

Prophet Forecaster

A comprehensive forecasting tool that leverages Facebook Prophet to generate accurate predictions for SKU sales in warehouse environments. This application supports multiple time granularities (daily, weekly, monthly) and provides customizable parameters for optimal forecasting results.

Features

- Upload CSV files with automatic delimiter detection (comma or semicolon)
- Process sales data with flexible date formats and SKU quantity columns
- Generate forecasts using Facebook Prophet with customizable parameters
- Support for multiple time granularities: daily, weekly, and monthly
- Outlier detection and handling using IQR method
- Holiday effects modeling for Brazilian holidays
- Cross-validation capabilities for model evaluation
- Interactive visualization of forecasts and components
- Export forecast data and comparison metrics as CSV

About Prophet

Facebook Prophet is a time series forecasting procedure designed for business forecasting tasks. It's particularly effective for:

- Data with strong seasonal effects and multiple seasons of historical data
- Missing data and outliers
- Shifts in trend and non-linear growth curves
- Holiday effects

Prophet decomposes time series into trend, seasonality, and holiday components, making it ideal for SKU sales forecasting in warehouse environments where these patterns are common.

Model Parameters by Time Granularity

Daily Granularity

- **Seasonality Mode:** Multiplicative - Captures percentage changes in seasonality as the trend increases
- **Seasonality Components:** Weekly seasonality with `fourier_order=5`
- **Regressors:** Weekend effect (`is_weekend`) with `prior_scale=25`
- **Changepoint Prior Scale:** 0.5 - Allows flexibility in trend changes to capture daily volatility
- **Confidence Interval:** 80% - Provides practical prediction bounds for operational planning
- **Benefits:** Captures day-of-week patterns, weekend effects, and daily fluctuations

Weekly Granularity

- **Seasonality Mode:** Additive - More stable for weekly aggregated data
- **Seasonality Components:** Custom yearly seasonality based on 52.18 weeks per year
- **Regressors:** Holiday effects only (weekend regressor not applicable)

- **Changepoint Prior Scale:** 0.05 - More conservative to prevent overfitting with fewer data points
- **Confidence Interval:** 80% - Balanced between capturing uncertainty and practical usability
- **Benefits:** Smooths daily noise, focuses on longer-term patterns, more stable for inventory planning

Monthly Granularity

- **Seasonality Mode:** Multiplicative - Captures percentage changes in monthly patterns
- **Seasonality Components:** Quarterly seasonality (period=3 months)
- **Changepoint Prior Scale:** 0.5 - Allows detection of significant trend changes
- **Confidence Interval:** 80% - Consistent with other granularities
- **Benefits:** Identifies quarterly and annual patterns, useful for long-term planning

Expected Results for Warehouse SKU Forecasting

A good forecast for warehouse SKU sales should demonstrate:

1. Accuracy Metrics:

- MAE (Mean Absolute Error): Lower values indicate better accuracy
- RMSE (Root Mean Square Error): Sensitive to large errors
- MAPE (Mean Absolute Percentage Error): Ideally below 50% for retail/warehouse SKUs

2. Pattern Recognition:

- Captures weekly patterns (higher sales on specific days)
- Accounts for seasonal trends (monthly/quarterly/annual)
- Identifies holiday effects on demand

3. Operational Utility:

- Confidence intervals narrow enough for practical inventory decisions
- Accurate prediction of demand spikes and troughs
- Reliable for both short-term (next week) and medium-term (next month) planning

Requirements

- Python 3.7+
- Dependencies listed in `requirements.txt`

Installation

1. Clone this repository
2. Install dependencies:

```
pip install -r requirements.txt
```

Usage

1. Run the Streamlit application:

```
streamlit run app.py
```

2. Open your browser and navigate to the URL shown in the terminal (usually `http://localhost:8501`)
3. Upload a CSV file with the following format:
 - Separator: semicolon (👉)
 - Required columns:
 - **DATA**: Date in dd-mm-yyyy format
 - One or more columns named **QUANTIDADE_VENDIDA_SKU_XXX** where XXX is the SKU ID
4. Adjust forecast parameters if needed
5. Click "Generate Forecasts" to see the results

CSV File Example

```
DATA;QUANTIDADE_VENDIDA_SKU_123;QUANTIDADE_VENDIDA_SKU_456
01-01-2023;10;15
02-01-2023;12;14
03-01-2023;15;16
...
```

Project Structure

- **app.py**: Main Streamlit application
- **src/util/forecaster.py**: Contains the forecasting functions
- **requirements.txt**: List of dependencies