Rossmann Sales Forecasting

Multiple Model Comparison Report

A Comprehensive Analysis Using ARIMA, SARIMA, SARIMAX, and VAR/VECM Models

Prepared by:

Ankit Birla

Capstone Project — Time Series Forecasting

Table of Contents

1. Problem Statement

Rossmann is a European drug distributor operating over 3,000 drug stores across seven European countries. A critical business challenge arises from the nature of pharmaceutical products — many drugs come with short shelf lives and do not have extended expiry dates. This makes accurate sales forecasting imperative for effective inventory management and reduced waste.

Currently, sales forecasting is managed by individual store managers who are tasked with predicting daily sales for the next six weeks. With thousands of managers forecasting based on their unique circumstances, local knowledge, and personal intuitions, the accuracy of forecasts varies significantly across the organization. This inconsistency leads to:

- Potential stockouts of high-demand medications
- Excess inventory leading to expired products
- Suboptimal resource allocation across stores
- Inconsistent customer service levels

To address this challenge, Rossmann has requested the development of a standardized, data-driven forecasting system that can provide accurate 6-week sales predictions across their store network.

2. Project Goals

The primary objective of this project is to build and compare multiple time series forecasting models to identify the most effective approach for predicting daily sales at Rossmann stores. The specific goals include:

2.1 Primary Objectives

1. **Model Development:** Build five distinct forecasting models — Naive Last-Value, ARIMA, SARIMA, SARIMAX, and VAR/VECM
2. **Model Comparison:** Systematically compare model performance using Mean Absolute Percentage Error (MAPE)
3. **Store-Level Analysis:** Apply models to nine key stores (IDs: 1, 3, 8, 9, 13, 25, 29, 31, 46)
4. **Future Forecasting:** Generate reliable 6-week ahead forecasts using the best-performing models

2.2 Success Criteria

- Achieve MAPE scores below 20% for majority of stores
- Identify the best model for each store based on test performance
- Provide actionable insights on factors affecting sales
- Create a reproducible forecasting pipeline

3. Data Description

The project utilizes two primary datasets from the Rossmann store network:

3.1 Training Dataset (train.csv)

Contains historical daily sales records for each store. Key variables include:

Variable	Description
Store	Unique identifier for each store
Date	Date of sales record (Jan 2013 – Jul 2015)
Sales	Daily turnover (target variable)
Customers	Number of customers on that day
Open	Binary indicator if store was open (0/1)
Promo	Binary indicator if promotion was running
StateHoliday	State holiday indicator (a, b, c, or 0)
SchoolHoliday	Binary indicator for school holiday

3.2 Store Metadata (store.csv)

Contains static store-level information including StoreType, Assortment level, CompetitionDistance, Promo2 participation status, and competition timeline details.

3.3 Data Scope

- **Time Period:** January 2013 to July 2015 (942 unique dates)
- **Stores Analyzed:** 9 key stores (IDs: 1, 3, 8, 9, 13, 25, 29, 31, 46)
- **Total Records:** 8,110 records after filtering

4. System Architecture

The forecasting system is designed as a modular pipeline with the following components:

4.1 Technology Stack

Component	Technology
Programming Language	Python 3.x
Data Processing	Pandas, NumPy
Time Series Modeling	Statsmodels (ARIMA, SARIMAX, VAR, VECM)
Preprocessing	Scikit-learn (StandardScaler)
Visualization	Matplotlib, Seaborn
Environment	Google Colab (High Memory)

4.2 Pipeline Architecture

The system follows a sequential data flow:

1. **Data Ingestion:** Load and merge train.csv with store.csv
2. **Data Filtering:** Filter to key stores and open days only
3. **Feature Engineering:** Extract temporal features (Year, Month, Day, WeekOfYear)
4. **Outlier Treatment:** Cap extreme values at 99th percentile
5. **Train/Test Split:** Last 42 days reserved for testing
6. **Model Training:** Fit five model types per store
7. **Evaluation:** Calculate MAPE and select best model
8. **Forecasting:** Generate 6-week predictions

5. Data Processing

5.1 Data Loading and Merging

The raw data was loaded from two CSV files and merged on the Store identifier. The Date column was parsed to datetime format for proper time series handling.

5.2 Missing Value Treatment

Initial null value analysis revealed significant missing data in promotional columns:

Column	Null Count	Null %
Promo2SinceWeek	5,652	69.69%
PromoInterval	5,652	69.69%
Promo2SinceYear	5,652	69.69%
CompetitionOpenSinceYear	1,700	20.96%

Missing values were handled through appropriate imputation strategies — zeros for promotional columns (indicating no promotion) and forward/backward filling for temporal gaps.

5.3 Filtering Criteria

- **Store Selection:** Filtered to 9 key stores of interest
- **Open Days Only:** Excluded closed days (Sales = 0) for meaningful analysis
- **Result:** 6,681 records for model training after filtering

6. Exploratory Data Analysis (EDA)

6.1 Column Classification

Variables were classified into categorical and numerical types for appropriate analysis:

Categorical Variables: DayOfWeek, Open, Promo, StateHoliday, SchoolHoliday, StoreType, Assortment, CompetitionOpenSinceMonth/Year, Promo2 indicators, Year, Month, Day, WeekOfYear, Weekend

Numerical Variables: Store, Sales, Customers, CompetitionDistance, DayOfYear

6.2 Univariate Analysis

Distribution analysis of numerical columns revealed the following key insights:

Sales Distribution: Right-skewed distribution with mean around €5,500–€7,000 per day. Some stores exhibit higher variability, indicating different store sizes or customer bases.

Customer Counts: Strong positive correlation with sales ($r > 0.9$). Average daily customers range from 400–800 depending on store.

Competition Distance: Varies significantly (310m to 14,130m), with most stores having competition within 5km.

6.3 Key Observations

- **Weekly Seasonality:** Clear patterns with higher sales on Mondays (often promo days) and Saturdays
- **Promotion Impact:** Promotional days show 20–40% higher average sales
- **Holiday Effects:** State and school holidays affect sales patterns differently across stores
- **Store Variability:** Significant heterogeneity in sales patterns across the 9 stores

7. Feature Engineering

7.1 Temporal Feature Extraction

From the Date column, the following temporal features were engineered to capture seasonality and trends:

Feature	Purpose
Year	Capture annual trends and year-over-year growth
Month	Capture monthly seasonality (holiday seasons, summer, etc.)
Day	Day of month (1–31)
WeekOfYear	Week number (1–52) for weekly patterns
DayOfYear	Day of year (1–365) for annual seasonality
Weekend	Binary flag for Saturday/Sunday

7.2 Exogenous Variables for SARIMAX

For the SARIMAX model, two exogenous regressors were prepared:

- **Promo:** Binary indicator for promotional activity
- **SchoolHoliday:** Binary indicator for school holiday periods

7.3 Multivariate Features for VAR/VECM

For multivariate modeling, a combined dataframe was created with Sales, Customers, and Promo variables, allowing the models to capture interdependencies between these related time series.

8. Model Selection

Five forecasting approaches were implemented to compare their effectiveness for this sales prediction task:

8.1 Model 1: Naive Last-Value Forecast

Approach: Uses the last observed value as the forecast for all future periods. Serves as the baseline for comparison.

Rationale: Simple benchmark that any sophisticated model should outperform.

8.2 Model 2: ARIMA (AutoRegressive Integrated Moving Average)

Approach: Univariate time series model combining autoregression (AR), differencing (I), and moving average (MA) components.

Parameters: Grid search over $p \in [0,3]$, $d \in [0,3]$, $q \in [0,3]$ with AIC-based selection.

8.3 Model 3: SARIMA (Seasonal ARIMA)

Approach: Extends ARIMA with seasonal components to capture weekly patterns (period=7 days).

Seasonal Order: (1, 1, 1, 7) — capturing weekly seasonality in retail sales.

8.4 Model 4: SARIMAX (SARIMA with Exogenous Variables)

Approach: Incorporates external factors (Promo, SchoolHoliday) that influence sales beyond historical patterns.

Advantage: Can account for known future promotions when forecasting.

8.5 Model 5: VAR/VECM (Vector Autoregression / Vector Error Correction)

Approach: Multivariate time series model analyzing Sales, Customers, and Promo jointly.

Model Selection: Johansen cointegration test determines whether to use VAR or VECM. If cointegration rank > 0, VECM is preferred.

9. Model Training

9.1 Train-Test Split

Data was split temporally with the last 42 days (6 weeks) held out as the test set, mirroring the actual forecasting requirement.

9.2 Hyperparameter Optimization

ARIMA/SARIMA/SARIMAX models used grid search with AIC criterion:

- p (AR order): 0 to 3
- d (Differencing): 0 to 3
- q (MA order): 0 to 3
- Total combinations evaluated: 63 per model type

9.3 Stationarity Testing

Augmented Dickey-Fuller (ADF) tests were conducted to assess stationarity. For non-stationary series, appropriate differencing was applied.

9.4 Cointegration Analysis (VAR/VECM)

The Johansen cointegration test was applied to determine relationships between Sales, Customers, and Promo. Results showed cointegration rank of 3 for all stores, indicating strong long-run equilibrium relationships.

Implication: VECM (Vector Error Correction Model) was used instead of VAR for all stores, as it accounts for cointegration.

10. Model Evaluation

10.1 Evaluation Metric

Mean Absolute Percentage Error (MAPE) was chosen as the primary metric:

$$MAPE = (1/n) \times \Sigma \, |Actual - Forecast| \, / \, |Actual| \times 100\%$$

Why MAPE? It provides an intuitive, scale-independent percentage error that is easily interpretable by business stakeholders. A MAPE of 15% means forecasts are off by 15% on average.

10.2 Model Comparison Results

Store	Naive %	ARIMA %	SARIMA %	SARIMAX %	VAR/VECM %
1	16.30	15.39	14.51	**11.14**	15.28
3	23.07	19.98	24.27	**13.76**	23.42
8	26.60	27.35	28.23	**15.98**	24.55
9	15.93	17.28	18.69	**14.61**	15.18
13	28.46	21.89	22.63	**16.29**	33.46
25	16.74	18.25	17.73	**14.32**	15.95
29	18.94	21.09	22.74	16.11	**15.44**
31	15.24	*	15.91	**8.73**	15.66
46	19.64	**13.68**	20.21	16.91	22.43

*Note: * ARIMA for Store 31 produced unstable results. Green cells indicate best model per store.*

11. Results & Interpretation

11.1 Best Model Summary

Store	Best Model	MAPE	Best Order
1	SARIMAX	11.14%	(2, 1, 3)
3	SARIMAX	13.76%	(1, 0, 3)
8	SARIMAX	15.98%	(3, 0, 2)
9	SARIMAX	14.61%	(0, 1, 3)
13	SARIMAX	16.29%	(1, 1, 3)
25	SARIMAX	14.32%	(1, 0, 3)
29	VAR/VECM	15.44%	Multivariate
31	SARIMAX	8.73%	(1, 0, 3)
46	ARIMA	13.68%	(3, 0, 3)

11.2 Key Findings

1. **SARIMAX Dominance:** SARIMAX achieved the best performance for 7 out of 9 stores, demonstrating the importance of incorporating promotional and holiday information.
2. **Promotional Impact:** The consistent outperformance of SARIMAX validates that promotions significantly affect sales and should be explicitly modeled.
3. **Store Variability:** Different stores require different optimal parameters, suggesting store-specific models rather than one-size-fits-all.
4. **MAPE Range:** Best model MAPE ranges from 8.73% (Store 31) to 16.29% (Store 13), indicating varying predictability across stores.
5. **Multivariate Value:** VAR/VECM performed best for Store 29, showing that joint modeling of Sales-Customers-Promo can capture interdependencies missed by univariate models.

11.3 Business Implications

- **Inventory Planning:** With MAPE under 15% for most stores, inventory can be sized within ±15% accuracy, reducing both stockouts and overstock.
- **Promotion Strategy:** Since SARIMAX captures promotion effects, future promotion schedules can be tested before implementation.
- **Staffing Optimization:** 6-week forecasts enable better staff scheduling aligned with expected customer traffic.

12. Appendix: Notebook Extracts

12.1 Key Libraries Used

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.api import VAR
from statsmodels.tsa.vector_ar.vecm import VECM, coint_johansen
```

12.2 MAPE Calculation Function

```
def mean_absolute_percentage_error(y_true, y_pred):
    """Compute MAPE (%) ignoring points where y_true == 0."""
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) / y_true[mask])) * 100
```

12.3 Data Files

- **train.csv:** Historical sales data (~1M records for all stores)
- **store.csv:** Store metadata (1,115 stores)

12.4 Execution Environment

- **Platform:** Google Colab (High Memory runtime)
- **Python Version:** 3.x
- **Execution Time:** ~45 minutes for full model comparison across 9 stores

— End of Report —