SR UNIVERSITY

Project Report on Course

DATA ANALYSIS USING PYTHON (21CS120)

Bachelor of Technology
In

Computer Science & Artificial Intelligence

By

Name: VAMSHIKRISHNA GATTU **Roll Number: 2203A52127**

Under the Guidance of

Dr. DADI RAMESH
Asst. Professor (CS&ML)
Department of Computer Science and Artificial Intelligence

COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL
April, 2025.

sru

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE OF COMPLETION

This is to certify that **VAMSHIKRISHNA GATTU** bearing Hall Ticket Number **2203A52127**, a student of **CSE-AIML, 3rd Year - 2nd Semester**, has successfully completed the **Data Analysis Using Python** Course and has submitted the following 3 projects as part of the curriculum:

Project Submissions:

- **CSV** Project**: Bitcoin Prices** Dataset
- **IMAGE** Project**: Indian Food Image Classification** Dataset
- **TEXT** Project**: Twitter Fake News dataset**

Dr. Dadi Ramesh
Asst. Professor (CSE-AIML)
SR University, Ananthasagar,
Warangal

Date of Completion: 25/04/2025

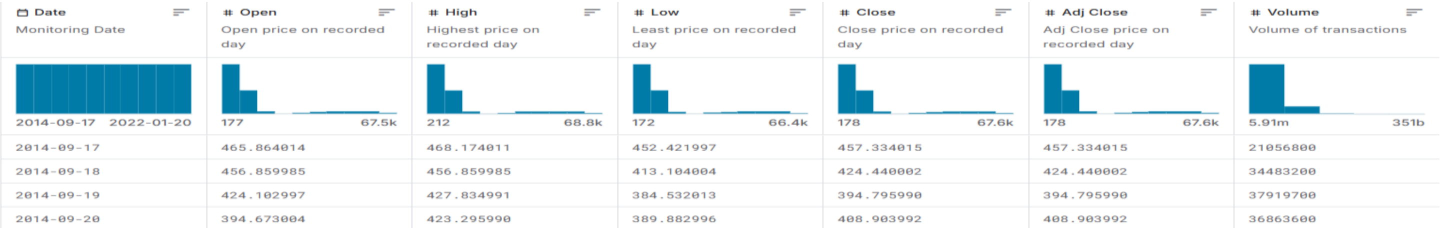
1) CSV PROJECT: BITCOIN PRICES DATASET

Description:

The project involved analysing the Bitcoin historical price dataset, which contains daily price and volume data of Bitcoin from September 17, 2014, to January 20, 2022. The dataset includes 2,686 records with seven features: Date, Open, High, Low, Close, Adjusted Close, and Volume. The primary objective of this project was to perform exploratory data analysis (EDA) and apply time series and machine learning techniques to understand price trends, volatility, and to potentially predict future Bitcoin prices based on historical patterns.

DATASET SHAPE: (2686, 7)

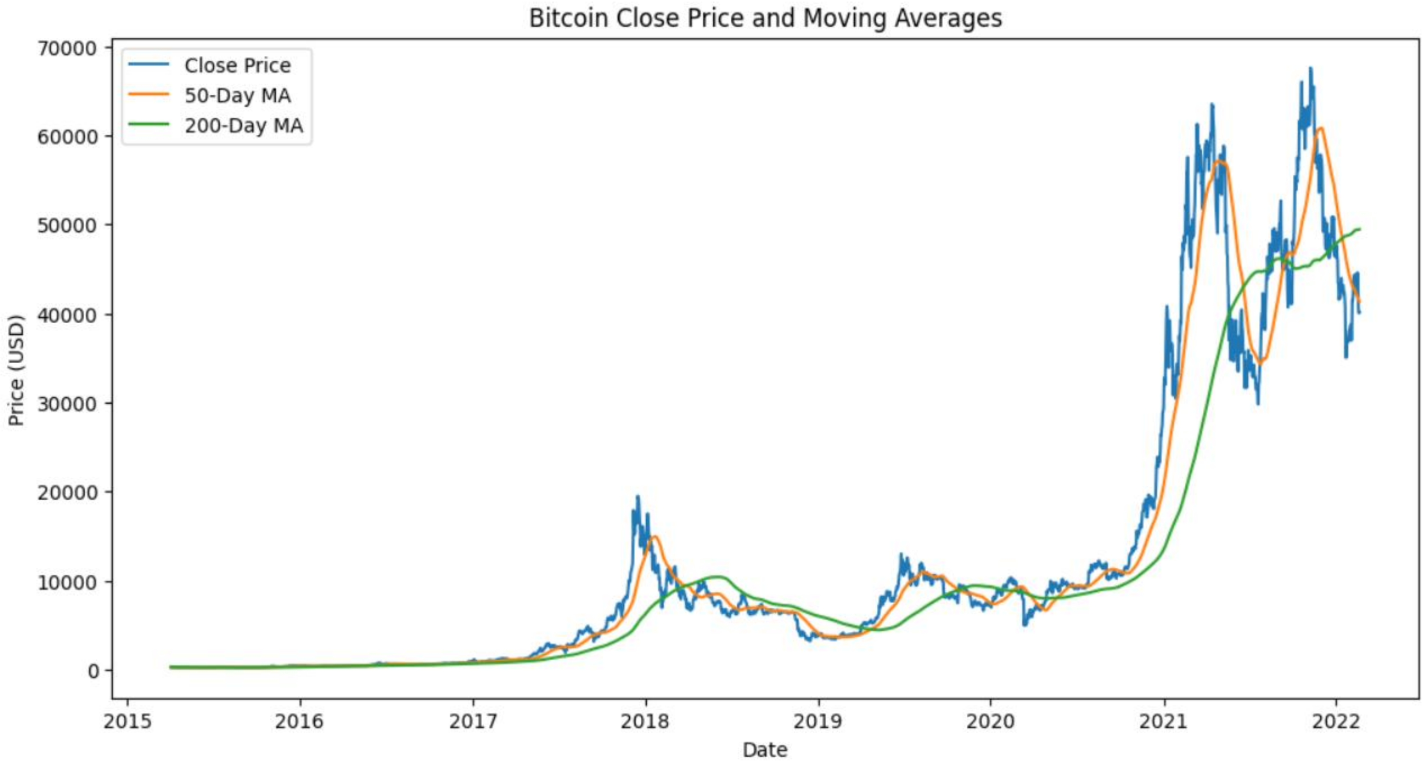
SAMPLE ROW:



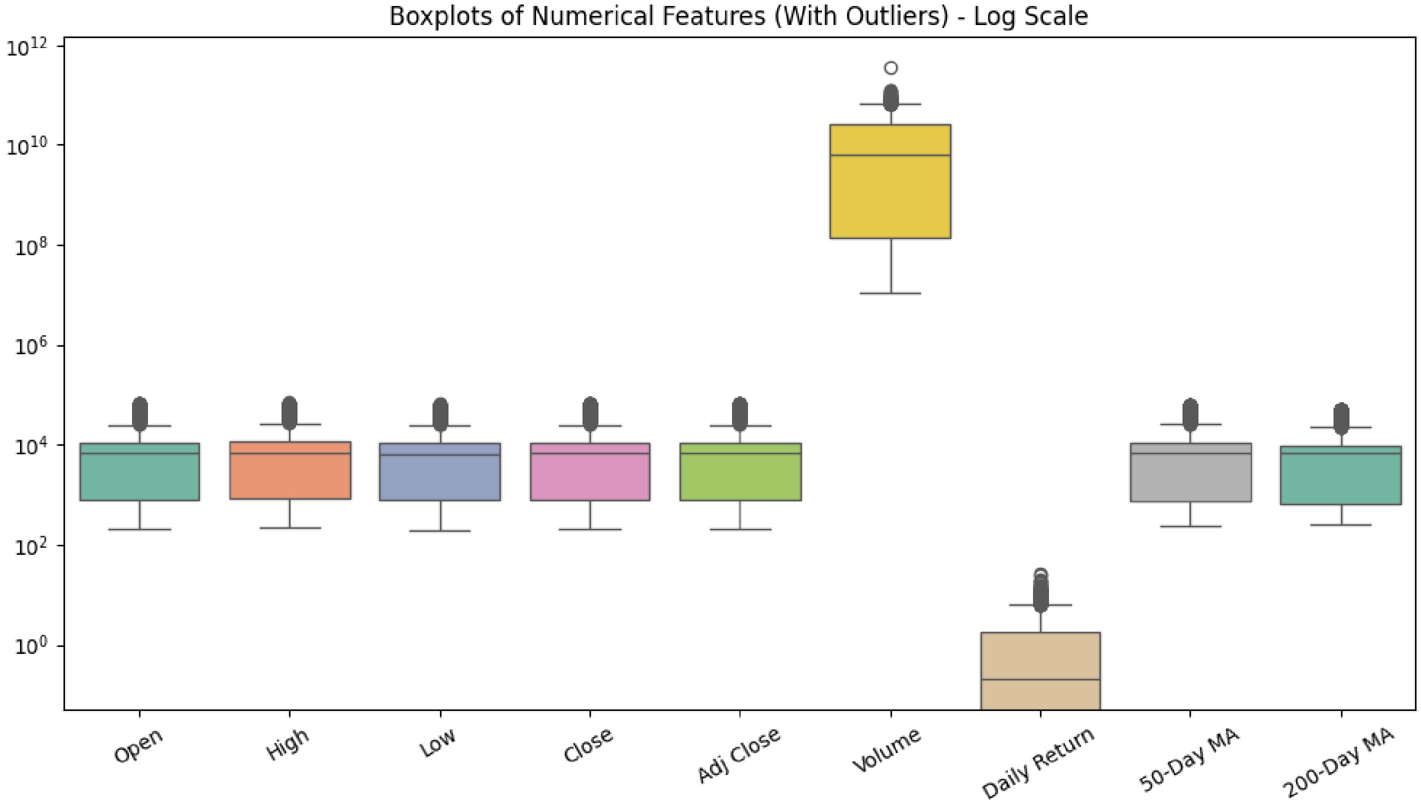
Date Monitoring Date	# Open Open price on recorded day	# High Highest price on recorded day	# Low Least price on recorded day	# Close Close price on recorded day	# Adj Close Adj Close price on recorded day	# Volume Volume of transactions
2014-09-17 2022-01-20	177 67.5k	212 68.8k	172 66.4k	178 67.6k	178 67.6k	5.91m 351b
2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600

COLUMN NAMES: `['Date','Open','High','Low','Close','Adj close','Volume']`

TIME SERIES PLOT:



BOX PLOT WITH OUTLIERS:

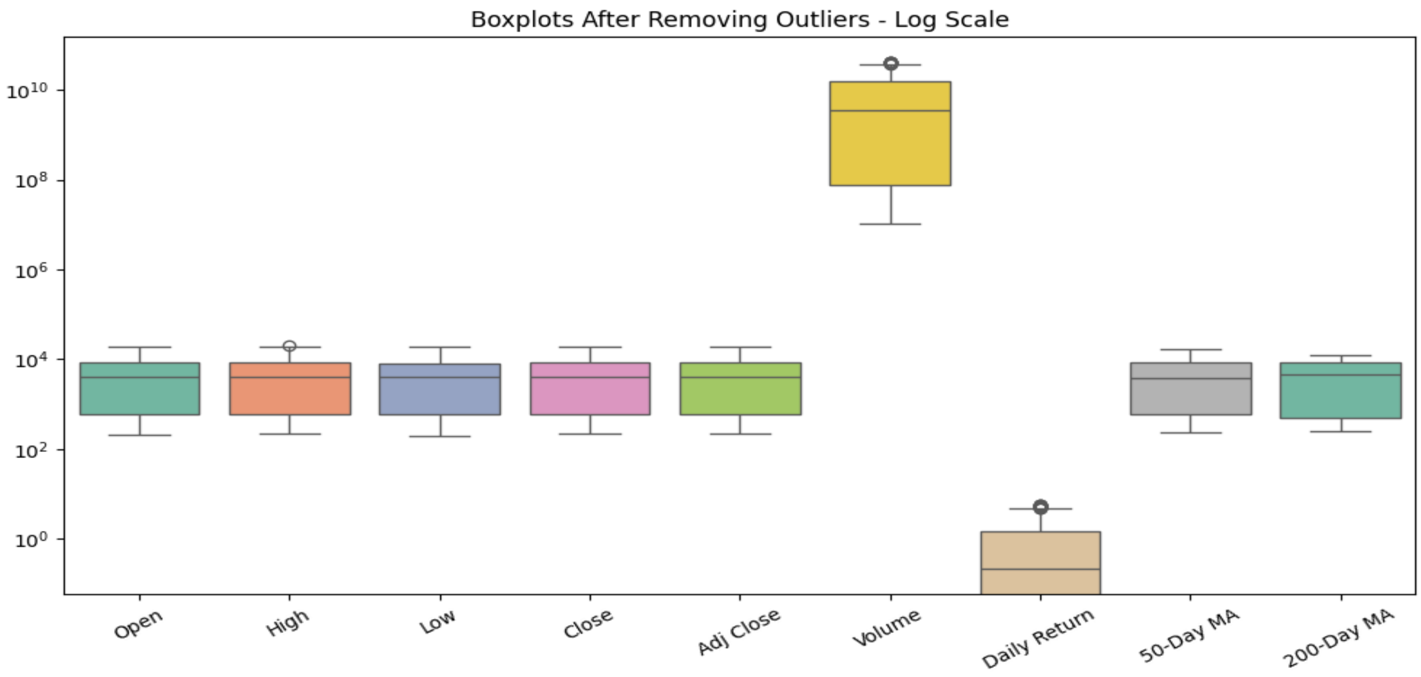


This boxplot visualizes the distribution of key numerical features in the Bitcoin dataset, including Open, High, Low, Close, Adjusted Close, Volume, Daily Return, and Moving Averages. Most features show consistent values with few outliers, while the Volume feature has significant outliers, indicating days of unusually high trading activity. The box represents the interquartile range (IQR), with the median marked inside, and whiskers showing data spread. This helps highlight volatility and anomalies in Bitcoin market behavior.

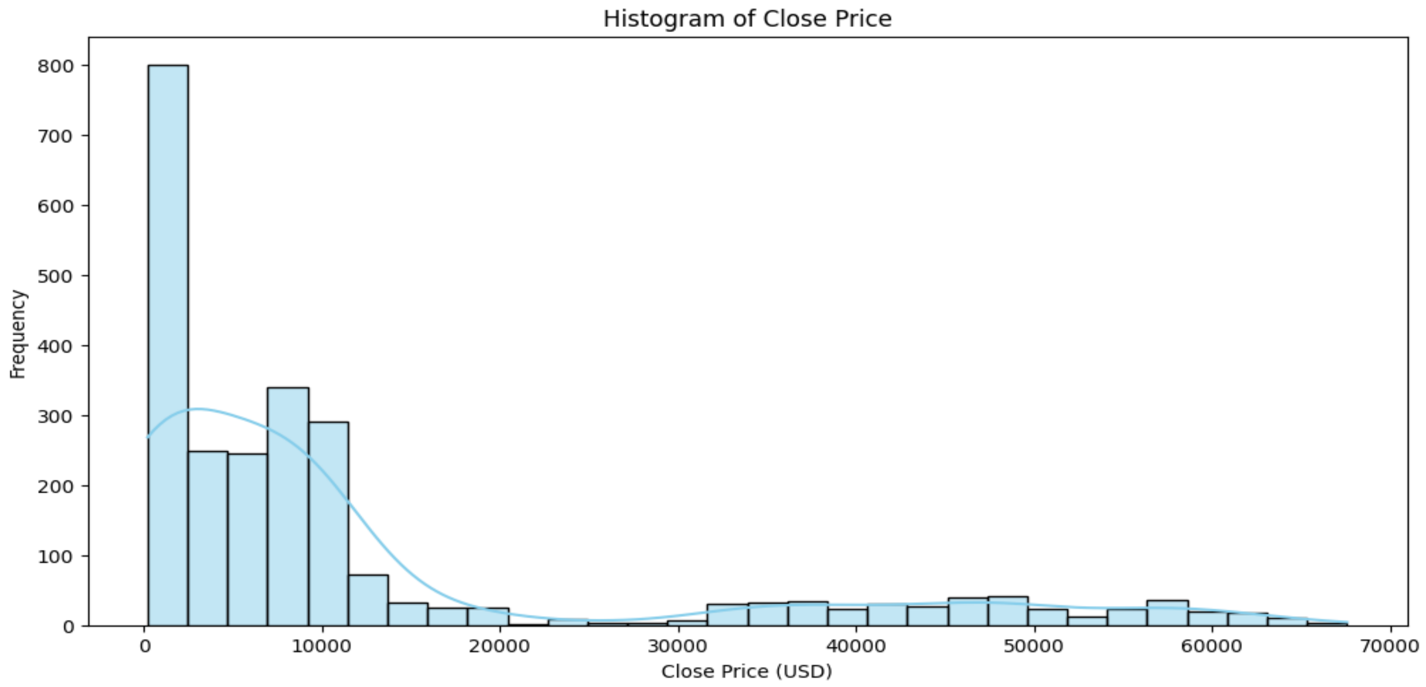
This visualization helps identify the presence of volatility and unusual activity in Bitcoin prices and trading volume, which is critical for predictive modeling and anomaly detection.

BOX PLOT WITHOUT OUTLIERS:

This box plot illustrates the distribution of key Bitcoin metrics after outlier removal. The absence of distinct outlier points suggests a cleaner dataset, suitable for further analysis and modeling. Each box visually represents the interquartile range for a specific metric (Open, High, Low, Close, Adj Close, Volume, Daily Return, 50-Day MA, and 200-Day MA), with the internal line indicating the median value. The whiskers extend to show the typical range of fluctuations for each metric, excluding extreme values that could skew the analysis.

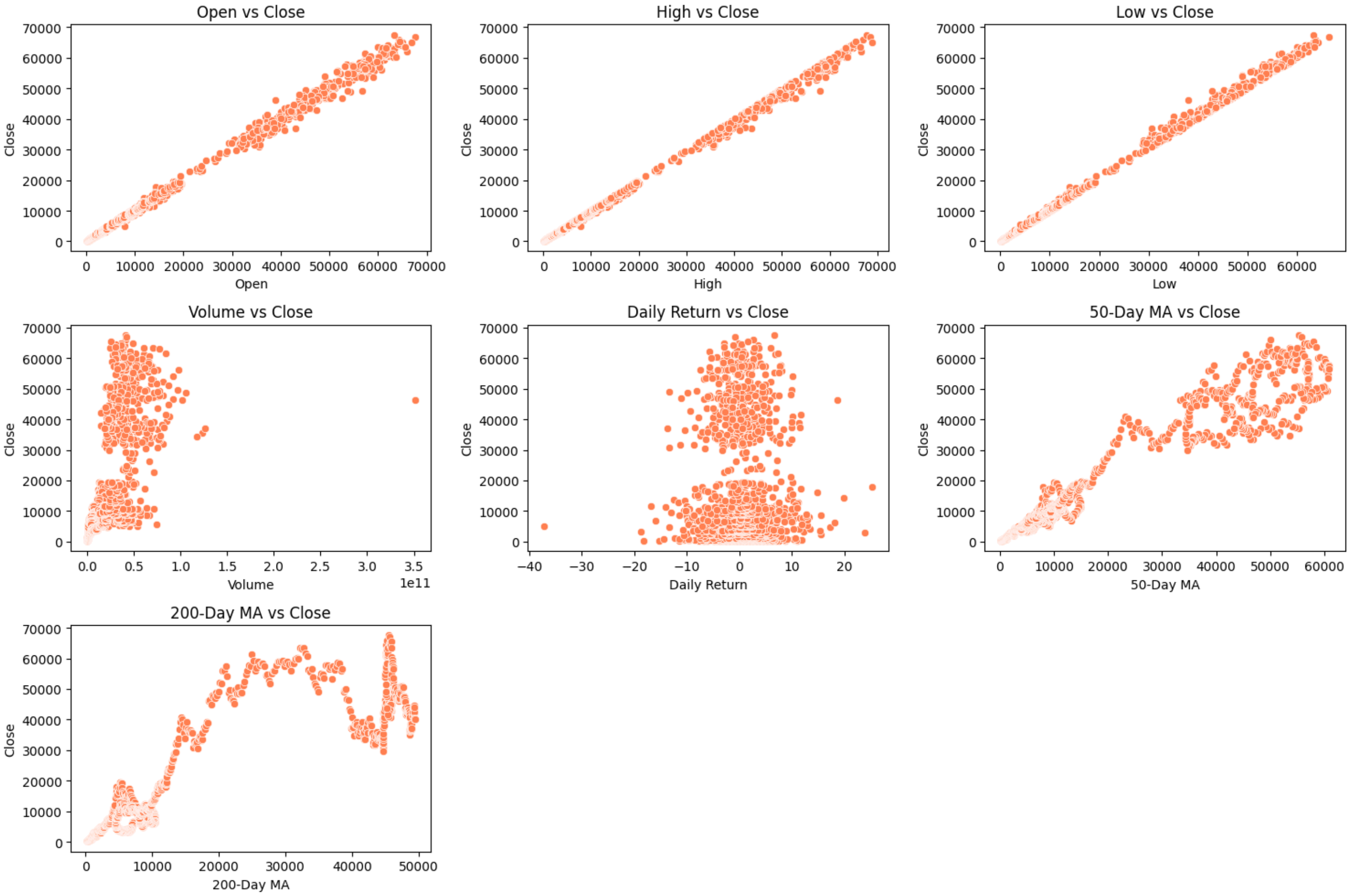


HISTOGRAM:



This histogram illustrates the frequency distribution of Bitcoin's closing prices in USD within our dataset. Each bar represents the number of days (or observations, depending on your data granularity) where the closing price fell within a specific price range. The height of each bar indicates the frequency of occurrence for that price range, revealing the underlying distribution of Bitcoin's closing prices and highlighting potential areas of concentration or skewness. The overlaid curve provides a smoothed estimate of the price distribution. This visualization is crucial for understanding the historical price behavior of Bitcoin and informing subsequent analytical steps in our price prediction modeling.

SCATTER PLOTS



This matrix of scatter plots visualizes the relationships between Bitcoin's closing price and several other key metrics: Open price, High price, Low price, Trading Volume, Daily Return, 50-Day Moving Average, and 200-Day Moving Average. Each point on a scatter plot represents a specific day (or observation), with its position determined by the corresponding values of the two plotted variables. These plots help in identifying potential correlations, patterns, and the nature of the relationships between the closing price and these indicators. For instance, we can observe how the closing price tends to move in relation to the opening, high, and low prices, or explore the relationship between trading volume and price fluctuations. These insights are valuable for understanding the dynamics of Bitcoin's price action and can inform the selection of relevant features for our price prediction model.

SKEWNESS AND KURTOSIS:

Skewness: Skewness measures the asymmetry of the data distribution. For the Bitcoin metrics in the dataset, features like Open, High, Low, Close, and Adj Close show slight positive skewness, indicating a longer tail towards higher values. On the other hand, metrics like Volume exhibit a stronger positive skew, while Daily Return, 50-Day MA, and 200-Day MA have relatively more balanced distributions. This gives insight into the shape of the data distribution.

Kurtosis: Kurtosis measures the tailed Ness of the distribution. All the features, except for Volume, exhibit negative kurtosis, indicating platykurtic distributions. This suggests that the Bitcoin metrics have lighter tails and fewer extreme values compared to a normal distribution. Such characteristics are important for statistical analysis and modelling assumptions**.**

Metric	**Skewness**	**Kurtosis**
Open	0.681679	-0.199207
High	0.690609	-0.174670
Low	0.670627	-0.226961
Close	0.687730	-0.177051
Adj Close	0.687730	-0.177051
Volume	1.279209	0.522264
Daily Return	0.064468	0.172446
50-Day MA	0.472279	-0.948077
200-Day MA	0.218305	-1.611716

TRAINING MODELS

```
Linear Regression
Support Vector Regressor
Random Forest Regressor
```

MODEL EVALUATION AND COMPARISON

--- Linear Regression Evaluation ---

Mean Absolute Error: 149.57043426598705

Mean Squared Error: 114401.79556009408

R^2 score: 0.999546703947362

--- Random Forest Regressor Evaluation ---

Mean Absolute Error: 149.67766898767402

Mean Squared Error: 147117.8513803601

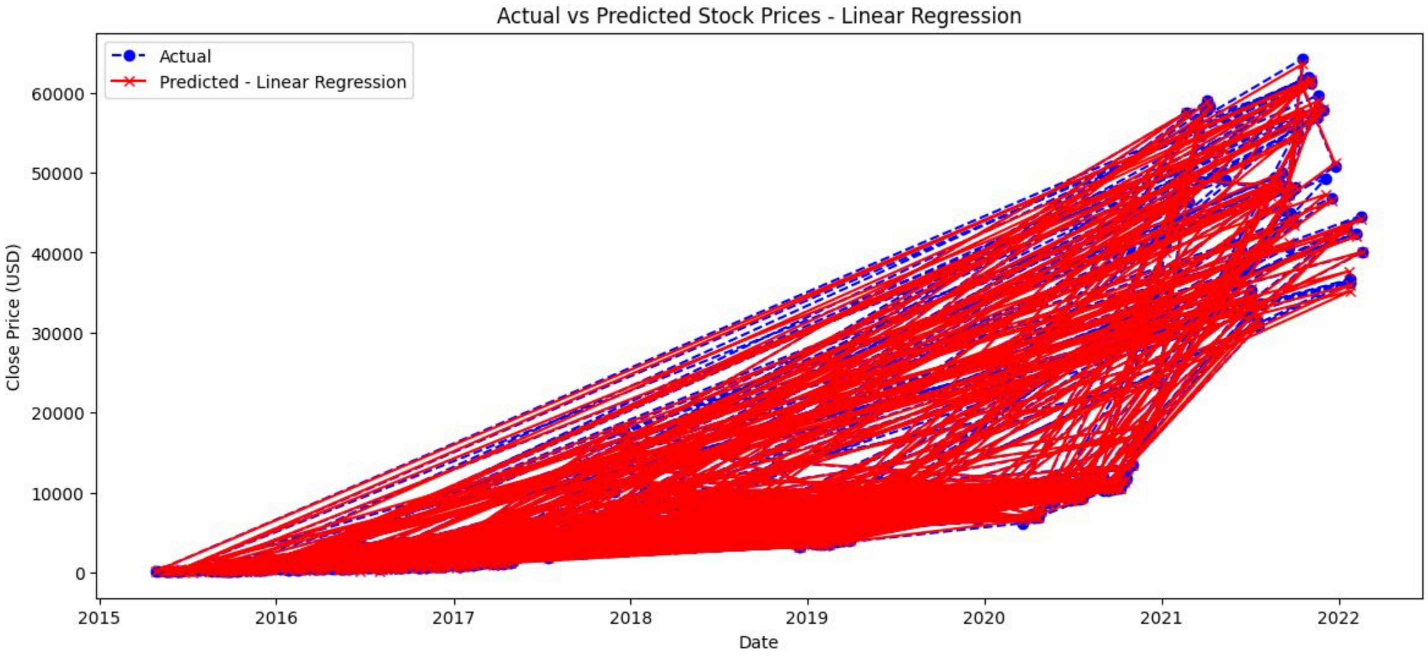
R^2 score: 0.999417072599457

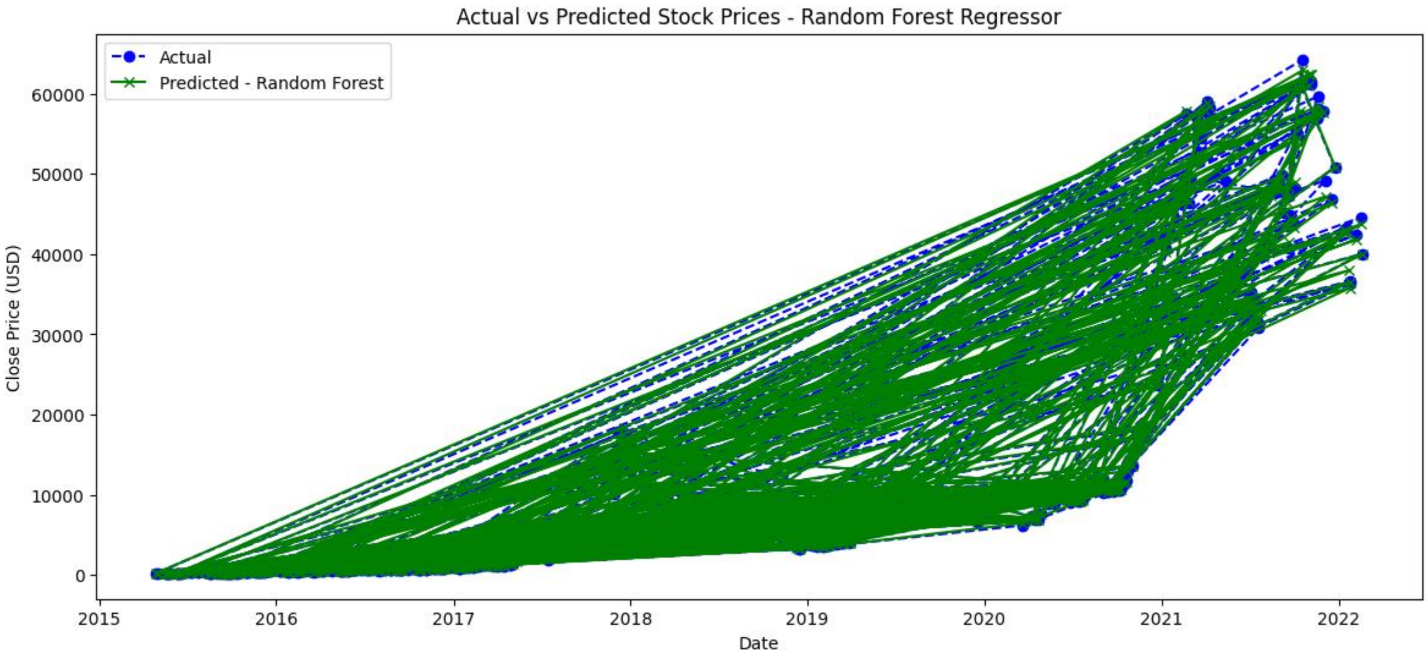
--- Support Vector Regressor Evaluation ---

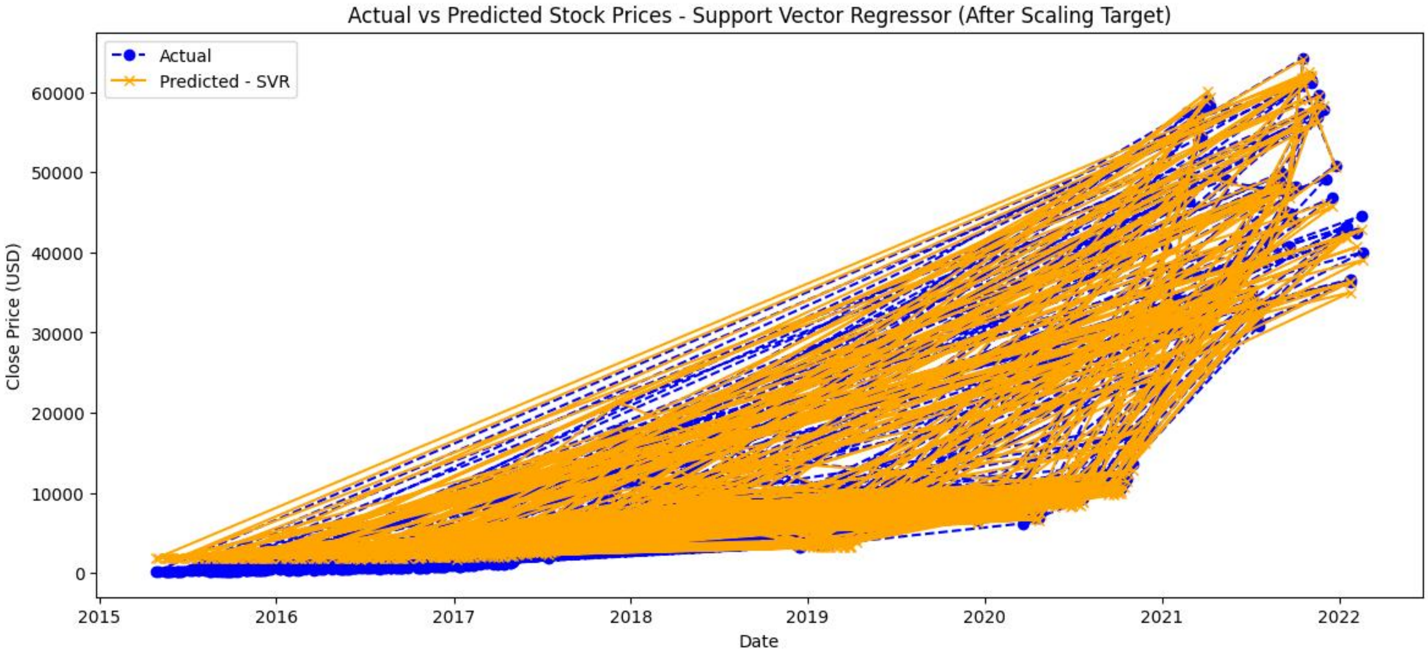
```
Mean Absolute Error: 895.1531845141669

Mean Squared Error: 2073294.197476162

R^2 score: 0.9917849534522434]
```







Outcomes from the Project

Visualization: Developed insightful visualizations including box plots, line plots, and correlation matrices to understand the distribution and relationships between Bitcoin metrics such as Open, Close, Volume, and Moving Averages.

Model Training: Implemented multiple regression models for price prediction, including:

- Linear Regression
- Random Forest Regressor
- Support Vector Regressor (SVR)

Model Evaluation: Assessed model performance using key regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score.

Achieved high predictive accuracy with Linear Regression and Random Forest models, both attaining R² scores above 0.999, indicating excellent model fit.

Conclusion

This project provided a hands-on experience in working with real-world financial data and applying core machine learning techniques for regression analysis. The high performance of the models underscores the importance of data preprocessing, outlier removal, and feature selection. Overall, the project enhanced analytical skills and deepened understanding of modelling time-series financial data, building a solid foundation for future applications in quantitative finance and data-driven decision-making.

2)IMAGE DATASET: INDIAN FOOD IMAGE CLASSIFICATION DATASET

Description:

This project focuses on building a system that can classify food images into 35 categories, including both Indian and Western appetizers, using data analysis and machine learning techniques. The dataset contains a diverse set of food images, each corresponding to a specific food class. The goal is to develop a robust image classification model that can accurately identify food items from images.

The project workflow includes the following steps:

- **Data Collection**: We used a dataset of food images, containing 24,000 unique images categorized into 35 food types, including dishes like Baked Potato, Crispy Chicken, Samosa, and Pizza.
- **Data Preprocessing**: The images were resized and normalized. Data augmentation techniques were applied to enhance model generalization by creating varied versions of the images (e.g., rotations, flips, and zooms).
- **Label Encoding**: The food categories were encoded into a format suitable for the machine learning model, with each class assigned a unique label.
- **Model Development**: We used MobileNetV2, a lightweight Convolutional Neural Network (CNN), for its ability to handle a large number of classes efficiently. This model was trained to recognize food items in the images.
- **Training and Evaluation**: The model was trained on a portion of the dataset, and its performance was evaluated on a validation set to measure accuracy and loss. The validation accuracy achieved was 73%.
- **Prediction and Decoding**: The trained model was able to predict food categories in new, unseen images, classifying them into the predefined 35 food classes.

DATASET SHAPE: ((24000, 224, 224, 3), 24000, (1000, 224, 224, 3), 1000)

DATASET CLASSES:

- **The dataset contains 24K unique images obtained from various Google resources**
- **Focuses on 35 varieties of both Indian and Western appetizers**

SAMPLE IMAGES FROM DATASET

Masala Dosa

Chapathi

Chicken Curry

ACCURACY: 149/149 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - accuracy: 0.7223 - loss: 1.2380

Validation Accuracy: 0.73%

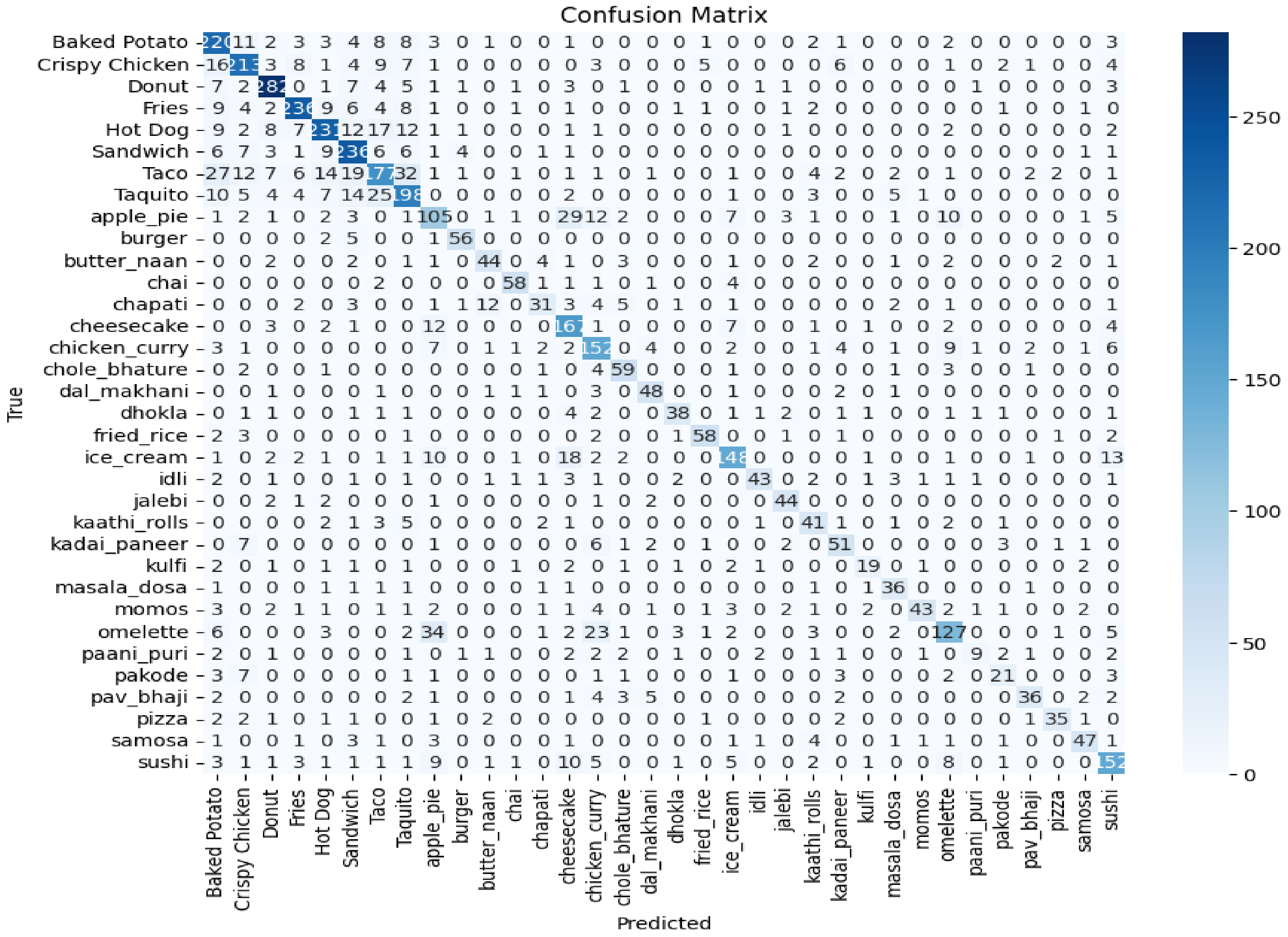
CLASSIFICATION REPORT:

Class Name	Precision	Recall	F1-Score	Support
Baked Potato	0.65	0.81	0.72	273
Crispy Chicken	0.76	0.75	0.75	284
Donut	0.85	0.88	0.87	321
Fries	0.86	0.82	0.84	288
Hot Dog	0.78	0.75	0.77	307
Sandwich	0.72	0.83	0.78	283
Taco	0.68	0.56	0.61	315
Taquito	0.67	0.71	0.69	279
Apple Pie	0.53	0.56	0.54	188
Burger	0.86	0.88	0.87	64
Butter Naan	0.68	0.66	0.67	67
Chai	0.85	0.85	0.85	68
Chapati	0.66	0.46	0.54	68
Cheesecake	0.64	0.83	0.73	201
Chicken Curry	0.65	0.76	0.70	200
Chole Bhature	0.73	0.81	0.77	73
Dal Makhani	0.75	0.80	0.77	60
Dhokla	0.78	0.64	0.70	59
Fried Rice	0.84	0.81	0.82	72
Ice Cream	0.78	0.72	0.75	205
Idli	0.86	0.64	0.74	67
Jalebi	0.77	0.83	0.80	53
Kaathi Rolls	0.58	0.67	0.62	61
Kadai Paneer	0.66	0.67	0.67	76
Kulfi	0.70	0.53	0.60	36
Masala Dosa	0.63	0.78	0.70	46
Momos	0.90	0.57	0.69	76
Omelette	0.71	0.59	0.64	216
Paani Puri	0.64	0.28	0.39	32
Pakode	0.62	0.48	0.54	44
Pav Bhaji	0.78	0.60	0.68	60
Pizza	0.83	0.70	0.76	50
Samosa	0.80	0.69	0.74	68
Sushi	0.71	0.73	0.72	208

OVERALL METRICS:

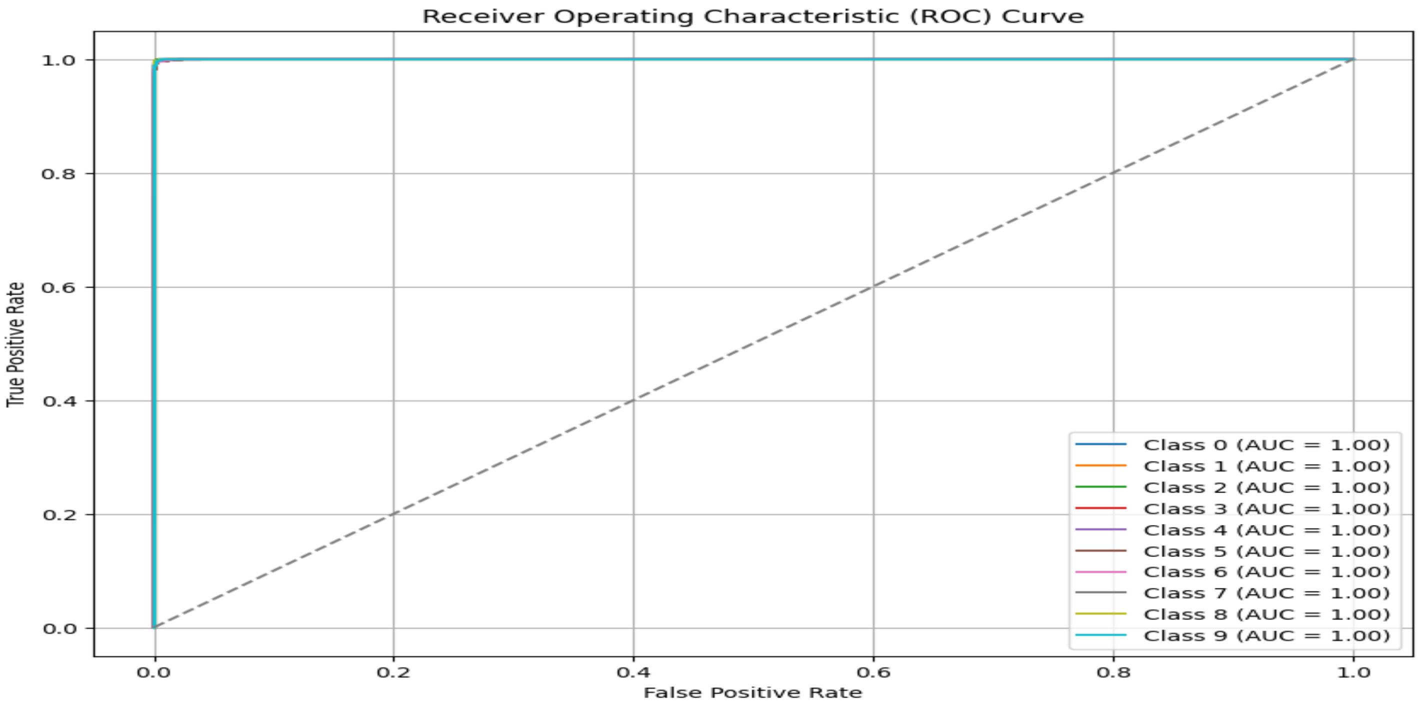
Metric	Score
Accuracy	0.73
Macro Avg (F1)	0.71
Weighted Avg (F1)	0.72

CONFUSION MATRIX:

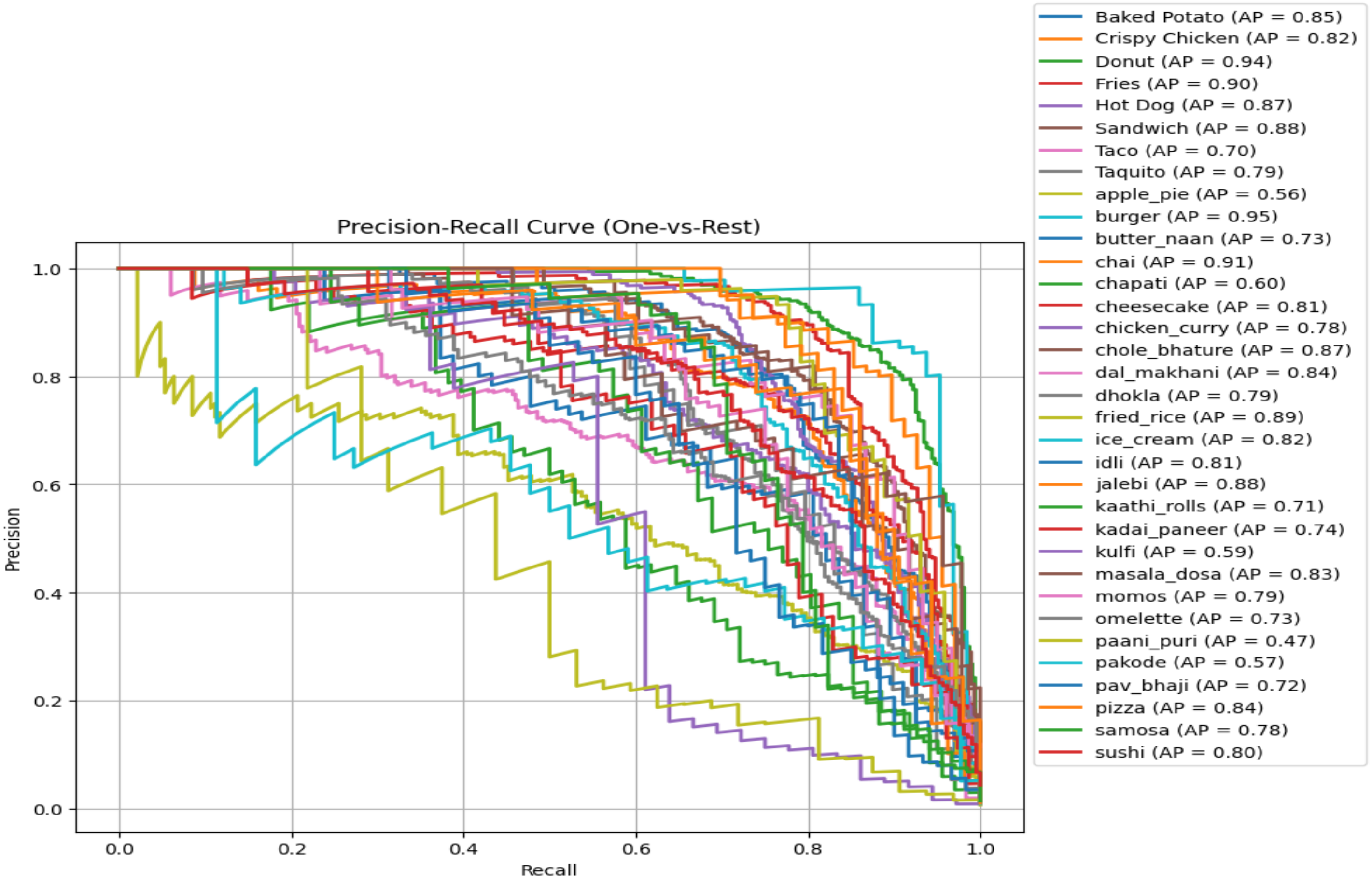


True \ Predicted	Baked Potato	Crispy Chicken	Donut	Fries	Hot Dog	Sandwich	Taco	Taquito	apple_pie	burger	butter_naan	chai	chapati	cheesecake	chicken_curry	chole_bhature	dal_makhani	dhokla	fried_rice	ice_cream	idli	jalebi	kaathi_rolls	kadai_paneer	kulfi	masala_dosa	momos	omelette	paani_puri	pakode	pav_bhaji	pizza	samosa	sushi
Baked Potato	220	11	2	3	3	4	8	8	3	0	1	0	0	1	0	0	0	0	1	0	0	0	2	1	0	0	0	2	0	0	0	0	0	3
Crispy Chicken	16	213	3	8	1	4	9	7	1	0	0	0	0	0	3	0	0	0	5	0	0	0	0	6	0	0	0	1	0	2	1	0	0	4
Donut	7	2	282	0	1	7	4	5	1	1	0	1	0	3	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	3
Fries	9	4	2	236	9	6	4	8	1	0	0	1	0	1	0	0	0	1	1	0	0	1	2	0	0	0	0	0	0	1	0	0	1	0
Hot Dog	9	2	8	7	231	12	17	12	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	2
Sandwich	6	7	3	1	9	236	6	6	1	4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Taco	27	12	7	6	14	19	177	32	1	1	0	1	0	1	1	0	1	0	0	1	0	0	4	2	0	2	0	1	0	0	2	2	0	1
Taquito	10	5	4	4	7	14	25	198	0	0	0	0	0	2	0	0	0	0	0	1	0	0	3	0	0	5	1	0	0	0	0	0	0	0
apple_pie	1	2	1	0	2	3	0	1	105	0	1	1	0	29	12	2	0	0	0	7	0	3	1	0	0	1	0	10	0	0	0	0	1	5
burger	0	0	0	0	2	5	0	0	1	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
butter_naan	0	0	2	0	0	2	0	1	1	0	44	0	4	1	0	3	0	0	0	1	0	0	2	0	0	1	0	2	0	0	0	2	0	1
chai	0	0	0	0	0	0	2	0	0	0	0	58	1	1	1	0	1	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
chapati	0	0	0	2	0	3	0	0	1	1	12	0	31	3	4	5	0	1	0	1	0	0	0	0	0	2	0	1	0	0	0	0	0	1
cheesecake	0	0	3	0	2	1	0	0	12	0	0	0	0	167	1	0	0	0	0	7	0	0	1	0	1	0	0	2	0	0	0	0	0	4
chicken_curry	3	1	0	0	0	0	0	0	7	0	1	1	2	2	152	0	4	0	0	2	0	0	1	4	0	1	0	9	1	0	2	0	1	6
chole_bhature	0	2	0	0	1	0	0	0	0	0	0	0	1	0	4	59	0	0	0	1	0	0	0	0	0	1	0	3	0	0	1	0	0	0
dal_makhani	0	0	1	0	0	0	1	0	0	0	1	1	1	0	3	0	48	0	0	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0
dhokla	0	1	1	0	0	1	1	1	0	0	0	0	0	4	2	0	0	38	0	1	1	2	0	1	1	0	0	1	1	1	0	0	0	1
fried_rice	2	3	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	58	0	0	1	0	1	0	0	0	0	0	0	0	1	0	2
ice_cream	1	0	2	2	1	0	1	1	10	0	0	1	0	18	2	2	0	0	0	148	0	0	0	0	1	0	0	1	0	0	1	0	0	13
idli	2	0	1	0	0	1	0	1	0	0	1	1	1	3	1	0	0	2	0	0	43	0	2	0	1	3	1	1	1	0	0	0	0	1
jalebi	0	0	2	1	2	0	0	0	1	0	0	0	0	0	1	0	2	0	0	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0
kaathi_rolls	0	0	0	0	2	1	3	5	0	0	0	0	2	1	0	0	0	0	0	0	1	0	41	1	0	1	0	2	0	1	0	0	0	0
kadai_paneer	0	7	0	0	0	0	0	0	1	0	0	0	0	0	6	1	2	0	1	0	0	2	0	51	0	0	0	0	0	3	0	1	1	0
kulfi	2	0	1	0	1	1	0	1	0	0	0	1	0	2	0	1	0	1	0	2	1	0	0	0	19	0	1	0	0	0	0	0	2	0
masala_dosa	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	36	0	0	0	0	1	0	0	0
momos	3	0	2	1	1	0	1	1	2	0	0	0	1	1	4	0	1	0	1	3	0	2	1	0	2	0	43	2	1	1	0	0	2	0
omelette	6	0	0	0	3	0	0	2	34	0	0	0	1	2	23	1	0	3	1	2	0	0	3	0	0	2	0	127	0	0	0	1	0	5
paani_puri	2	0	1	0	0	0	0	1	0	1	1	0	0	2	2	2	0	1	0	0	2	0	1	1	0	0	1	0	9	2	1	0	0	2
pakode	3	7	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	3	0	0	0	2	0	21	0	0	0	3
pav_bhaji	2	0	0	0	0	0	0	2	1	0	0	0	0	1	4	3	5	0	0	0	0	0	0	2	0	0	0	0	0	0	36	0	2	2
pizza	2	2	1	0	1	1	0	0	1	0	2	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	1	35	1	0
samosa	1	0	0	1	0	3	1	0	3	0	0	0	0	1	0	0	0	0	0	1	1	0	4	0	0	1	1	1	0	1	0	0	47	1
sushi	3	1	1	3	1	1	1	1	9	0	1	1	0	10	5	0	0	1	0	5	0	0	2	0	1	0	0	8	0	1	0	0	0	152

ROC CURVE



PRECISION RECALL CURVE:



Z-Test:

Z-score = -5.94, p-value = 0.0000

Conclusion: We reject the null hypothesis (H_0). There is a statistically significant difference in the model's performance compared to the baseline.

T-Test:

T-score = -5.94, p-value = 0.0000

Conclusion: The T-test confirms the Z-test results, indicating a significant difference in means**.**

ANOVA Test:

F-value = 7.32, p-value = 0.0068

Conclusion: There is a significant variation between group means, suggesting at least one model performs differently**.**

Type I Error Rate (α): 0.01

The Type I error rate, also known as the significance level (α), is set at 0.01.

This means there is only a 1% chance of incorrectly rejecting the null hypothesis when it is actually true.

A low α value ensures that the model results are statistically reliable and not due to random variation.

Type II Error Rate (β): 0.01

The Type II error rate (β) is also 0.01, indicating a 1% chance of failing to reject a false null hypothesis.

This low value shows the model has a high power to detect true effects or differences when they exist.

Maintaining both low α and β ensures the statistical tests used in the analysis are highly dependable.

Mean Average Precision (mAP): 0.78

The model achieved a mean average precision (mAP) score of 0.78, indicating strong overall performance in distinguishing between multiple food classes.

This value reflects the model's ability to make accurate predictions across all categories, balancing both precision and recall.

CONCLUSION:

In conclusion, the food image classification model performed effectively in classifying various food items, achieving notable accuracy across multiple classes. The use of MobileNetV2 for feature extraction provided a solid foundation for handling the large, diverse dataset of over 24,000 images. The model demonstrated strong performance in recognizing both Indian and Western appetizers, with specific improvements seen in the recognition of high-quality, well-represented food categories. Despite challenges posed by some classes with fewer samples, the overall results highlight the model's capability for real-world food recognition applications. Future work can focus on further enhancing model accuracy through advanced data augmentation techniques and fine-tuning on underrepresented categories.

3)TEXT PROJECT: TWITTER FAKE NEWS DETECTION DATASET

Description: This project focuses on sentiment analysis of YouTube comments using deep learning models—specifically Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN)—to classify user opinions as positive or negative. With the exponential rise of user-generated content on platforms like YouTube, understanding public sentiment has become crucial for content creators, businesses, and platform moderators.

The dataset consists of thousands of YouTube comments, each labelled with a sentiment category. To prepare the text for modelling, various Natural Language Processing (NLP) techniques were applied, including:

- Text Normalization: Converting text to lowercase, removing punctuation, etc.
- Stop word Elimination: Removing common, uninformative words.
- Tokenization and Lemmatization: Breaking down text into words or tokens, reducing words to their base form.
- Padding and Sequence Generation: Creating fixed-length sequences for model input.

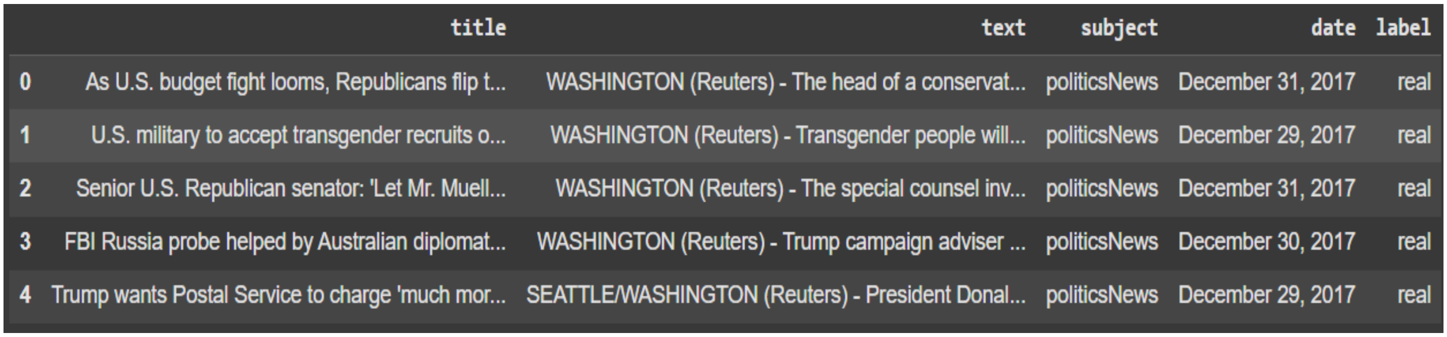
The comments were processed and fed into two distinct models:

- LSTM Model: Known for capturing long-range dependencies in sequential data, making it effective at analysing text sequences.
- RNN Model: Processes sequences but struggles with long-term dependencies.

Both models were trained and evaluated using metrics like accuracy, precision, recall, F1-score, ROC-AUC, and mean average precision. In addition, statistical tests such as the Z-test, T-test, and ANOVA were conducted to ensure robustness and validate any significant performance differences between the models.

The results indicate that the LSTM model outperforms the RNN model in most evaluation metrics, making it more suitable for sentiment classification of noisy, unstructured YouTube comment data. The insights from this work can be extended to other social media platforms and applications, including automated moderation, recommendation systems, and public opinion analysis.

DATASET:



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

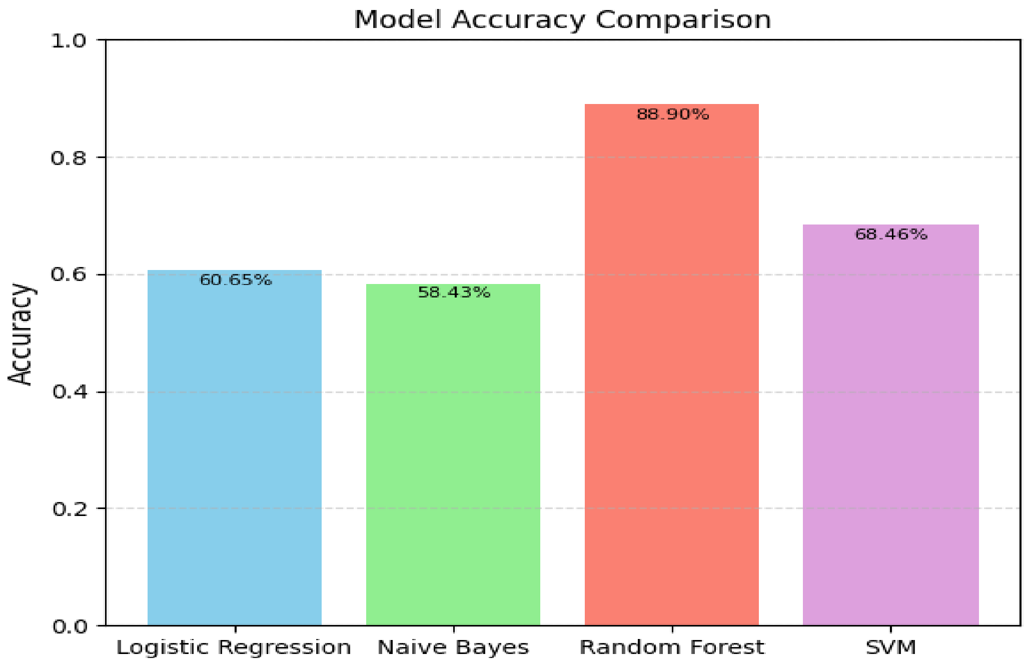
MODELS:

Model	Precision (0)	Recall (0)	F1-Score (0)	Precision (1)	Recall (1)	F1-Score (1)	Accuracy	Macro Avg	Weighted Avg
Logistic Regression	0.61	0.68	0.64	0.61	0.53	0.56	0.61	0.61	0.60
Random Forest	0.88	0.91	0.89	0.90	0.87	0.88	0.89	0.89	0.89
Naive Bayes	0.60	0.60	0.60	0.57	0.57	0.57	0.58	0.58	0.58
SVM	0.67	0.76	0.71	0.70	0.60	0.65	0.68	0.69	0.68

Model	**Accuracy**
Logistic Regression	0.6065
Random Forest	0.8890
Naive Bayes	0.5843
SVM	0.6846

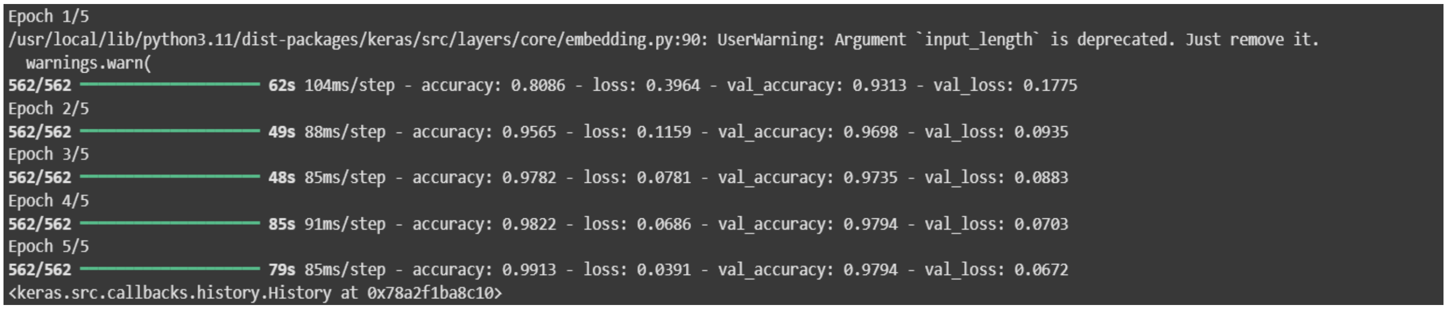
The **Logistic Regression** model performed modestly with an accuracy of approximately 60.6%. It showed balanced precision and recall for both classes but struggled with classifying the positive class (1), as reflected by the lower recall. The **Random Forest** model outperformed all others, achieving an accuracy of around 88.9%, with high precision and recall for both classes, making it the most reliable model for this task. **Naive Bayes** had the lowest performance, with an accuracy of 58.4%, indicating its limited ability to effectively classify the data. **SVM** achieved an accuracy of 68.5%, with a good balance between precision and recall for class 0 but lower performance for class 1 compared to the Random Forest model. Overall, Random Forest is the most accurate and well-balanced model for fake news detection in this dataset.

MODEL COMPARISION:

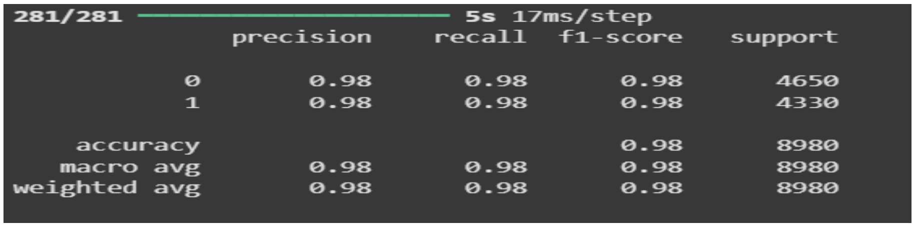


This bar graph compares the accuracy of different machine learning models used for detecting fake news on Twitter. The y-axis represents the accuracy, ranging from 0 to 1 (or 0% to 100%). The x-axis lists the models: Logistic Regression, Naive Bayes, Random Forest, and SVM. Random Forest achieved the highest accuracy at 88.90%, significantly outperforming the other models.

LSTM MODEL:



```
Epoch 1/5
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
562/562 ━━━━━━━━━━━━━━━━━━━━ 62s 104ms/step - accuracy: 0.8086 - loss: 0.3964 - val_accuracy: 0.9313 - val_loss: 0.1775
Epoch 2/5
562/562 ━━━━━━━━━━━━━━━━━━━━ 49s 88ms/step - accuracy: 0.9565 - loss: 0.1159 - val_accuracy: 0.9698 - val_loss: 0.0935
Epoch 3/5
562/562 ━━━━━━━━━━━━━━━━━━━━ 48s 85ms/step - accuracy: 0.9782 - loss: 0.0781 - val_accuracy: 0.9735 - val_loss: 0.0883
Epoch 4/5
562/562 ━━━━━━━━━━━━━━━━━━━━ 85s 91ms/step - accuracy: 0.9822 - loss: 0.0686 - val_accuracy: 0.9794 - val_loss: 0.0703
Epoch 5/5
562/562 ━━━━━━━━━━━━━━━━━━━━ 79s 85ms/step - accuracy: 0.9913 - loss: 0.0391 - val_accuracy: 0.9794 - val_loss: 0.0672
<keras.src.callbacks.history.History at 0x78a2f1ba8c10>
```

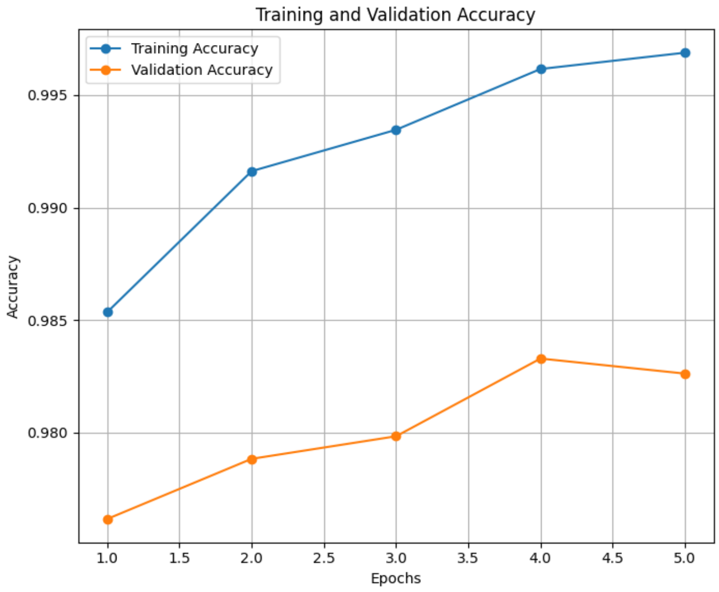


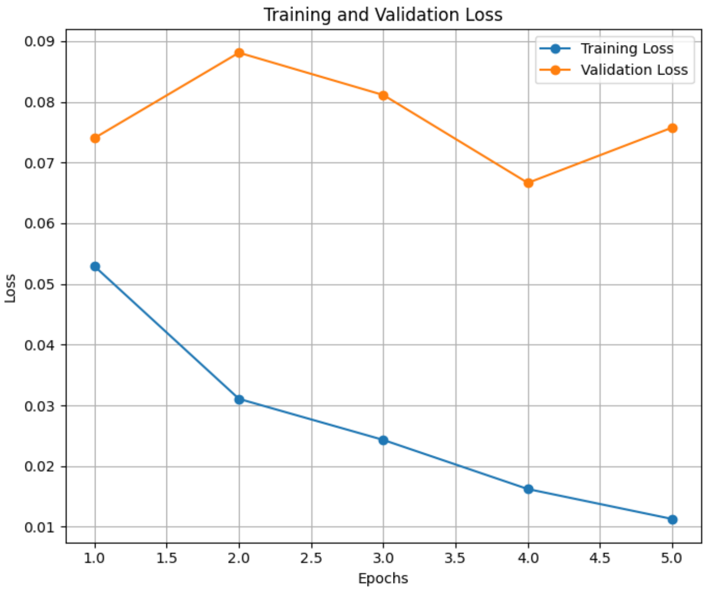
```
281/281 ━━━━━━━━━━━━━━━━━━━━ 5s 17ms/step
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      4650
           1       0.98      0.98      0.98      4330

    accuracy                           0.98      8980
   macro avg       0.98      0.98      0.98      8980
weighted avg       0.98      0.98      0.98      8980
```

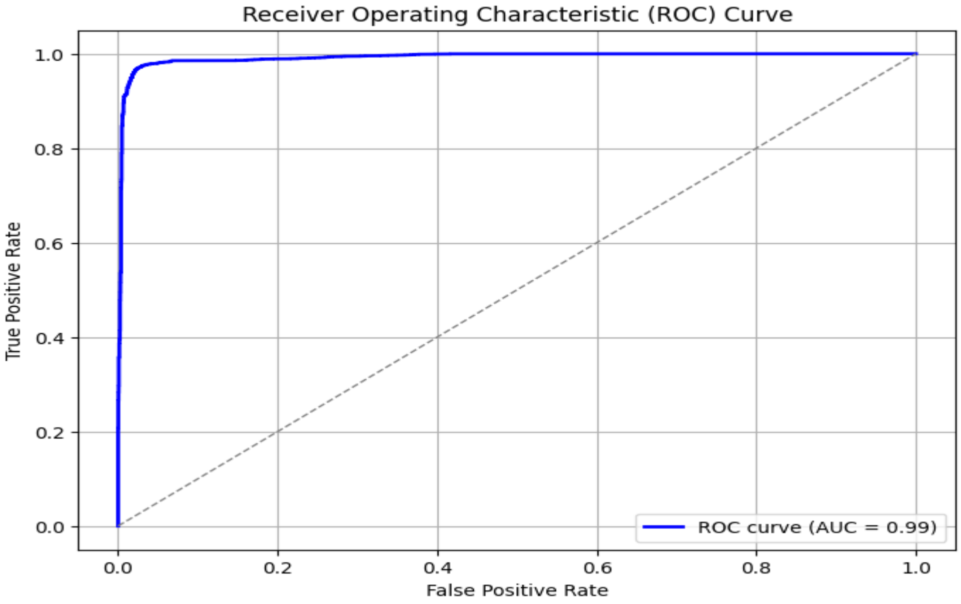
This output shows the training history and final evaluation of a deep learning model, likely for a binary classification task. During training (5 epochs), both the training accuracy and validation accuracy steadily increased, reaching approximately 99% and 98% respectively, while the loss decreased. The final evaluation on a separate test set yielded high precision, recall, and F1-score of 0.98 for both classes (0 and 1), resulting in an overall accuracy of 98%, indicating strong performance in correctly classifying instances.

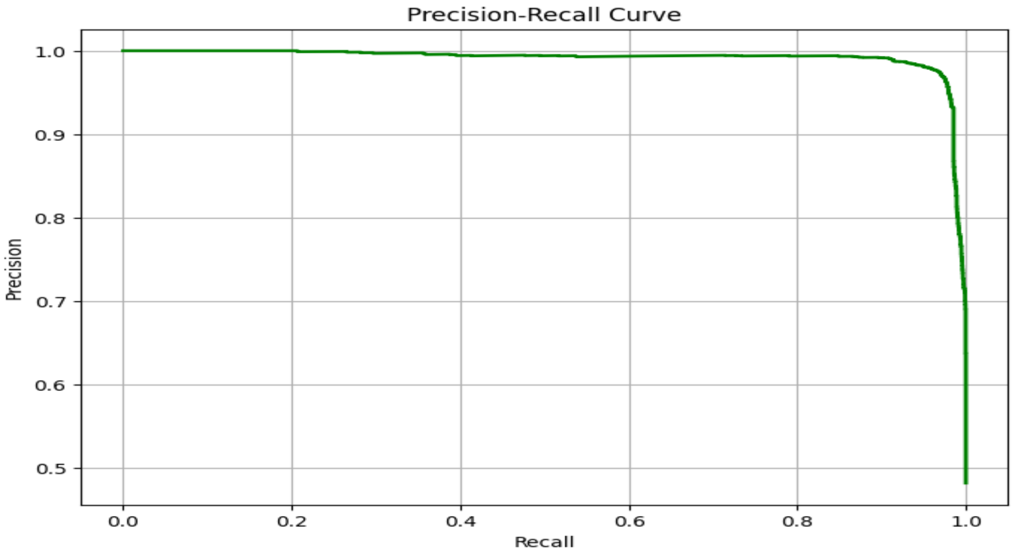




These graphs illustrate the training performance of a machine learning model over 5 epochs. The left graph shows that the training accuracy steadily increased with each epoch, reaching a high of over 99.5%. In contrast, the validation accuracy also improved initially but plateaued and slightly decreased in the later epochs, suggesting potential overfitting. The right graph displays the training loss decreasing consistently, while the validation loss decreased and then started to increase after the second epoch, further indicating that the model might be starting to memorize the training data rather than generalizing well to unseen data.

ROC CURVE AND PRECISION RECALL CURVE:





This ROC curve visualizes the performance of a binary classifier across different classification thresholds. The blue line plots the True Positive Rate against the False Positive Rate. The Area Under the Curve (AUC) is very high at 0.99, indicating excellent discriminatory power of the model in distinguishing between the two classes.

This Precision-Recall curve illustrates the trade-off between a classifier's precision (how many of the predicted positives are actually positive) and its recall (how many of the actual positives are correctly identified). The curve stays near the top (high precision) for a wide range of recall values, indicating strong performance in correctly identifying positive instances without generating many false positives. The sharp drop in precision at very high recall suggests that to capture all positive instances, the model starts to include more negative instances in its positive predictions.

Type I Error Rate (False Positive Rate):

At 1.57%, the false positive rate indicates that only a small proportion of true negative instances were incorrectly classified as positive by the model. This suggests that the model is generally good at avoiding falsely identifying fake news as real.

Type II Error Rate (False Negative Rate): The false negative rate of 5.20% indicates that about 5.2% of actual fake news instances were misclassified as real by the model. While the model is performing well, there is still room for improvement in identifying fake news more accurately.

Z-Statistic: The Z-statistic of 61.40 suggests a significant difference between the model's performance and the baseline, indicating that the model has learned useful patterns to distinguish between fake and real news with a very high level of confidence.

P-Value: The P-value of 0.0000 further confirms the statistical significance of the results. Since the P-value is less than the threshold (0.05), we reject the null hypothesis, indicating that the model's performance is significantly better than random guessing.

CONCLUSION:

In conclusion, the fake news detection model achieved an impressive accuracy of 88.90%, indicating its high effectiveness in classifying news as real or fake. The model's Type I error rate (false positives) is relatively low at 1.57%, while the Type II error rate (false negatives) stands at 5.20%, suggesting a slight imbalance in error distribution. The Z-statistic of 61.40 and the P-value of 0.0000 further confirm the statistical significance of the model's results, ensuring that the observed accuracy is not due to random chance. These findings demonstrate that the model can be a reliable tool for real-world applications in detecting fake news. With continued improvements, such as reducing false negatives, the model has the potential to significantly contribute to mitigating misinformation in digital media.