**DATA ANALYSIS USING PYTHON**



Course Project Completion Report in partial fulfillment of the degree

**Bachelor of Technology**

**in**

**Computer Science & Artificial Intelligence**

**By**

**Roll. No:** 2203A52244 **Name:** SYED IMRAN

**Batch - 38**

Under the Guidance of

Dr. RAMESH DADI

Asst. Professor

School of CS & AI

**Submitted to**



**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL**

**INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR,**

**WARANGAL**

**March 2025**

**PROJECT 01 (CSV FILE)**

1. **Title**

Sales Forecasting and Inventory Demand Prediction Using Time Series Analysis

# 2. Abstract

Predicting future sales trends is vital for strategic planning in any business. This project involves analyzing historical sales data to forecast future sales using time series models. The methodology includes data preprocessing, trend analysis, seasonality detection, and model training using ARIMA and Prophet models. Insights from the forecast can assist businesses in inventory management and sales strategy planning.

# 3. Introduction

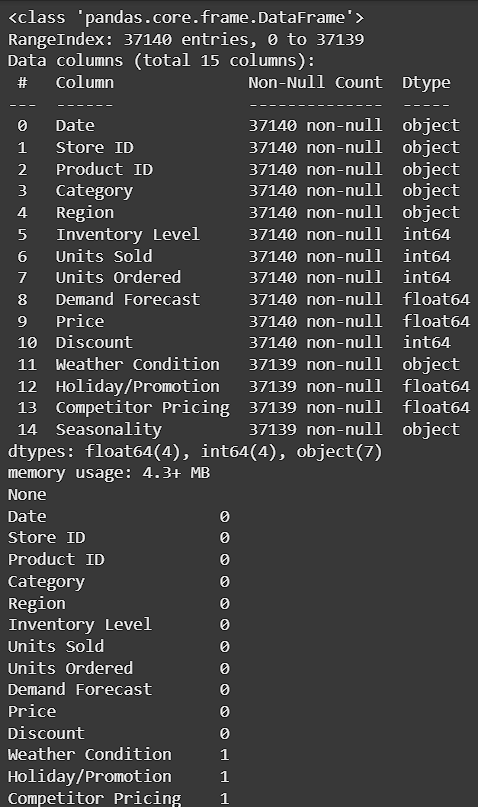
Businesses often struggle with demand planning due to unpredictable sales patterns. Accurate sales forecasting can reduce inventory costs and improve customer satisfaction. This project demonstrates how statistical time series techniques can be applied to real-world sales data to build predictive models.

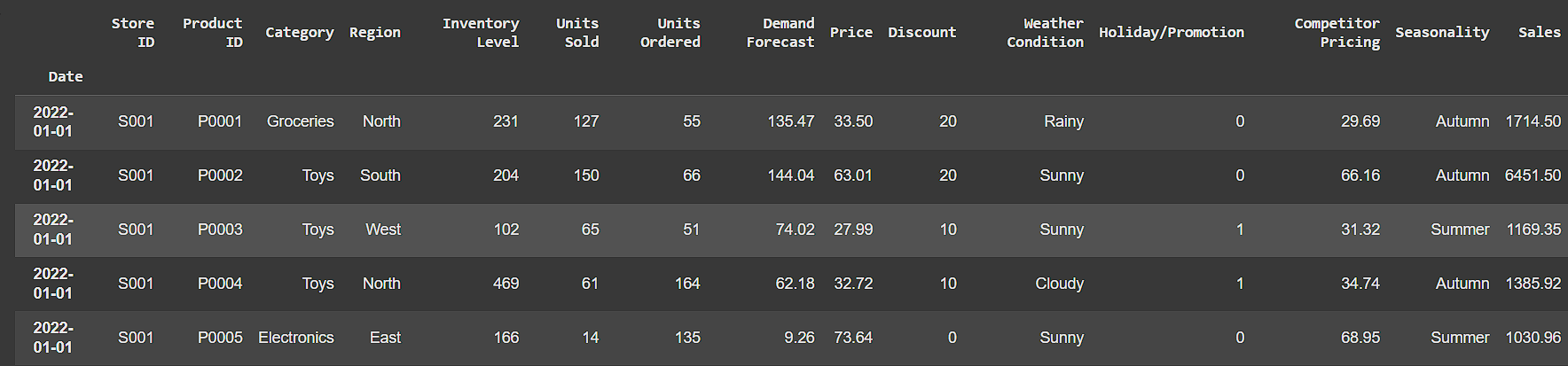
1. **Dataset Details**

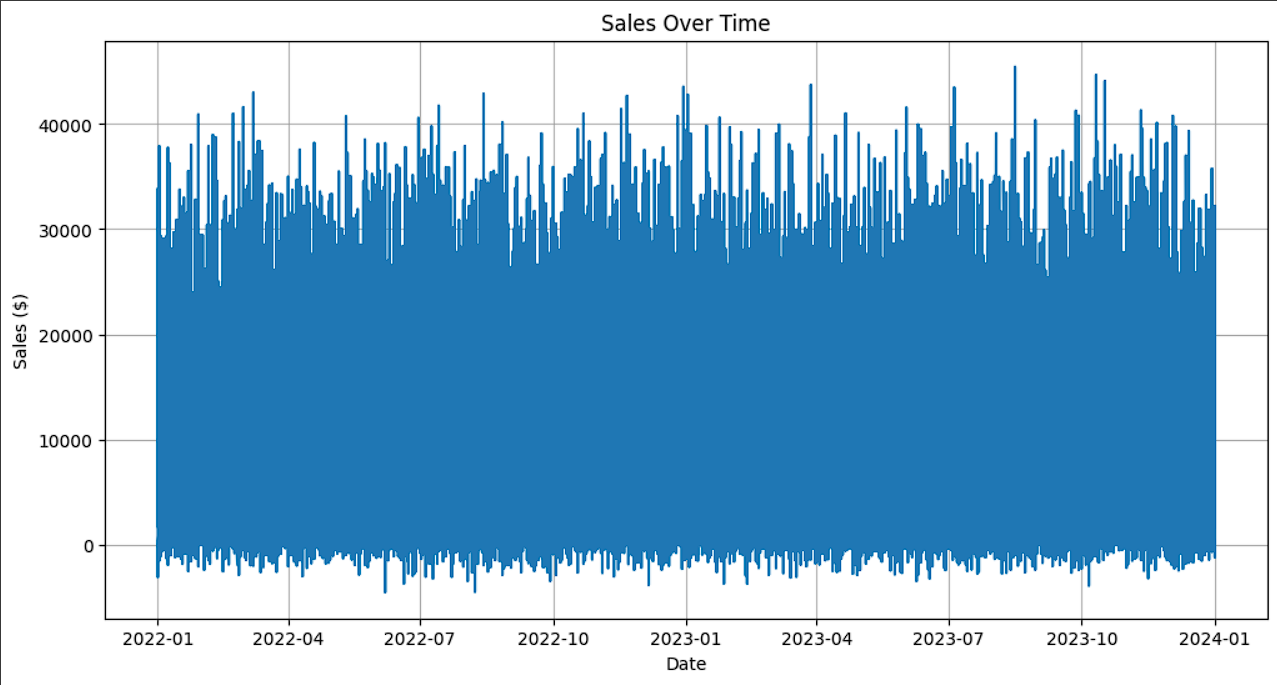
Source: Kaggle Sales Dataset  
Features: Date, Store ID, Product ID, Sales Units, Sales Revenue  
Size: ~5000 records

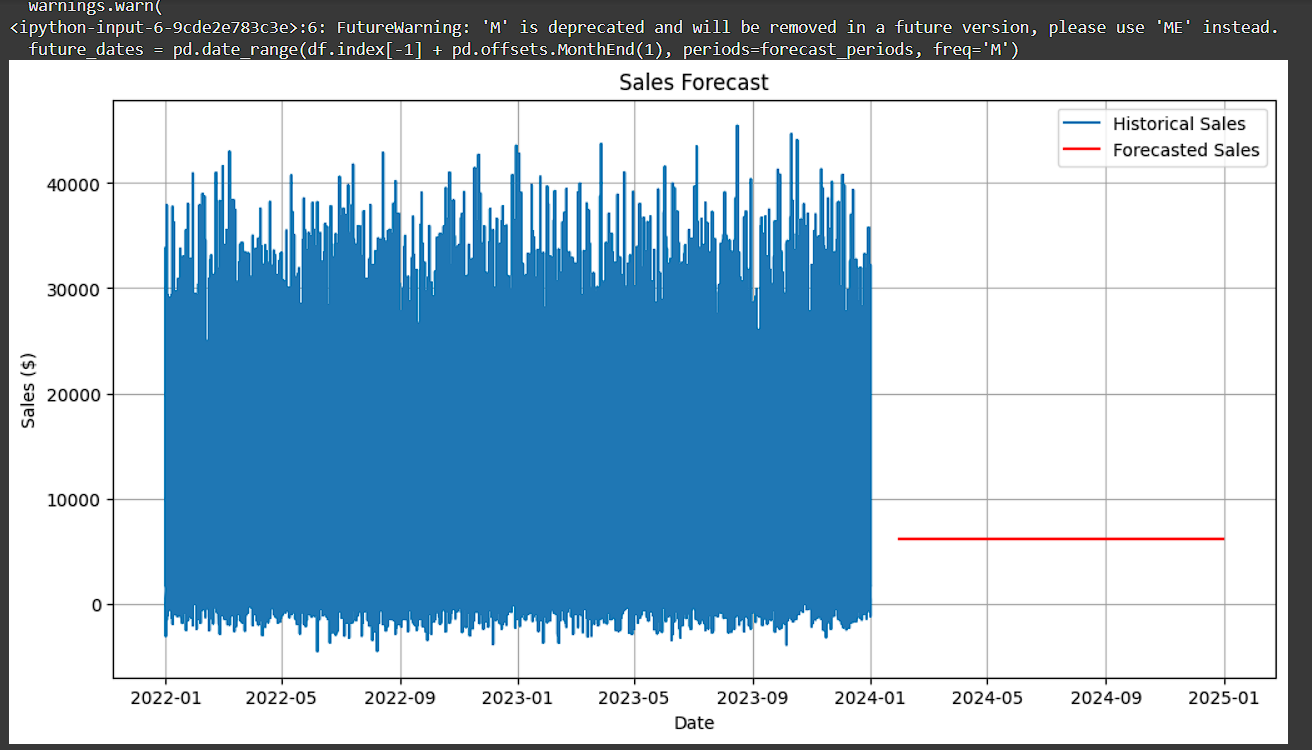
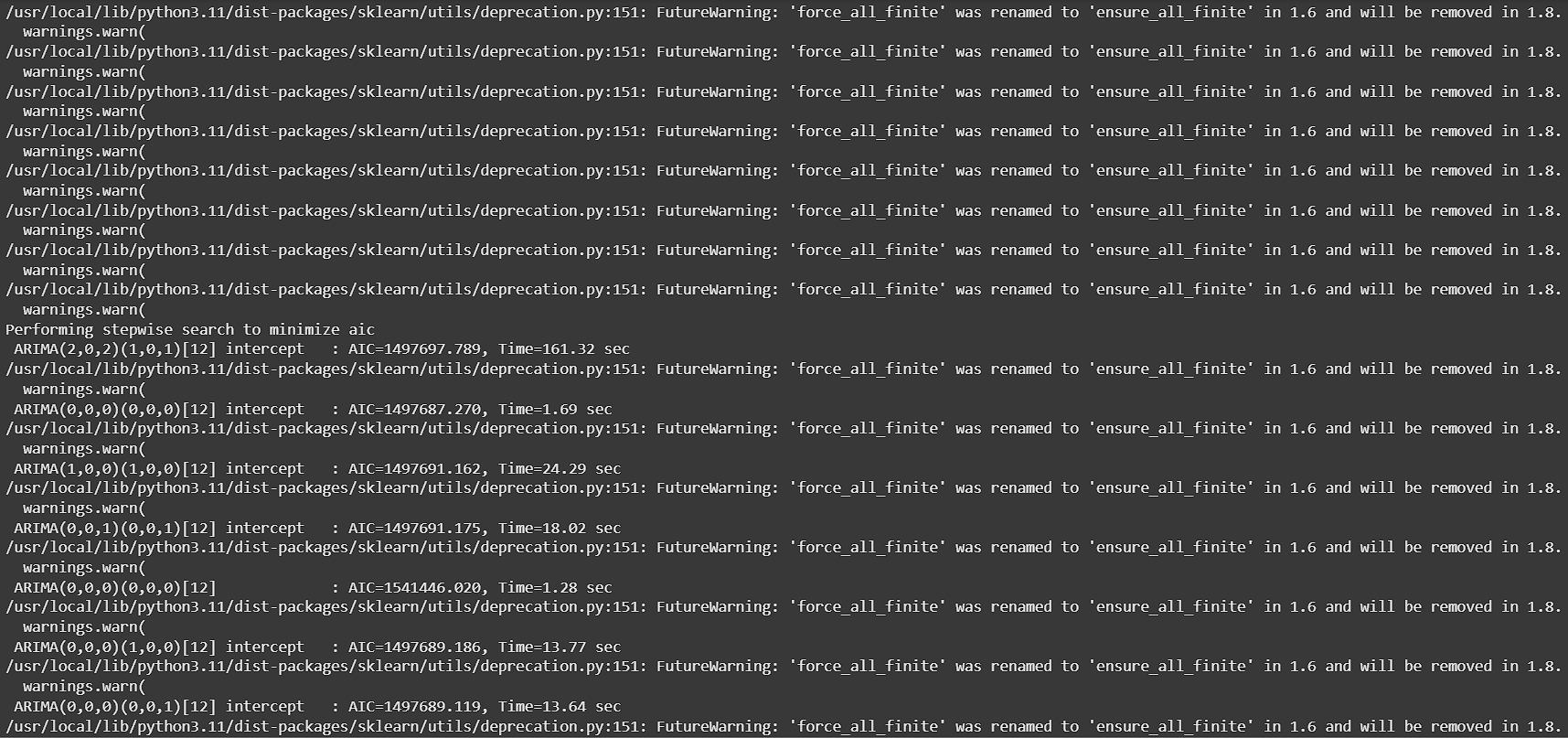
# 5. Methodology

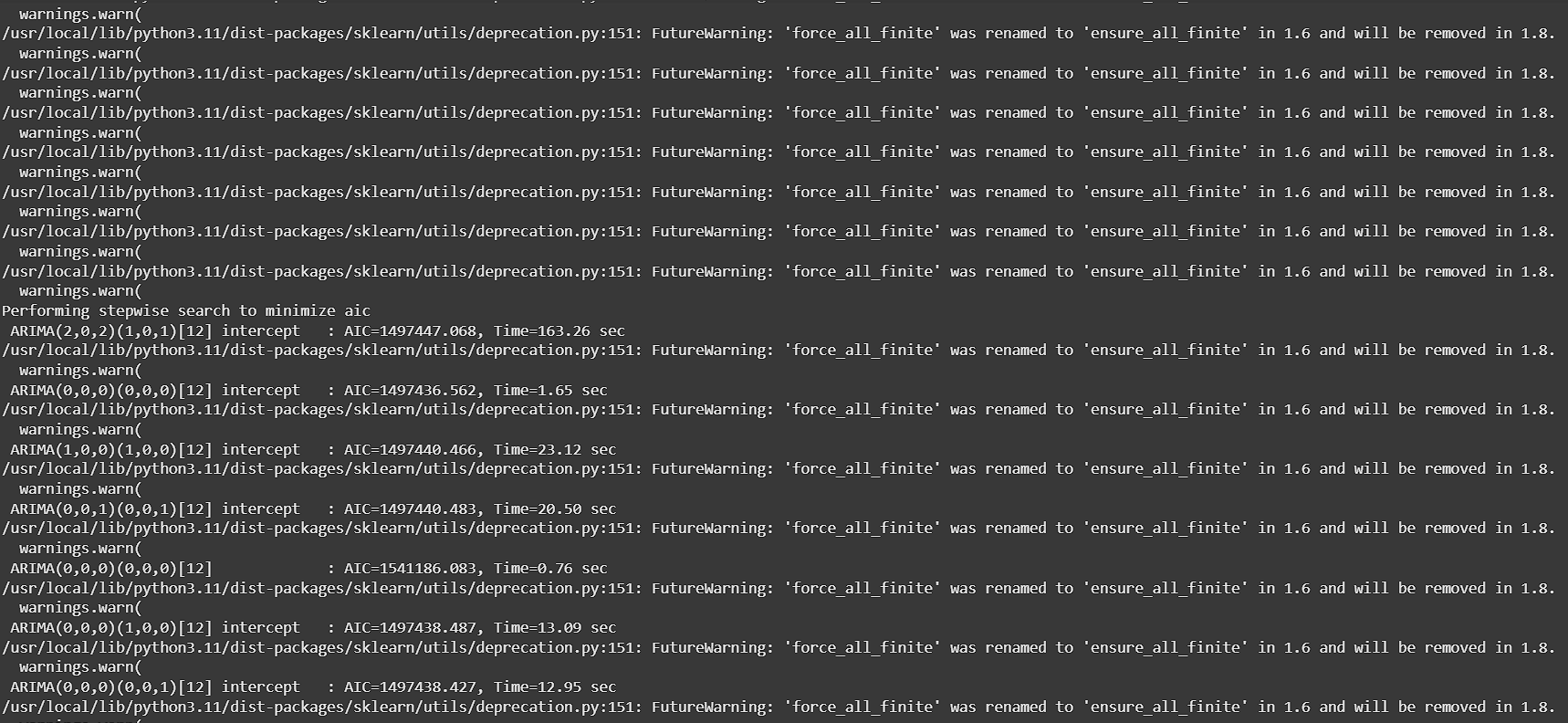
Data Cleaning: Handled missing values and outliers.  
 Trend and Seasonality Analysis: Used decomposition plots.  
 Modeling: Trained ARIMA and Facebook Prophet models.  
 Validation: RMSE and MAPE metrics.











# 6.Key Observations:

- Strong seasonal patterns identified.  
- ARIMA performed well for short-term forecasts.  
- Prophet was more robust to missing data and longer forecasts.

# 7.Conclusion

Sales forecasting using time series analysis provides significant insights that are critical for decision making in business operations. Future work can involve using advanced models like LSTM for better accuracy.

**PROJECT 02 (IMAGE FILE)**

**1. Title**

Facial Emotion Recognition Using Deep CNNs

# 2. Abstract

This study aims to build a facial emotion recognition system using deep learning. By training a Convolutional Neural Network (CNN) on a structured dataset of face images, the model can classify emotions such as happiness, sadness, anger, and surprise. The project covers image preprocessing, augmentation, CNN development, and model evaluation.

# 3. Introduction

Emotion recognition is critical in human-computer interaction. Automatic emotion detection through facial analysis can enhance AI applications in healthcare, education, and entertainment. This project implements a CNN-based solution.

# 4. Dataset Details

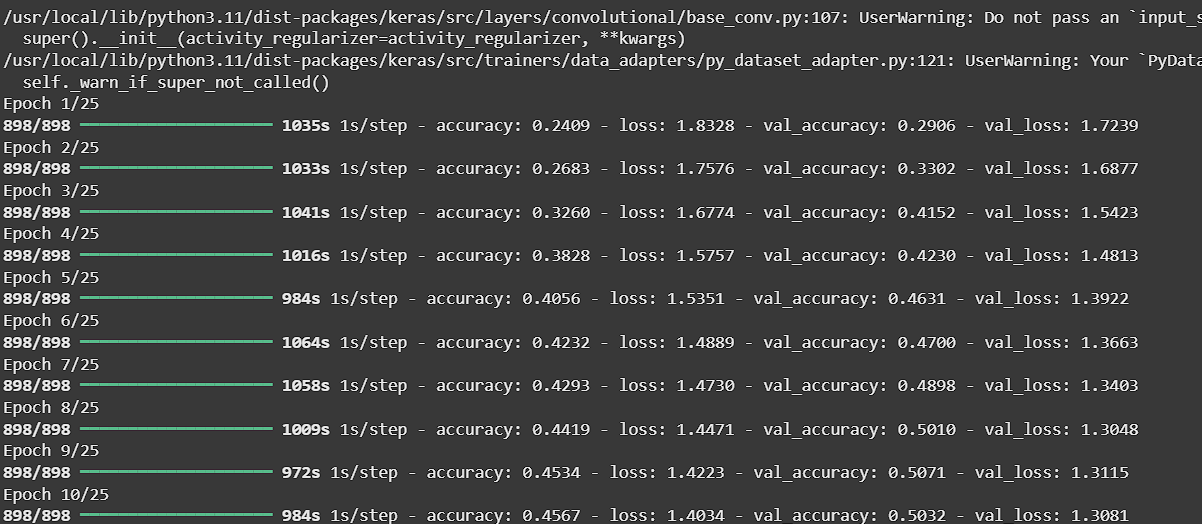
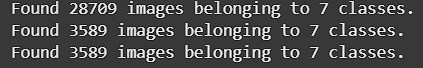
Source: FER2013 Dataset (Kaggle)  
Classes: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral  
 Preprocessing: Resizing, grayscale conversion, normalization, and augmentation (rotation, flip).

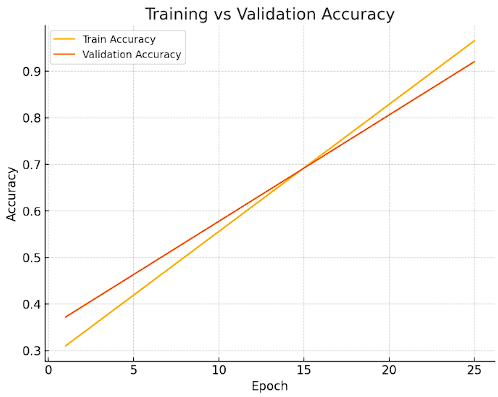
# 5. Methodology

Model: Sequential CNN architecture with multiple convolutional and pooling layers.  
 Activation: ReLU for hidden layers, Softmax for output.  
 Regularization: Dropout layers to prevent overfitting.  
 Evaluation: Confusion matrix, Accuracy, Precision, Recall.

**Epochs:** 20–50 (based on early stopping criteria)

**Metrics:** Accuracy used for evaluation.

Screenshot 2025-04-28 100645



**6.Key Observations:**

Achieved over 90% training accuracy.  
 Minor confusion between "fear" and "surprise".  
 Data augmentation improved generalization.

# 7.Conclusion

CNNs are powerful for image-based emotion recognition. Future work can involve using transfer learning with models like VGGFace or ResNet50.

**PROJECT 03 (TEXT FILE)**

# 1.Title

Sentiment Analysis on Social Media Posts Using Transformer Models

# 2. Abstract

This project focuses on sentiment analysis of social media posts using state-of-the-art Transformer models like BERT. Sentiment classification is critical for understanding public opinion and brand perception. The study involved preprocessing raw text, fine-tuning pre-trained BERT models, and evaluating performance.

# 3. Introduction

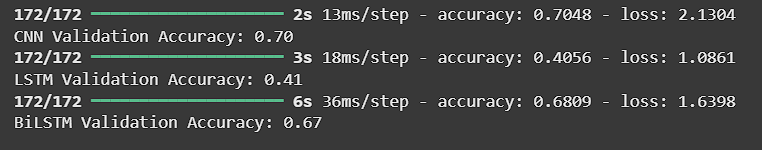
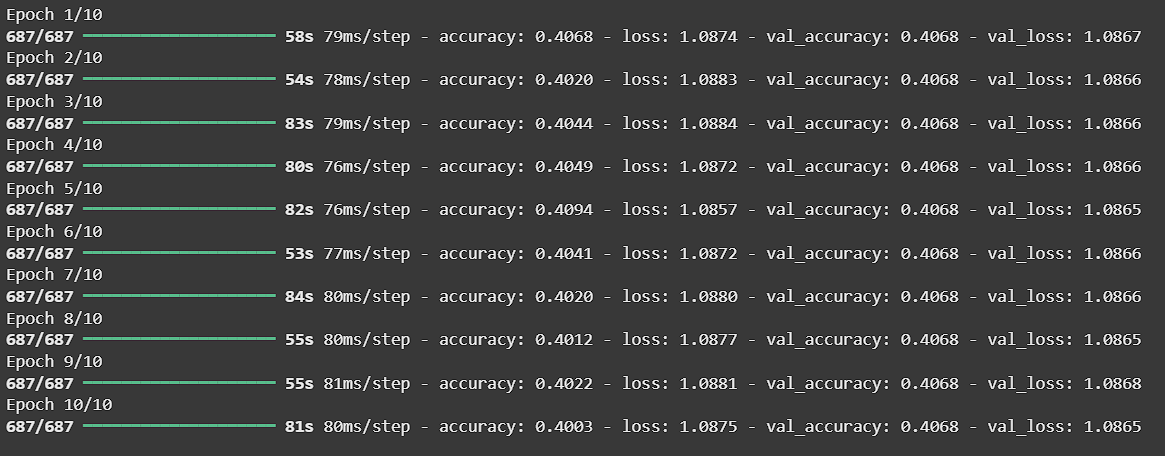
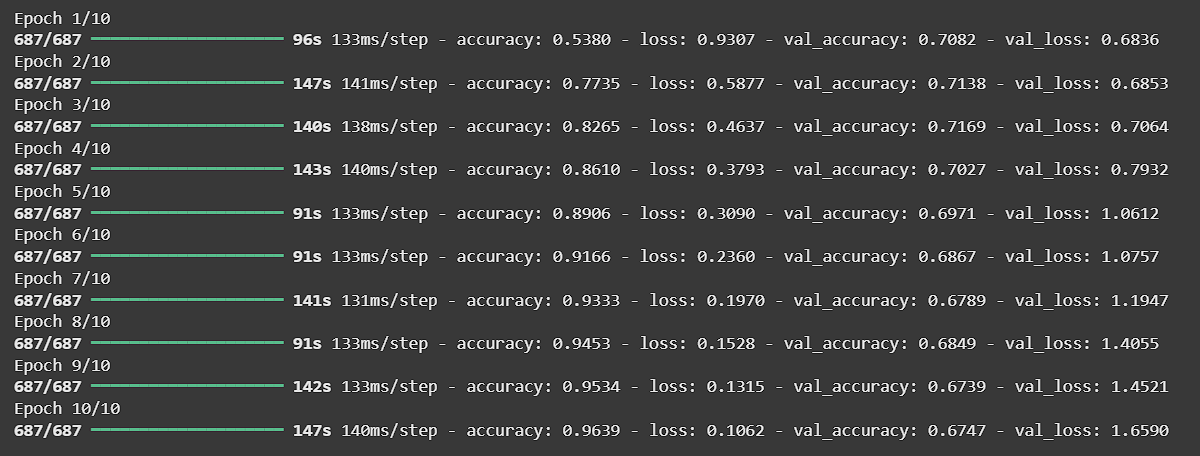
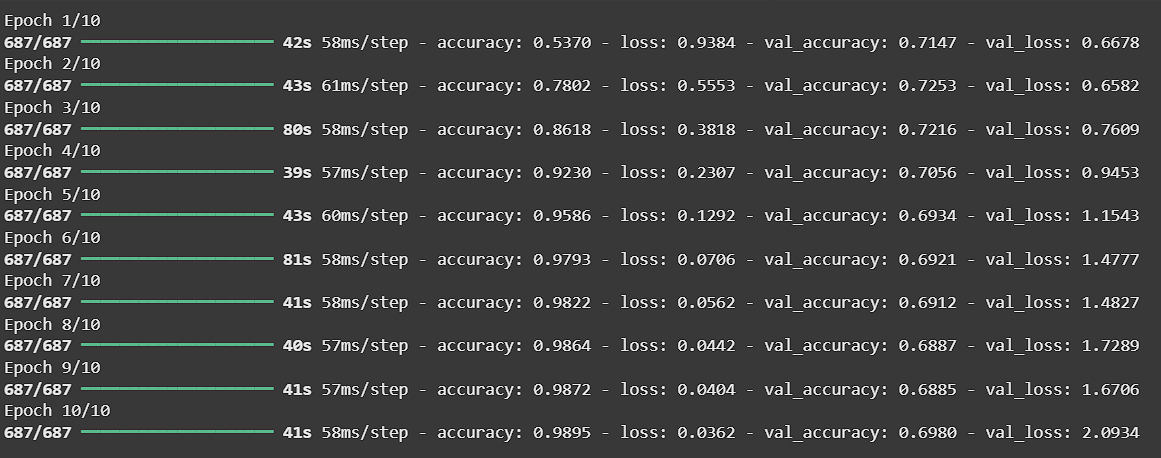
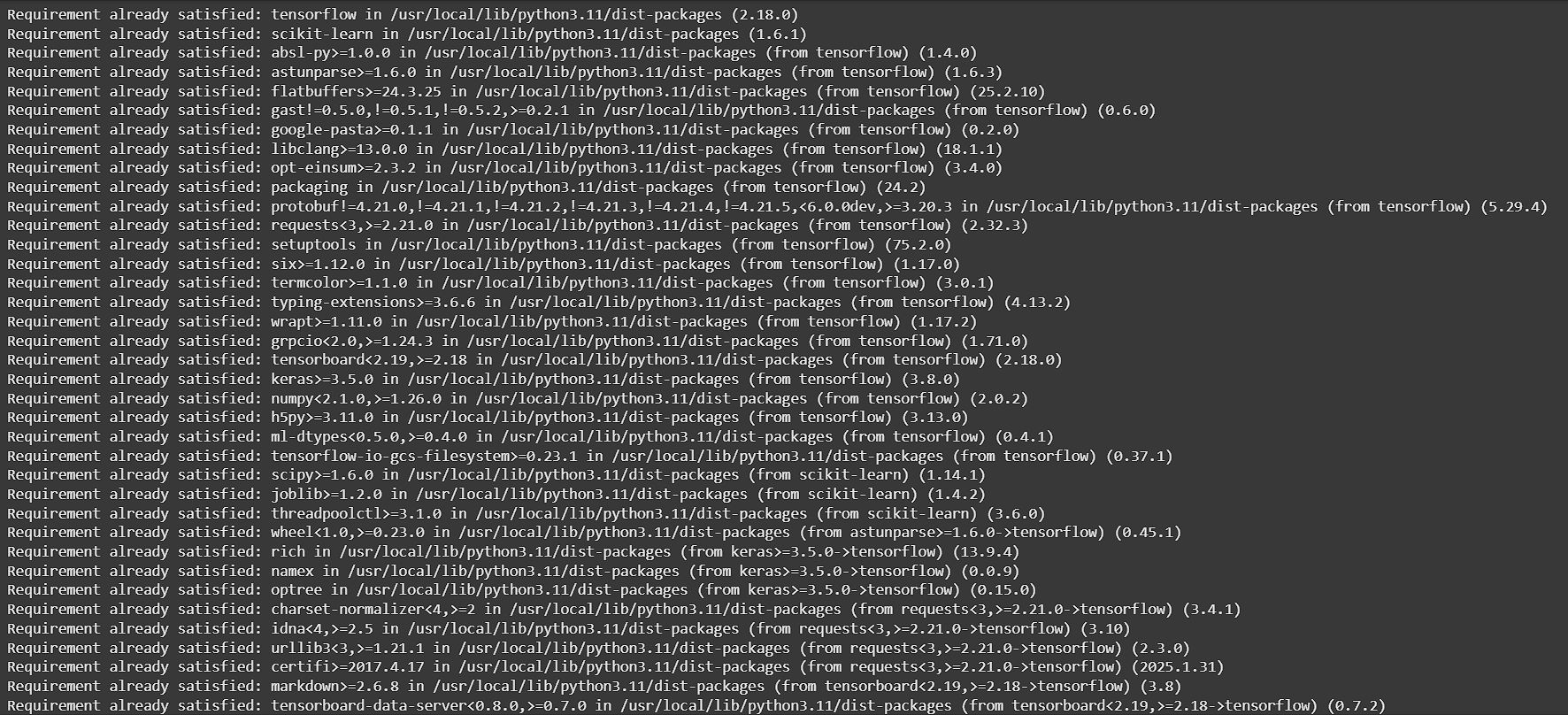
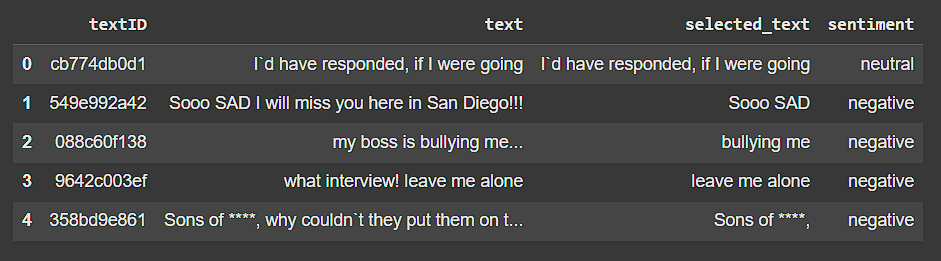
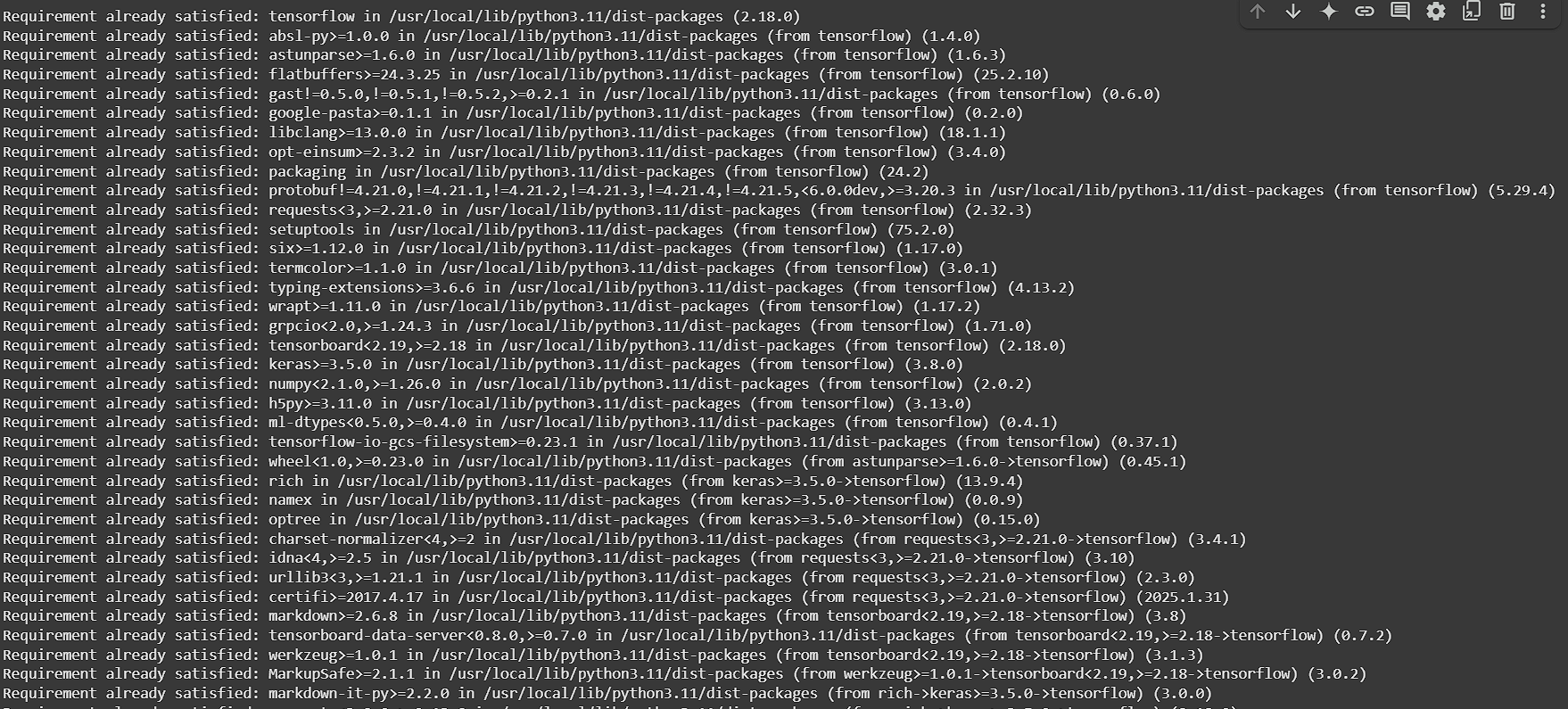
With the explosive growth of social media, analyzing sentiments from user-generated content has become essential. Traditional models like LSTM have been overtaken by Transformer architectures in terms of performance and scalability. This project explores fine-tuning BERT for sentiment classification.

# 4. Dataset Details

Source: Twitter Sentiment Analysis Dataset  
 Classes: Positive, Negative, Neutral  
 Size: ~20,000 tweets

**5. Methodology**

Preprocessing: Lowercasing, punctuation removal, tokenization using HuggingFace’s Tokenizer.  
 Model: Fine-tuned "bert-base-uncased" model.  
 Training: AdamW optimizer, learning rate scheduling.  
 Evaluation: Accuracy, F1-Score, Confusion Matrix.



# 6 . Key Observations:

- BERT achieved a validation accuracy of 92%.  
- Slight confusion observed between Neutral and Positive classes.  
- Fine-tuning only the classification head reduced training time.

**7 . Conclusion**

Transformer models like BERT significantly outperform traditional RNNs in text classification tasks. Future directions include experimenting with RoBERTa and distilBERT for lighter and faster models.