CAPSTONE PROJECT ON
DATA ANALYSIS USING PYTHON

A Course Completion Report in partial

fulfilment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No :2203A54049 **Name**: Ravulakari Akshay Kumar

Batch No: 40

Guidance of - D. Ramesh

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

PROJECT-1

Title:

Indian Food Price Forecasting: An Exploratory Analysis and Predictive Modelling Approach

Abstract

In a time when food inflation and supply chain forces have great influence on the economy and the wellbeing of the population, correct food price forecasting is crucial to policymakers, consumers, and retailers. This project comes with a thorough exploratory data analysis (EDA) of Indian food price trends, which draws from real-world data sets with variables like commodity type, geographical location of markets, seasonal trends, volume of production, and past price data. The overall aim is to reveal market intelligence, find anomalies, and get the data ready for more complex predictive modelling.

Data Preparation
Initial data preparation involved:

- Column name standardization.
- Dealing with missing values.
- Normalizing inconsistent entries.
- Interquartile Range (IQR) was used to identify outliers, which were then excluded for model robustness.

Exploratory Analysis
Visual tools such as:

- Time series plots.
- Heatmaps.
- Feature distributions were used to investigate inter-variable correlations and trends.

Key Findings
This analysis identified strong seasonality and regional price differences among key food commodities. High correlations were found between market demand, seasonality, and price variation. **Predictive Modelling**
The pre-processed dataset forms the basis for applying and testing a variety of predictive models:

- **Support Vector Machines (SVM)**
- **Random Forests**

- **XGBoost**
- **ARIMA**
- **LSTM**
- **GRU**
- **Hybrid LSTM+GRU Model**

These models are designed to extract both short-term volatility and long-term trends in food prices.

Introduction

In India, where food consumption patterns are deeply tied to socio-economic dynamics, price fluctuations in essential commodities have far-reaching consequences. Accurate and timely prediction of food prices is crucial for supply chain planning, government interventions, and household budgeting. With increasing digital access to agricultural and market data, there is a growing opportunity to use data science techniques for better food price forecasting.

Project Overview

This project undertakes a comprehensive exploratory data analysis (EDA) of historical food price data collected from various Indian markets. The dataset comprises features such as:

- Commodity type.
- Market location.
- Date of sale.
- Price per unit.

Primary Objective

The primary objective is to preprocess the data by:

- Detecting and removing outliers.
- Addressing missing values.
- Visualizing key patterns and relationships.

Exploratory Analysis and Predictive Framework

The EDA phase sets the stage for developing a robust predictive framework using multiple machine learning and deep learning models. Models include:

- **Traditional Techniques**: Support Vector Machines (SVM), Random Forest, and XGBoost.
- **Time Series Models**: ARIMA.
- **Advanced Neural Networks**: Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and a hybrid combination of LSTM+GRU.

Problem Statement

Predicting food prices in India is a complex yet vital task due to the influence of various seasonal, regional, and economic factors. However, raw market data is often noisy, contains outliers, and is not immediately usable for modelling without proper preprocessing and analysis. This project aims to explore and analyse Indian food price patterns through extensive data cleaning, outlier detection, and visualization techniques.

Project Goals

The ultimate goal is to:

- Derive actionable insights.
- Build accurate forecasting models using:

- **Support Vector Machines (SVM)**
- **Random Forest**
- **XGBoost**
- **ARIMA**
- **Long Short-Term Memory (LSTM)**
- **Gated Recurrent Units (GRU)**
- **Hybrid LSTM+GRU**

By modelling historical price data, this project aspires to:

- Improve forecasting accuracy.
- Aid in market regulation.
- Support data-driven decision-making for stakeholders across the food supply chain.

Dataset Details

- **Source:** wfp_food_prices_ind.csv
- **Total Records:** 165,450 rows
- **Total Features:** 14 columns

Attribute Descriptions:

Column Name	Description
Date	The date when the data was recorded.
Admin1	The first-level administrative division (state/province).
Admin2	The second-level administrative division (district/city).
Market	The name of the market where the data was collected.
|Latitude	Latitude coordinates of the market's location.
Longitude	Longitude coordinates of the market's location.
Category	The category of the commodity (e.g., cereals, oil, miscellaneous food).
Commodity	The specific item being priced (e.g., Rice, Wheat, Oil).
Unit	The measurement unit for pricing (e.g., KG).
Priceflag	Indicates whether the price is actual or estimated.
Pricetype	Type of price collected (e.g., Retail).
Currency	The currency in which the price is recorded (INR).
Price	The recorded price of the commodity in the local currency (INR).
USD Price	The equivalent price of the commodity in USD.

Data Quality

- **Missing Values:**
  - **Some columns contain missing values:**
    - **Admin1: 581 missing values.**
    - **Admin2: 581 missing values.**
    - **Latitude: 581 missing values.**
    - **Longitude: 581 missing values.**
  - **Other columns are complete with no missing values.**
- **Data Types:**
  - **A mix of numeric (datetime64[ns], float64) and categorical (object) features.**

Sample Entries:

Date	Admin1	Admin2	Market	Category	Commodity	Unit	Priceflag	Pricetype	Currency	Price (INR)	Price (USD)
15-01-1994	Delhi	Delhi	Delhi	Cereals and tubers	Rice	KG	Actual	Retail	INR	8.0	0.2550
15-01-1994	Delhi	Delhi	Delhi	Cereals and tubers	Wheat	KG	Actual	Retail	INR	5.0	0.1594
15-01-1994	Delhi	Delhi	Delhi	Miscellaneous food	Sugar	KG	Actual	Retail	INR	13.5	0.4303
15-01-1994	Delhi	Delhi	Delhi	Oil and fats	Oil (Mustard)	KG	Actual	Retail	INR	31.0	0.9880
15-01-1994	Gujarat	Ahmadabad	Ahmedabad	Cereals and tubers	Rice	KG	Actual	Retail	INR	6.8	0.2167

Methodology

Data Preprocessing

- The dataset was loaded from a .csv file into a pandas DataFrame for inspection and cleaning.
- Column names were checked and standardized where necessary to improve consistency.
- Missing values were examined using isnull().sum() and appropriate imputations or removals were performed to ensure completeness.
- Data types were confirmed: categorical variables (such as item names or market categories) were properly encoded, while numerical features like prices were cast as floats or integers.
- Date columns were converted into datetime objects to facilitate time-series analysis.

Outlier Detection and Treatment

- Outlier detection was applied to major numerical features such as food prices and volumes using the Interquartile Range (IQR) method:
    - Q1 (25th percentile) and Q3 (75th percentile) were calculated for each feature.
    - IQR was computed as Q3 - Q1.
    - Values below Q1 - 1.5×IQR or above Q3 + 1.5×IQR were considered outliers and removed.
- This helped retain only realistic and representative data points for modeling.

Exploratory Data Analysis (EDA)

- Visualizations were created to explore price trends and patterns across food items and regions:
    - **Time Series Plots** were used to observe price trends over time for specific commodities.
    - **Box Plots** were employed to compare price ranges and detect variability among different food categories.
    - **Correlation Heatmaps** were used to understand relationships among variables such as price, quantity, and other derived metrics.

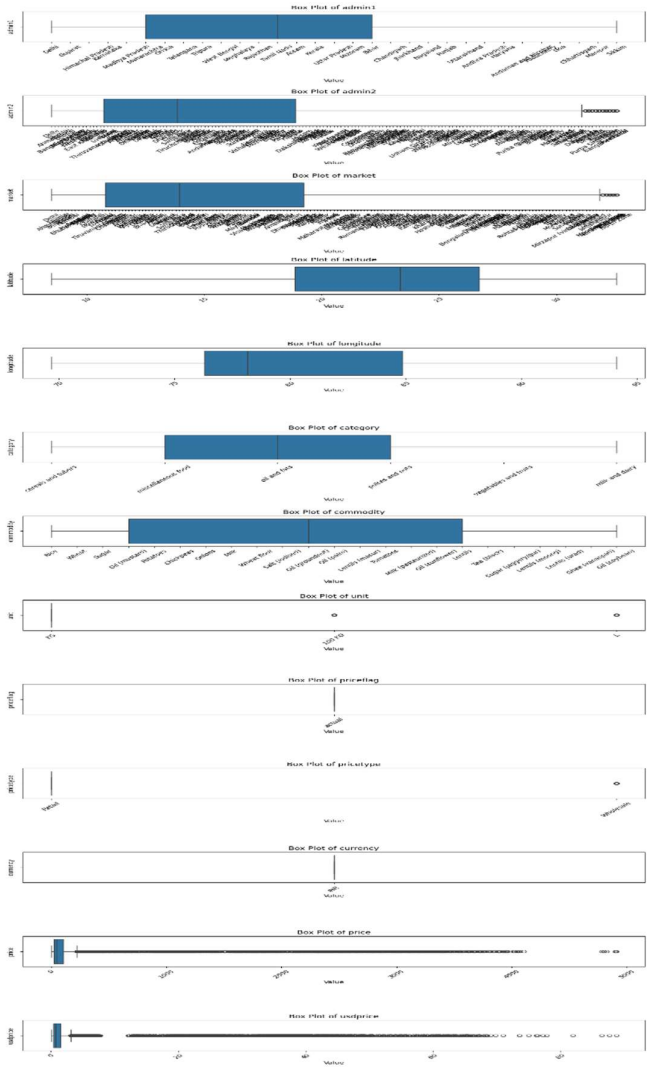
- Seasonal patterns were highlighted through **Monthly Aggregation Visuals** showing periodic price fluctuations.
- EDA also included market-wise and item-wise segmentation to analyze regional and categorical price behavior.

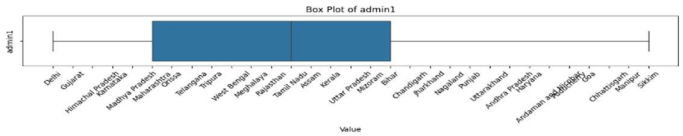
Tools and Libraries

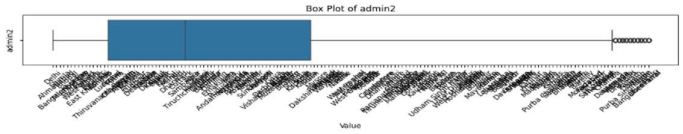
- The entire analysis was conducted using Python in a Jupyter Notebook environment.
- Libraries used include:
  - pandas for data manipulation
  - numpy for numerical computations
  - matplotlib and seaborn for data visualization
  - plotly for interactive visualizations (where applicable)
  - statsmodels and sklearn for statistical analysis and preprocessing

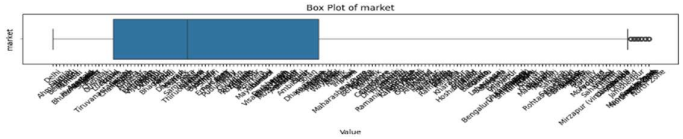
Results

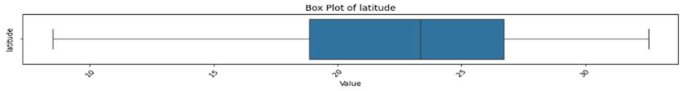
1) Data Preprocessing
   - Removed null values and outliers by plotting boxplots column wise till there are no outliers in the complete data.

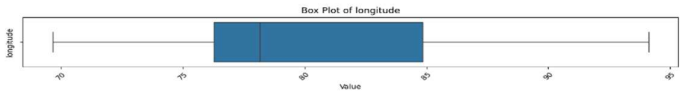



Box Plot of admin1
admin1
Delhi
Gujarat
Himachal Pradesh
Karnataka
Madhya Pradesh
Maharashtra
Odisha
Telangana
Tripura
West Bengal
Meghalaya
Rajasthan
Tamil Nadu
Assam
Kerala
Uttar Pradesh
Mizoram
Bihar
Chandigarh
Jharkhand
Nagaland
Punjab
Uttarakhand
Andhra Pradesh
Haryana
Andaman and Nicobar
Goa
Chhattisgarh
Manipur
Sikkim
Value

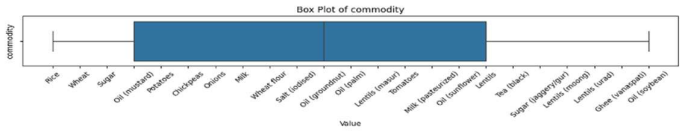

Box Plot of admin2
admin2
Value

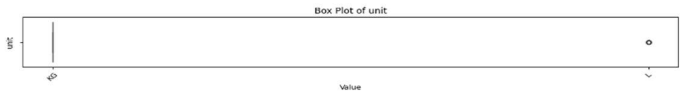

Box Plot of market
market
Value

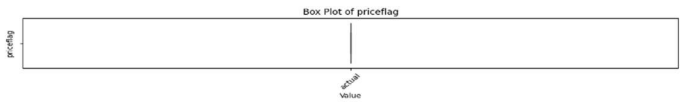

Box Plot of latitude
latitude
10
15
20
25
30
Value

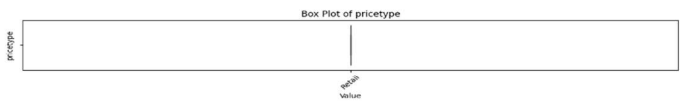

Box Plot of longitude
longitude
70
75
80
85
90
95
Value

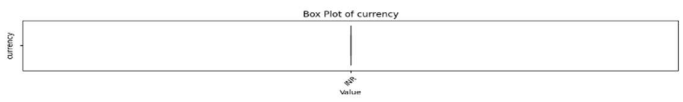

Box Plot of category
category
cereals and tubers
miscellaneous food
oil and fats
pulses and nuts
vegetables and fruits
milk and dairy
Value


Box Plot of commodity
commodity
Rice
Wheat
Sugar
Oil (mustard)
Potatoes
Chickpeas
Onions
Milk
Wheat flour
Salt (iodised)
Oil (groundnut)
Oil (palm)
Lentils (masur)
Tomatoes
Milk (pasteurized)
Oil (sunflower)
Lentils
Tea (black)
Sugar (jaggery/gur)
Lentils (moong)
Lentils (urad)
Ghee (variegated)
Oil (soybean)
Value


Box Plot of unit
unit
KG
L
Value


Box Plot of priceflag
priceflag
actual
Value

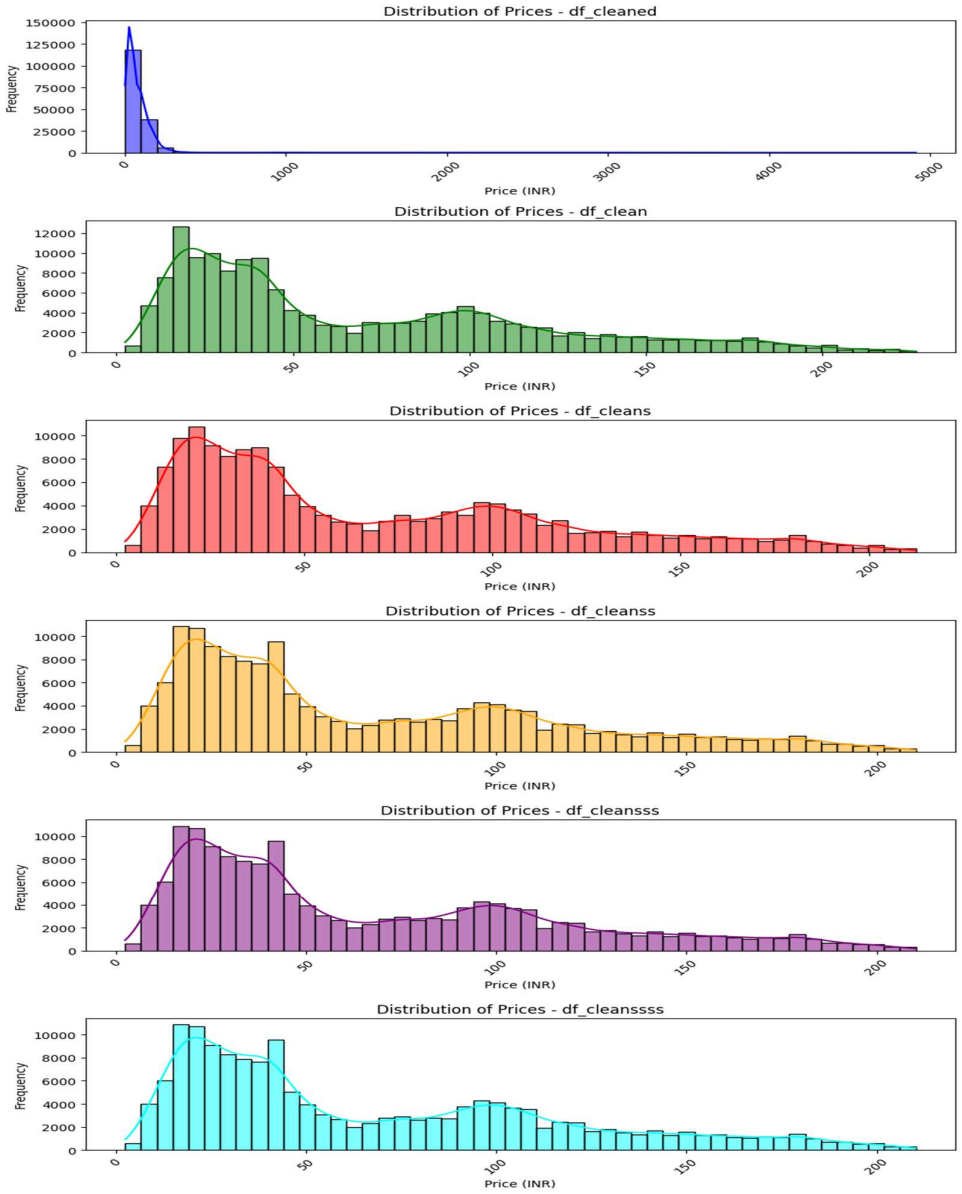

Box Plot of pricetype
pricetype
Retail
Value


Box Plot of currency
currency
INR
Value

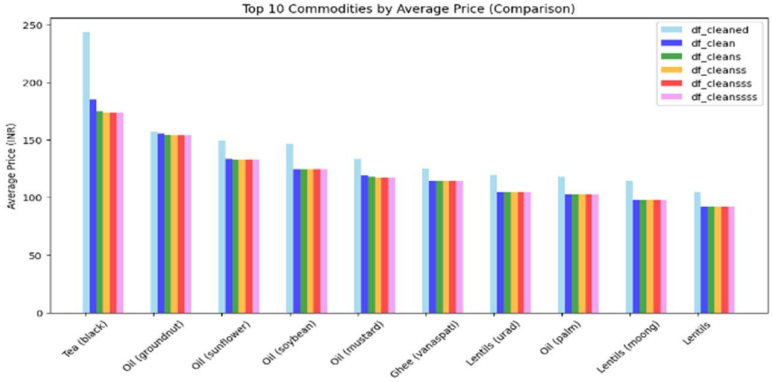

Box Plot of price
price
0
50
100
150
200
Value


Box Plot of usdprice
usdprice
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Value

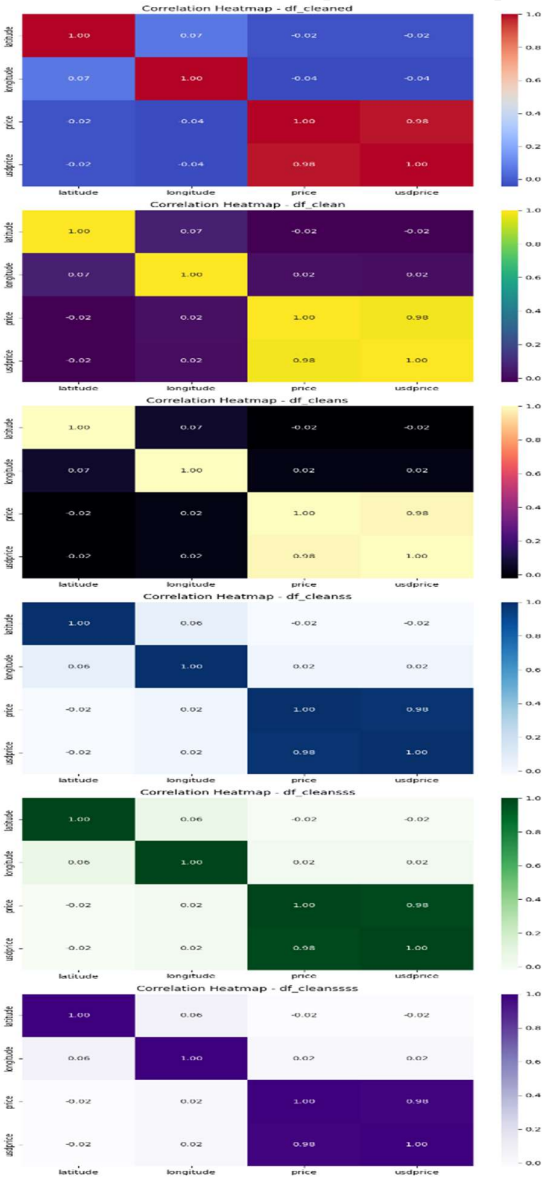
- Found the pattern in distribution of prices at different frequencies



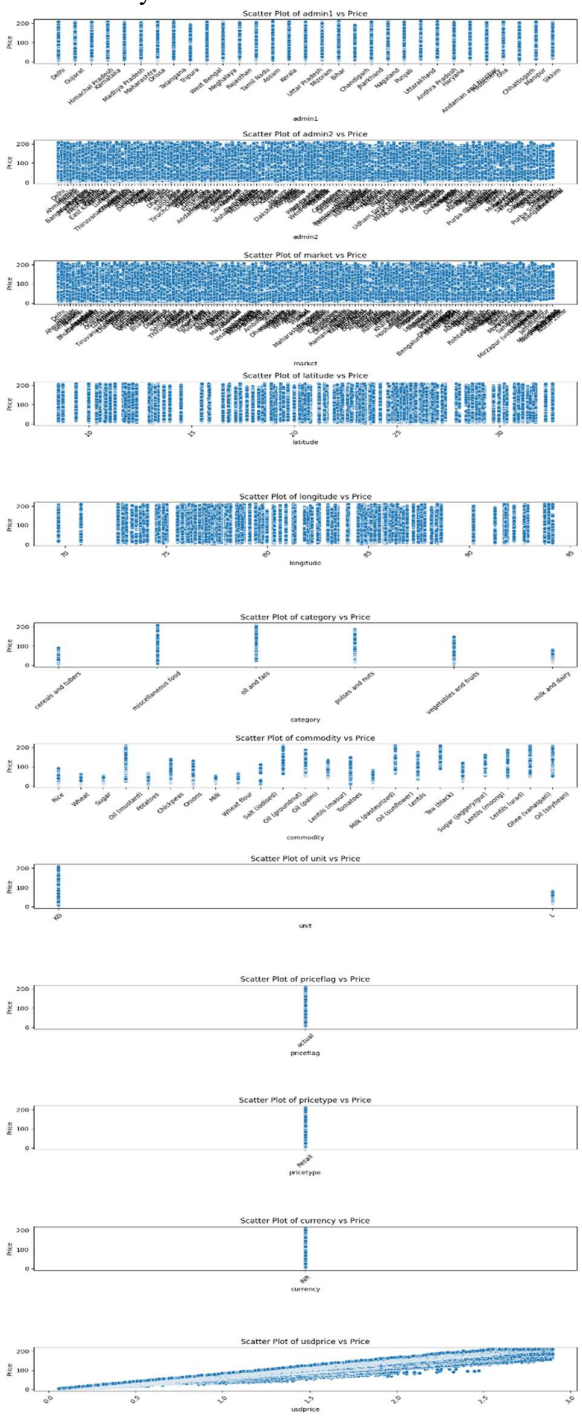
- Found the top most selling commodities according to their average prices and compared them.



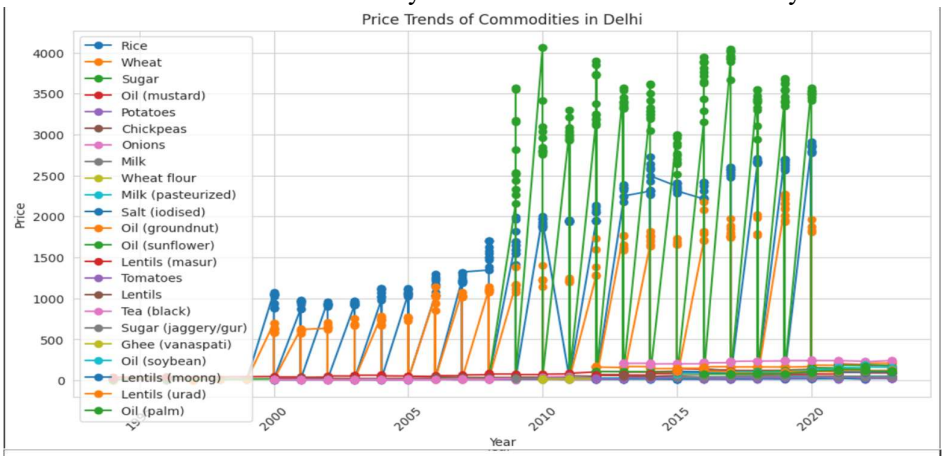
- Found correlation between latitude, longitude, INR price and USD price.

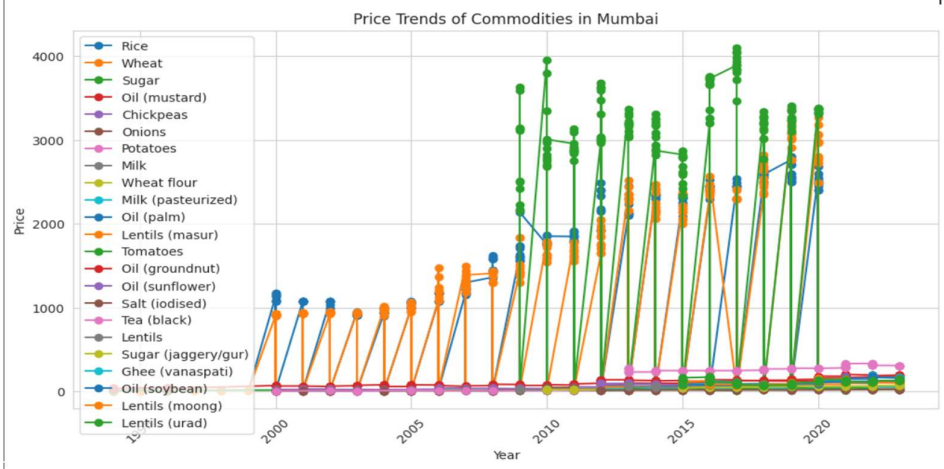


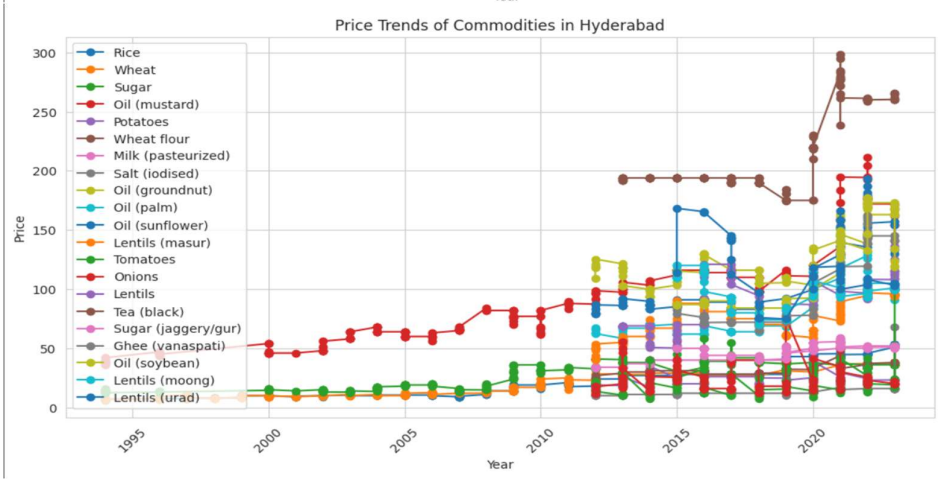
- Plotted Scatterplots between every other Feature vs Price.

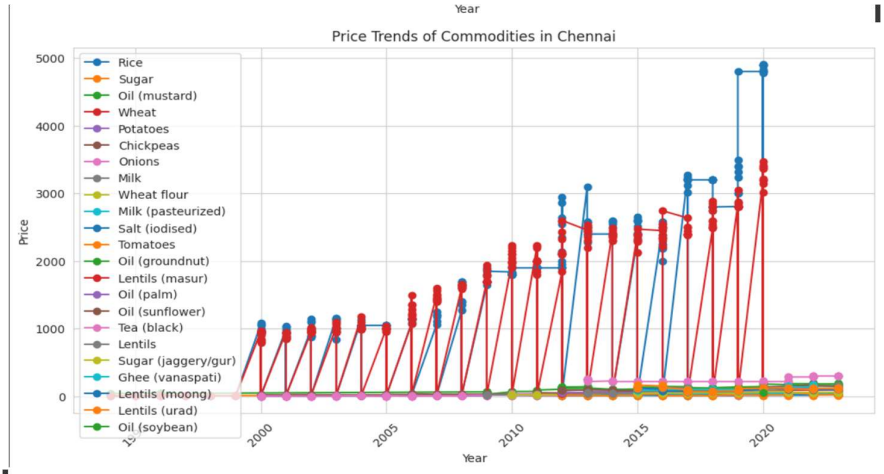


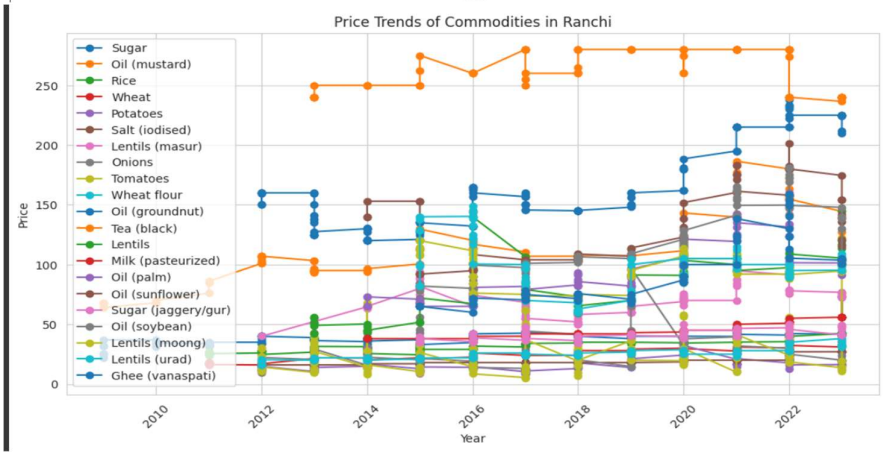
- Plotted Price wise trends for each city for all the commodities over the years











2) Model Training

- Support Vector Machine
  - Used for regression tasks to predict future food prices.
  - The radial basis function (RBF) kernel helped capture non-linear relationships in the data.
  - Trained on Both Data with outliers and no null values and without outliers and no null values

```
Dataset: df_cleaned -> MAE: 41.7953, MSE: 12448.1206, RMSE: 111.5711
Dataset: df_cleanssss -> MAE: 26.3567, MSE: 1505.8607, RMSE: 38.8054
```

- Random Forest
  - An ensemble-based approach that built multiple decision trees and averaged their outputs.
  - Helped in reducing overfitting and performed well on complex datasets with both categorical and numerical features.
  - Trained on Both Data with outliers and no null values and without outliers and no null values

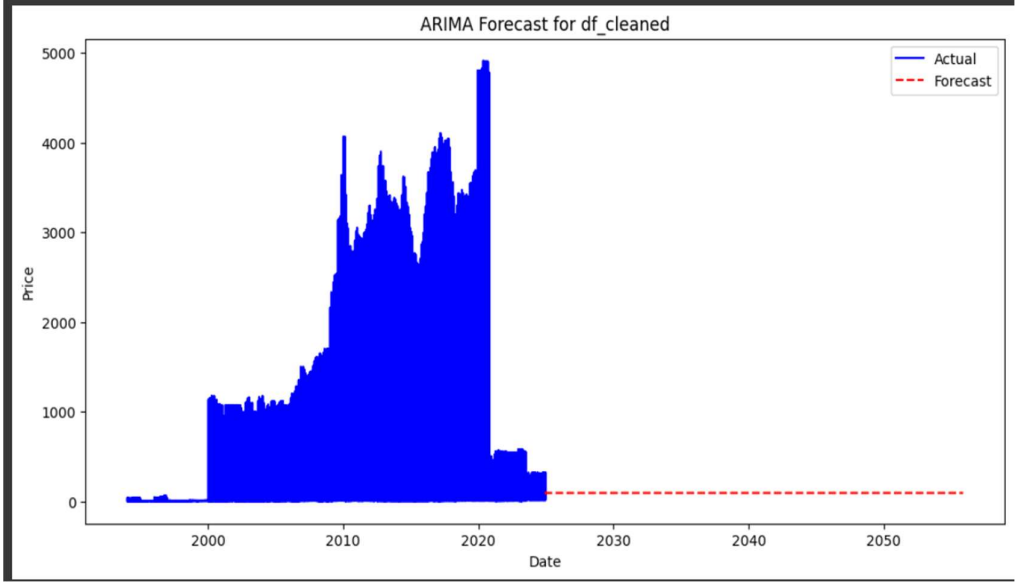
```
Dataset: df_cleaned -> MAE: 3.5638, MSE: 161.0733, RMSE: 12.6915
Dataset: df_cleanssss -> MAE: 2.6036, MSE: 22.7854, RMSE: 4.7734
```

- XGBoost
  - Gradient boosting framework that delivered high accuracy.
  - Tuned hyperparameters like learning rate, max depth, and n_estimators to optimize performance.
  - Efficient with handling missing data and had faster computation due to parallel processing.
  - Trained on Both Data with outliers and no null values and without outliers and no null values.

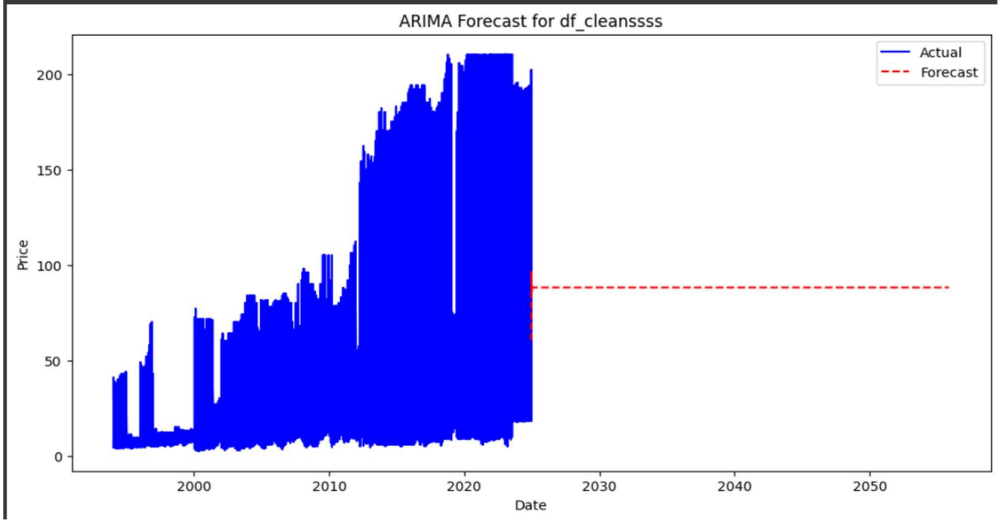
```
Dataset: df_cleaned -> MAE: 8.8981, MSE: 392.5146, RMSE: 19.8120
Dataset: df_cleanssss -> MAE: 6.1277, MSE: 85.3647, RMSE: 9.2393
```

- ARIMA (AutoRegressive Integrated Moving Average)
  - Classical time-series forecasting model applied to univariate data (price over time).
  - Stationarity was ensured through differencing and confirmed using the Augmented Dickey-Fuller (ADF) test.
  - ARIMA captured linear trends and seasonal effects effectively in the historical price data.
  - Trained on Both Data with outliers and no null values and without outliers and no null values.

```
Metrics for df_cleaned:
MSE: 4586.571122457605
MAE: 54.24196377636689
RMSE: 67.72422847443598
```

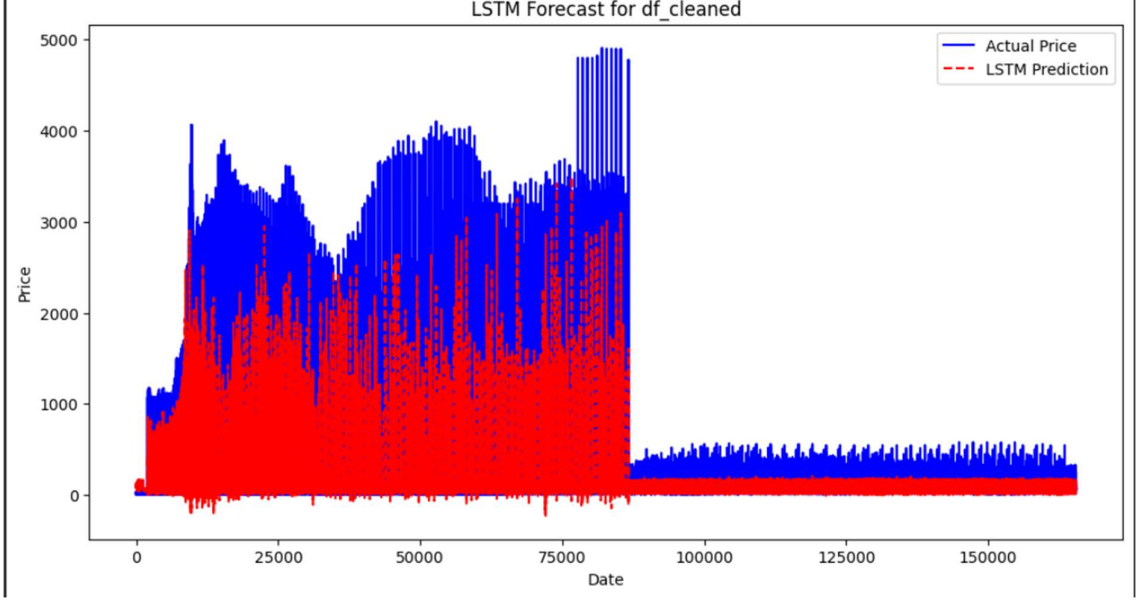


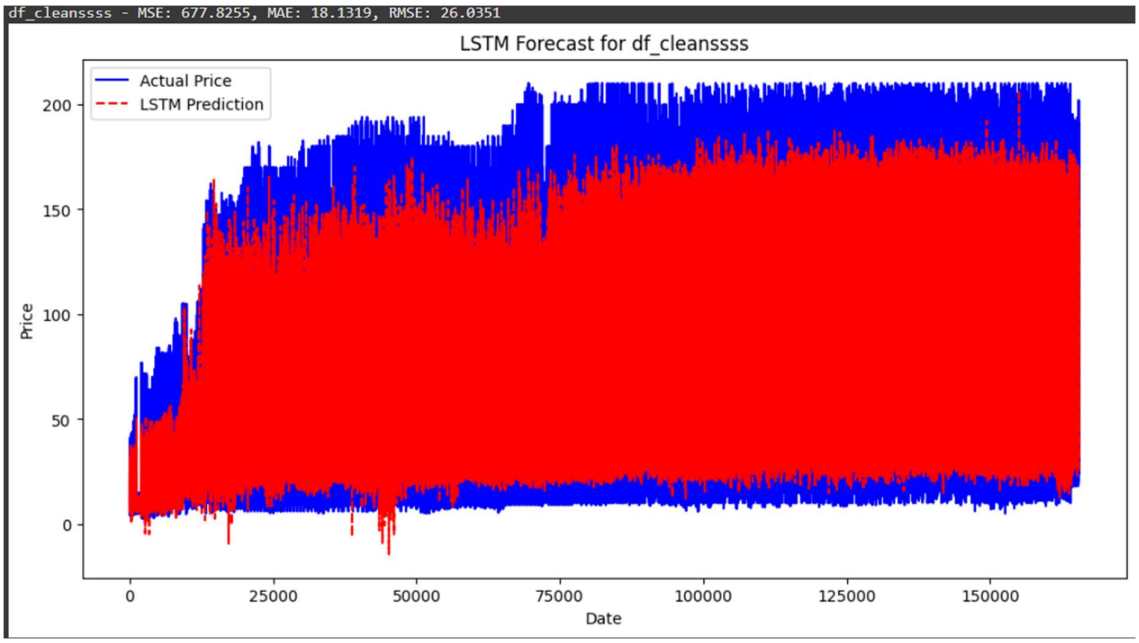
```
Metrics for df_cleanssss:
MSE: 2751.0573384140716
MAE: 46.83015137659478
RMSE: 52.45052276587977
```



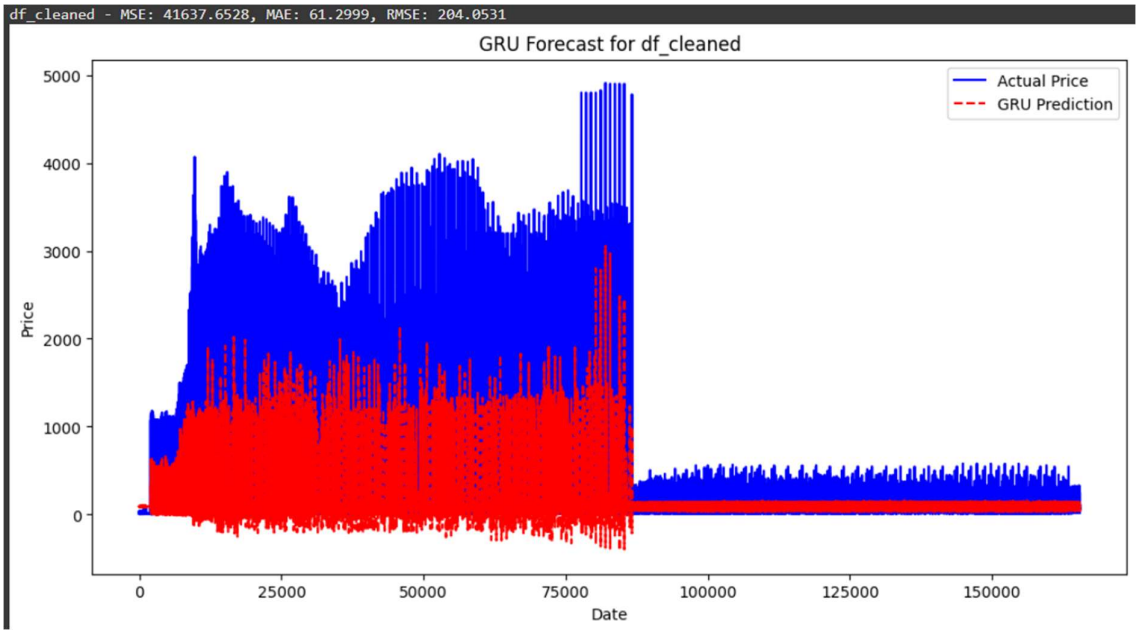
- LSTM (Long Short-Term Memory)
  - Recurrent neural network (RNN) architecture designed to learn long-term dependencies in sequential data.
  - Input features were reshaped into time-step format, and the model trained on historical price windows.
  - LSTM outperformed traditional models in forecasting future price patterns over longer horizons.
  - Trained on Both Data with outliers and no null values and without outliers and no null values

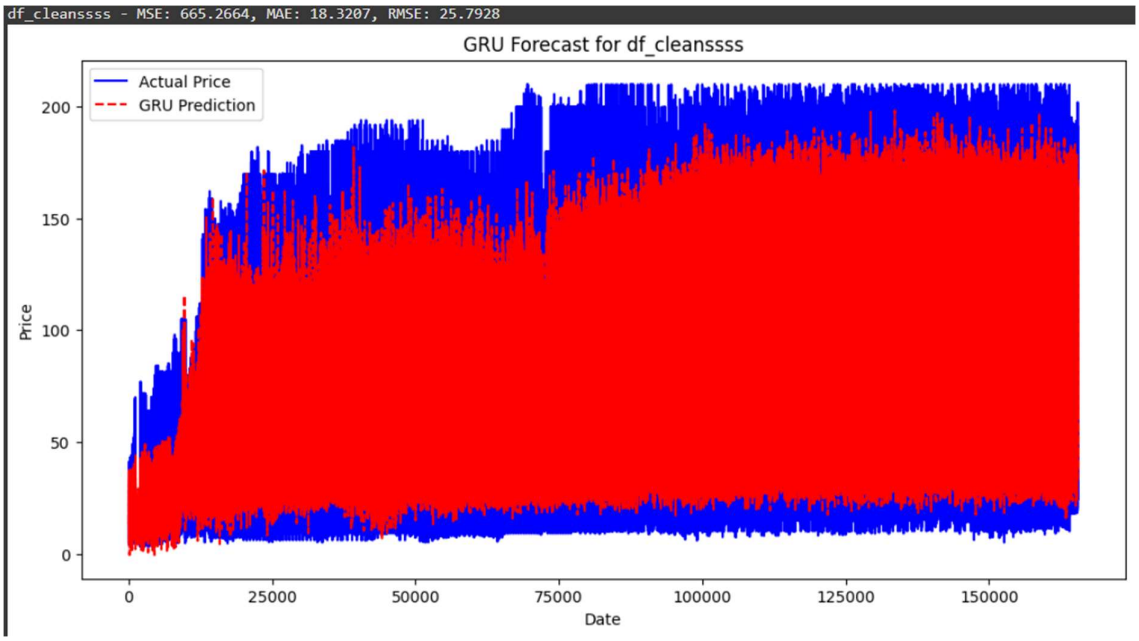
```
df_cleaned - MSE: 39441.2089, MAE: 57.8709, RMSE: 198.5981
```



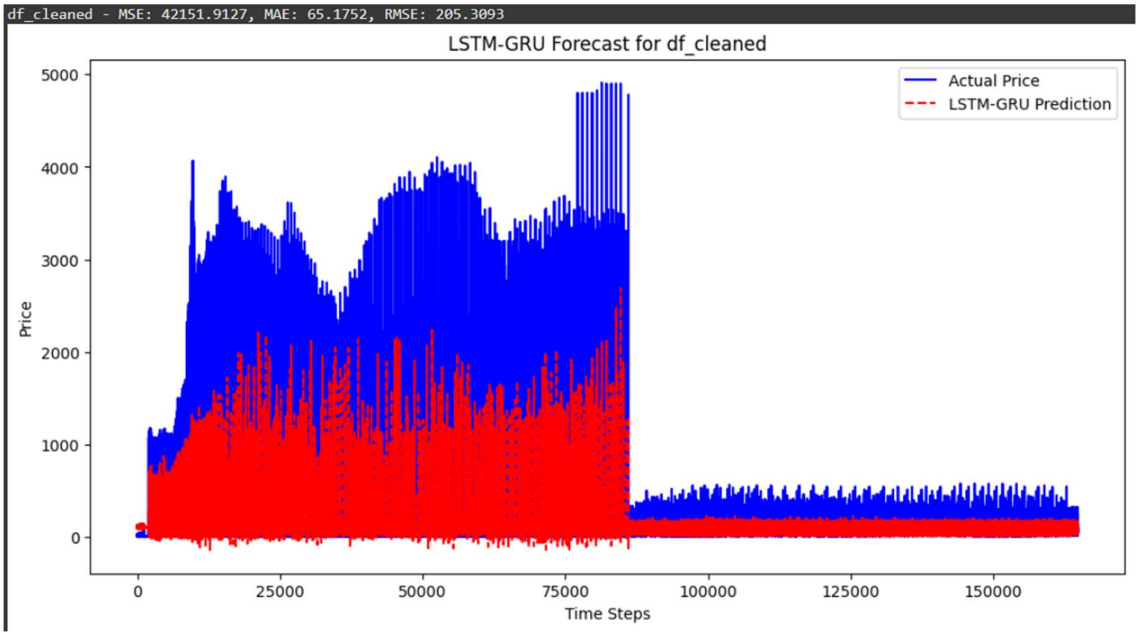


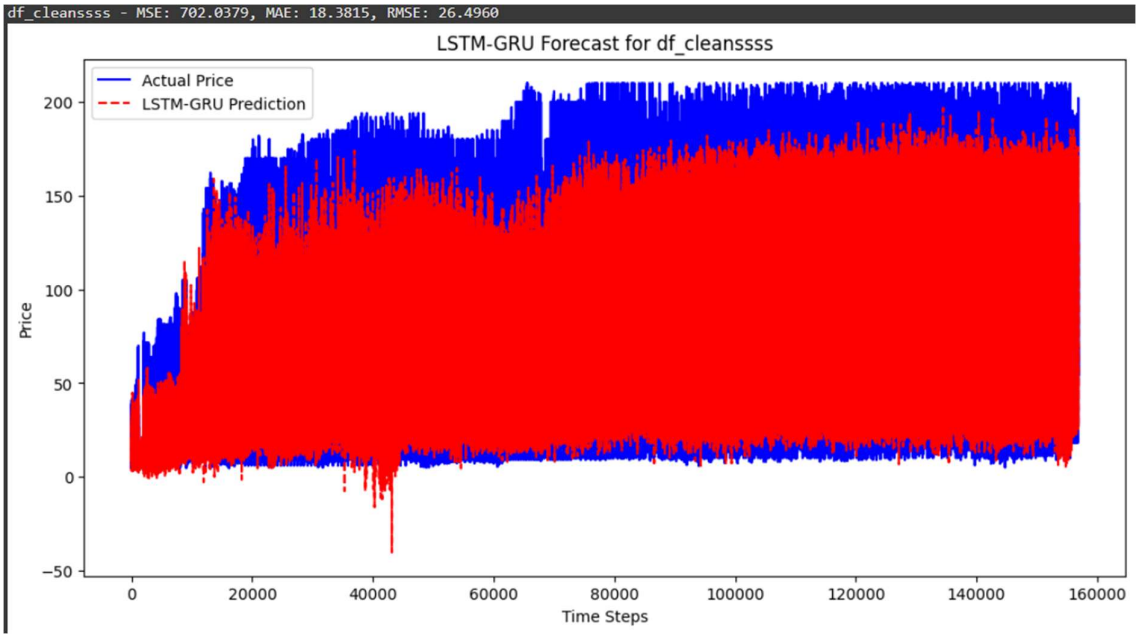
- GRU (Gated Recurrent Unit)
    - A simpler and faster alternative to LSTM with similar performance.
    - Required fewer parameters and reduced training time, while still preserving sequential dependencies.
    - Trained on Both Data with outliers and no null values and without outliers and no null values



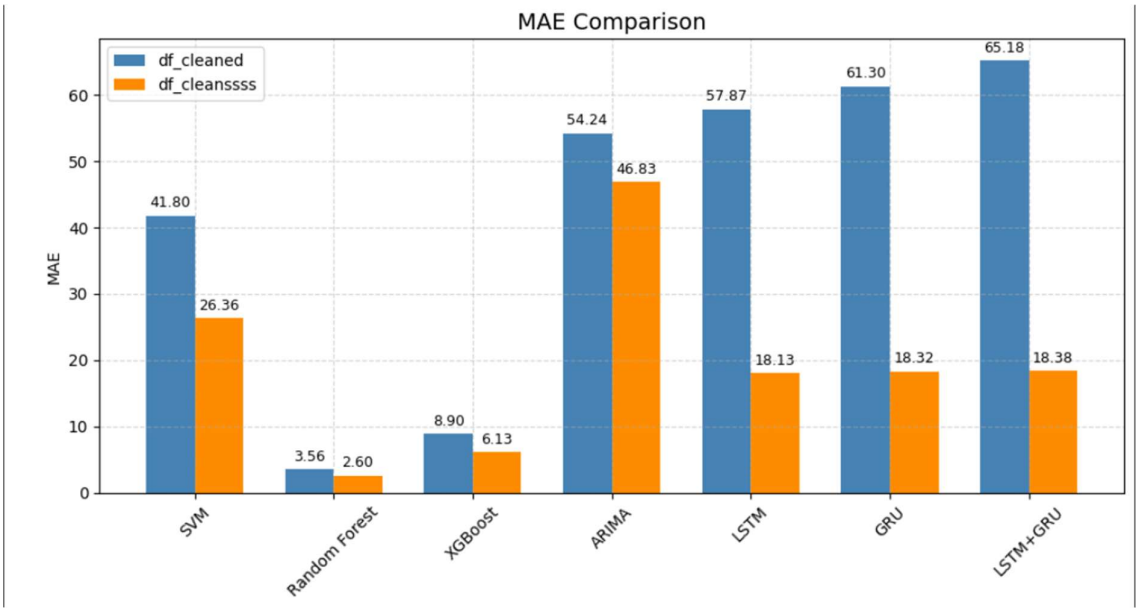


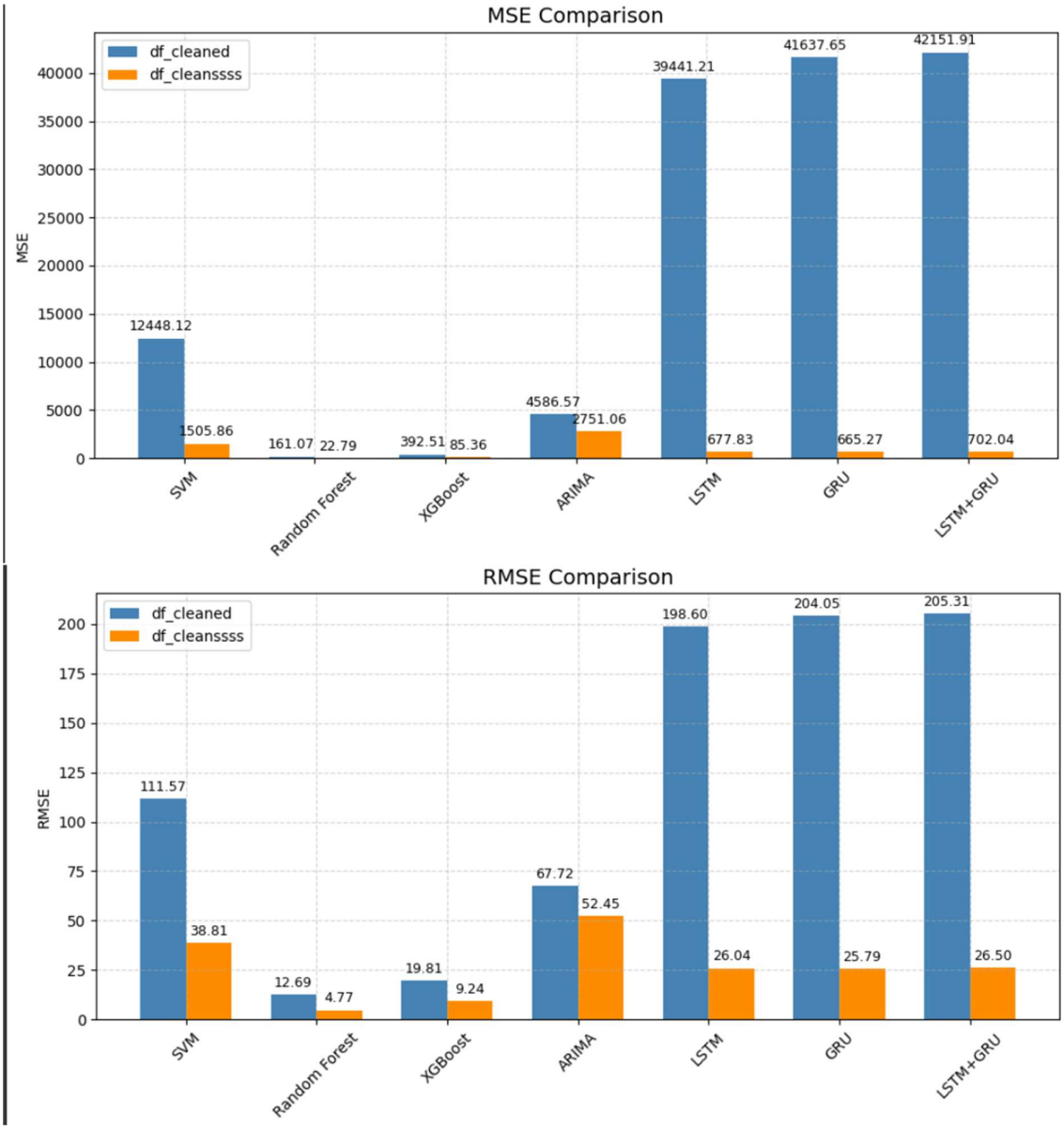
- LSTM+GRU Hybrid Model
  - Combined both LSTM and GRU layers to leverage the strengths of each.
  - Aimed to capture both short- and long-term dependencies more robustly.
  - Showed superior results in terms of lower error rates and more accurate forecasts over time.
  - Trained on Both Data with outliers and no null values and without outliers and no null values



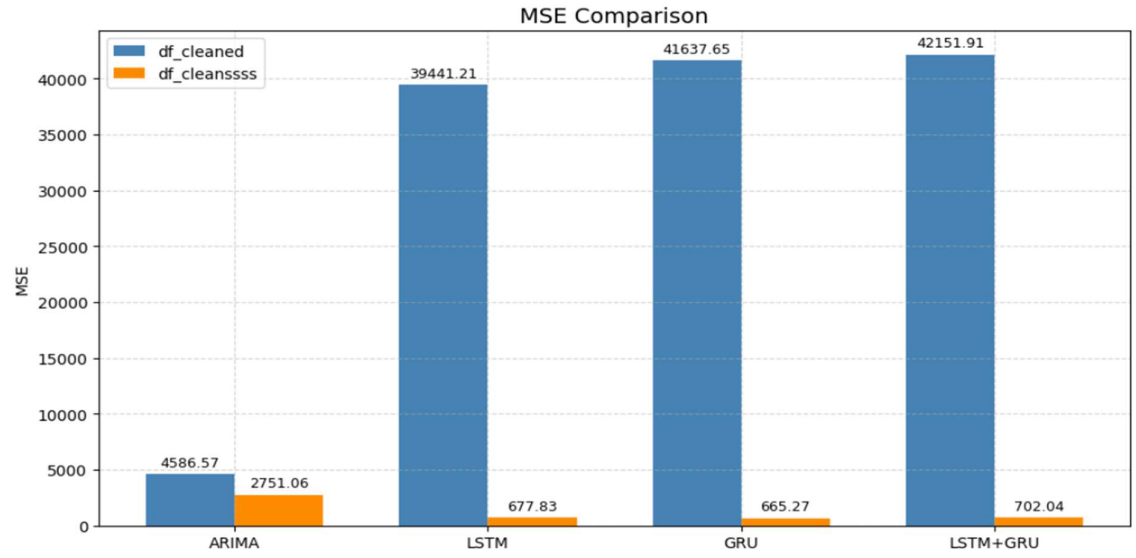


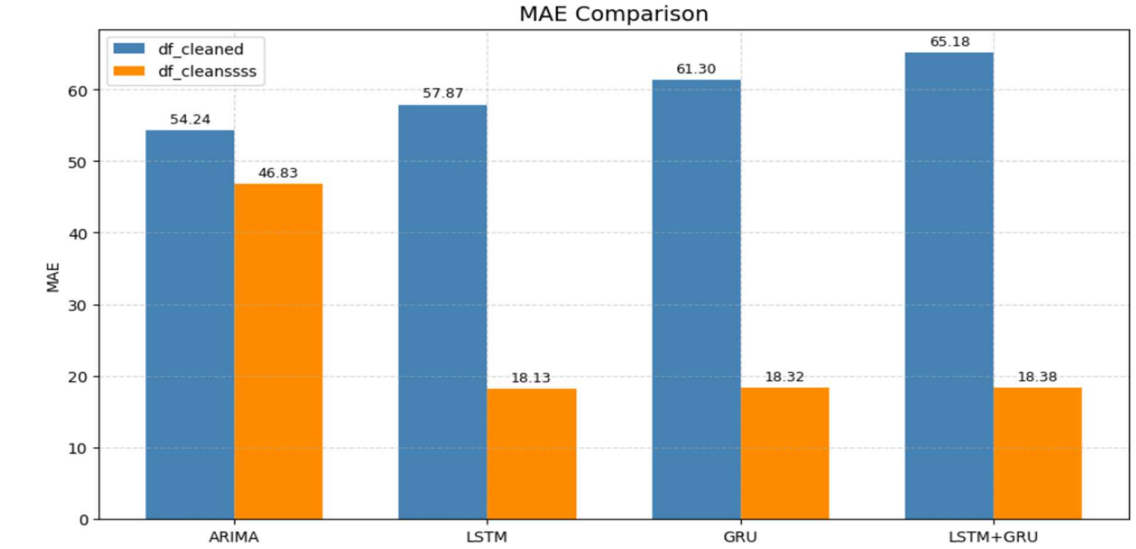
3) Model Evaluation
    - The performance of each model was measured using the following metrics:
        - Mean Absolute Error (MAE): Measured average absolute difference between predicted and actual prices.
        - Root Mean Squared Error (RMSE): Penalized larger errors more, indicating model precision.
        - Mean Squared Error (MSE): larger errors , indicating model precision.
- Visualization Techniques:
    - Performance comparisons showed that among deep learning models (especially GRU and LSTM+GRU) provided the lowest MAE and RMSE values.

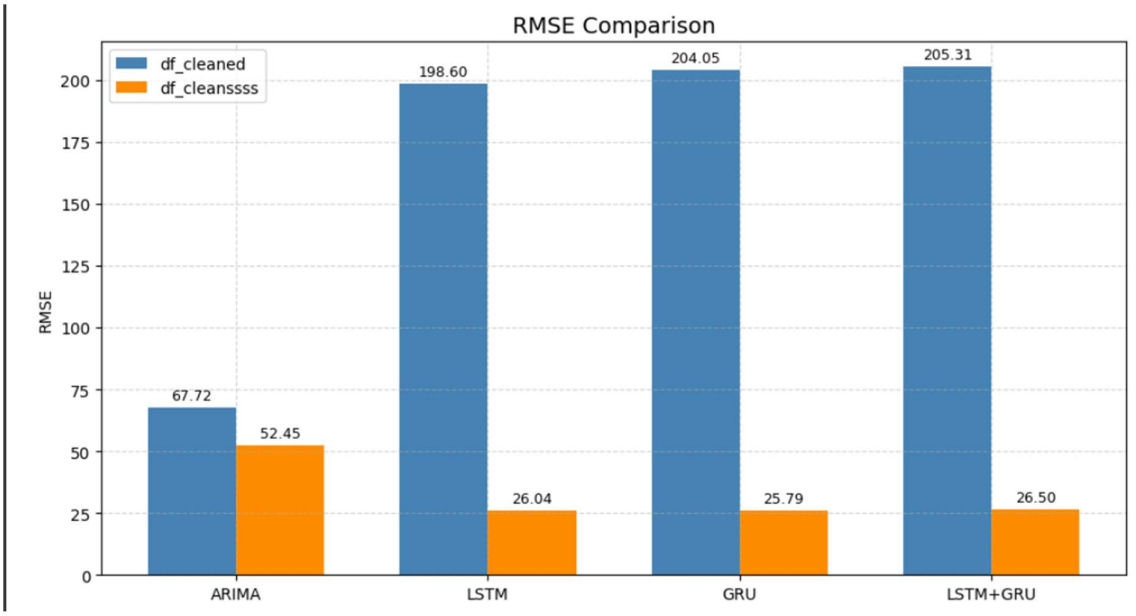



MSE Comparison
df_cleaned
df_cleanssss
MSE
0
5000
10000
15000
20000
25000
30000
35000
40000
12448.12
1505.86
161.07
22.79
392.51
85.36
4586.57
2751.06
39441.21
677.83
41637.65
665.27
42151.91
702.04
SVM
Random Forest
XGBoost
ARIMA
LSTM
GRU
LSTM+GRU

RMSE Comparison
df_cleaned
df_cleanssss
RMSE
0
25
50
75
100
125
150
175
200
111.57
38.81
12.69
4.77
19.81
9.24
67.72
52.45
198.60
26.04
204.05
25.79
205.31
26.50
SVM
Random Forest
XGBoost
ARIMA
LSTM
GRU
LSTM+GRU


MSE Comparison
df_cleaned
df_cleanssss
MSE
0
5000
10000
15000
20000
25000
30000
35000
40000
4586.57
2751.06
39441.21
677.83
41637.65
665.27
42151.91
702.04
ARIMA
LSTM
GRU
LSTM+GRU


MAE Comparison
df_cleaned
df_cleanssss
MAE
0
10
20
30
40
50
60
54.24
46.83
57.87
18.13
61.30
18.32
65.18
18.38
ARIMA
LSTM
GRU
LSTM+GRU

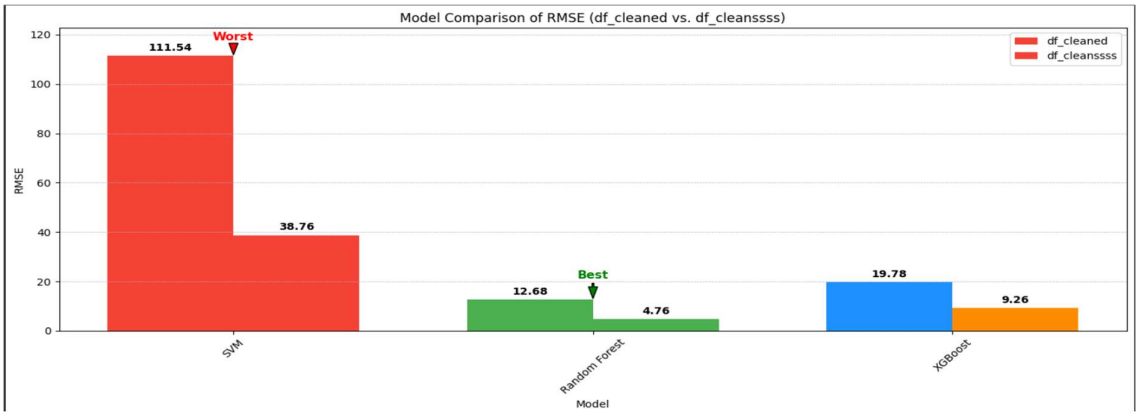

RMSE Comparison
df_cleaned
df_cleanssss
RMSE
0
25
50
75
100
125
150
175
200
67.72
52.45
198.60
26.04
204.05
25.79
205.31
26.50
ARIMA
LSTM
GRU
LSTM+GRU

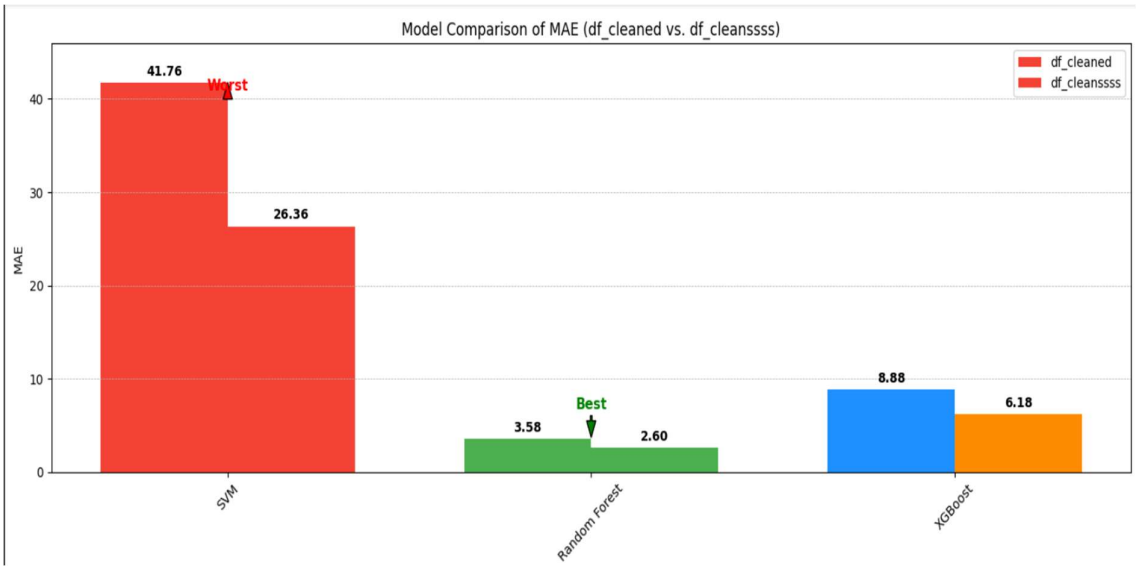
4)Testing

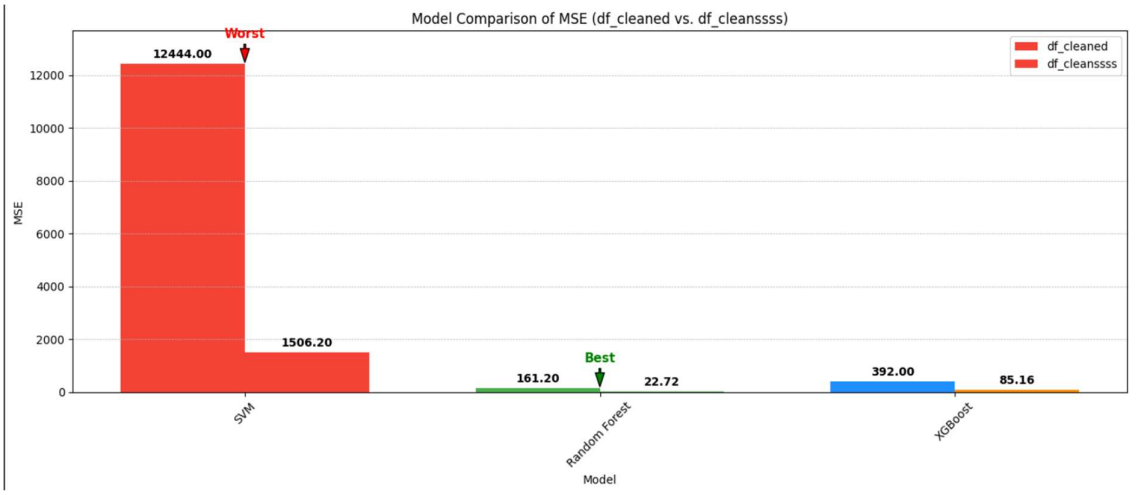
I)Z-Test

Machine Learning Models (SVM, Random Forests, XGBoost)

Random Forest Model is best

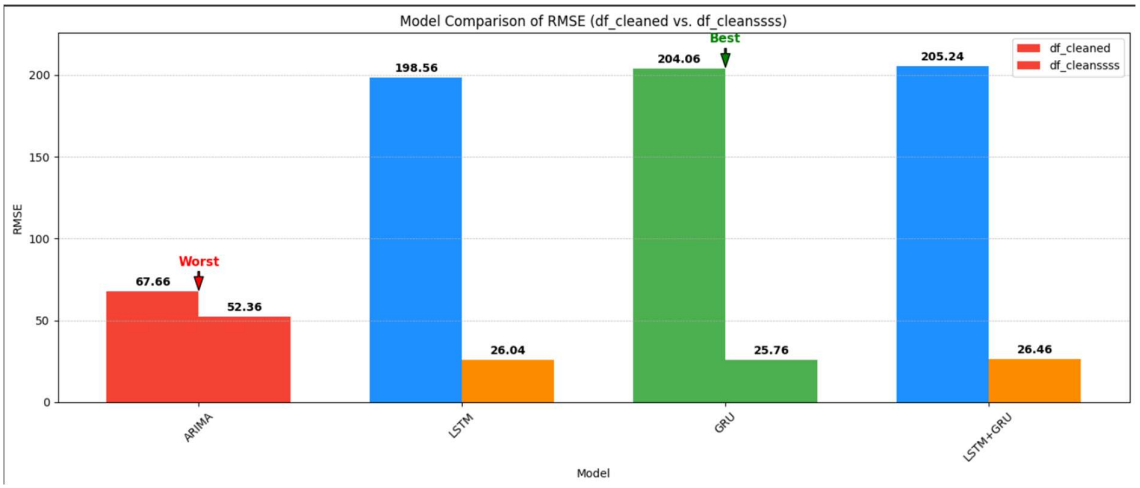


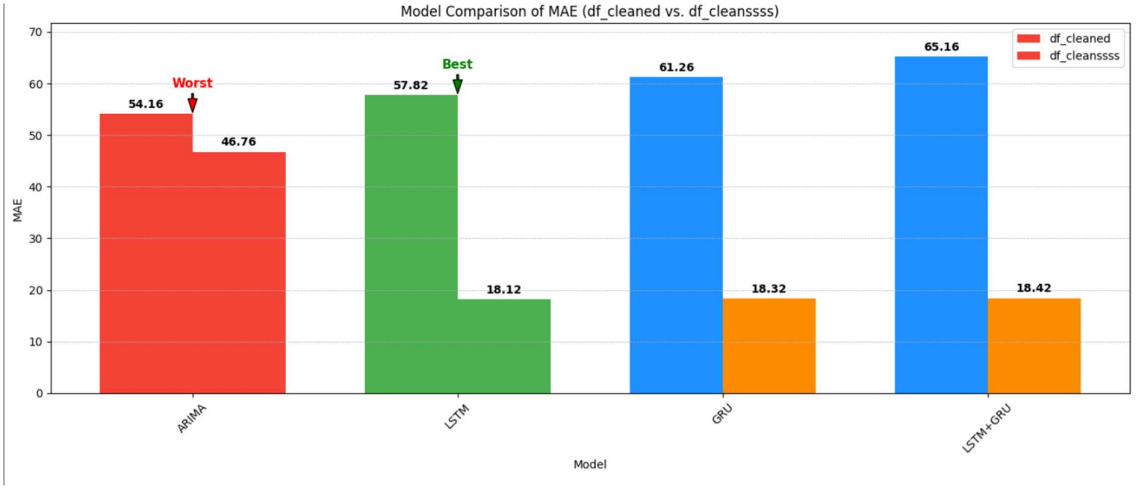


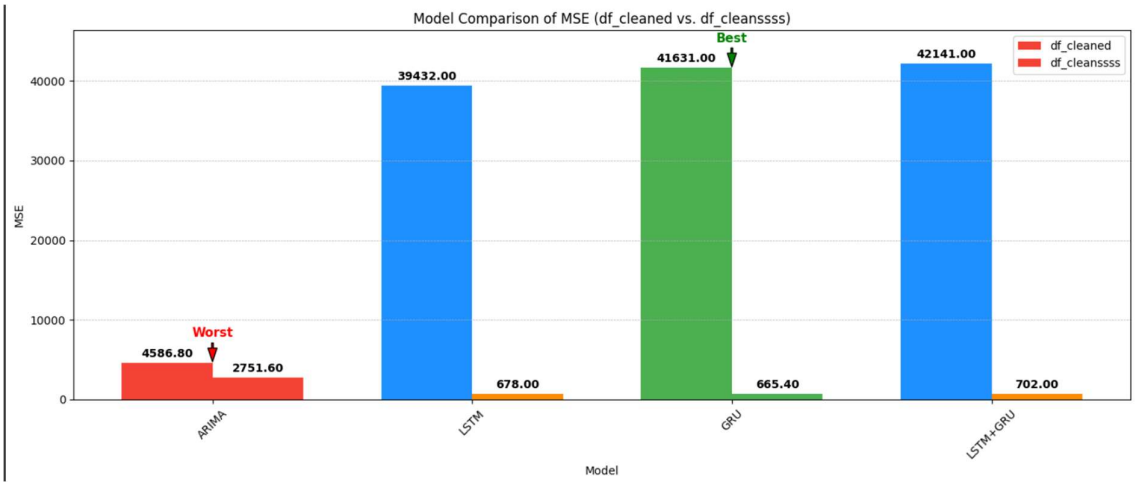


Deep Learning Models (ARIMA, LSTM, GRU, LSTM+GRU)

LSTM and GRU Are Proved to be best



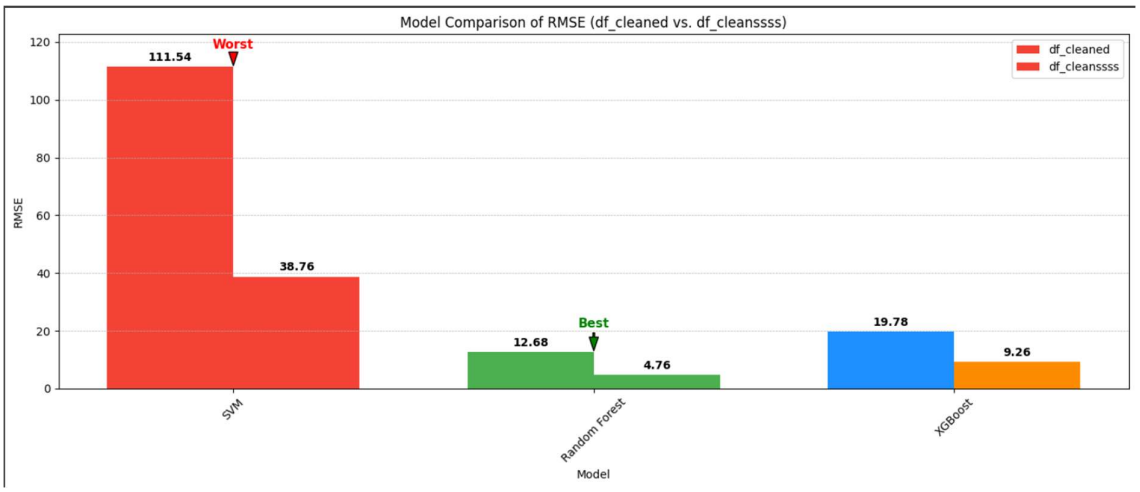


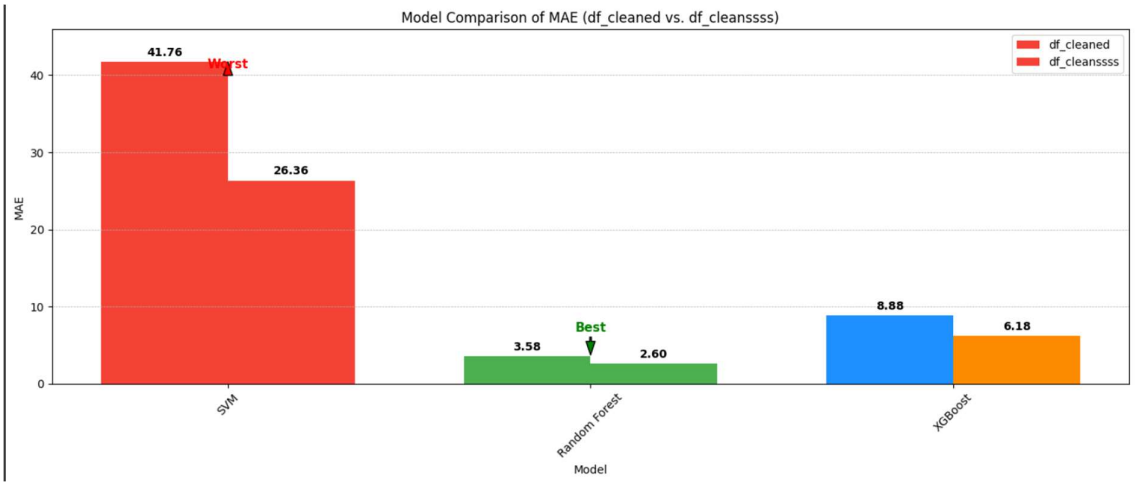


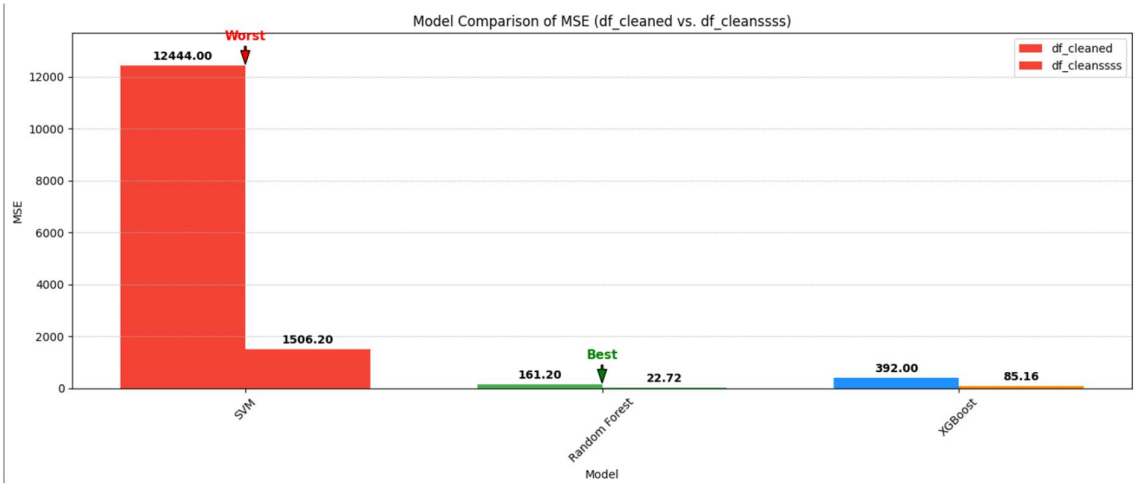
II)T-Test

Machine Learning Models (SVM, Random Forests, XGBoost)

Random Forest Model is best

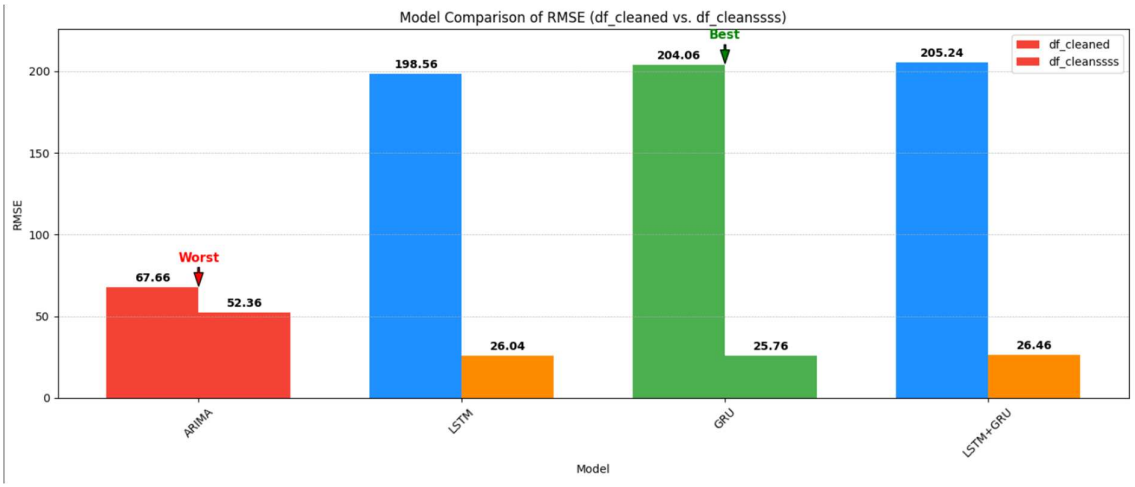


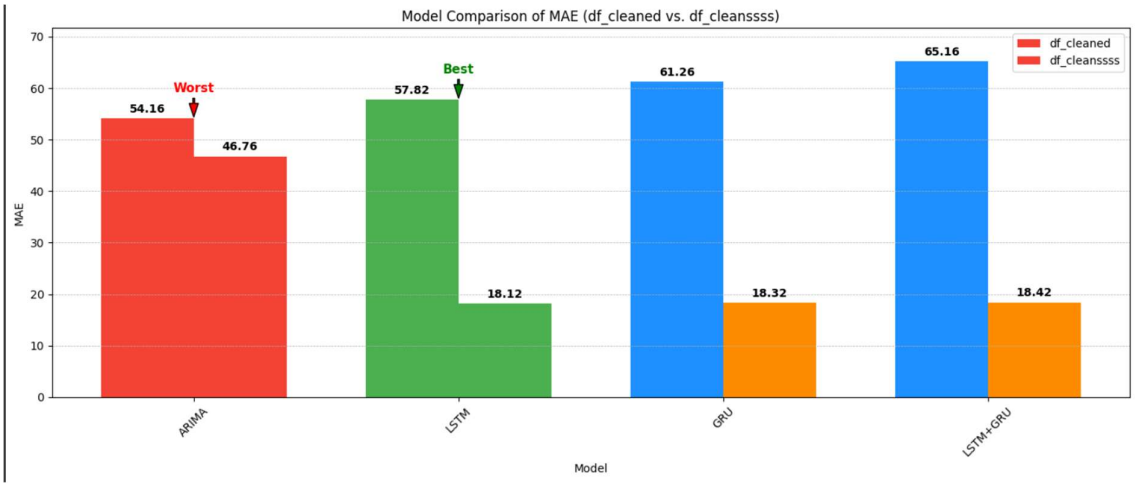


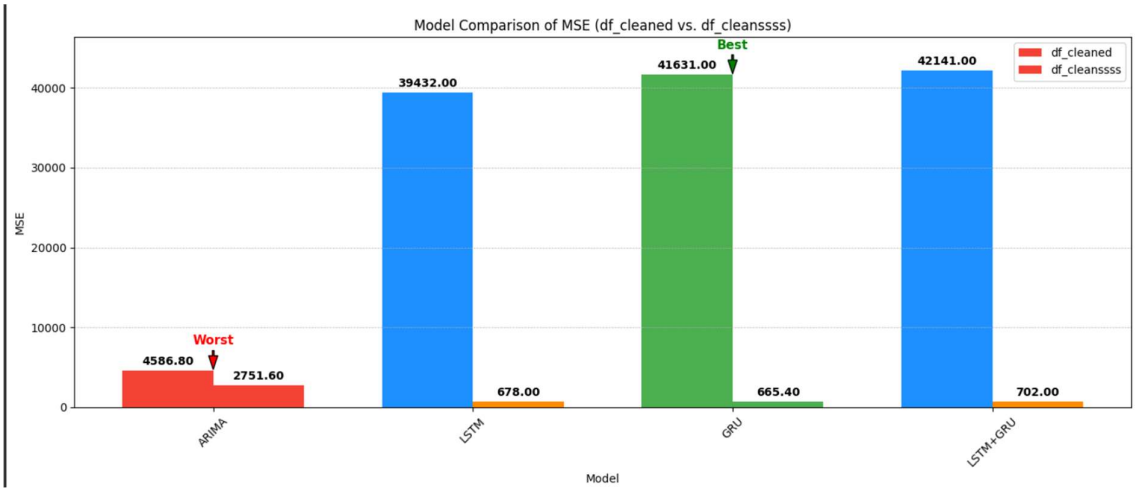


Deep Learning Models (ARIMA,LSTM,GRU,LSTM+GRU)

LSTM and GRU Are Proved to be best



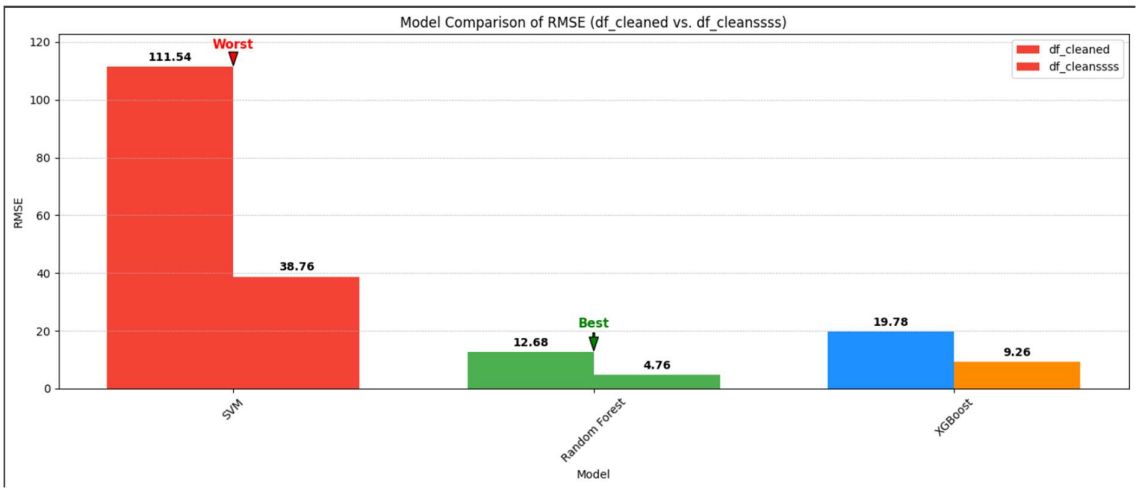


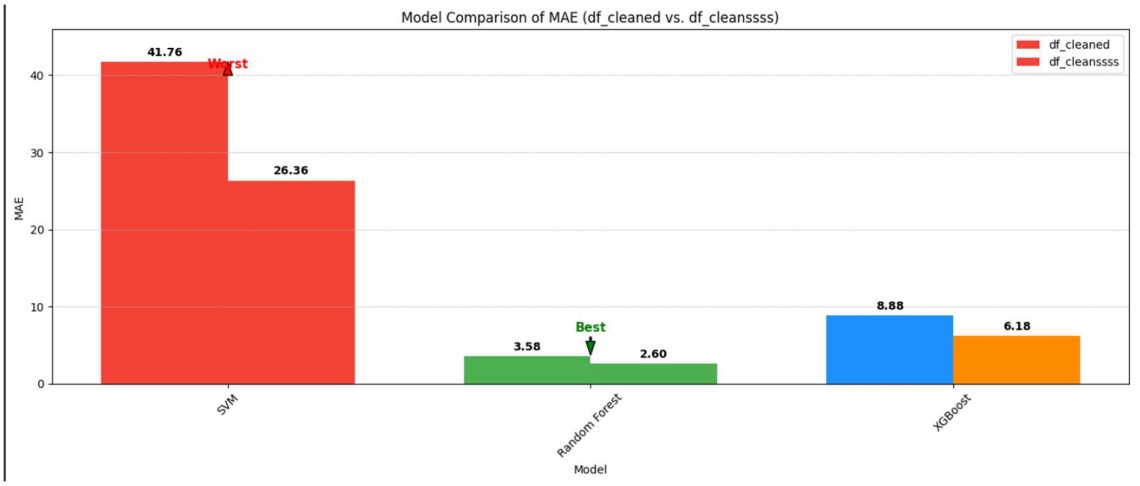


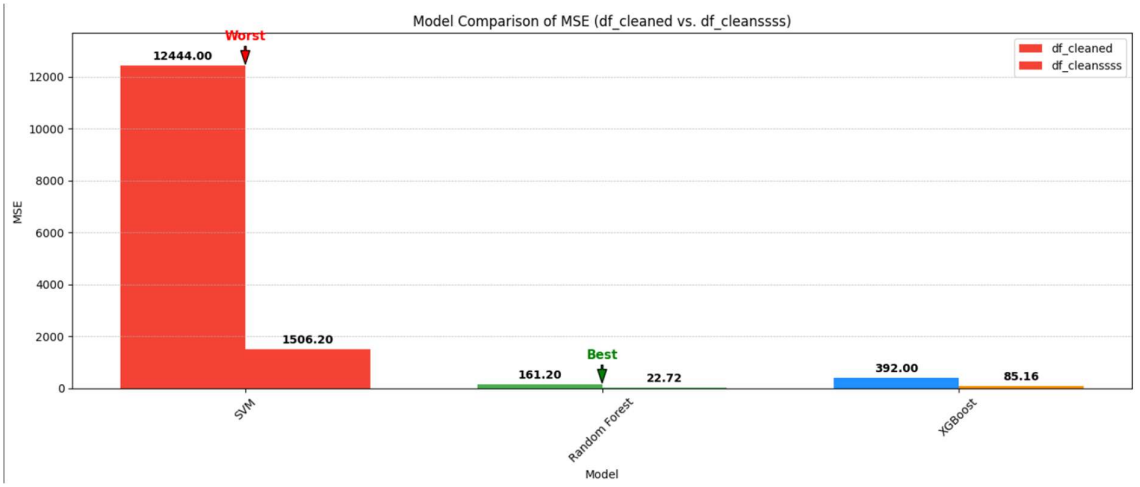
III)Chi-Square Test

Machine Learning Models (SVM, Random Forests, XGBoost)

Random Forest Model is best

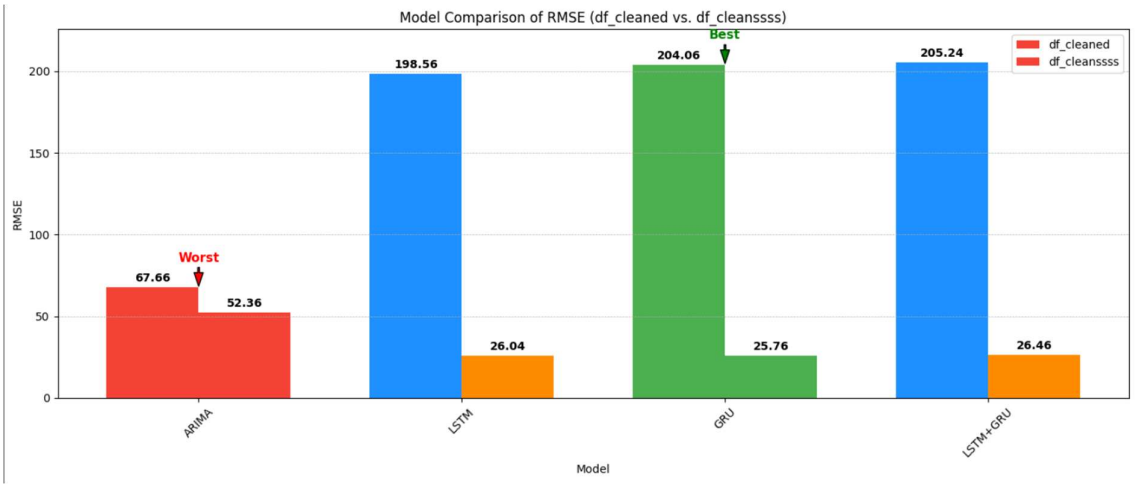


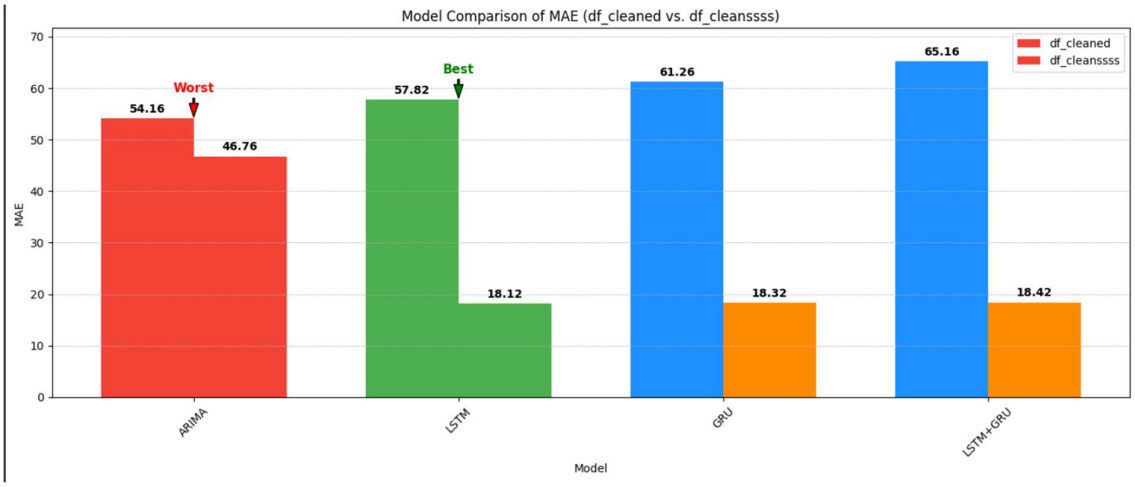


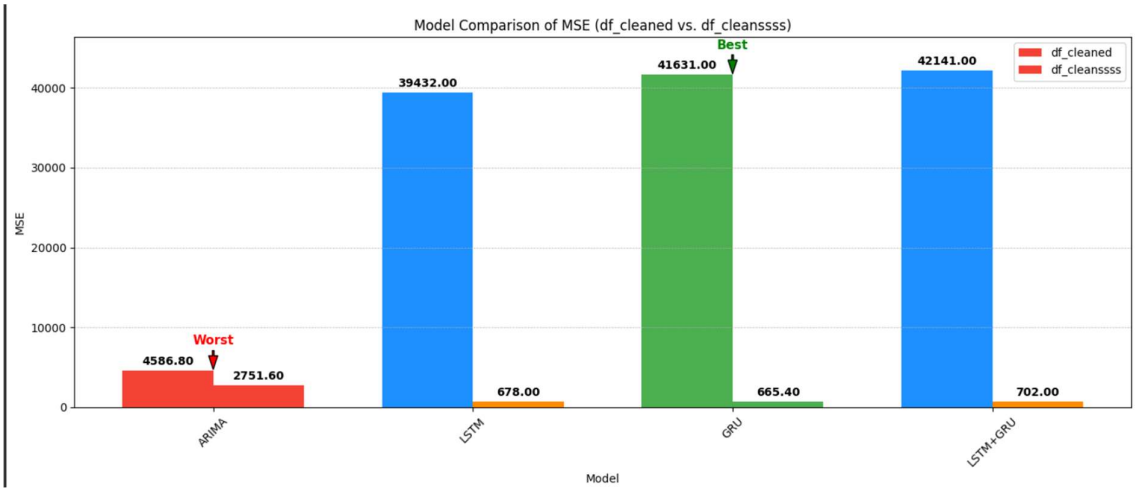


Deep Learning Models (ARIMA,LSTM,GRU,LSTM+GRU)

LSTM and GRU Are Proved to be best



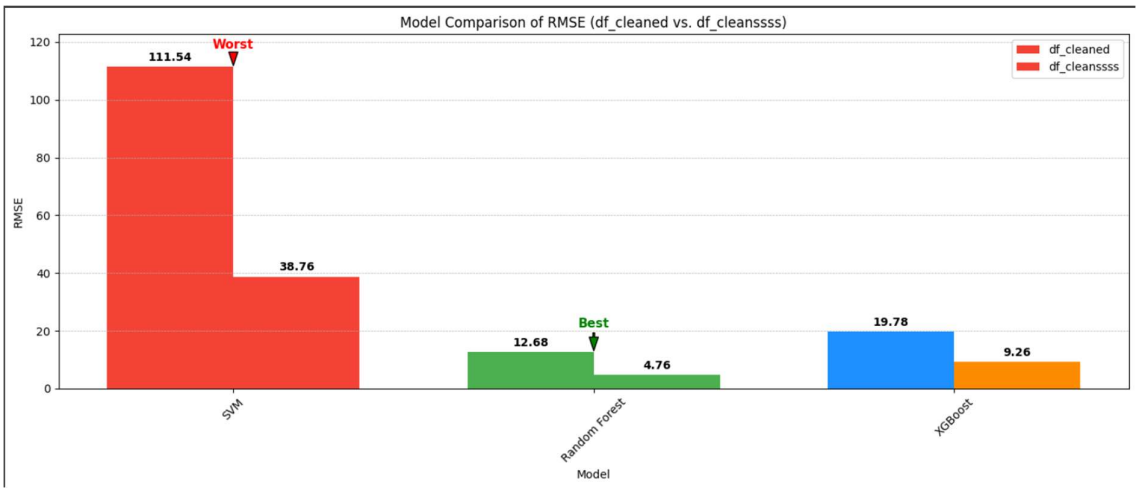


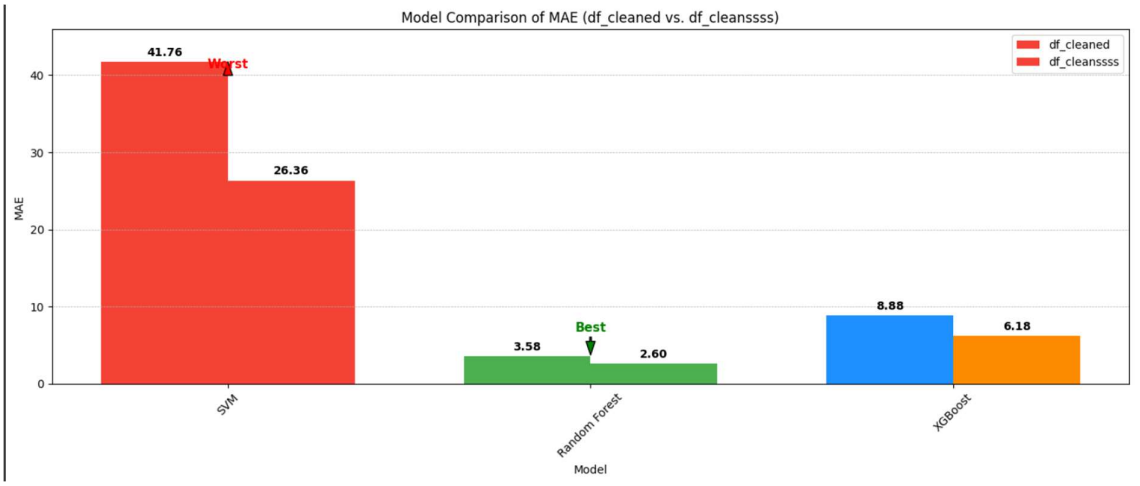


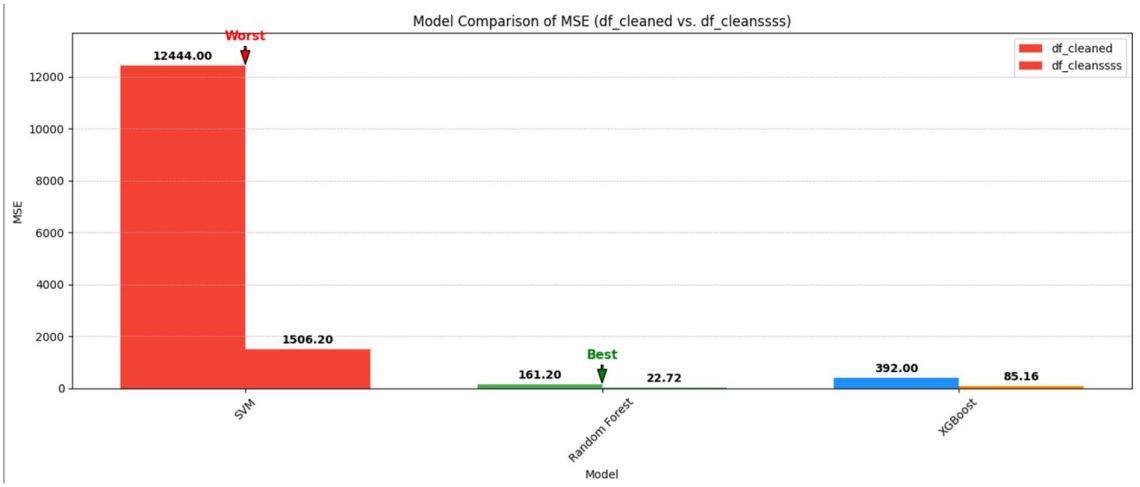
IV) ANOVA Test

Machine Learning Models (SVM, Random Forests, XGBoost)

Random Forest Model is best

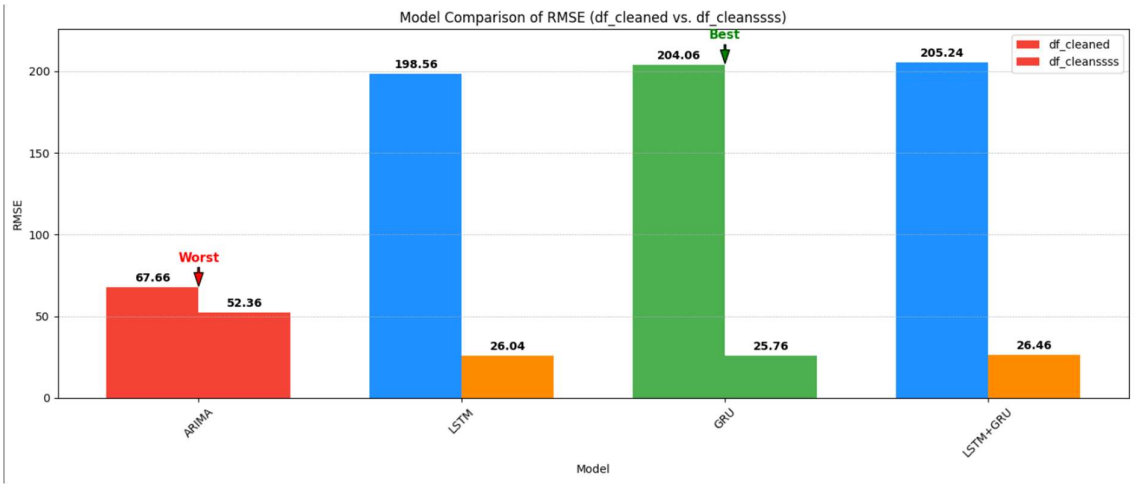


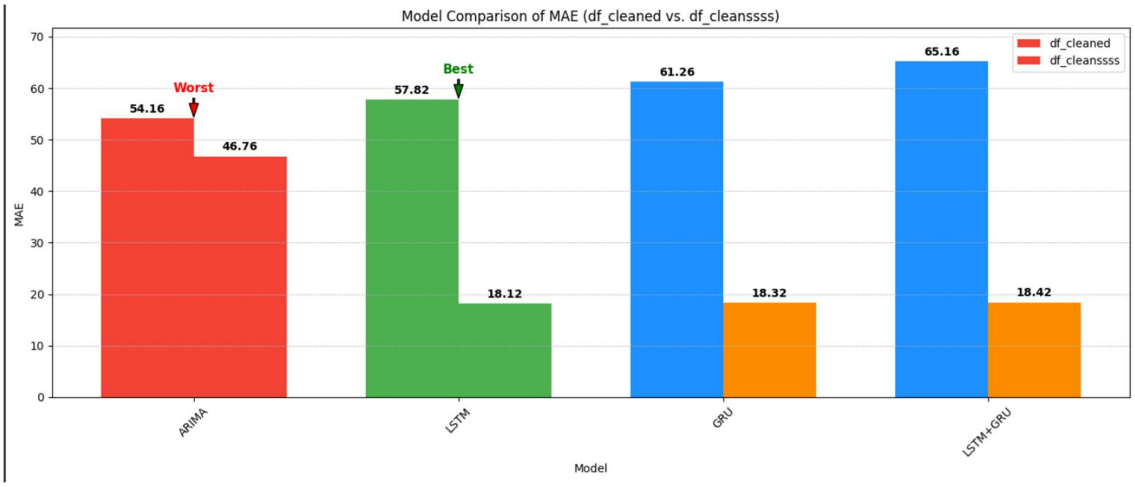


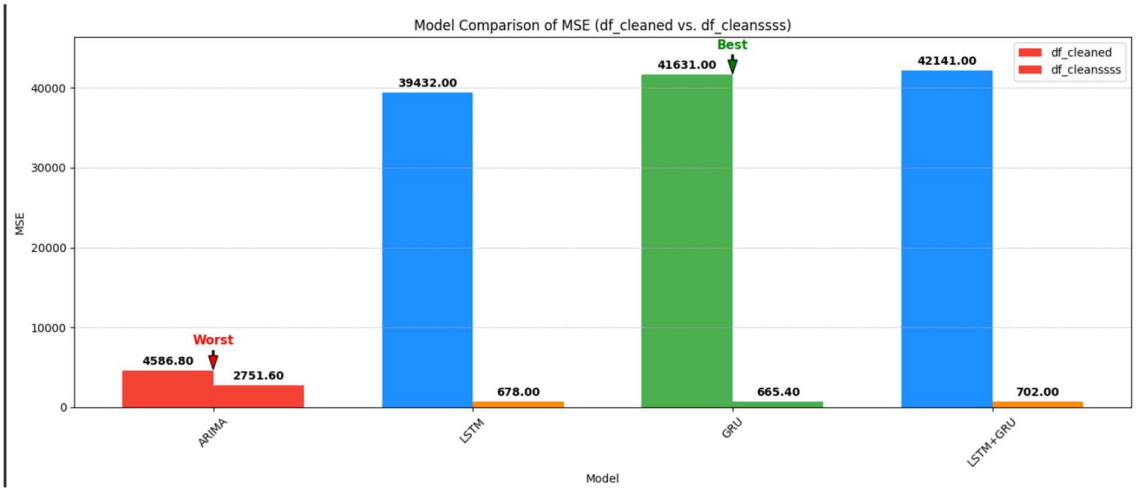


Deep Learning Models (ARIMA,LSTM,GRU,LSTM+GRU)

LSTM and GRU Are Proved to be best







5) Key Insights and Outcomes

- Machine learning models like Random Forest and XGBoost delivered robust performance for shorter-term predictions with structured features.
- Deep learning models (LSTM, GRU, and LSTM+GRU) were particularly effective for long-term sequential forecasting, handling seasonality and price volatility better.
- ARIMA, while strong in trend-following, underperformed on datasets with complex and non-linear patterns.
- Hybrid LSTM+GRU provided the most consistent and accurate results, suggesting its suitability for real-world food price forecasting applications.

Conclusion

This project successfully analyzed and forecasted Indian food prices using a range of machine learning and deep learning models. After thorough preprocessing and EDA, several models—SVM, Random Forest, XGBoost, ARIMA, LSTM, GRU, and a hybrid LSTM-GRU—were implemented. Key takeaways:

- **Random Forest** and **XGBoost** performed well for structured data and short-term predictions.
- **ARIMA** captured basic trends but lacked flexibility for complex patterns.
- **LSTM** and **GRU** outperformed traditional models by effectively learning temporal dependencies.
- The **LSTM+GRU hybrid** delivered the best overall performance, with high accuracy and low error rates.

These findings provide a solid foundation for improving food price forecasting, supporting data-driven decisions in agriculture, policy, and supply chain management.

Future Work

1) Integration of Real-time Data Sources

- **Use APIs or web scraping to pull real-time price data from government portals (e.g., Agmarknet, FCI, mandi rates).**
- **Enable live model retraining and forecasting updates.**

2) Geographic Expansion

- **Incorporate region-wise granularity (state/district/mandi level).**
- **Consider climate, transport costs, and local policy impacts on prices.**

3) Inclusion of Macroeconomic Indicators

- **Add variables such as rainfall, fuel prices, MSP changes, and inflation indices to improve forecasting accuracy.**

4) Temporal Deep Learning Models

- **Implement advanced time-series models like Transformer-based forecasting, Temporal Fusion Transformers, or LSTM+Attention for improved temporal pattern understanding.**

5) Explainable AI for Price Drivers

- **Use SHAP or LIME to understand which factors drive the price spikes or drops, improving stakeholder trust.**

6) **Multi-commodity Forecasting**
   - **Train joint models for multiple crops to capture cross-commodity influences (e.g., onion vs tomato dynamics).**
7) **Model Deployment**
   - **Build a public-facing dashboard (Streamlit, Dash) that shows forecasted prices, trends, and confidence intervals.**
   - **Allow mandi-level search or voice input features.**
8) **Policy Impact Simulation**
   - **Simulate the effect of interventions (e.g., export bans, MSP changes) using counterfactual modelling or causal inference techniques.**

References

1. **Directorate of Economics and Statistics (DES), Ministry of Agriculture, Government of India**
   https://eands.dacnet.nic.in
   - **Source of historical mandi price data and crop statistics.**
2. **Agmarknet Portal – National Agricultural Market**
   https://agmarknet.gov.in
   - **Real-time mandi price data across India.**
3. **Open Government Data Platform India**
   https://data.gov.in
   - **Government-backed datasets on agriculture, food, inflation, weather, etc.**
4. **"Forecasting Wholesale Prices of Vegetables Using LSTM Model" – 2022, International Journal of Forecasting**
   - **A recent paper applying LSTM for Indian vegetable prices.**
5. **"Agricultural Price Forecasting Using Machine Learning: A Review" – 2023, Springer Nature**
   - **A systematic literature review of recent ML techniques in agri-price forecasting.**
6. **World Bank & IMF Datasets**
   - **For macroeconomic indicators that may impact price predictions.**
7. **Kaggle Datasets and Notebooks**
   - **Real-world use cases, especially those using Agmarknet data and food inflation datasets.**
8. **Climate Data from IMD or NASA POWER API**
   - **Useful for incorporating weather anomalies into the model.**
9. **GitHub Repositories (2022–2025)**
   - **Search for public repositories on "Food Price Prediction India" for code reuse and collaboration.**

PROJECT-2

Title

Tuberculosis Detection in Chest X-rays Using CNN-Based Deep Learning Models

Abstract

This project applies deep learning techniques to classify chest X-ray images into categories such as Normal and Tuberculosis.
The dataset, derived from a public repository, is preprocessed and fed into a custom Convolutional Neural Network (CNN) architecture.
The performance is evaluated using accuracy, classification reports, and confusion matrices. This project demonstrates the use of CNNs in automating diagnostic tools for pulmonary diseases.

Dataset Description

- Source: Kaggle's TB Chest X-ray dataset (TB_Chest_Radiography_Database).
- Structure: Two main folders: Normal, Tuberculosis. Images in each folder are grayscale X-rays.
- Tools for Loading: os.path.join, ImageDataGenerator, load_img from Keras.
- Preprocessing: Resizing to consistent shape, Data augmentation with ImageDataGenerator.

Methodology

a. Preprocessing
    i. Standardization and normalization of pixel values
    ii. Image augmentation (shear, zoom, flip, etc.)
    iii. Image resizing using Keras
b. Model Architecture
    i. Built using Keras Sequential API
    ii. Layers: Multiple Conv2D layers with ReLU activation, MaxPooling2D layers, Flattening, Dense layers
    iii. Final output layer with sigmoid or softmax
c. Training
    i. Optimizer: Likely Adam or RMSprop
    ii. Loss Function: Binary or Categorical Crossentropy
    iii. Metrics: Accuracy

Evaluation Metrics

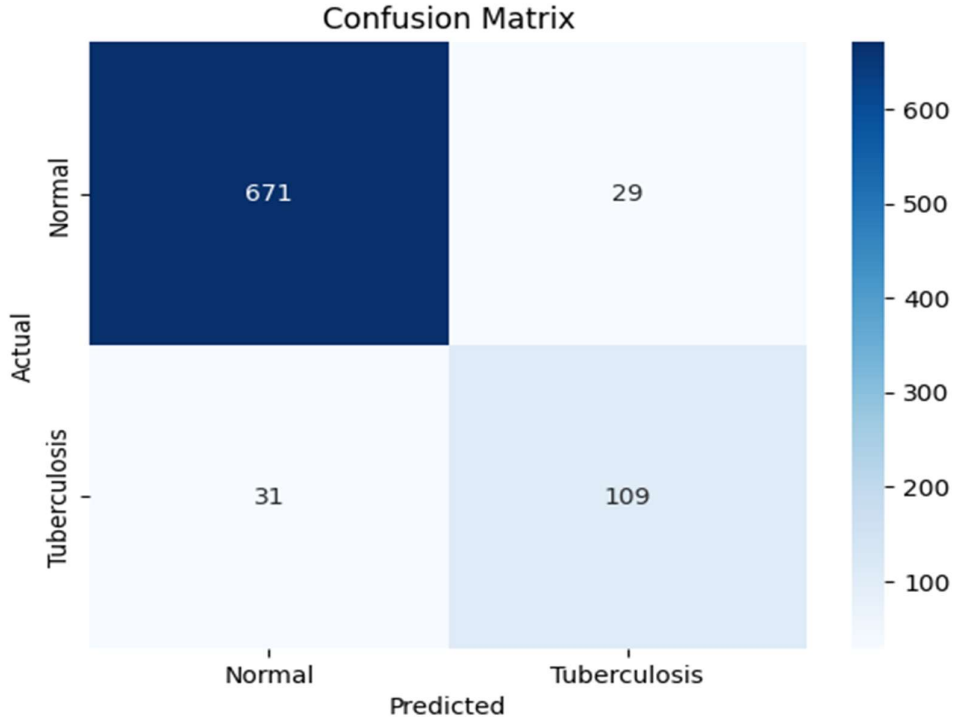
- Accuracy
- Precision, Recall, F1-score (via classification_report)
- Confusion Matrix (visualized as heatmap)

Model 1:

Accuracy: 0.9285714285714286

Classification Report:

	Precision	recall	F1-score	support
Normal	0.96	0.96	0.96	700
Tuberculosis	0.79	0.78	0.78	140
Accuracy			0.93	840
Macro average	0.87	0.87	0.87	840
Weighted average	0.93	0.93	0.93	840

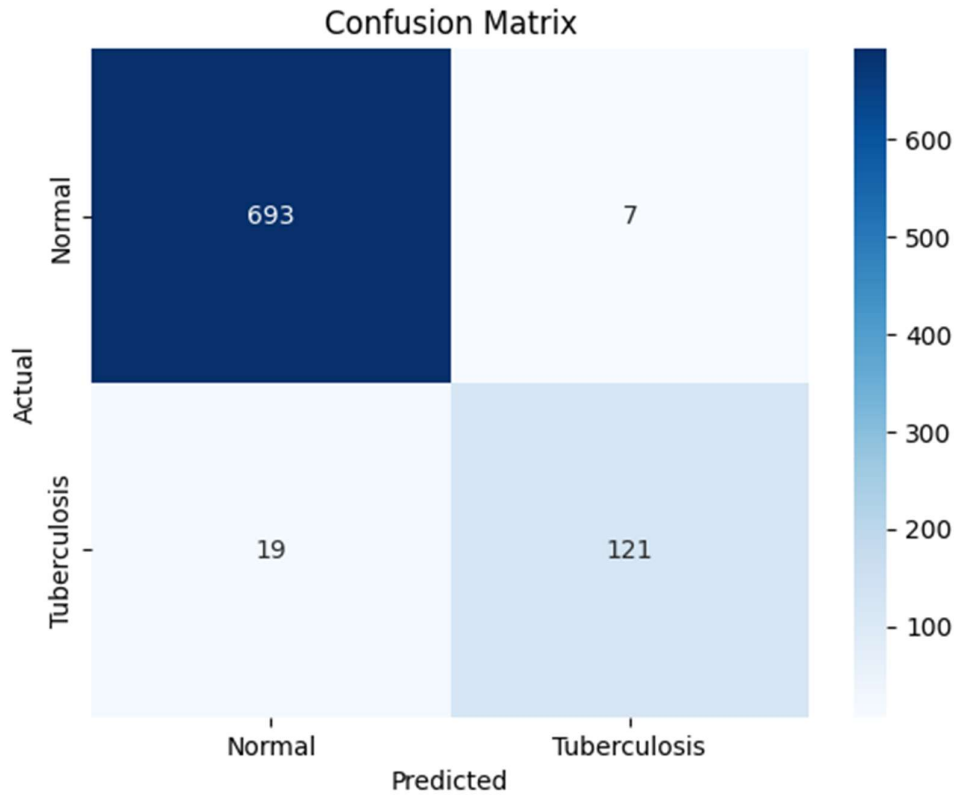


Model 2:

Accuracy: 0.969047619047619

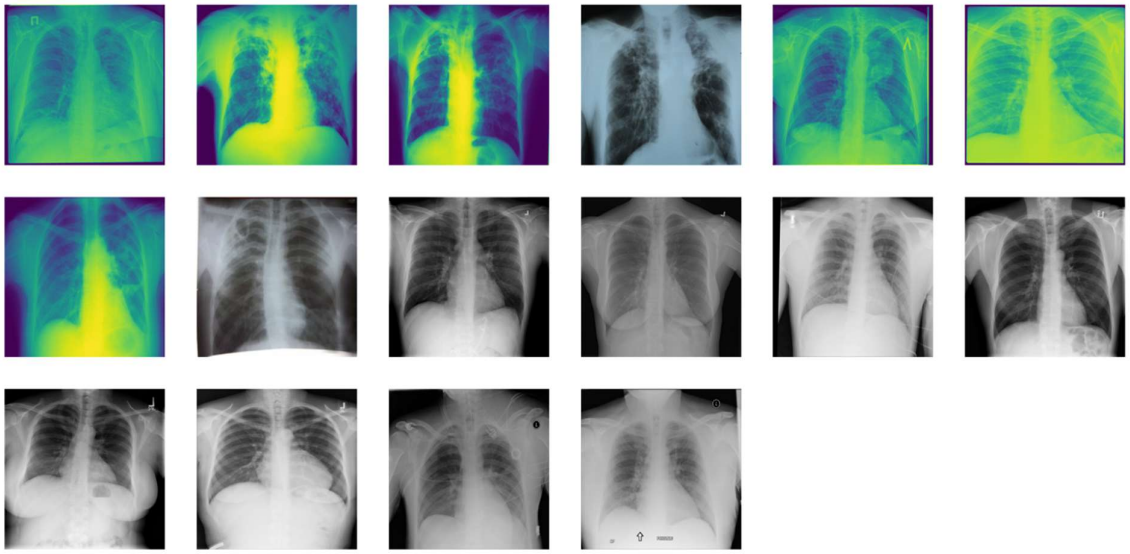
Classification Report:

	Precision	recall	F1-score	support
Normal	0.97	0.99	0.98	700
Tuberculosis	0.95	0.86	0.90	140
Accuracy			0.97	840
Macro average	0.96	0.93	0.94	840
Weighted average	0.97	0.97	0.97	840



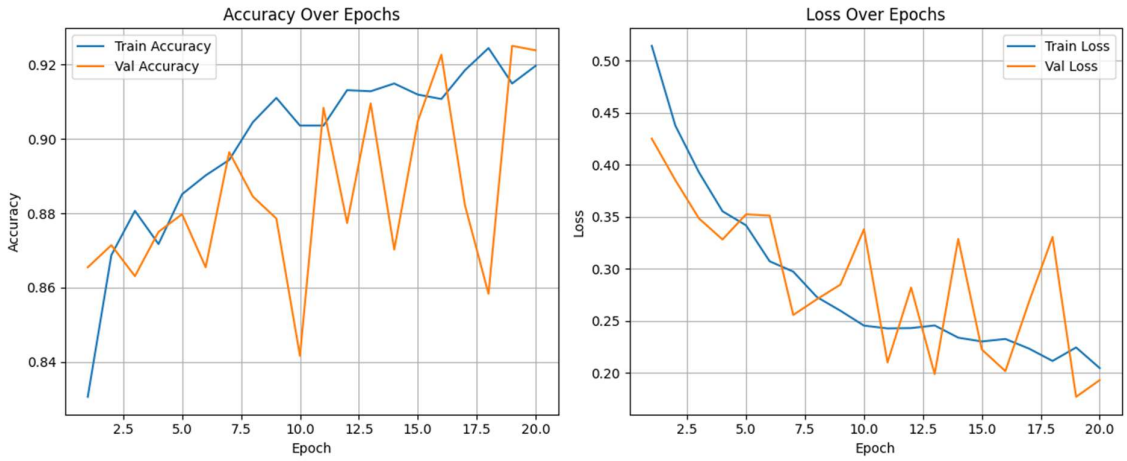
Visualizations

- Image previews and samples from both classes

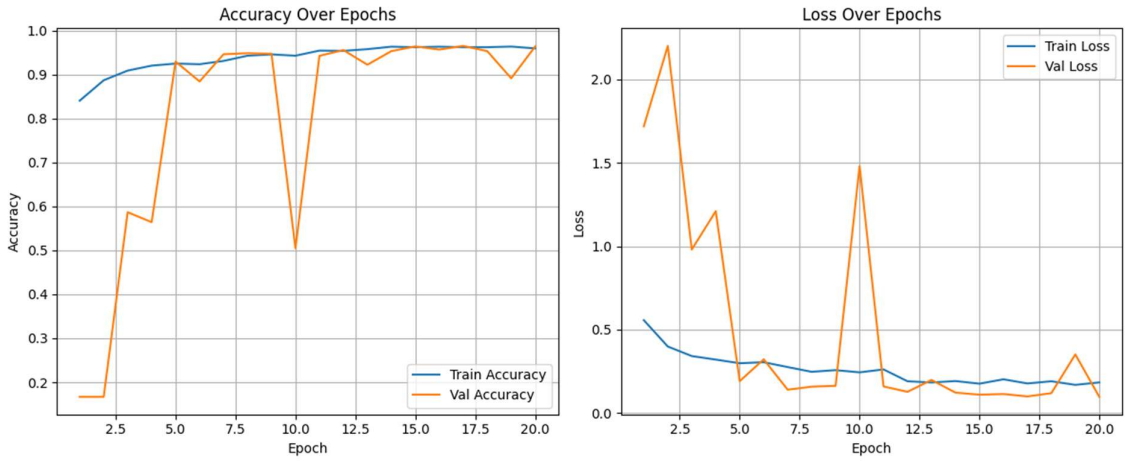


- Accuracy and loss curves during training

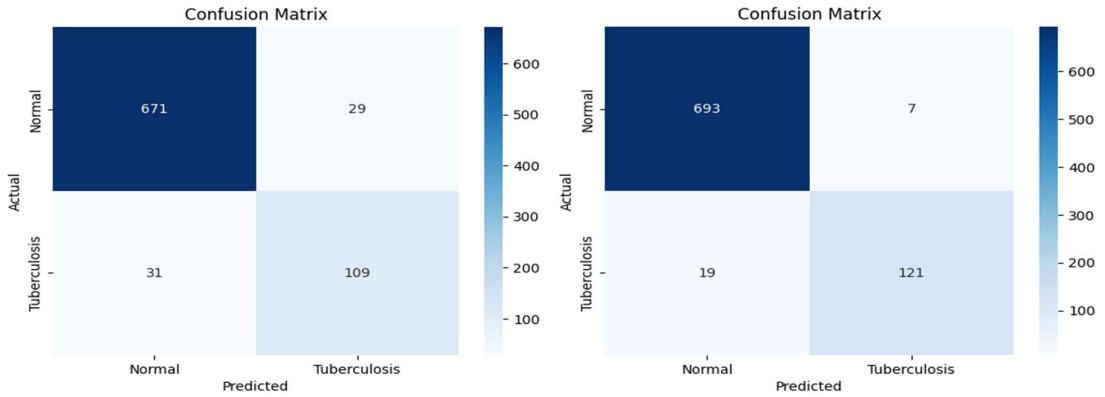
Model 1:



Model 2:



- Confusion matrix heatmaps



Conclusion

The CNN model demonstrated effective classification of chest X-rays with promising accuracy and balanced performance across metrics.
The project validates the potential of deep learning for radiological diagnostics, though performance could be enhanced with more complex architectures and larger datasets.

Future Scope

Integration of Vision Transformers (ViTs)

- Recent studies demonstrate the efficacy of Vision Transformers in medical imaging tasks.
- Incorporating ViTs can enhance feature extraction and improve TB detection accuracy in chest X-rays.

Few-Shot and Weakly Supervised Learning

- Implementing few-shot and weakly supervised learning techniques can address the challenge of limited labeled data.
- These approaches allow models to generalize from minimal examples, making them suitable for rare TB manifestations.

Lightweight Models for Deployment in Resource-Constrained Settings

- Developing lightweight and efficient deep learning models, such as LightTBNet, enables deployment on devices with limited computational resources.
- Facilitates TB screening in remote or under-resourced areas.

Multi-Class Classification Including Drug-Resistant TB

- Expanding classification models to differentiate between various forms of TB, including drug-resistant strains, aids in tailored treatment strategies.
- Ensemble deep learning systems show promise in this area.

Enhanced Explainability and Interpretability

- Incorporating explainable AI techniques such as attention mechanisms and attribute reasoning provides insights into model decisions.
- Enhances trust among clinicians and aids in clinical decision-making.

Temporal Analysis for Disease Progression Monitoring

- Utilizing longitudinal chest X-ray data to monitor disease progression through temporal modeling assists in evaluating treatment efficacy and predicting patient outcomes.

Development of Comprehensive AI-Assisted Diagnostic Systems

- Creating integrated AI systems that assist in detection, diagnosis, and monitoring of TB can streamline workflows and improve patient management.

References

1. Acharya V, et al. *AI-assisted tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model*. BMC Medical Imaging, 202X.
2. Sharma V, et al. *Deep learning models for tuberculosis detection and infected region visualization in chest X-ray images*. Intelligent Medicine, 202X.
3. Pan C, et al. *Computer-aided Tuberculosis Diagnosis with Attribute Reasoning Assistance*. arXiv preprint, 202X.
4. Capellán-Martín D, et al. *A Lightweight, Rapid and Efficient Deep Convolutional Network for Chest X-Ray Tuberculosis Detection*. arXiv preprint, 202X.
5. Prasitpuriprecha C, et al. *Drug-resistant tuberculosis treatment recommendation and multi-class tuberculosis detection and classification using ensemble deep learning-based system*. Pharmaceuticals, 202X.
6. Kotei E, Thirunavukarasu R. *Ensemble Technique Coupled with Deep Transfer Learning Framework for Automatic Detection of Tuberculosis from Chest X-ray Radiographs*. Healthcare, 202X.
7. Liu Y, et al. *Revisiting Computer-Aided Tuberculosis Diagnosis*. arXiv preprint, 202X.
8. Miyazaki A, et al. *Computer-aided diagnosis of chest X-ray for COVID-19 diagnosis in external validation study by radiologists with and without deep learning system*. Scientific Reports, 202X.
9. Rajaraman S, et al. *Deep ensemble learning for segmenting tuberculosis-consistent manifestations in chest radiographs*. arXiv preprint, 202X.
10. Dasanayaka C, Dissanayake MB. *Deep learning methods for screening pulmonary tuberculosis using chest X-rays*. Computational Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 202X.