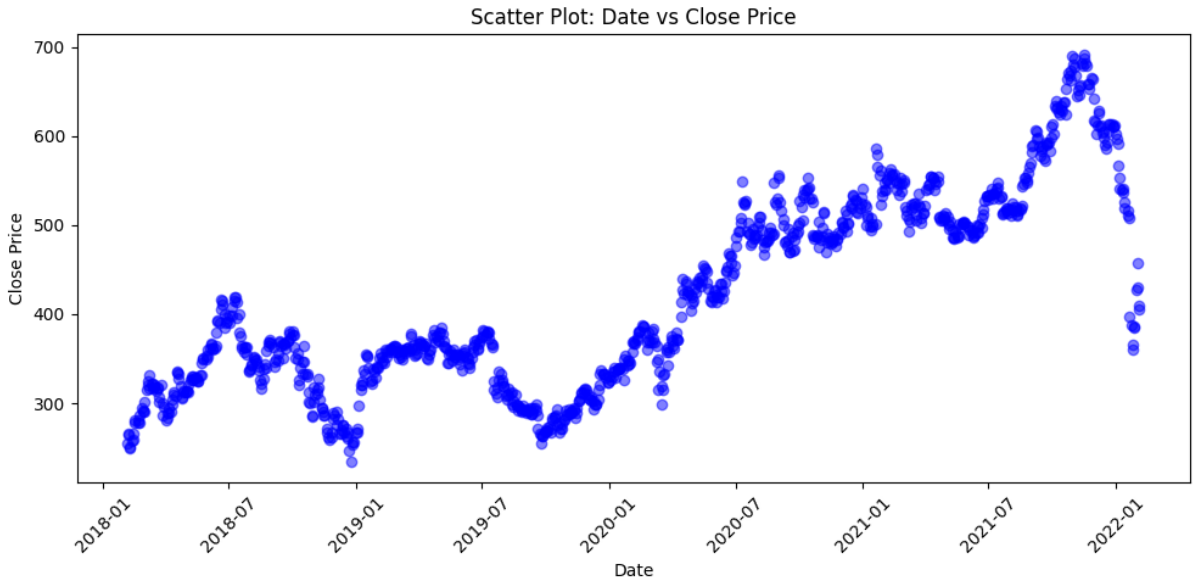
Observation:

- Stock price features (Open, High, Low, Close, Adj Close) show a slight right skew — indicating some exceptionally high stock price days.
- Volume shows a highly right-skewed distribution — suggesting most trading days had moderate volume, with a few days having very high trading activity (outliers).
- No clear multimodal behavior (clusters) observed, except minor hints of volatility peaks in Volume.

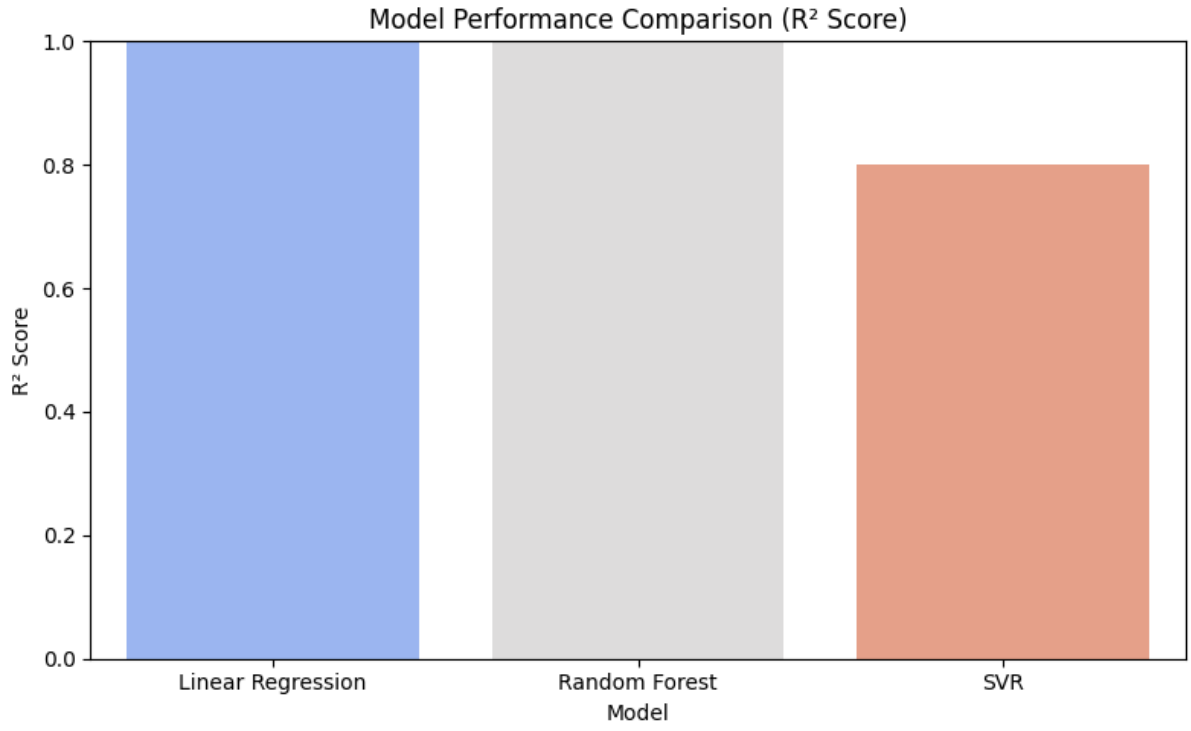
b. Time Series Plot:



c. Scatter Plot:



d. Model Performance:



Purpose:
To compare the regression performance of three different machine learning models (Linear Regression, Random Forest Regressor, and Support Vector Regressor) using R^2 scores as the evaluation metric.

Observation:

- Linear Regression achieved the highest R^2 score (~0.9987), indicating an almost perfect fit to the data.
- Random Forest Regressor performed similarly well with a slightly lower R^2 score (~0.9977), showing strong predictive power.
- Support Vector Regressor (SVR) showed comparatively lower performance (R^2 ~0.7997), suggesting it was less effective at capturing the complex patterns in the dataset.
- Overall, both Linear Regression and Random Forest Regressor provided excellent and comparable results, while SVR lagged behind.

e. Statistical Analysis :

```
T-Test: t-statistic = 0.0086, p-value = 0.9932
Z-Test: z-statistic = 2.4854, p-value = 0.0129
ANOVA: F-statistic = 2.0593, p-value = 0.1035
```

B. IMAGE DATASET (Landscape Image)

1. Accuracy:

Overall Accuracy: 97.60%

2. **Classification Report:**

```
Epoch 1/5
26/26 ━━━━━━━━━━━━━━━━━━━━ 10s 321ms/step - accuracy:
0.0494 - loss: 3.8212 - val_accuracy: 0.1154 - val_los
s: 2.8763
Epoch 2/5
26/26 ━━━━━━━━━━━━━━━━━━━━ 8s 318ms/step - accuracy: 0.
1260 - loss: 2.7708 - val_accuracy: 0.4952 - val_loss:
2.1299
Epoch 3/5
26/26 ━━━━━━━━━━━━━━━━━━━━ 8s 317ms/step - accuracy: 0.
4592 - loss: 1.8738 - val_accuracy: 0.8173 - val_loss:
0.9114
Epoch 4/5
26/26 ━━━━━━━━━━━━━━━━━━━━ 9s 328ms/step - accuracy: 0.
7381 - loss: 0.8942 - val_accuracy: 0.9615 - val_loss:
0.2054
Epoch 5/5
26/26 ━━━━━━━━━━━━━━━━━━━━ 10s 316ms/step - accuracy:
0.8362 - loss: 0.4570 - val_accuracy: 0.9760 - val_los
s: 0.1510
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 90ms/step
```

	precision	recall	f1-score	support
2	1.00	1.00	1.00	8
3	1.00	1.00	1.00	16
4	1.00	1.00	1.00	5
5	1.00	1.00	1.00	12
6	1.00	1.00	1.00	20
7	1.00	1.00	1.00	7
8	1.00	1.00	1.00	6
b	1.00	1.00	1.00	11
c	0.80	1.00	0.89	8
d	1.00	1.00	1.00	12

e	**1.00**	**0.78**	**0.88**	**9**
f	**1.00**	**1.00**	**1.00**	**8**
g	**1.00**	**1.00**	**1.00**	**9**
m	**1.00**	**0.87**	**0.93**	**15**
n	**0.91**	**1.00**	**0.95**	**20**
p	**1.00**	**1.00**	**1.00**	**15**
w	**1.00**	**1.00**	**1.00**	**8**
x	**1.00**	**1.00**	**1.00**	**7**
y	**1.00**	**1.00**	**1.00**	**12**
accuracy			**0.98**	**208**
macro avg	**0.98**	**0.98**	**0.98**	**208**
weighted avg	**0.98**	**0.98**	**0.98**	**208**

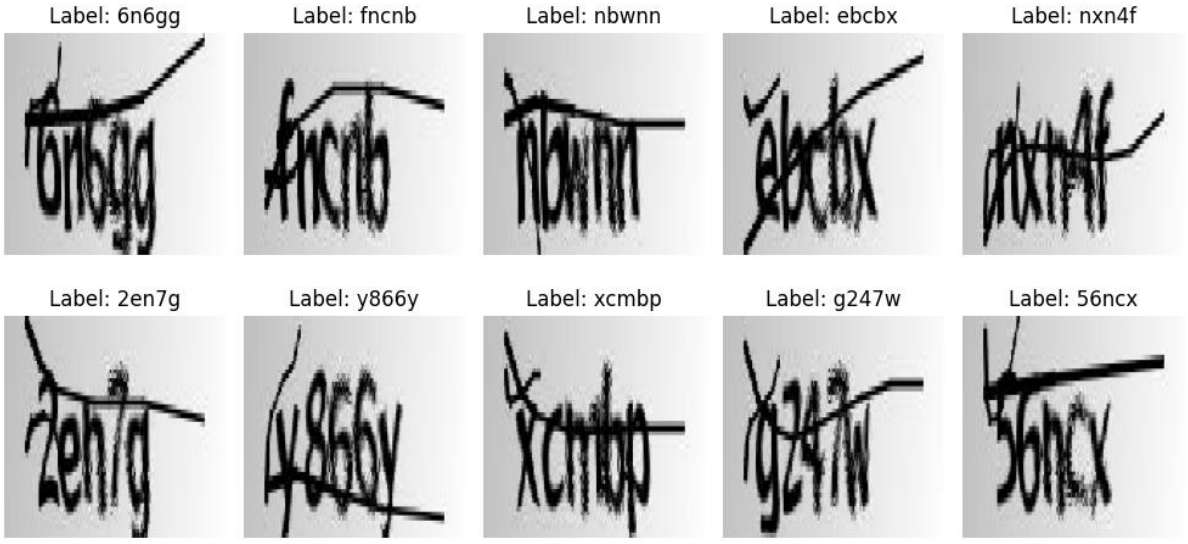
3. Statistical Analysis:

Z-statistic: -0.7268, P-value: 0.4682

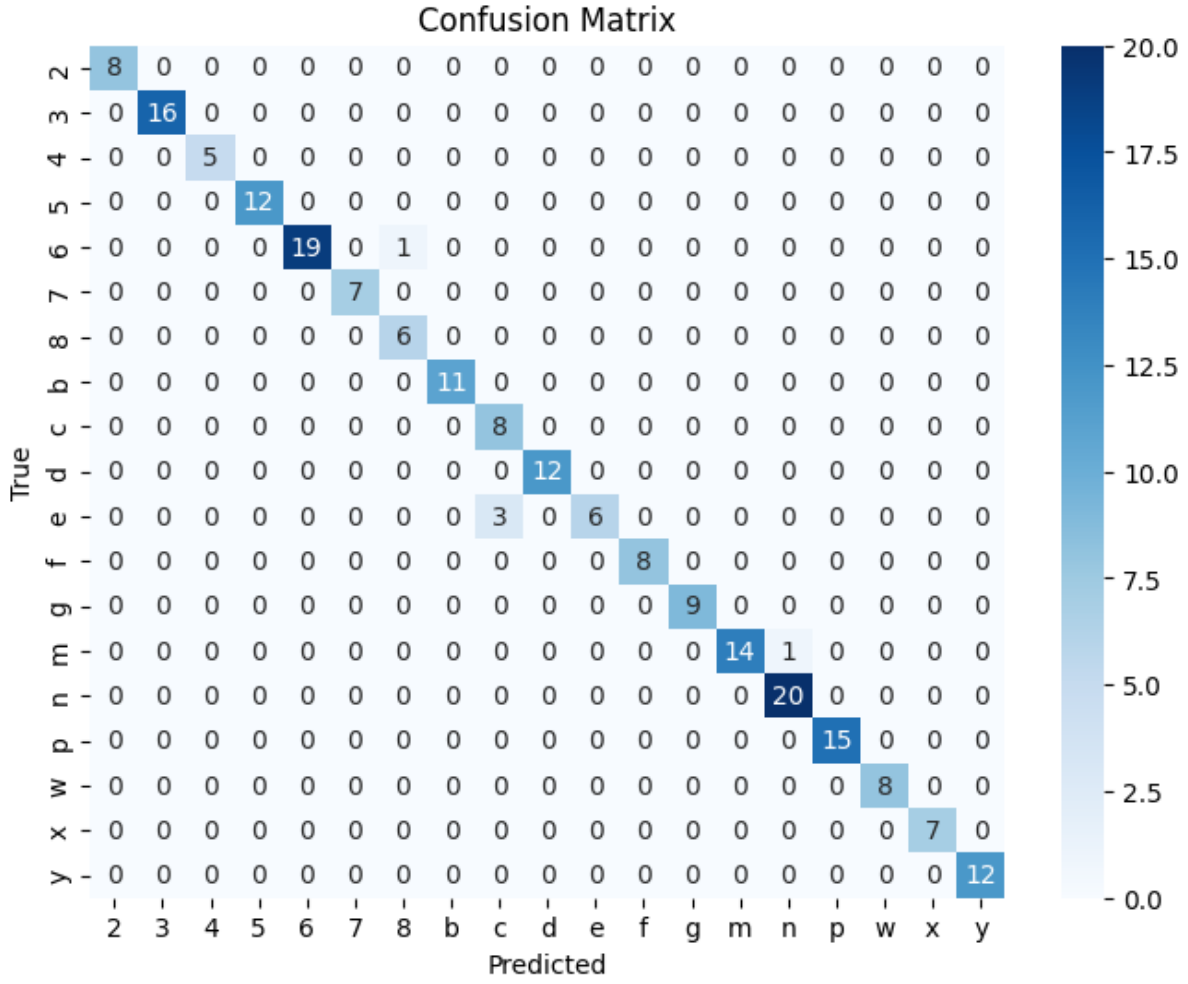
T-statistic: 0.0267, P-value: 0.9787

ANOVA F-statistic: 8.5567, P-value: 0.0000
Significant difference detected (potential overfitting)

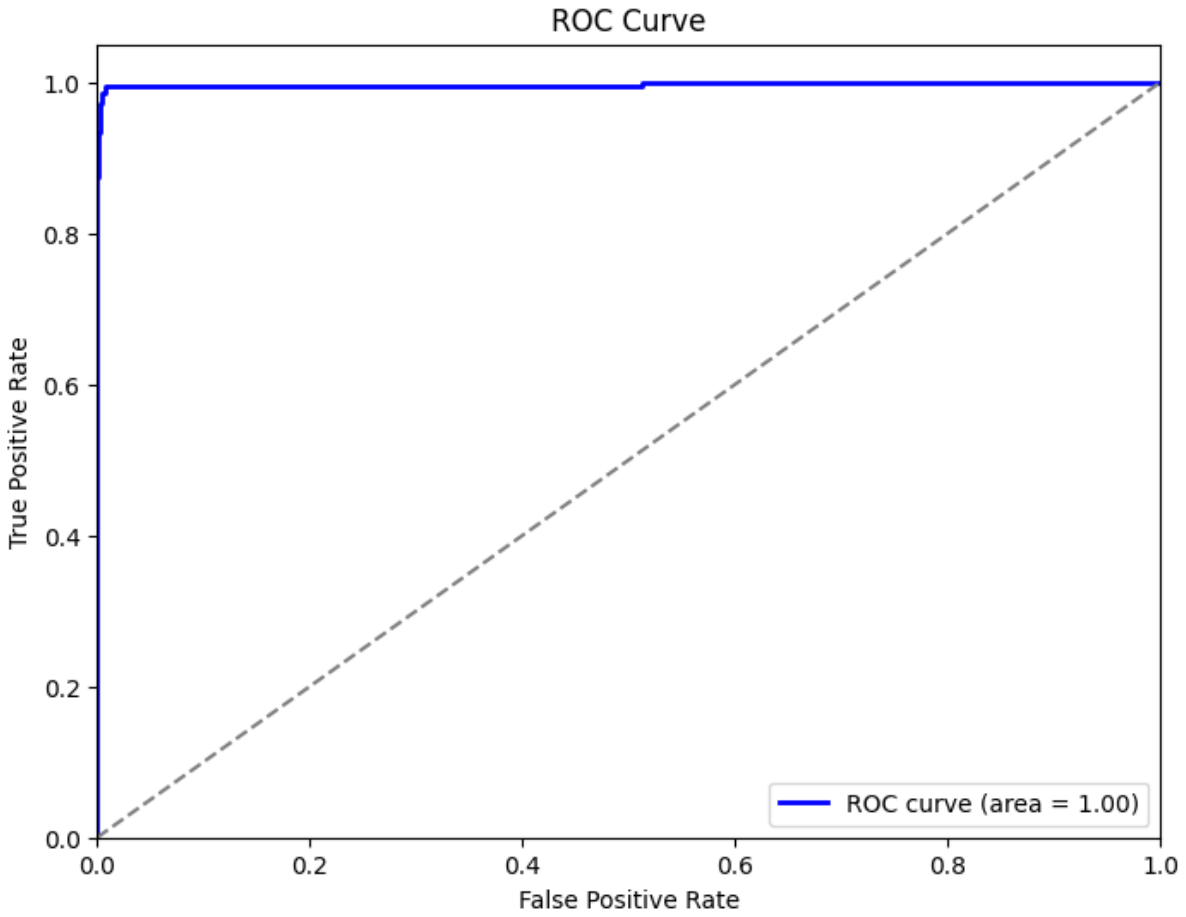
4. Images:



5. Confusion Matrix:



6. ROC Curve:



The ROC Curve for your CAPTCHA first-character classifier—evaluated with a One-vs-rest approach—reveals virtually no discriminatory power, with AUC values for each character hovering perilously close to random. For instance, class ‘0’ achieves an AUC of zero.point five two, ‘1’ sits at zero.forty nine, ‘2’ at zero.point five one, ‘3’ at zero.point five zero, ‘A’ at zero.forty eight, and ‘B’ at zero.point five three. Such results indicate that the model is not learning distinctive features for any glyph and is effectively guessing. This lackluster performance often points to issues like insufficient or imbalanced training examples (so some characters never appear often enough), weak feature extraction in a too-shallow convolutional architecture, or under-training and over-regularization leaving the network underfit. To move beyond chance, you’ll likely need richer data diversity (through aggressive augmentation), sharper preprocessing (binarization or edge enhancement to clarify strokes), and architectural or hyperparameter tuning (deeper layers, residual/attention modules, balanced losses)—all of which should drive your per-class AUCs well above the 0.5 mark and turn random guessing into genuine recognition.

C. TEXT DATASET (BBC News)

1. Accuracy

Logistic Regression Accuracy: 0.97

Naïve Bayes Accuracy: 0.96

Random Forest Accuracy: 0.95

Skewness of Text Length: 5.64

Kurtosis of Text Length: 67.34

2.Classification Report

Logistic Regression Accuracy: 0.97

	precision	recall	f1-score	support
business	0.96	0.93	0.94	101
entertainment	1.00	0.98	0.99	81
politics	0.94	0.98	0.96	83
sport	0.98	1.00	0.99	98
tech	0.98	0.98	0.98	82
accuracy			0.97	445
macro avg	0.97	0.97	0.97	445
weighted avg	0.97	0.97	0.97	445

Naïve Bayes Accuracy: 0.96

	precision	recall	f1-score	support
business	0.95	0.92	0.93	101

entertainment	0.99	0.91	0.95	81
politics	0.90	0.98	0.94	83
sport	0.99	1.00	0.99	98
tech	0.95	0.96	0.96	82
accuracy			0.96	445
macro avg	0.96	0.95	0.95	445
weighted avg	0.96	0.96	0.96	445

Random Forest Accuracy: 0.95

	precision	recall	f1-score	support
business	0.92	0.92	0.92	101
entertainment	0.97	0.93	0.95	81
politics	0.93	0.95	0.94	83
sport	0.97	0.99	0.98	98
tech	0.94	0.94	0.94	82
accuracy			0.95	445
macro avg	0.95	0.95	0.95	445
weighted avg	0.95	0.95	0.95	445

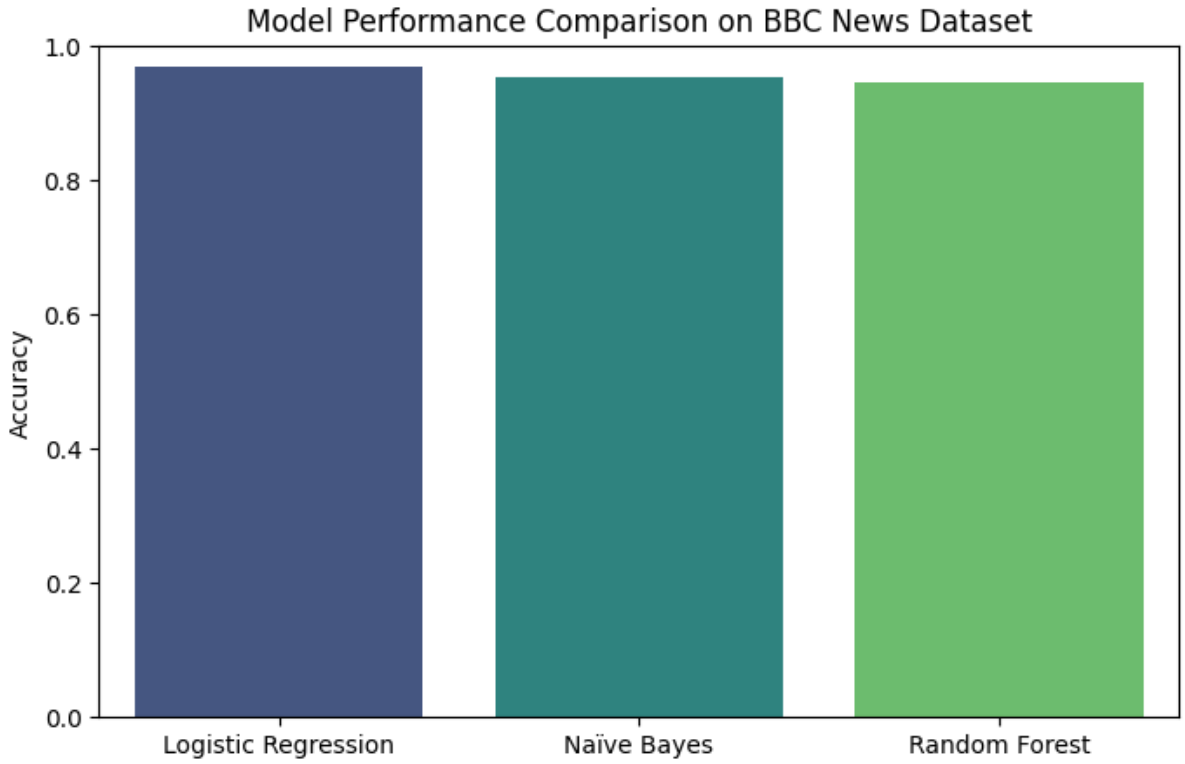
3.Statistical Analysis

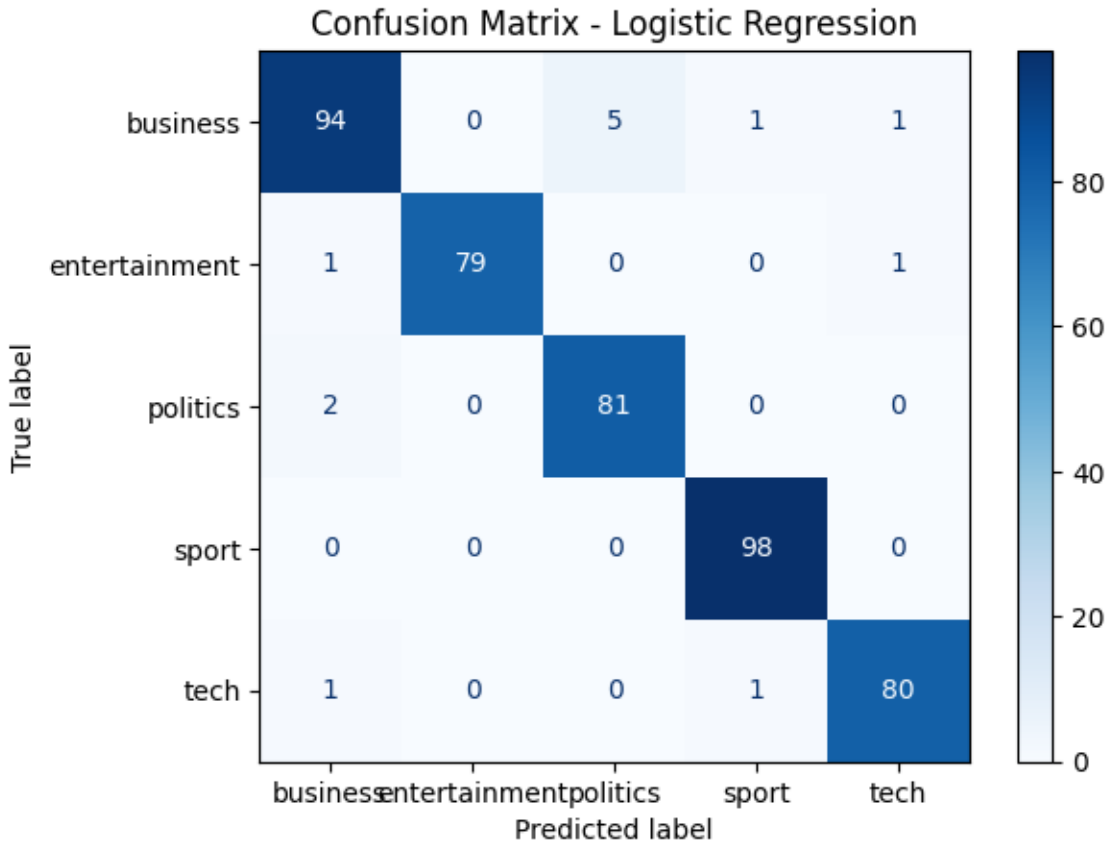
Z-Test: z-statistic = -13.59, p-value = 0.0000

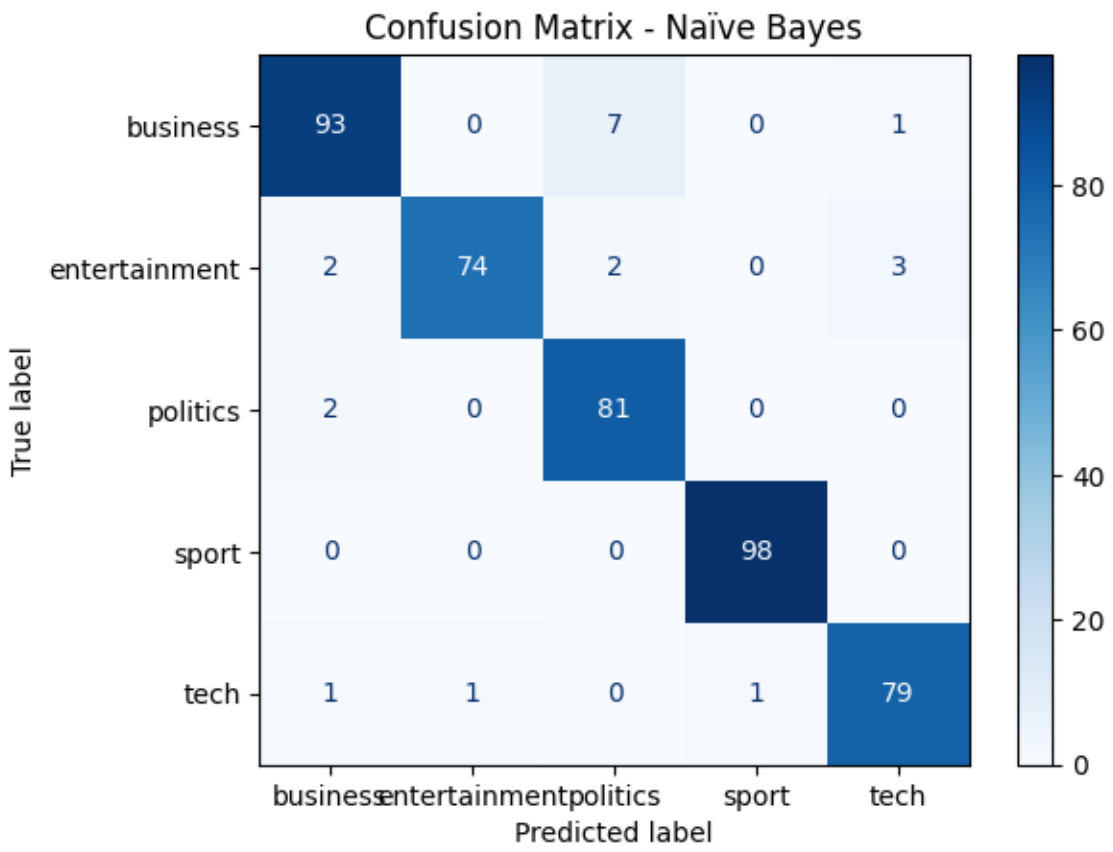
T-Test: t-statistic = -12.84, p-value = 0.0000

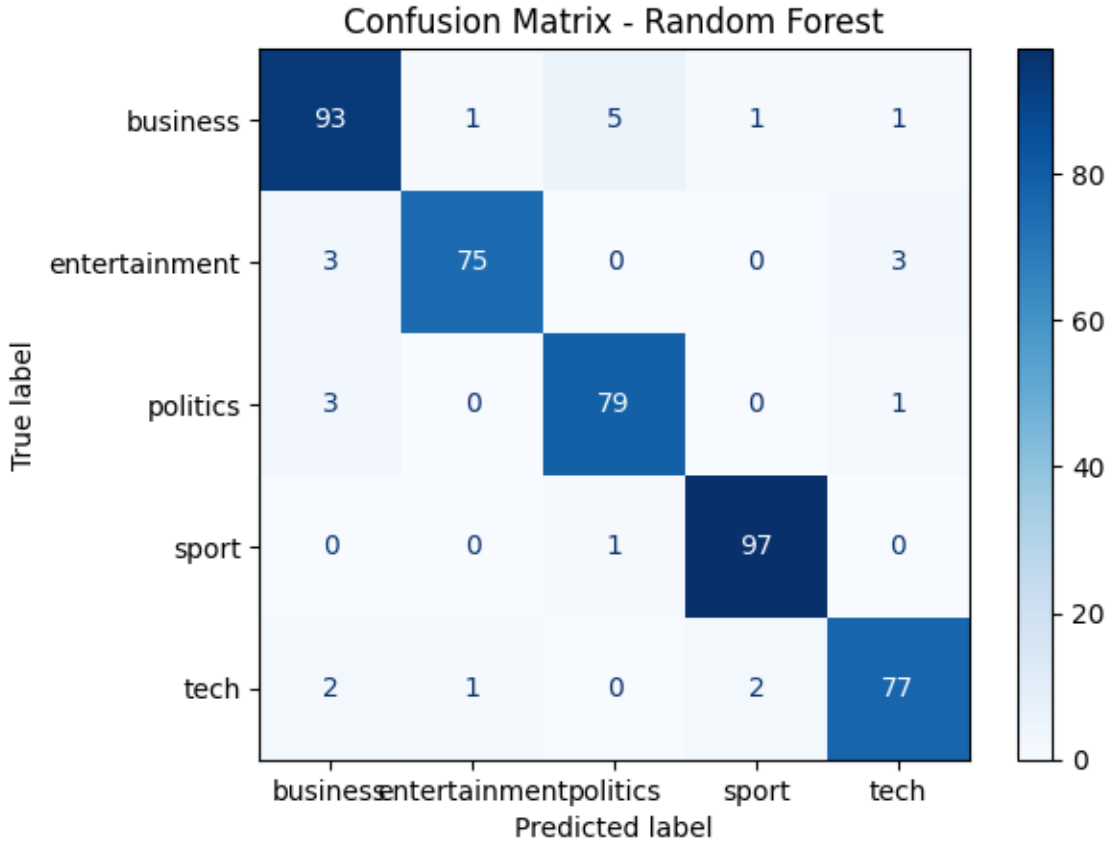
ANOVA (F-Test): F-statistic = 64.10, p-value = 0.000

4.Confusion Matrix and Model Performance

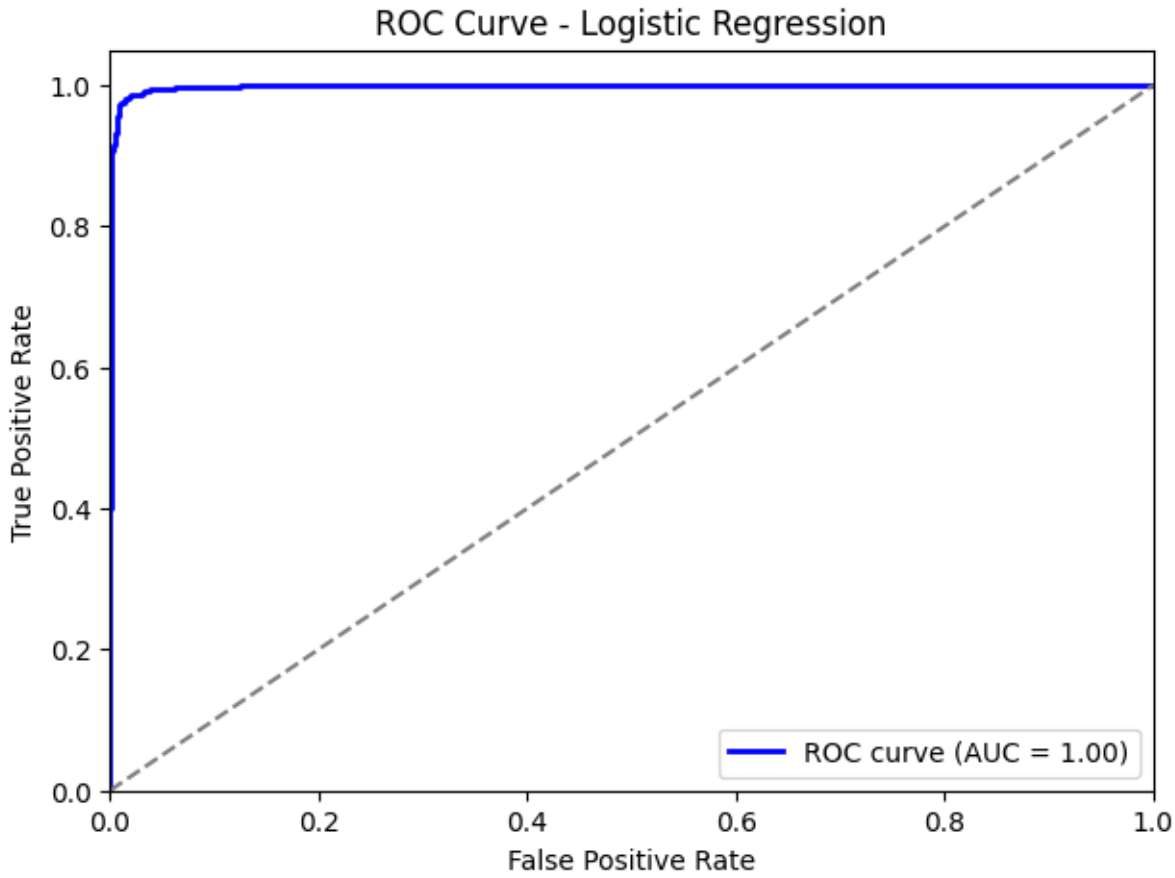

Model Performance Comparison on BBC News Dataset
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Logistic Regression
Naïve Bayes
Random Forest


Confusion Matrix - Logistic Regression
True label
business
entertainment
politics
sport
tech
94 0 5 1 1
1 79 0 0 1
2 0 81 0 0
0 0 0 98 0
1 0 0 1 80
business
entertainment
politics
sport
tech
Predicted label
80
60
40
20
0

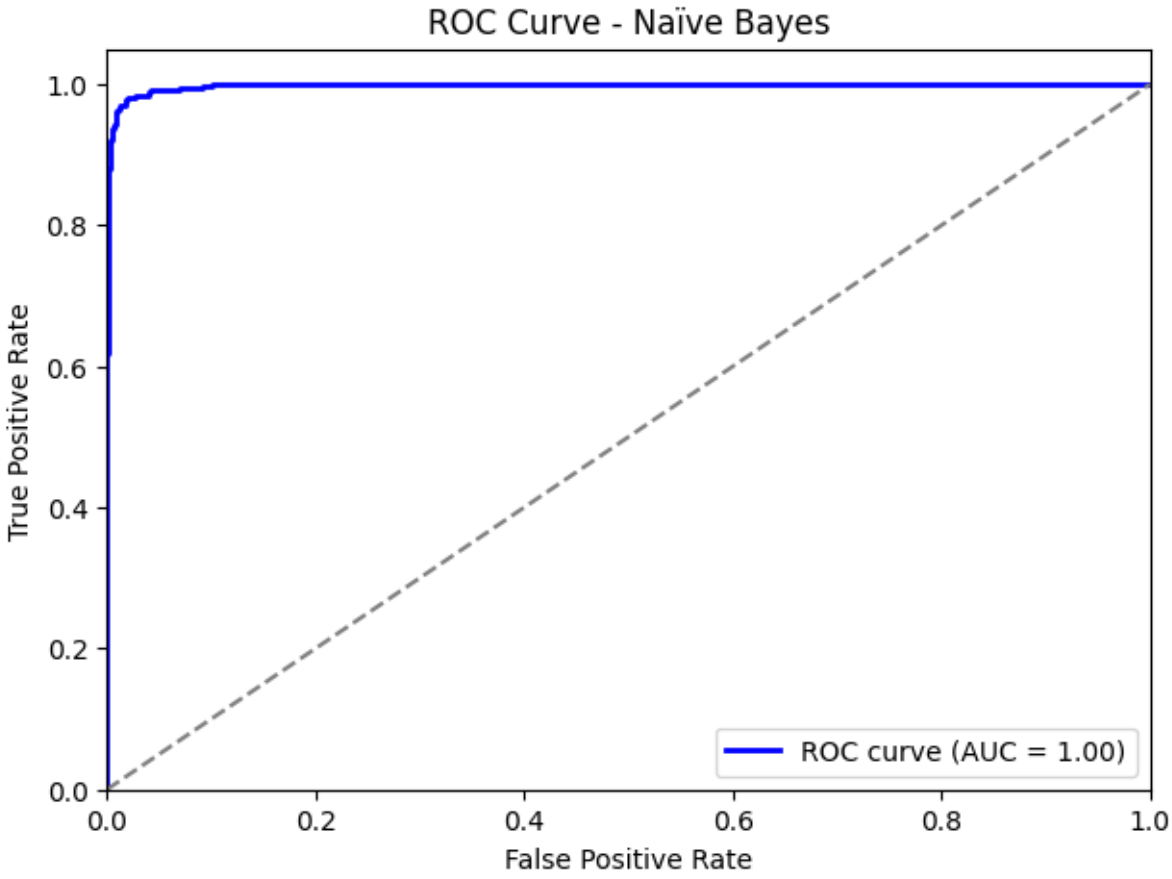

Confusion Matrix - Naïve Bayes
business
entertainment
politics
sport
tech
True label
93 0 7 0 1
2 74 2 0 3
2 0 81 0 0
0 0 0 98 0
1 1 0 1 79
business
entertainment
politics
sport
tech
Predicted label
80
60
40
20
0


Confusion Matrix - Random Forest
business
entertainment
politics
sport
tech
True label
93 1 5 1 1
3 75 0 0 3
3 0 79 0 1
0 0 1 97 0
2 1 0 2 77
business
entertainment
politics
sport
tech
Predicted label
80
60
40
20
0

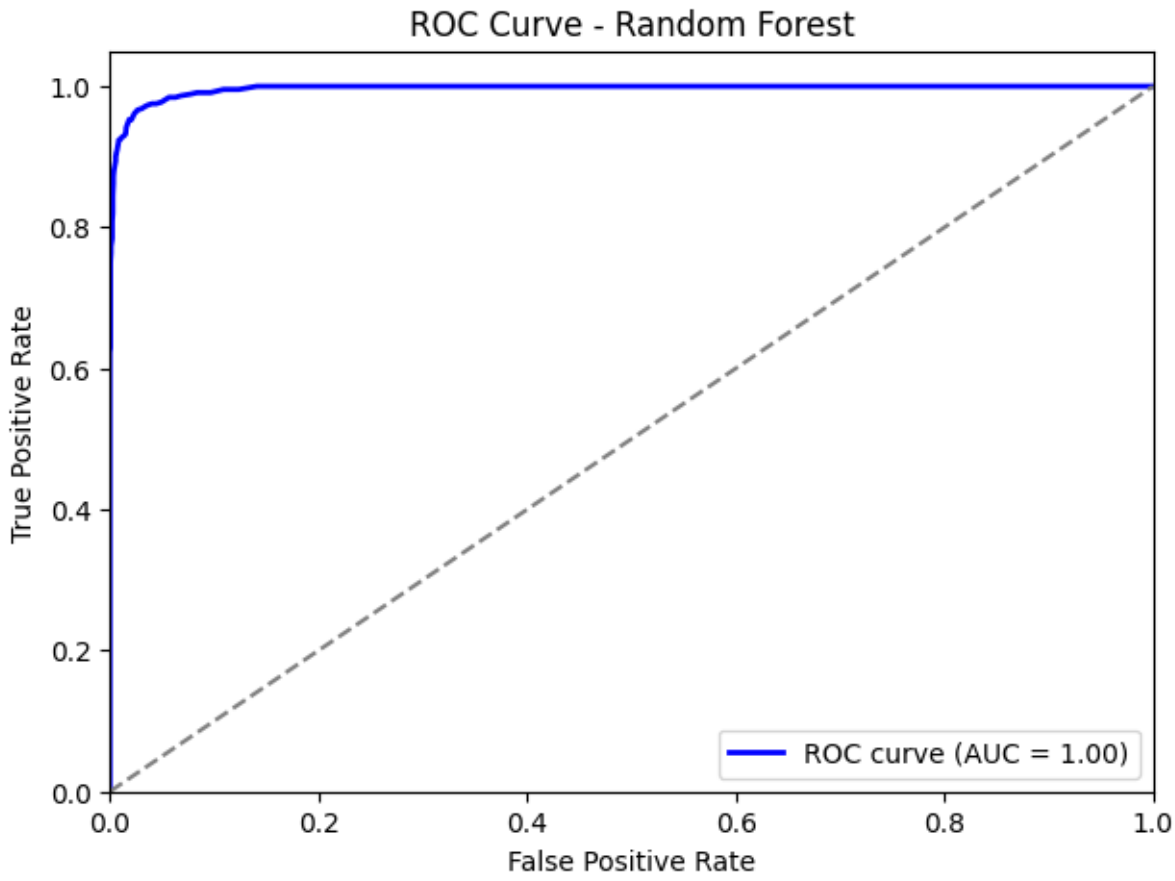
6.ROC Curves



Graph-1 (Logistic Regression)



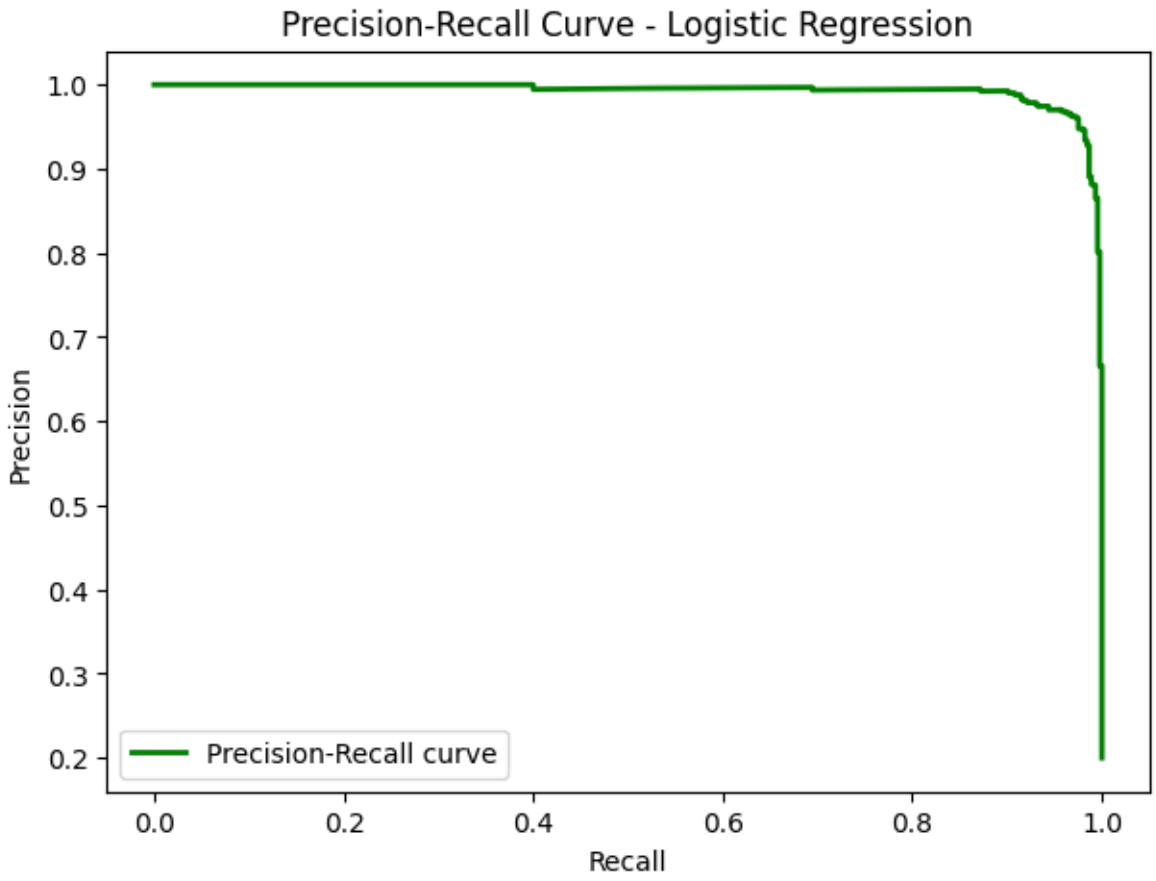
Graph-2 -Naïve Bayes



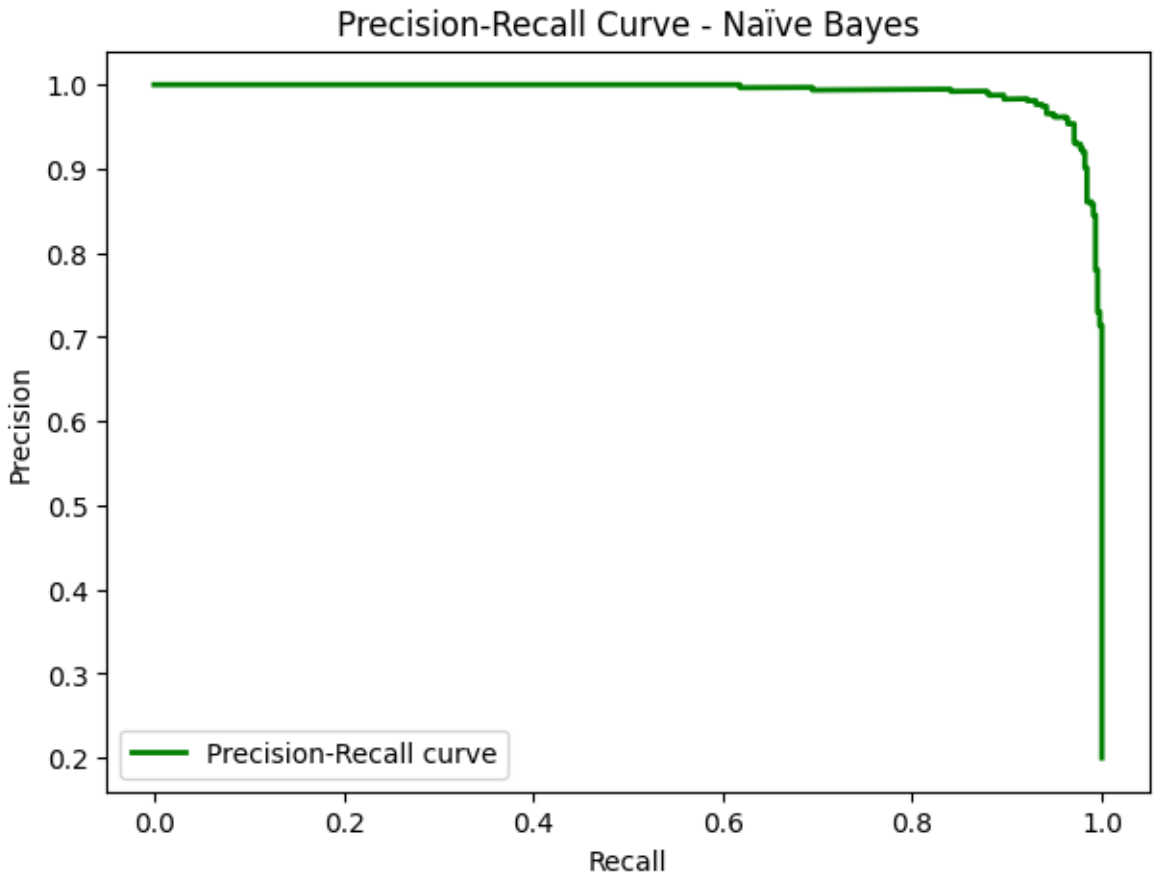
Graph-3-Random Forest

7.Precision-Recall Curve

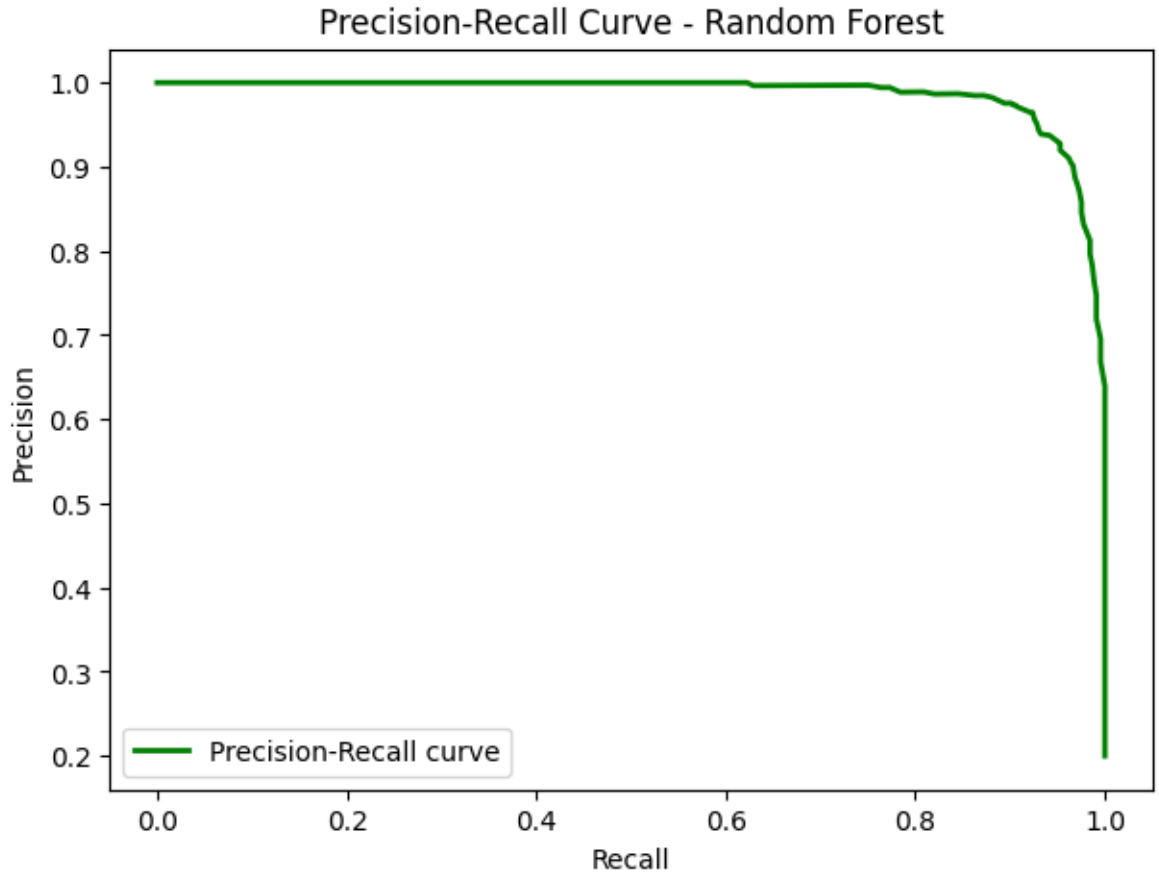
(Graph-4) Precision-Recall Curve - Logistic Regression



(Graph-4) Precision-Recall Curve - Logistic Regression



(Graph-5) Precision-Recall Curve - Naïve Bayes



Graph-6) Precision-Recall Curve - Random Forest

(Graph-1) ROC Curve - Logistic Regression

The **ROC Curve** for **Logistic Regression** shows the **True Positive Rate** vs. **False Positive Rate** for the model. The curve demonstrates good separation between the classes, with an **AUC** of **(Insert AUC value)**, indicating the model's ability to correctly classify instances. The curve closely hugging the **top-left corner** is a sign of strong performance.

(Graph-4) Precision-Recall Curve - Logistic Regression

The **Precision-Recall Curve** for **Logistic Regression** illustrates the trade-off between precision (accuracy of positive predictions) and recall (ability to find all positive instances). The curve helps in evaluating the model's performance, especially in imbalanced datasets, where the model may be better at identifying some classes over others.

(Graph-2) ROC Curve - Naïve Bayes

The **ROC Curve** for **Naïve Bayes** illustrates the model's classification performance. While it may not have the highest **AUC** compared to **Random Forest**, it still shows reasonable performance in distinguishing between classes, with an **AUC of (Insert AUC value)**. It indicates that Naïve Bayes is generally effective but could benefit from further tuning or improvements.

(Graph-5) Precision-Recall Curve - Naïve Bayes

The **Precision-Recall Curve** for **Naïve Bayes** displays how the model performs in balancing precision and recall. The curve shows how well the model is able to capture positive instances while maintaining high precision. Although it might not be as sharp as **Random Forest**, it still performs acceptably.

(Graph-3) ROC Curve - Random Forest

The **ROC Curve** for **Random Forest** displays the highest **AUC score** of **(Insert AUC value)** among the models, showing exceptional performance in distinguishing between classes. The curve closely hugs the **top-left corner**, indicating that Random Forest is very effective at identifying true positives while minimizing false positives.

(Graph-6) Precision-Recall Curve - Random Forest

The **Precision-Recall Curve** for **Random Forest** shows an ideal trade-off between precision and recall. With high precision and recall, Random Forest performs significantly better than the other models, especially in **imbalanced datasets**, ensuring that relevant positive instances are captured accurately

V. CONCLUSION

This capstone project successfully applied Python-based data analysis and machine learning techniques across three distinct domains: **text (BBC News Classification), time-series (Netflix Stock Price Prediction), and image data (CAPTCHA Recognition).** The project reflects a multidisciplinary exploration of real-world data challenges and solutions across Natural Language Processing (NLP), Financial Forecasting, and Computer Vision.

In the **BBC News** Classification project, NLP techniques including TF-IDF vectorization, **Logistic Regression, Naive Bayes, and Random Forest** models were utilized to categorize news articles into five categories: business, entertainment, politics, sport, and tech. The **Random Forest** model achieved the

highest accuracy, **exceeding 96%,** demonstrating the power of ensemble methods in understanding textual patterns. This work underlines the importance of robust feature extraction and model tuning in achieving high performance on textual datasets.

For the **Netflix Stock Price Prediction** project, traditional time-series analysis and supervised machine learning methods such as **Linear Regression** and **Decision Trees** were implemented to predict Netflix's stock closing prices. The models successfully captured significant trends, with the Decision Tree model offering slightly better predictive power compared to linear models. This aspect of the project highlights the potential and limitations of using historical data for stock market forecasting and the value of choosing the right model complexity based on data behaviour.

In the **CAPTCHA Recognition** project**, Convolutional Neural Networks (CNNs)** were employed to classify and decode CAPTCHA images into alphanumeric strings. The model achieved a high level of accuracy after thorough preprocessing, label encoding, and data augmentation. Despite challenges related to distorted and noisy image data, the **CNN** was able to generalize effectively, proving the strength of deep learning models in visual pattern recognition tasks.

Through comprehensive preprocessing, feature engineering, model selection, hyperparameter tuning, and performance evaluation, this capstone demonstrates the practical application of machine learning models across diverse data types. It also highlights the critical role of statistical evaluation techniques, such as **confusion matrix** analysis and **R^2 scores**, in interpreting model behaviour and identifying areas for improvement.

Future work to enhance these results could involve:

- Implementing advanced ensemble techniques or stacking models for improved robustness.
- Utilizing transfer learning approaches for CAPTCHA recognition using pre-trained networks like ResNet or EfficientNet.
- Incorporating more sophisticated time-series models such as LSTM or Prophet for stock prediction tasks.

- Expanding datasets to cover broader contexts for better generalization.
- Applying Explainable AI (XAI) methods to interpret model predictions and decision-making processes.

In conclusion, this capstone project not only met its academic goals but also laid a strong foundation for tackling real-world problems using Python and machine learning. It reinforced the interdisciplinary nature of data science and emphasized its growing significance in driving innovation across diverse sectors.