Manual

Walmart Sales Forecasting Web Applications

Author:	Hemanth Jadiswami Prabhakaran
Matriculation No.:	7026000
Course of Studies:	Mechanical Engineering
First examiner:	Prof. Dr. Elmar Wings
Submission date:	May 5, 2025

University of Applied Sciences Emden/Leer · Faculty of Technology · Mechanical Engineering Department
Constantiaplatz 4 · 26723 Emden · `http://www.hs-emden-leer.de`

Contents

List of Figures

List of Tables

Acronyms

1 Introduction

This manual guides you through using the Walmart Sales Forecasting System, a two-application solution for predicting weekly sales data.

1.1 What is the Walmart Sales Forecasting System?

The system consists of two complementary applications:

- **Training App**: Runs locally on your computer to train forecasting models
- **Prediction App**: Accessible online to generate 4-week sales forecasts

1.2 Purpose and Benefits

This system enables:

- Training custom forecasting models using your historical sales data
- Generating 4-week sales forecasts with interactive visualizations
- Downloading predictions in multiple formats for further analysis
- Comparing different forecasting models (Auto ARIMA and Exponential Smoothing)

1.3 Target Audience

This manual is designed for:

- Business analysts needing sales forecasts
- Data analysts working with time series data
- Students learning about forecasting methods
- Anyone requiring sales predictions without deep technical knowledge

1.4 Quick Start Overview

The typical workflow involves:

1. Setting up the Training App on your local computer
2. Uploading historical sales data and training models
3. Saving trained models
4. Accessing the online Prediction App
5. Loading models or using defaults to generate forecasts

1.5 Manual Organization

This manual covers:

- System overview and architecture
- Local Training App setup and usage
- Cloud-based Prediction App usage
- Data requirements and formats
- Model types and selection guidance
- Common troubleshooting solutions
- Frequently asked questions

2 System Overview

2.1 Architecture Overview

The Walmart Sales Forecasting System comprises two distinct applications that work together to provide end-to-end forecasting capabilities.

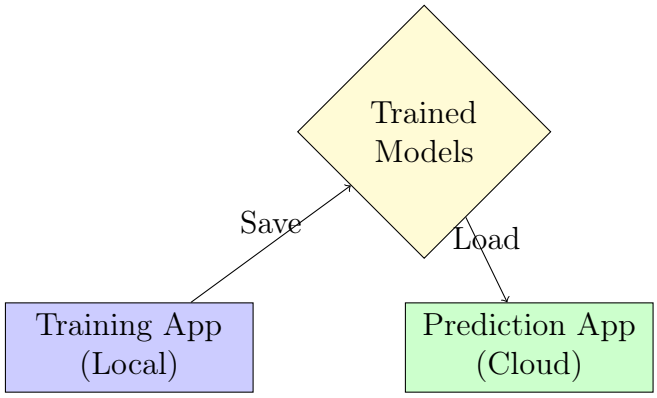

Figure 2.1: High-Level System Architecture

2.2 Application Roles

2.2.1 Training Application (Local)

- Installed and run on your personal computer
- Processes historical sales data
- Trains forecasting models using selected algorithms
- Saves models to disk for later use

2.2.2 Prediction Application (Cloud)

- Accessible through any web browser
- No installation required
- Uses trained models to generate forecasts
- Provides interactive visualization and data export

2.3 Key Features

2.3.1 Model Training

- Support for two modeling algorithms:
  - Auto ARIMA (automatic parameter selection)
  - Exponential Smoothing (Holt-Winters method)
- Customizable hyperparameters
- Performance evaluation using WMAE metric

2.3.2 Prediction Generation

- 4-week forecast generation
- Interactive visualizations
- Multiple export formats (CSV, JSON)
- Summary statistics

2.4 System Requirements

2.4.1 Training App

- Windows, macOS, or Linux
- Python 3.8+
- Internet connection for initial setup
- 2GB RAM minimum

2.4.2 Prediction App

- Modern web browser (Chrome, Firefox, Safari, Edge)
- Stable internet connection
- No additional software required

3 Local App Setup

3.1 Prerequisites

Before installing the Training App, ensure you have:

- Python 3.8 or higher installed
- Basic command line familiarity
- Internet connection for downloading dependencies

3.2 Installation Steps

3.2.1 Step 1: Download the Application

Download the Training App files to a local directory on your computer.

3.2.2 Step 2: Install Dependencies

Open a terminal/command prompt and navigate to the application directory:

```
cd path/to/streamlit_app_training
```

Install required packages:

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

3.2.3 Step 3: Launch the Application

Start the Training App:

```
streamlit run app.py
```

The application will open in your default web browser at `http://localhost:8501`.

3.3 Verification

After successful installation, you should see:

- The Training App interface in your browser
- File upload sections for CSV files
- Model selection options
- Training controls

3.4 Troubleshooting Installation

3.4.1 Common Issues

Problem	**Solution**
Python not found	Ensure Python is in your system PATH
Module import errors	Check Python version (3.8+) and reinstall dependencies
Port already in use	Use a different port: `streamlit run app.py --server.port=8502`

Table 3.1: Installation Troubleshooting

3.5 Optional: Test Data Generation

To test the application without real data:

```
python utils/test_data_generator.py
```

This generates sample CSV files for testing.

3.6 Updating the Application

To update dependencies:

1. Navigate to the application directory
2. Run: `pip install -r requirements.txt --upgrade`
3. Restart the application

4 Model Training

4.1 Data Upload

4.1.1 Required Files

Upload three CSV files in this order:

1. **train.csv**: Historical sales data
2. **features.csv**: Store features and markdown data
3. **stores.csv**: Store information

4.1.2 File Upload Process

- Click each `Browse files` button or drag-and-drop files
- Wait for the green success message
- Review data information displayed

4.2 Model Selection

4.2.1 Available Models

Auto ARIMA

- Automatically finds optimal parameters
- Best for complex seasonal patterns
- May take longer to train

Exponential Smoothing

- Simpler and faster to train
- Good for data with clear trends
- More straightforward interpretation

4.2.2 Choosing a Model

Select your preferred model from the dropdown menu. For beginners, start with Exponential Smoothing.

4.3 Hyperparameter Configuration

4.3.1 Auto ARIMA Parameters

- **Start p/q**: Initial values for parameter search
- **Max p/q**: Maximum values to search
- **Seasonal P/Q**: Parameters for seasonal components

4.3.2 Exponential Smoothing Parameters

- **Seasonal periods**: Number of weeks in a season (default: 20)
- **Seasonal type**: Additive or multiplicative
- **Trend type**: Additive, multiplicative, or none
- **Damped**: Whether to use damped trend

4.4 Training Process

4.4.1 Starting Training

1. Click `Start Training`
2. Monitor the progress indicator
3. Wait for training to complete

4.4.2 During Training

- Do not close the browser or stop the application
- Training time varies based on data size and model complexity
- ARIMA typically takes longer than Exponential Smoothing

4.5 Results and Evaluation

4.5.1 Understanding Results

After training completes, you'll see:

- **WMAE score**: Lower values indicate better accuracy
- **Diagnostic plots**: Visual comparison of predictions vs actual data
- **Model information**: Parameters used and performance metrics

Figure 4.1: Example Diagnostic Plot

4.5.2 Interpreting WMAE

- WMAE: Weighted Mean Absolute Error
- Lower values indicate better model performance
- Compare across different models and parameters

4.6 Saving Models

4.6.1 Automatic Saving

Models are automatically saved to:

```
models/default/
|- auto_arima.pkl
‘- exponential_smoothing.pkl
```

4.6.2 Downloading Models

1. Click the `Download Model` button
2. Save the .pkl file to your computer
3. Upload to Prediction App when needed

4.7 Best Practices

- Start with default parameters
- Try both models to compare performance
- Keep track of WMAE scores for comparison
- Save successful models with descriptive names
- Retrain models quarterly for best accuracy

5 Cloud App Usage

5.1 Accessing the Prediction App

The Prediction App is available online at: `https://your-app-url.streamlit.app`

5.1.1 No Installation Required

- Open any modern web browser
- Navigate to the application URL
- Begin using immediately

5.2 Loading Models

5.2.1 Using Default Models

1. Click on the `Default Models` tab
2. Choose between:
    - `Load Auto ARIMA Model`
    - `Load Exponential Smoothing Model`
3. Wait for the success message

5.2.2 Uploading Custom Models

1. Click on the `Upload Model` tab
2. Select model type from dropdown
3. Click `Choose file` and select your .pkl file
4. Click `Load Uploaded Model`

5.3 Generating Forecasts

5.3.1 Creating Predictions

1. Ensure a model is loaded (default or uploaded)
2. Click `Generate 4-Week Forecast`
3. Wait for processing to complete

5.3.2 Viewing Results

The forecast displays:

- Interactive line chart showing 4-week predictions
- Data table with exact values
- Summary statistics

Figure 5.1: 4-Week Sales Forecast Visualization

5.4 Working with Results

5.4.1 Interactive Visualization

- Hover over chart for detailed values
- Pan and zoom using Plotly controls
- View predictions by week

5.4.2 Summary Statistics

Review key metrics:

- Total predicted sales
- Average weekly sales
- Highest sales week
- Growth rate percentage

5.5 Downloading Forecasts

5.5.1 Available Formats

- **CSV**: For spreadsheet applications
- **JSON**: For programming use

5.5.2 Download Process

1. Scroll to the `Download Results` section
2. Click desired format button
3. Save file to your computer

5.6 Session Management

5.6.1 Model Sessions

- Models remain loaded during browser session
- Sessions reset after 30 minutes of inactivity
- Reload models as needed

5.6.2 Multiple Predictions

- Generate multiple forecasts in one session
- Switch between models without page refresh
- Download each forecast separately

5.7 Best Practices

- Use browser bookmarks for quick access
- Save frequently used models on your computer
- Download forecasts immediately after generation
- Keep browser tabs open during active use
- Review forecast accuracy over time

6 Data Requirements

6.1 Input Data Formats

6.1.1 CSV File Requirements

All input files must be:

- Comma-separated values (.csv)
- UTF-8 encoded
- Headers must match exactly as specified

6.1.2 File Specifications

train.csv

Column	Type	Description
Store	Integer	Store identifier
Date	String	Date (YYYY-MM-DD)
Weekly_Sales	Float	Sales for the week
IsHoliday	Boolean	Holiday flag

Table 6.1: train.csv Structure

features.csv

Column	Type	Description
Store	Integer	Store identifier
Date	String	Date (YYYY-MM-DD)
Temperature	Float	Temperature in Fahrenheit
Fuel_Price	Float	Gas price
MarkDown1-5	Float	Markdown amounts
CPI	Float	Consumer Price Index
Unemployment	Float	Unemployment rate
IsHoliday	Boolean	Holiday flag

Table 6.2: features.csv Structure

stores.csv

Column	Type	Description
Store	Integer	Store identifier
Type	String	Store type (A, B, C)
Size	Integer	Store size in sq ft

Table 6.3: stores.csv Structure

6.2 Data Quality Requirements

6.2.1 Valid Data

- No negative sales values
- Consistent date formats
- Complete store identifiers
- Proper number formats (no text in numeric fields)

6.2.2 Data Completeness

- Minimum 2 years of historical data
- Complete weekly coverage (no missing weeks)
- All stores must be present in all files

6.3 Model File Formats

6.3.1 Trained Model Files

- File extension: .pkl
- Saved using joblib serialization
- Binary format, not human-readable

6.3.2 File Naming

- Default: `auto_arima.pkl` or `exponential_smoothing.pkl`
- Custom: Use descriptive names (e.g., `store_1_arima_2024.pkl`)

6.4 Output Formats

6.4.1 Forecast CSV

```
Week,Date,Predicted_Sales
Week 1,2024-01-08,15234.67
Week 2,2024-01-15,16123.45
Week 3,2024-01-22,15987.23
Week 4,2024-01-29,16345.89
```

6.4.2 Forecast JSON

```
[
{
 "Week": "Week 1",
 "Date": "2024-01-08",
 "Predicted_Sales": 15234.67
},
...
]
```

6.5 Data Validation

6.5.1 Automatic Checks

The system automatically validates:

- Column names and types
- Date format consistency
- Numeric value ranges
- Required fields presence

6.5.2 Error Messages

Common validation errors:

- `"Missing required column: X"`
- `"Invalid date format"`
- `"Negative sales values found"`
- `"Store ID mismatch between files"`

6.6 Data Preparation Tips

- Verify all CSV files before upload
- Use consistent date formats (YYYY-MM-DD)
- Remove any special characters from numeric fields
- Ensure all required columns are present
- Check for duplicate entries

7 Model Details

7.1 Model Types Overview

7.1.1 Auto ARIMA

Description: Automatic ARIMA (AutoRegressive Integrated Moving Average) model that automatically determines optimal parameters.

Strengths:

- Handles complex seasonal patterns automatically
- Adapts to various data characteristics
- No manual parameter tuning required

Limitations:

- Longer training time
- May overfit small datasets
- Complex output interpretation

7.1.2 Exponential Smoothing

Description: Holt-Winters exponential smoothing method for time series forecasting.

Strengths:

- Fast training and prediction
- Good for stable patterns
- Simple to interpret

Limitations:

- Limited handling of complex seasonality
- Requires manual parameter selection
- Less adaptable to irregular patterns

7.2 Model Selection Guidelines

7.2.1 When to Use Auto ARIMA

- Data shows complex seasonal patterns
- Multiple seasonalities present
- Irregular time series patterns
- Historical data varies significantly

7.2.2 When to Use Exponential Smoothing

- Data shows clear trends and seasonality
- Fast results needed
- Simple interpretation required
- Consistent historical patterns

7.3 Understanding Parameters

7.3.1 Auto ARIMA Parameters

Parameter	Description
p, q	Non-seasonal AR and MA orders
P, Q	Seasonal AR and MA orders
d, D	Differencing orders
max_p, max_q	Search limits for parameters
seasonal	Enable seasonal components

Table 7.1: Auto ARIMA Parameters

7.3.2 Exponential Smoothing Parameters

7.4 Performance Evaluation

7.4.1 WMAE Metric

Weighted Mean Absolute Error (WMAE):

- Measures prediction accuracy

Parameter	Description
seasonal_periods	Number of periods in a season
seasonal	Type: additive or multiplicative
trend	Trend component type
damped	Apply damping to trend

Table 7.2: Exponential Smoothing Parameters

- Weights errors by importance
- Lower values indicate better performance

Typical WMAE Ranges:

- Excellent: < 0.05
- Good: 0.05 - 0.10
- Acceptable: 0.10 - 0.20
- Poor: > 0.20

7.4.2 Diagnostic Plots

- Training data (blue line)
- Test data (orange line)
- Model predictions (green line)
- Visual assessment of model fit

7.5 Model Limitations

7.5.1 General Limitations

- Cannot predict external shocks
- Historical patterns may not continue
- Accuracy decreases with forecast horizon
- Requires sufficient historical data

7.5.2 Specific Constraints

- 4-week forecast limit
- Weekly granularity only
- No multi-store forecasting
- No external variable integration

7.6 Model Versioning

7.6.1 Version Tracking

- Include date in model filenames
- Keep WMAE scores for comparison
- Document parameter changes
- Archive old models for reference

7.6.2 Model Updates

- Retrain quarterly for best accuracy
- Update with new data regularly
- Compare performance against existing models
- A/B test new vs. old models

8 Troubleshooting

8.1 Common Issues and Solutions

8.1.1 File Upload Problems

Problem: CSV file not accepted
Solutions:

- Verify file extension is .csv
- Check file encoding (must be UTF-8)
- Ensure column names match requirements exactly
- Remove any special characters

Problem: Data validation errors
Solutions:

- Check for missing required columns
- Verify date format (YYYY-MM-DD)
- Look for negative sales values
- Ensure consistent store IDs across files

8.1.2 Training Failures

Problem: Training times out or fails
Solutions:

- Reduce max_p and max_q values for ARIMA
- Try Exponential Smoothing instead
- Check data quality (remove outliers)
- Ensure sufficient historical data

Problem: Poor model performance
Solutions:

- Try different hyperparameters
- Switch to alternative model type
- Clean and preprocess data better
- Add more historical data

8.1.3 Model Loading Issues

Problem: Model file won't load in Prediction App
Solutions:

- Verify model type matches selection
- Check file corruption (redownload)
- Ensure file saved correctly in Training App
- Verify joblib compatibility

Problem: Default models not found
Solutions:

- Clear browser cache
- Refresh page and try again
- Check internet connection
- Contact support if persistent

8.1.4 Prediction Generation Errors

Problem: Forecast fails to generate
Solutions:

- Ensure model is properly loaded
- Check for browser console errors
- Restart Prediction App session
- Try with different model

Error Message	Meaning and Solution
`Invalid model file`	Model format incorrect; use Training App to recreate
`Missing required column: X`	CSV file missing column X; verify file format
`Training failed: timeout`	Training took too long; reduce complexity
`Default model not found`	Default model missing; try custom upload
`Invalid date format`	Use YYYY-MM-DD format

Table 8.1: Common Error Messages

8.2 Error Message Guide

8.2.1 Common Error Messages

8.3 Performance Issues

8.3.1 Slow Performance

Training App is slow:

- Reduce data size for testing
- Lower max parameter values
- Close other applications
- Check system resources (RAM/CPU)

Prediction App is slow:

- Check internet connection speed
- Clear browser cache
- Try different browser
- Avoid peak usage times

8.4 System Recovery

8.4.1 Emergency Fixes

Training App won't start:

```
# Reinstall dependencies
pip install -r requirements.txt --force-reinstall

# Use different port
streamlit run app.py --server.port=8502
```

Browser crashes:

1. Clear browser cache and cookies
2. Restart browser
3. Try incognito/private mode
4. Test in different browser

8.5 Contacting Support

8.5.1 Before Contacting Support

Gather this information:

- Error messages (exact text)
- Steps taken before error
- Browser type and version
- Operating system details
- Screenshots if applicable

8.5.2 Support Channels

- Email: support@walmart-forecasting.example.com
- Documentation: Check README files
- FAQ section: See Chapter 9
- Community forum: [URL if available]

9 Frequently Asked Questions

9.1 General Questions

Q: Do I need both apps to use the system?

A: Not necessarily. You can use pre-trained models in the Prediction App without using the Training App. However, for custom forecasts, you'll need to train your own models.

Q: Can I retrain models?

A: Yes. You can retrain models at any time using the Training App. We recommend quarterly retraining for optimal accuracy.

Q: Is internet required for both apps?

A: The Training App needs internet only for initial setup. The Prediction App requires continuous internet connection since it's cloud-based.

Q: How secure is my data?

A: Data uploaded to the Training App remains local. The Prediction App processes data temporarily in session only.

9.2 Technical Questions

Q: What's the maximum data size I can upload?

A: Training App: Limited by your local system resources. Prediction App: Models up to 50MB. CSV files up to 200MB.

Q: Why do my forecasts differ from expectation?

A: Several factors affect forecasts:

- Data quality and completeness
- Model selection
- Hyperparameter settings
- Unexpected market changes

Q: Can I use different date formats?

A: No. Use YYYY-MM-DD format only for all date fields.

Q: Where are models stored?

A: Training App: `models/default/` directory locally. Prediction App: Temporary session storage.

9.3 Usage Questions

Q: How often should I retrain models?
A: Recommend quarterly retraining or when accuracy drops below acceptable levels.
Q: Can I forecast for specific stores?
A: Currently, the system provides aggregate forecasts. Individual store forecasting may be added in future versions.
Q: What happens to my data after session ends?
A: Session data is automatically cleared. Download important forecasts before closing.
Q: Can I use the system offline?
A: Training App: Yes, after initial setup. Prediction App: No, requires internet connection.

9.4 Model Questions

Q: Which model should I choose?
A: Start with Exponential Smoothing for simplicity. Use Auto ARIMA for complex patterns.
Q: How do I improve model accuracy?
A: Consider:

- More historical data
- Data cleaning and preprocessing
- Parameter tuning
- Different model selection

Q: Can I combine multiple models?
A: Not directly in the current version. Consider averaging results manually.

9.5 Troubleshooting FAQs

Q: My model won't save. What to do?
A: Check write permissions for the models/default/ directory. Ensure sufficient disk space.
Q: Prediction App shows 'session expired'. Fix?
A: Reload the page and re-upload your model. Sessions timeout after 30 minutes of inactivity.
Q: Training is very slow. Normal?
A: Auto ARIMA can take significant time. Reduce max parameters or switch to Exponential Smoothing.

9.6 Advanced Usage

Q: Can I automate the process?
A: Advanced users can script interactions with the Training App. Prediction App requires manual interaction.

Q: Where do I find logs?
A: Training App: Console output during runtime. Prediction App: Browser developer console (F12).

Q: Can I extend the system?
A: The Training App is open for local modifications. Prediction App is hosted and requires different architecture.

9.7 Support Questions

Q: Where to report bugs?
A: Contact support@walmart-forecasting.example.com with detailed error information.

Q: Will there be updates?
A: System undergoes regular updates. Major changes announced in documentation.

Q: Training materials available?
A: This manual, plus optional video tutorials and documentation examples.

10 Glossary

10.1 Technical Terms

ARIMA Auto-Regressive Integrated Moving Average. A statistical model for time series forecasting that uses historical data to predict future values.

Exponential Smoothing A forecasting method that uses weighted averages of past observations, giving more weight to recent data.

Forecast Horizon The length of time into the future for which predictions are made. In this system, fixed at 4 weeks.

Hyperparameters Configuration settings that control the learning process of the models. Examples include max_p, max_q for ARIMA.

Joblib A set of tools for lightweight pipelining in Python, primarily used for efficient saving/loading of models.

Seasonality Patterns that repeat over specific time periods (e.g., weekly, monthly, yearly) in time series data.

Time Series A sequence of data points indexed in time order, typically at equally spaced intervals.

WMAE Weighted Mean Absolute Error. A metric used to measure the accuracy of forecasting models.

10.2 File Types

.csv Comma-Separated Values file. Plain text format for tabular data.

.pkl Pickle file. Python serialization format used for saving machine learning models.

10.3 Model Components

Differencing A transformation applied to time series data to make it stationary by removing trends.

Moving Average (MA) Component of ARIMA that models the dependency between an observation and a residual error from a moving average model.

Auto-Regressive (AR) Component of ARIMA that models the relationship between an observation and a number of lagged observations.

Trend Long-term direction in time series data (increasing, decreasing, or stable).

10.4 Application Features

Diagnostic Plot Visual representation comparing actual vs. predicted values to assess model performance.

Session State Temporary storage of data and models during an active browser session.

Training Loop The iterative process of training a model using historical data.

Validation Process of checking data quality and format before processing.

10.5 Data Processing Terms

Aggregation Combining data points over time periods (e.g., daily to weekly).

Data Merging Combining multiple datasets based on common fields.

Feature Engineering Creating new data features from existing ones to improve model performance.

Outlier Data point that differs significantly from other observations.

10.6 Performance Metrics

Accuracy How well the model's predictions match actual values.

Error Difference between predicted and actual values.

Training Error Model error calculated on training data.

Test Error Model error calculated on unseen test data.

10.7 System Components

Client The user's web browser accessing the Prediction App.

Local Environment The user's computer running the Training App.

Session A user's continuous interaction with the web application.

Server Remote computer hosting the Prediction App.

10.8 Business Terms

Store Type Classification of Walmart stores (Type A, B, or C).

Markdown Price reduction applied to products.

CPI Consumer Price Index. Economic indicator measuring price changes.

Unemployment Rate Percentage of the labor force that is unemployed.

10.9 Software Terms

Streamlit Python library used for creating web applications.

Pandas Python library for data manipulation and analysis.
Plotly Python library for creating interactive visualizations.
Environment The setup of software and libraries needed to run the applications.

11 References

11.1 External Resources

11.1.1 Documentation

- Streamlit Documentation: `https://docs.streamlit.io`
- pmdarima Documentation: `https://alkaline-ml.com/pmdarima/`
- Statsmodels Documentation: `https://www.statsmodels.org/stable/`
- Pandas Documentation: `https://pandas.pydata.org/docs/`

11.1.2 Theoretical Background

- Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice. `https://otexts.com/fpp3/`
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control.

11.2 System Resources

11.2.1 GitHub Repository

Source code and documentation: `https://github.com/your-repo/walmart-forecasting`

11.2.2 Application URLs

- Prediction App: `https://your-app-url.streamlit.app`
- Support Portal: `https://support.walmart-forecasting.example.com`

11.3 Dataset Information

11.3.1 Walmart Dataset

- Original Source: Kaggle Walmart Sales Dataset

- Data Period: 2010-2012
- Stores: 45 locations
- Features: Sales, markdowns, holidays, CPI, unemployment

11.3.2 Data Processing

- Weekly aggregation
- Missing value handling
- Outlier detection and treatment

11.4 Software Dependencies

11.4.1 Training App Dependencies

```
streamlit>=1.28.0
pandas>=2.0.0
numpy>=1.24.0
matplotlib>=3.7.0
seaborn>=0.12.0
statsmodels>=0.14.0
pmdarima>=2.0.4
joblib>=1.3.0
scikit-learn>=1.3.0
```

11.4.2 Prediction App Dependencies

```
streamlit>=1.28.0
pandas>=2.0.0
numpy>=1.24.0
plotly>=5.17.0
statsmodels>=0.14.0
pmdarima>=2.0.4
joblib>=1.3.0
scikit-learn>=1.3.0
```

11.5 Version History

11.5.1 System Versions

- v1.0.0: Initial release

- v1.1.0: Added Exponential Smoothing support
- v1.2.0: Improved error handling and validation
- v1.3.0: Enhanced visualization features

11.6 Contact Information

11.6.1 Support Contacts

- Email: support@walmart-forecasting.example.com
- Phone: +1-800-EXAMPLE
- Business Hours: Monday-Friday, 9 AM - 5 PM EST

11.6.2 Development Team

- Project Lead: [Name]
- Technical Lead: [Name]
- Data Science Team: [Names]

11.7 License and Legal

11.7.1 Software License

11.7.2 Terms of Use

- Personal and educational use permitted
- Commercial use requires license
- Data privacy policy available at website
- No warranty provided as-is

11.8 Acknowledgments

11.8.1 Open Source Software

Special thanks to the developers and communities of:

- Python Software Foundation
- Streamlit team
- Data science libraries used
- Open source contributors

11.8.2 Beta Testers

Thanks to our beta testing team for valuable feedback and suggestions.

11.9 Future Roadmap

11.9.1 Planned Features

- Individual store forecasting
- Longer forecast horizons
- Additional forecasting models
- Automated model selection
- API integration

11.9.2 Research and Development

- Machine learning model ensembles
- Real-time data integration
- Advanced visualization options
- Custom holiday modeling