



**Online Retail Store**  
**Time Series Analysis**  
**Project Report**



ADNAN RAHMAN

## **1. Objective & Problem Statement**

The primary objective of this project is to develop reliable time series forecasting models to predict future sales for a retail business using historical transaction data. The problem statement revolves around predicting monthly sales for an online retail store, which is crucial for inventory management, resource allocation, and strategic planning. Retail sales often exhibit patterns such as trends, seasonality, and cyclicity due to factors like holidays, economic conditions, and consumer behavior. Inaccurate forecasts can lead to overstocking, stockouts, or lost revenue opportunities.

The specific goals of this analysis are:

- To preprocess and aggregate transactional retail data.
- To build multiple forecasting models ranging from baseline naive methods to advanced techniques like exponential smoothing methods, ARIMA and machine learning models.
- To compare forecast accuracy using standard metrics: RMSE and  $R^2$ .
- To interpret forecast outputs and provide practical insights for the retail store.

This report walks through all major analytical steps performed in the retail\_store\_TSA.ipynb notebook, including preprocessing, modeling, result interpretation and accuracy evaluation.

## **2. Description of the Data & Pre-processing**

The dataset used consists of two sheets:

- Year 2009-2010
- Year 2010-2011

These were loaded separately and then combined using: `Final_df = pd.concat([df2, df1], axis=0)`

The dataset contains:

- InvoiceDate - Timestamp of the transaction
- Quantity - units sold
- Price - unit price

### **Preprocessing Steps:**

#### **a. Cleaning Date and Filtering Values**

Sales Revenue is computed by `Final_df["Sales"] = Final_df["Quantity"] * Final_df["Price"]`

#### **b. Aggregation to Monthly Sales**

Since forecasting daily sales would be noisy and influenced by short-term spikes, monthly aggregation was chosen and as December 2011 data has only 10 days data, so it is omitted from the monthly\_sales dataframe. This produced a stable and meaningful time series suitable for exponential smoothing and trend-based models.

### **3. Model(s) Used and Justification**

A diverse set of models was employed to capture different aspects of the time series: trend, seasonality, and irregularities. Baseline models provided simple benchmarks, while advanced ones handled complexity.

#### **3.1. Baseline Models:**

**Naive Forecast:** Assumes future values equal the last observed value. Justified as a simple sanity check for more complex models.

**Average Method:** Uses the mean of historical data. Suitable for stable series but expected to underperform due to trends.

**Moving Average:** Smooths data over a window 5 months. Helps to identify trends but ignore seasonality.

#### **3.2. Exponential Smoothing Models:**

**Simple Exponential Smoothing (SES):** Weights recent observations more heavily. Ideal for non-trending, non-seasonal data.

**Holt's Linear Trend Method:** Extends SES with a trend component. Chosen to model the observed upward sales growth.

**Holt-Winters:** Incorporates trend and seasonality (additive/multiplicative). Justified by clear seasonal patterns (e.g., holiday peaks), making it suitable for retail data.

#### **3.3. ARIMA-based Models:**

**ARIMA (AutoRegressive Integrated Moving Average):** Handles stationarity via differencing, with autoregressive and moving average terms. Parameters (p,d,q) were selected via ACF/PACF plots.

**SARIMA:** Extends ARIMA with seasonal components (P,D,Q,s). Essential for capturing monthly seasonality (s=12).

**AutoARIMA:** Automates parameter selection using Pyramid's auto\_arima. Justified for efficiency in hyperparameter tuning.

#### **3.5. Regression Models**

**Linear Regression:** Fits a linear trend using time as a predictor. Simple for capturing overall growth.

**Non-Linear Regression:** Includes quadratic terms and monthly dummies. Justified to model non-linear trends and seasonality explicitly.

**Exponential Trend Model:** Applies log transformation to sales for multiplicative effects. Useful for exponentially growing series.

#### **4. Plots/Outputs of the Forecast**

Several visualizations were generated to illustrate model fits and forecasts:

**Original Sales Time Series Plot:** A line plot of monthly sales showed an increasing trend from ~£400,000 in late 2009 to peak over £1,000,000 in November 2011, with dips in early months and spikes in Q4, confirming seasonality.

##### **Fitted vs. Original Plots:**

**Holt-Winters Forecast:** Extended the series 6-12 months ahead, showing predicted upward trends with seasonal oscillations.

**For Non-Linear and Exponential Models:** Similar plots (e.g., original sales vs. fitted values) demonstrated better capture of peaks, with the exponential model using log-scale for stability.

**Forecast Comparison Plot:** Overlaid forecasts from all models against test data, revealing Holt-Winters closely tracking actuals, while baselines like Naive deviated significantly during peaks.

#### **5. Interpretation of Results and Insights**

The analysis highlighted several important patterns. Sales followed a strong upward linear trend, increasing by roughly 13,555 units per month based on the regression results. Clear seasonal effects were also present, with November consistently showing peak sales, nearly double the annual average likely driven by holiday demand. Aside from these stable patterns, only minor irregularities appeared, such as the notable dip in 2010, which may reflect broader economic conditions.

##### **Insights:**

**Trend:** Consistent growth suggests expanding market or product range; non-linear models captured acceleration in later months.

**Seasonality:** Q4 dominance implies holiday-focused strategies; exponential models highlighted multiplicative effects.

**Model Limitations:** ARIMA struggled with seasonality (high RMSE), while LSTM showed promise but required more data for optimization. Baselines confirmed the need for advanced methods.

##### **Business Implications:**

**Seasonality plays a major role:** Peaks were consistently seen during the holiday months (Nov–Dec). Holt-Winters successfully reproduced these peaks in its forecast.

**Sales trend is positive:** The dataset shows a consistent upward trend over the two years. Holt and Holt-Winters both captured this well.

**Holt-Winters is the most realistic forecasting tool for this dataset:** Based on accurate metrics and visualization, this model should be preferred for operational use.

Overall, results underscore the importance of seasonality in retail TSA, with hybrid models like Holt-Winters balancing simplicity and accuracy.

## **6. Discussion of Accuracy Metrics (RMSE, $R^2$ )**

Model performance was evaluated on the test set using RMSE and  $R^2$  :

- **Holt-Winters:** Lowest RMSE (54,933) and highest  $R^2$  (0.957), indicating excellent fit (95.7% variance explained). It outperformed others by handling seasonality effectively.
- **Non-Linear Regression:** Strong contender (RMSE 57,444,  $R^2$  0.953), capturing monthly dummies well.
- **Exponential Trend:** Moderate (RMSE 157,907,  $R^2$  0.644), better for trends but missed fine seasonality.
- **Baselines (e.g., Naive):** RMSE (225,582,  $R^2$  0.304), highlighting poor performance on trending/seasonal data.
- **ARIMA Variants:** Higher RMSE (252,839-322,407), low  $R^2$  (0.055 to -0.484), due to inadequate seasonality modeling; SARIMA's negative  $R^2$  suggests worse than mean prediction.
- **LSTM:** RMSE 249,402,  $R^2$  0.228, underperformed possibly due to limited training data.

Holt-Winters' superiority aligns with retail's predictable cycles. Future improvements could involve hyperparameter tuning or ensemble methods to further reduce RMSE.

## **7. References**

- Tools/Libraries: Python 3.12, pandas (data manipulation), NumPy (numerics), Matplotlib (plots), statsmodels (TSA models), pmdarima (AutoARIMA), scikit-learn (regressions), TensorFlow/Keras (LSTM).
- External Materials: UCI Machine Learning Repository (similar retail datasets).