Pharma Sales Analysis and Forecasting

> **A comprehensive time-series analysis and forecasting case study for pharmaceutical product sales at small scale, benchmarking ARIMA, Prophet, and LSTM against baseline methods.**

Project Overview

This project implements a **complete time-series forecasting pipeline** for pharmaceutical sales data, addressing the research question:

> *Can modern time-series forecasting methods outperform Naïve baselines for small-scale pharmaceutical sales prediction?*

Key Capabilities

- **Seasonality Analysis** — Annual, weekly, and daily pattern detection
- **Stationarity Testing** — ADF and KPSS statistical tests
- **Autocorrelation Analysis** — ACF/PACF for parameter selection
- **Multiple Forecasting Methods** — ARIMA, SARIMA, Prophet, LSTM
- **Benchmark Comparison** — Against Naïve, Seasonal Naïve, and Average baselines

Repository Structure

```
Pharma-Sales-Analysis-and-Forecasting-main/
│
├── pharma_sales_data_analysis_and_forecasting.ipynb  # Main notebook (100
cells, 1200+ lines)
├── salesdaily.csv                                    # Daily aggregated
sales
├── salesweekly.csv                                   # Weekly aggregated
sales (primary)
├── salesmonthly.csv                                  # Monthly aggregated
sales
├── saleshourly.csv                                   # Hourly aggregated
sales
├── README.md                                         # Original project
readme
└── my_readme.md                                      # This comprehensive
guide
```

Dataset Description

Source

- **Origin:** Point-of-Sale system from a single pharmacy
- **Period:** 6 years (2014–2019)
- **Raw Data:** 600,000 transactional records
- **Aggregation:** Classified into 8 ATC drug categories

Time Series Granularity

File	Rows	Frequency	Use Case
`saleshourly.csv`	~52,560	Hourly	Daily pattern analysis
`salesdaily.csv`	~2,190	Daily	Seasonality analysis
`salesweekly.csv`	302	Weekly	**Primary forecasting dataset**
`salesmonthly.csv`	~72	Monthly	Trend analysis

Drug Categories (ATC Classification)

Code	Category	Description
M01AB	Anti-inflammatory	Acetic acid derivatives (e.g., Diclofenac)
M01AE	Anti-inflammatory	Propionic acid derivatives (e.g., Ibuprofen)
N02BA	Analgesics	Salicylic acid derivatives (e.g., Aspirin)
N02BE	Analgesics	Pyrazolones and Anilides (e.g., Paracetamol)
N05B	Psycholeptics	Anxiolytic drugs
N05C	Psycholeptics	Hypnotics and sedatives
R03	Respiratory	Drugs for obstructive airway diseases
R06	Antihistamines	Antihistamines for systemic use

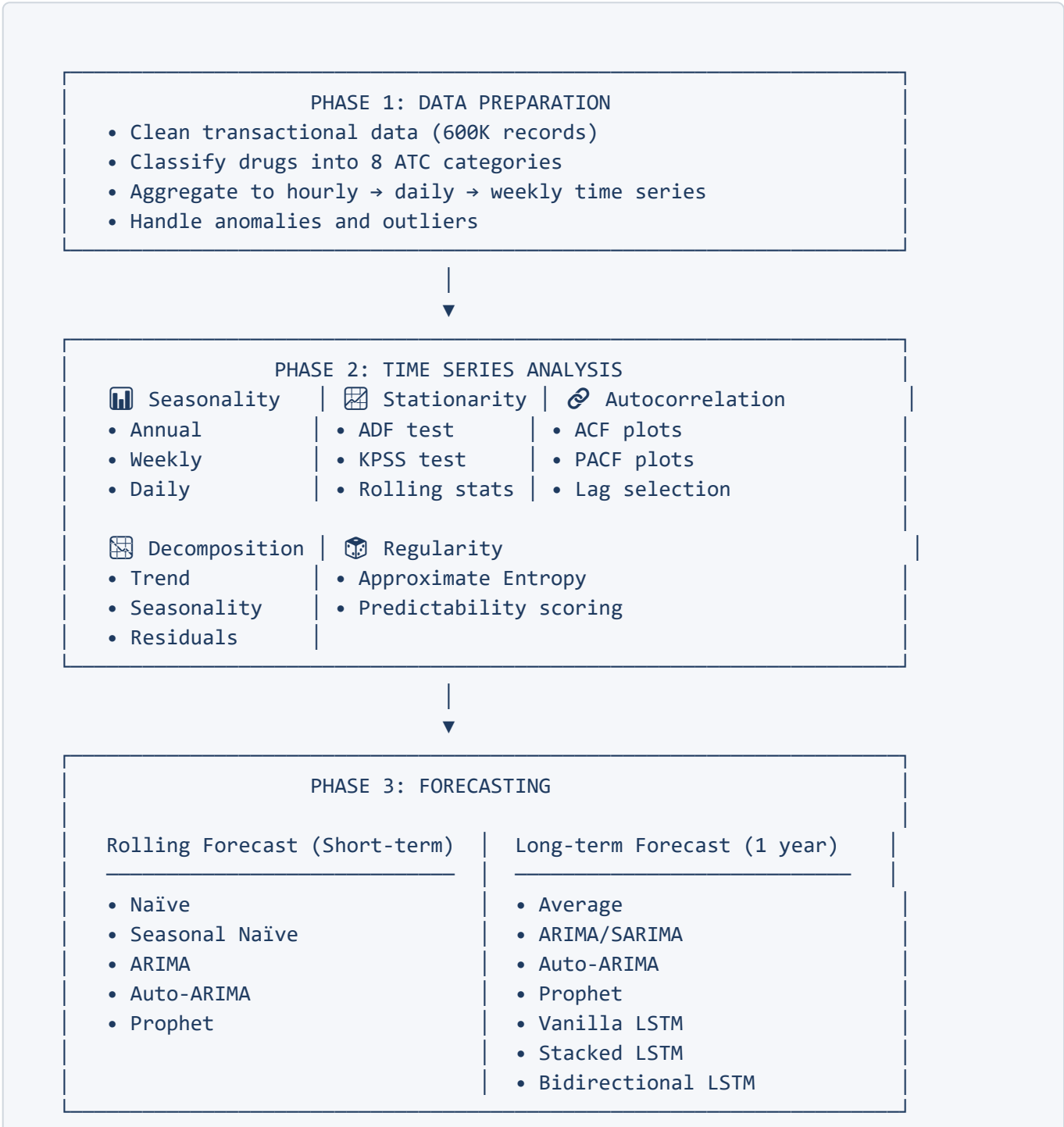
Data Schema

Column	Type	Description
`datum` / `DATE`	datetime	Timestamp of aggregation period
`M01AB`	float	Sales quantity for category
`M01AE`	float	Sales quantity for category
`N02BA`	float	Sales quantity for category
`N02BE`	float	Sales quantity for category
`N05B`	float	Sales quantity for category
`N05C`	float	Sales quantity for category

Column	Type	Description
`R03`	float	Sales quantity for category
`R06`	float	Sales quantity for category
`Month`	int	Month (1-12)
`Year`	int	Year
`Weekday Name`	str	Day of week

Methodology

Three-Phase Pipeline



Train-Test Split

Parameter	Value
Total Observations	302 weeks
Training Set	250 weeks (≈83%)
Test Set	52 weeks (1 year)
Split Method	Chronological (time-based)

📈 Time Series Analysis Results

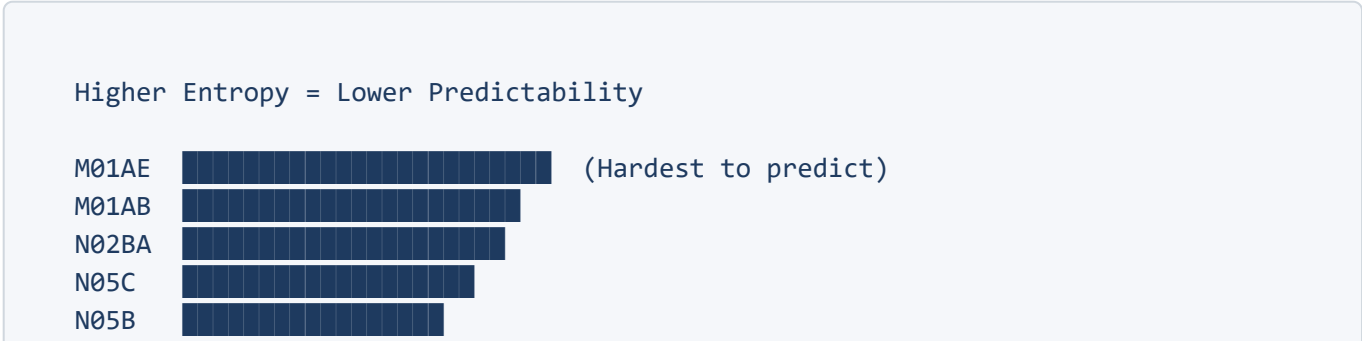
Seasonality Detection

Category	Annual	Weekly	Daily	Outliers
M01AB	◯ Weak	◯ Weak	☑ Yes	Medium
M01AE	◯ Weak	◯ Weak	☑ Yes	High
N02BA	◯ Weak	◯ Weak	☑ Yes	Medium
N02BE	☑ **Strong**	☑ Yes	☑ Yes	Low
N05B	✗ None	✗ None	◯ Weak	Medium
N05C	✗ None	✗ None	◯ Weak	High
R03	☑ **Strong**	◯ Weak	☑ Yes	High
R06	☑ **Strong**	◯ Weak	☑ Yes	Medium

Stationarity Tests

Test	Result Summary
ADF Test	All stationary except N02BA (P=0.249)
KPSS Test	Trend non-stationarity in N02BE, R03, R06

Approximate Entropy (Predictability)



```
R03    ████████████
R06    ██████████
N02BE  ████████                (Easiest to predict)
```

Residuals Analysis (% of Observed)

Category	Residuals %	Interpretation
N05C	~35%	High noise, low predictability
R03	~30%	Significant residuals
M01AB	~25%	Moderate noise
M01AE	~25%	Moderate noise
N02BE	~15%	Low noise, higher predictability
R06	~15%	Low noise, higher predictability

Forecasting Methods

1. Baseline Methods

Method	Description	Best For
Naïve	$f_{t+1} = o_t$	Random walk data
Seasonal Naïve	$f_{t+1} = o_{t-m}$	Seasonal data
Average	$f = \bar{o}_{train}$	Long-term baseline

2. ARIMA/SARIMA

ARIMA(p, d, q) Parameters:

- `p` — AR order (PACF cutoff lag)
- `d` — Differencing degree (0 if stationary)
- `q` — MA order (ACF cutoff lag)

SARIMA(p, d, q)(P, D, Q, m) Additional Parameters:

- `P` — Seasonal AR order
- `D` — Seasonal differencing
- `Q` — Seasonal MA order
- `m` — Seasonal period (52 for weekly data)

3. Facebook Prophet

Hyperparameter	Description

Hyperparameter	Description
`growth`	Linear or logistic trend
`changepoint_prior_scale`	Trend flexibility
`seasonality_prior_scale`	Seasonality flexibility
`fourier_order`	Seasonality complexity
`interval_width`	Uncertainty interval

4. LSTM Architectures

```
┌─────────────────────────────────────────────────────────────┐
│                      VANILLA LSTM                           │
│  Input → [LSTM Layer] → Dense → Output                      │
└─────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────┐
│                      STACKED LSTM                           │
│  Input → [LSTM] → [LSTM] → [LSTM] → Dense → Output          │
└─────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────┐
│                    BIDIRECTIONAL LSTM                       │
│  Input → [Forward LSTM ⇔ Backward LSTM] → Dense → Output    │
└─────────────────────────────────────────────────────────────┘
```

LSTM Data Preparation:

1. Transform to stationary series
2. Sequence to supervised format: `[X_{t-n}...X_{t-1}] → [y_t]`
3. Scale (normalize/standardize)
4. Fixed random seeds for reproducibility

📊 Evaluation Metrics

Metric	Formula	Purpose
MSE	$\frac{1}{n}\sum(y - \hat{y})^2$	Primary accuracy metric
MAPE	$\frac{100}{n}\sum\|\frac{y - \hat{y}}{y}\|$	Percentage interpretation
MAE	$\frac{1}{n}\sum\|y - \hat{y}\|$	Absolute error

Results Tracking

Rolling Forecast Methods (5):

- Naïve, Seasonal Naïve, ARIMA, Auto-ARIMA, Prophet

Long-term Forecast Methods (7):

- Average, ARIMA, Auto-ARIMA, Prophet, Vanilla LSTM, Stacked LSTM, Bidirectional LSTM

🚀 Quick Start

Installation

```
# Clone repository
git clone <repository-url>
cd Pharma-Sales-Analysis-and-Forecasting-main

# Create virtual environment
python -m venv pharma_env
source pharma_env/bin/activate  # Windows: pharma_env\Scripts\activate

# Install dependencies
pip install numpy pandas matplotlib seaborn scikit-learn
pip install statsmodels pyramid-arima prophet
pip install tensorflow keras
```

Dependencies

```
numpy>=1.21.0
pandas>=1.3.0
matplotlib>=3.4.0
seaborn>=0.11.0
scikit-learn>=1.0.0
statsmodels>=0.13.0
pmdarima>=2.0.0          # Auto-ARIMA
prophet>=1.1.0           # Facebook Prophet
tensorflow>=2.8.0        # LSTM models
keras>=2.8.0
```

Run Notebook

```
jupyter notebook pharma_sales_data_analysis_and_forecasting.ipynb
```

📓 Notebook Workflow

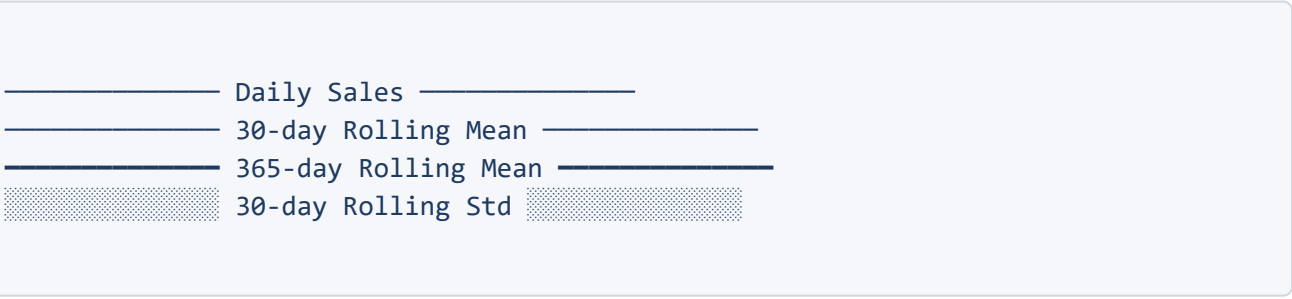
Section	Description	Key Outputs
1. Introduction	Problem statement & research question	Context
2. Methodology	Pipeline overview	Approach definition
3.1 Seasonality	Boxplots, rolling means, decomposition	Seasonal patterns
3.2 Stationarity	ADF, KPSS tests	Stationarity flags
3.3 Regularity	Approximate Entropy	Predictability scores
3.4 Autocorrelation	ACF/PACF plots	ARIMA parameters
3.5 Distribution	Daily sales patterns	Time-of-day insights
4.1 Baselines	Naïve, Seasonal Naïve, Average	Benchmark MSE/MAPE
4.2 ARIMA	Grid search optimization	Tuned ARIMA models
4.3 Prophet	Hyperparameter tuning	Prophet forecasts
4.4 LSTM	Vanilla, Stacked, Bidirectional	Neural network forecasts
5. Results	Comparison tables	Best methods per category

📊 Expected Visualizations

1. Seasonality Boxplots

Monthly and weekly distribution of sales by drug category.

2. Rolling Statistics



3. STL Decomposition

- **Trend:** Long-term direction
- **Seasonality:** Repeating patterns
- **Residuals:** Random noise

4. ACF/PACF Correlograms

Visual lag selection for ARIMA parameters.

5. Forecast Comparison Plots

Actual vs Predicted for each method and category.

Business Applications

Sales Strategy Recommendations

Finding	Recommendation
R03, R06, N02BE show annual seasonality	Plan inventory for seasonal peaks
Weekend sales drop	Optimize staffing for weekdays
Morning/afternoon peaks	Schedule promotions for high-traffic hours
N05B/N05C irregular	Maintain safety stock buffer

Potential Explanatory Variables (Future Work)

Variable	Impact
Weather data	Atmospheric pressure → M01AB/M01AE sales
Drug prices	Discounts → sales spikes
Pension dates	State pension payoff → sales peaks
Holidays	Non-working days → Sunday-like patterns

⚠ Limitations & Assumptions

Limitation	Implication
Single pharmacy	Results may not generalize to chains
Univariate forecasting	External factors not modeled
Fixed train-test split	Cross-validation could improve estimates
No hyperparameter tuning for LSTM	LSTM performance potentially suboptimal
High residuals for some categories	N05B, N05C remain difficult to predict

---

Theoretical Background

Time Series Components

$$y(t) = Level + Trend + Seasonality + Noise \quad \text{(Additive)}$$

$$y(t) = Level \times Trend \times Seasonality \times Noise \quad \text{(Multiplicative)}$$

Stationarity Tests

- **ADF Test:** Null hypothesis = non-stationary. P < 0.05 → reject → stationary

- **KPSS Test:** Null hypothesis = trend-stationary. P < 0.05 → reject → non-stationary

ARIMA Selection Rules

Plot	Observation	Action
PACF	Sharp cutoff at lag k	Set p = k
ACF	Sharp cutoff at lag k	Set q = k
ACF	Positive lag-1 autocorrelation	Consider AR term
ACF	Negative lag-1 autocorrelation	Consider MA term

🔧 Troubleshooting

Issue	Solution
`pyramid-arima` not found	`pip install pmdarima` (renamed package)
Prophet installation fails	Use `conda install -c conda-forge prophet`
TensorFlow GPU errors	Set `CUDA_VISIBLE_DEVICES=""` for CPU
Memory error with LSTM	Reduce batch size or sequence length
Convergence warnings	Increase `max_iter` in ARIMA

📚 References

- statsmodels Time Series Documentation
- Facebook Prophet
- TensorFlow LSTM Guide
- ATC Classification System
- Time Series Forecasting with Python

👤 Credits

Research Focus: Small-scale pharmaceutical sales forecasting
Data Source: Single pharmacy Point-of-Sale system (2014-2019)
Methodology: Problem-neutral time series forecasting pipeline

---

🤝 Contributing

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/enhancement`)
3. Commit changes (`git commit -m 'Add enhancement'`)
4. Push to branch (`git push origin feature/enhancement`)
5. Open a Pull Request

Star this repo if you find it useful!

Made with for Pharmaceutical Analytics