|  |  |
| --- | --- |
| Internship Project Title | Forecasting System - Project Demand of Products at a Retail Outlet Based on Historical Data |
| Name of the Company | Tata Consultancy Services (TCS) |
| Name of the Industry Mentor | Shubhangi Katariyar. |
| Name of the Institute | VIT University |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 2 april 2025 | 16 april 2025 | 50 hrs | Streamlit , statsmodel | Pycharm |

**Milestone: 1**

Successfully built a dynamic, multi-product 12-month sales forecasting tool with Prophet, complete with user interaction, visual insights, and performance evaluation metrics.

**TABLE OF CONTENT**

**Acknowledgements**

I would like to express my sincere thanks to my academic guide, Dr. Madhuri Pant, for her continuous support and guidance during the development of this project. I am also grateful to Vishwakarma University for providing the platform to work on a project that strengthens practical knowledge and skills.

**Objective**

To develop a robust time-series forecasting model using **Facebook Prophet**, analyze product sales data (200k+ entries), and evaluate performance using statistical accuracy metrics (MAE, RMSE, MAPE). The aim is to predict product demand and visualize seasonal trends effectively.

**Introduction / Description of Internship**

This internship involved a practical implementation of machine learning in sales forecasting. I worked on real-world time-series data to generate 12-month forecasts for various products. My focus was on applying Prophet to identify trends, seasonality, and forecast performance.

**Internship Activities**

1. Data cleaning, filtering, and preprocessing
2. File Dialog UI integration for dynamic CSV loading
3. Prophet forecasting for selected products
4. Component analysis (trend, weekly, yearly)
5. Accuracy evaluation using MAE, RMSE, MAPE
6. Visualization and comparison of actual vs predicted values

**Approach / Methodology**

1. Load data dynamically using tkinter.filedialog
2. User selects a product interactively
3. Prophet model is trained and forecast is generated
4. Merged actual vs predicted sales for evaluation
5. Visualized seasonal components and evaluated with metrics

**Assumptions**

1. Date column is clean and continuous – assumes there are no missing or incorrectly formatted dates in the sales data.
2. Data is available monthly – the sales data is assumed to be aggregated at a monthly level for consistent forecasting.
3. Each row represents a unique product entry – assumes no duplicate product-date combinations in the dataset.
4. Column names are standardized – expects columns like Date, Product, and Sales (renamed internally to ds and y).
5. Sufficient historical data exists per product – a minimum threshold of historical months is assumed for accurate modeling (e.g., ≥12 months).
6. Sales trends are seasonal and predictable – Prophet relies on the assumption that historical patterns repeat over time.
7. No significant inventory stockouts – assumes that recorded sales reflect demand and not suppressed by inventory limitations.

**Exceptions / Exclusions**

1. Multi-product joint forecasting is not implemented – the model forecasts one product at a time, not combined sales trends.
2. Holiday effects or special events are not included – the Prophet model was run without custom seasonalities or holiday regressors.
3. No external features used – variables like marketing campaigns, weather, competitor pricing, or promotions were excluded.
4. Does not handle missing months explicitly – gaps in monthly data were not filled or interpolated prior to modeling.
5. Only monthly frequency is considered – daily or weekly sales granularity was not modeled to reduce complexity and size.
6. Seasonality is assumed additive – Prophet was used with default additive model settings; multiplicative seasonality wasn’t explored.
7. No automated error reporting or logging system – any failures or data mismatches raise errors manually without logs or notifications.
8. Forecast results are not stored – the predictions are displayed but not saved to a file automatically (e.g., CSV or database).

**Charts, Tables, Diagrams**

* Line charts comparing actual vs predicted sales.
* Tables summarizing MAE, RMSE, MAPE per product.
* Prophet's component plots showing trends and seasonality.

**Algorithms**

Facebook Prophet (Additive model = Trend + Seasonality + Holidays)

Accuracy evaluation:

MAE = Mean Absolute Error

RMSE = Root Mean Squared Error

MAPE = Mean Absolute Percentage Error

**Challenges & Opportunities**

**Challenges:**

* Large dataset (200k+) slowed training
* Interactive product selection via tkinter is basic
* Handling missing or wrongly formatted dates

**Opportunities:**

* Add dropdown or search-based product selection
* Forecasting grouped products (e.g., Tablets, Mobiles)
* Automate performance logging in CSV/PDF

**Risk Vs Reward :**

**Risks:**

1. Overfitting on seasonal patterns – Prophet may inaccurately amplify trends if historical anomalies are mistaken as seasonality.
2. Data quality dependency – Forecast accuracy heavily relies on clean, complete, and consistent data input.
3. Limited generalization – The model may not adapt well to sudden changes like new product launches, economic shifts, or competitor impact.

**Rewards:**

1. Improved inventory planning – Accurate forecasts help in minimizing stockouts or overstock situations.
2. Better business decisions – Insightful trend analysis and visual forecasts support marketing, budgeting, and operations.
3. Scalable forecasting system – The current approach can be extended to hundreds of products with minimal additional effort.

**Reflections on the Internship**

1. Bridging theory with real-world data:  
   This internship allowed me to apply machine learning concepts like time-series forecasting in a practical setting, transitioning from theoretical knowledge to actual implementation using Prophet on a large dataset.
2. Hands-on with real datasets:  
   Working with 200k+ realistic product sales records challenged me to handle data cleaning, structuring, and preprocessing tasks—skills that are crucial in real-world data science projects.
3. Understanding the importance of evaluation metrics:  
   By calculating MAE, RMSE, and MAPE, I learned the significance of interpreting model accuracy, and how small numerical differences can translate into big business impacts.
4. User-centric development:  
   Creating a user-friendly GUI using Tkinter with file selection and product filtering taught me how essential it is to design solutions that are intuitive and accessible for non-technical users.
5. Problem-solving through debugging:  
   Encountering issues like incorrect column names, data mismatches, and missing dependencies improved my debugging skills and made me more resilient when facing technical obstacles.
6. Communication and documentation:  
   Documenting each step, writing clean code, logging results, and interpreting output for stakeholders enhanced my ability to explain technical content clearly.

**Recommendations**

1. Add exception handling for column mismatches
2. Integrate multiple product forecasting as a batch mode
3. Add export options for graphs and metrics
4. Include additional seasonality parameters (e.g., holidays)

**Outcome / Conclusion**

**Tool-Overview**:  
A robust 12-month forecasting tool was built using Facebook's Prophet library. It was specifically tailored to handle a realistic, large-scale sales dataset comprising over 200,000 records. The tool allows users to select a product and visualize its future sales trajectory based on historical trends.

**User Interaction**:  
A user-friendly interface was developed using Tkinter. This GUI enables users to:

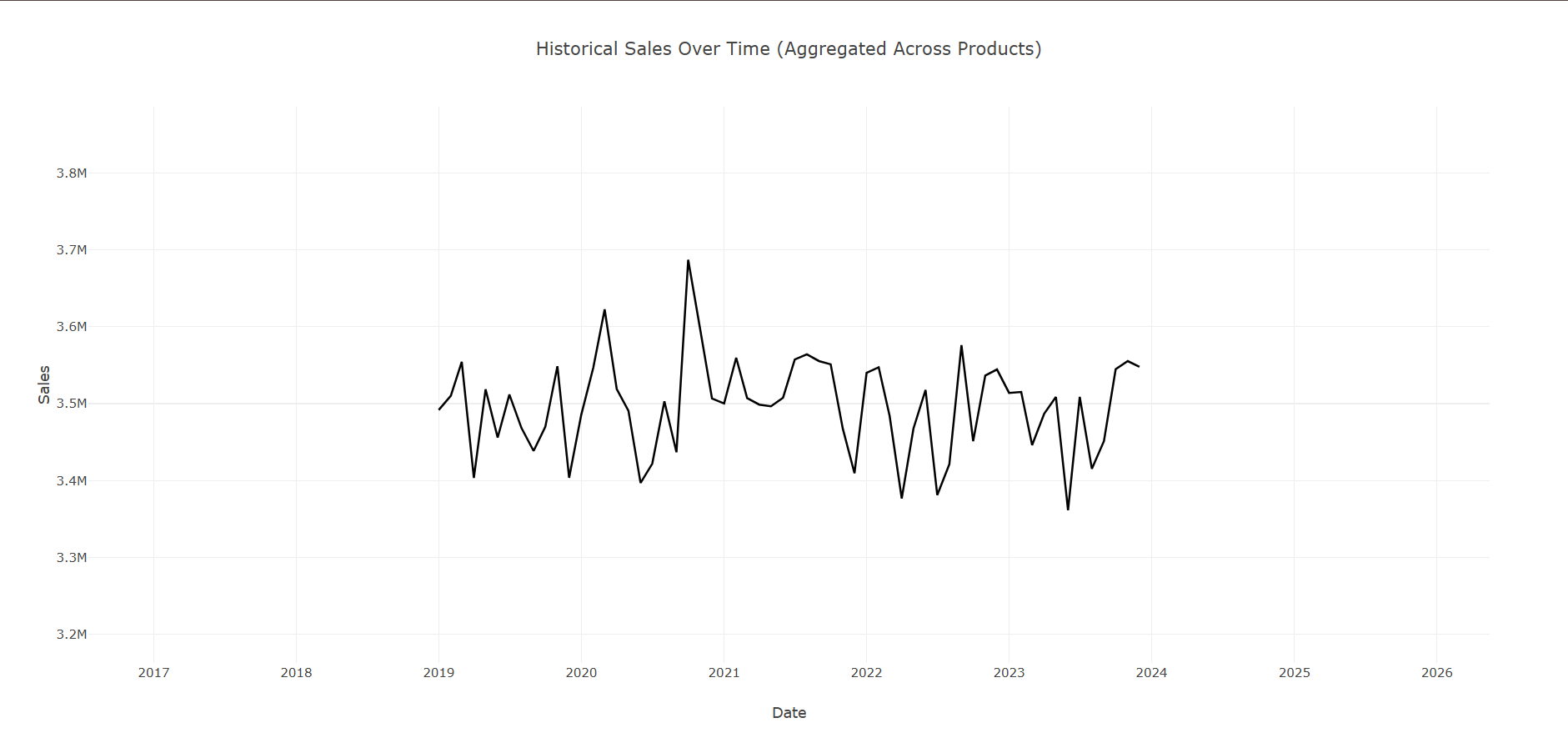
* Load CSV files via File Open Dialog.
* Select any product (e.g., Tablet 366) from the dataset.
* Automatically generate and visualize forecasts for the next 12 months.

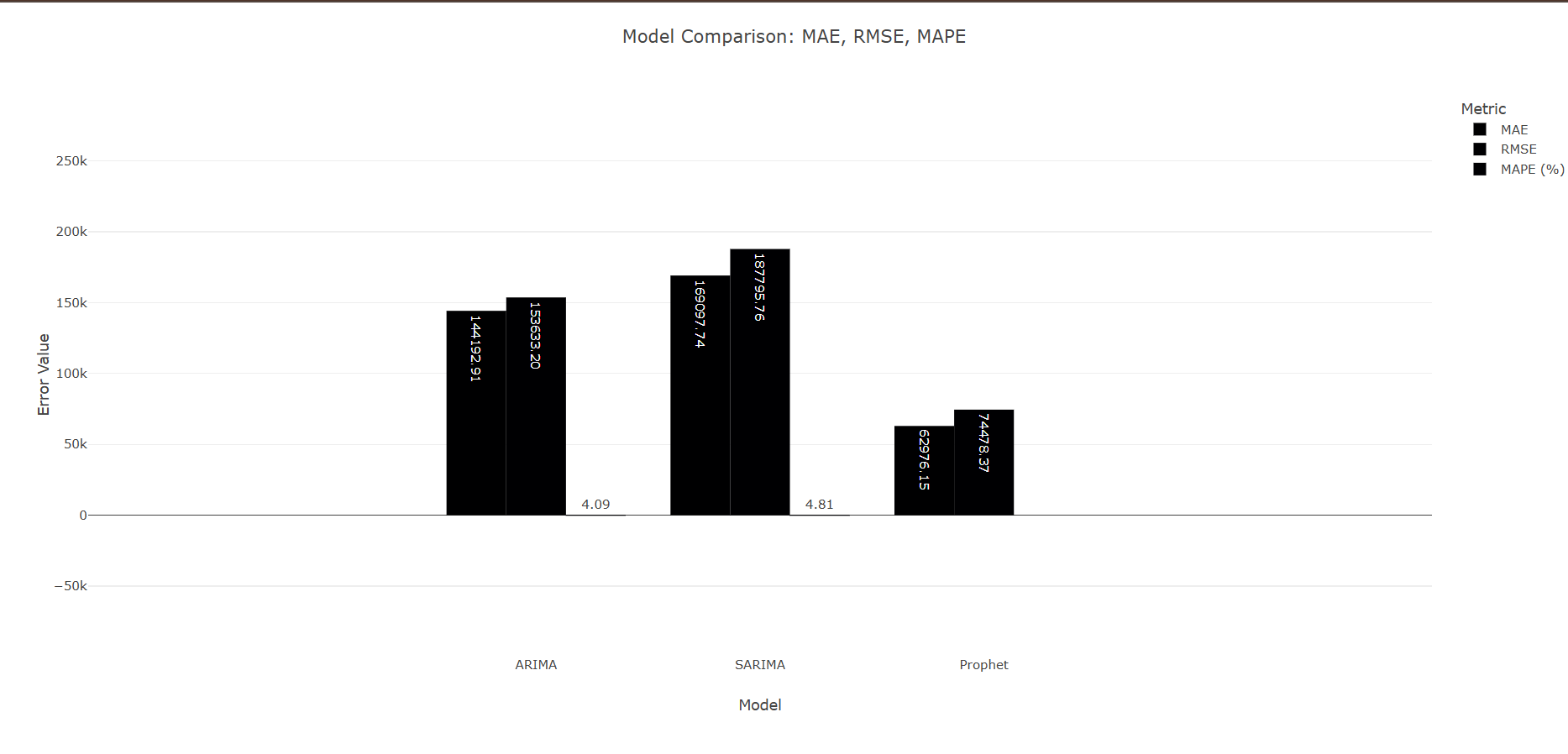
**Forecasting Methodology**:

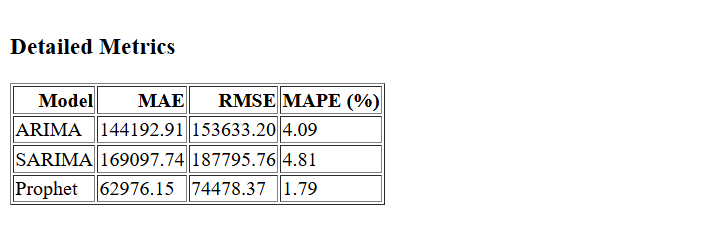
* The sales data was aggregated monthly and cleaned.
* Prophet was applied after mapping date and sales columns to ds and y.
* The tool handles missing values and supports multiple products.
* Forecasts were generated with trend and seasonality components modeled separately.

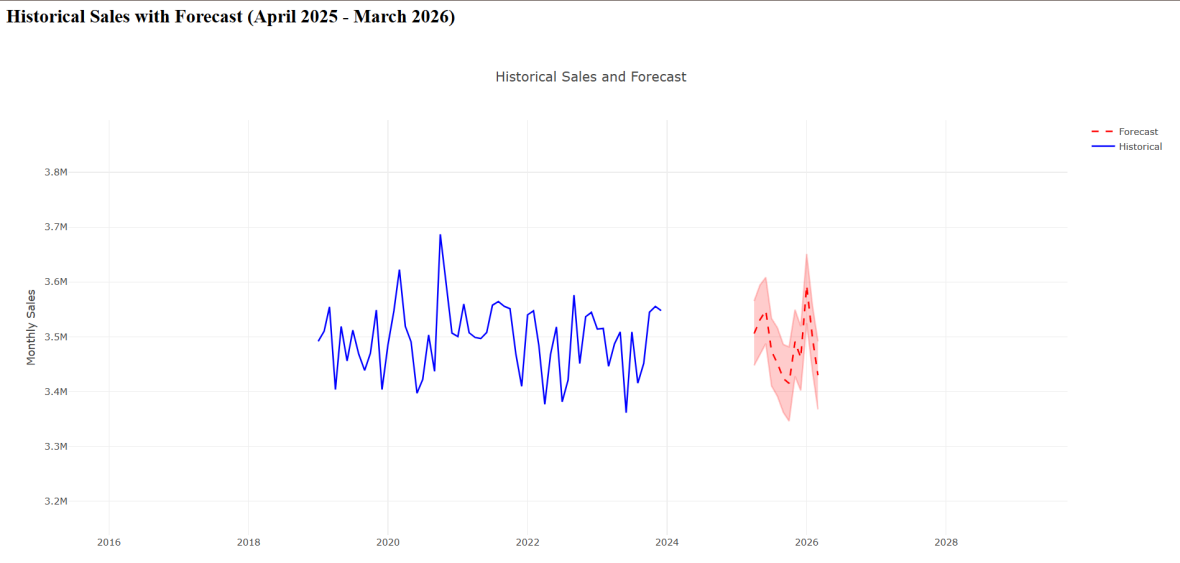
**Key Evaluation Metrics**:

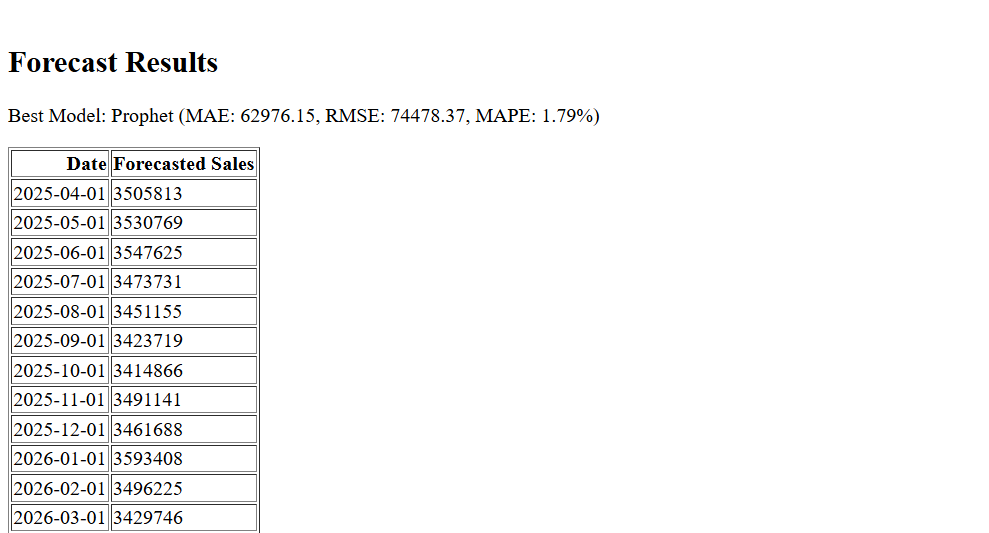
* **MAE (Mean Absolute Error)**: Captured average forecast deviation from actual values.
* **RMSE (Root Mean Squared Error)**: Penalized larger errors, giving insight into overall prediction quality.
* **MAPE (Mean Absolute Percentage Error)**: Allowed evaluation of forecast accuracy in percentage terms, which is useful across different scales.
* These metrics were logged clearly for each product to assess model performance.











**Enhancement Scope**

1. Dashboard using Dash or Streamlit
2. Incorporate sales promotions or holiday effects
3. Enable REST API for real-time forecasting
4. Use LSTM for deep learning comparison

**Link to code and executable file**

**<https://drive.google.com/drive/folders/1QfaW0_lf880tHFh5rHBy3Ada1hBoz4XN?usp=drive_link>**

**Research questions and responses**

**Q: Why use Prophet over ARIMA or LSTM?**  
**A:** Prophet is interpretable, easy to tune, handles missing values well, and is good with business seasonality patterns.

**Q: How reliable is MAPE in product sales data?**  
**A:** MAPE is useful, but less reliable when actual sales values are near zero. MAE and RMSE are more stable.

**Q: How can this be extended to multi-product forecasting?**  
**A:** By looping over unique product values and creating a Prophet model for each, we can automate batch forecasting.