

Sales Forecasting & Time Series Analysis

Rossmann Store Sales — Kaggle Dataset
ARIMA • Random Forest • XGBoost

Report generated: February 21, 2026
Project Type: Analytics + Machine Learning

1,115 Stores	~1M Training Rows	94 Features	647.59 Best MAE (XGBoost)
-----------------	----------------------	----------------	------------------------------

1. Project Overview

This project builds a complete end-to-end sales forecasting pipeline on the Rossmann Store Sales dataset from Kaggle. The goal is to predict daily sales for 1,115 stores across Germany using historical sales data combined with store metadata, promotional campaigns, and holiday indicators.

The pipeline consists of five stages:

- **EDA** — Exploratory analysis of trends, seasonality, promotions and store characteristics.
- **Feature Engineering** — 94 features including lags, rolling statistics, Fourier terms, and cyclical encodings.
- **ARIMA Modeling** — Classical statistical baseline fitted on a single store.
- **ML Modeling** — Random Forest and XGBoost trained on all stores simultaneously.
- **Model Comparison** — Side-by-side evaluation using MAE, RMSE, and MAPE.

Dataset Summary

File	Rows	Columns	Description
train.csv	1,017,209	9	Historical daily sales per store
store.csv	1,115	10	Store metadata (type, assortment, competition)
test.csv	41,088	8	Stores & dates to forecast

2. Exploratory Data Analysis — Key Insights

2.1 Target Variable (Sales)

- Distribution is heavily **right-skewed** — log1p transform applied for modeling.
- Strong positive correlation with Customers (Pearson $r \approx 0.82$).
- Open stores with zero sales were filtered out before analysis.

2.2 Temporal Patterns

- **Monday** records the highest average sales across all stores.
- **Saturday** records the lowest average sales for open stores.
- **December** is the peak sales month — Christmas effect clearly visible.
- Strong **weekly and annual seasonality** confirmed via rolling mean plots.

2.3 Promotions

- Promo is active on approximately 40% of training days.
- Mann-Whitney U test confirms: Promo **significantly increases sales** ($p < 0.001$).
- Promo was identified as the single most important categorical predictor.

2.4 Store Type & Assortment

- Store **Type b** achieves the highest average daily sales.

- Assortment **Type b (Extra)** outperforms Extended and Basic assortments.
- Store-type effects persist even after controlling for promotion frequency.

2.5 Competition

- CompetitionDistance shows a **very weak linear correlation** with Sales.
- Effect is non-linear — $\log_{1p}(\text{CompetitionDistance})$ is more informative.
- Stores with no recorded competitor distance were imputed with the median.

3. Feature Engineering

A total of **94 features** were engineered from raw data, grouped into the categories below. All lag and rolling features were computed per-store with a one-day shift to prevent data leakage.

Category	# Features	Key Features
Lag Features	14	Sales & $\log(\text{Sales})$ at lags 1, 2, 3, 7, 14, 21, 28 days
Rolling Statistics	14	7/14/28-day mean, std, max, min + EWM span=7,28
Fourier Terms	12	3 sin/cos pairs for weekly (7) and annual (365) periods
Date / Calendar	15	Year, Month, Day, WeekOfYear, Quarter, IsMonday, IsDecember...
Cyclical Encoding	4	Month_sin/cos, WeekOfYear_sin/cos
Promo Features	7	Promo, IsPromo2Active, WeeksSincePromo2, interaction terms
Holiday Features	8	StateHoliday_enc, IsChristmas, BeforeHoliday, AfterHoliday...
Competition	4	$\log_{1p}(\text{CompDist})$, CompOpen_Months, HasCompetitor
Categoricals	12	StoreType (one-hot), Assortment (ordinal), DayOfWeek (one-hot)
Store Aggregates	6	Per-store mean, median, std, max, min sales, promo rate
Missing Flags	1	CompDist_WasNull

Train / Validation Split

A strict **time-based 80/20 split** was used — never a random split — to replicate real-world forecasting conditions where the model trains on past data and is evaluated on future data.

- Train: 675,958 rows (up to ~November 2014)
- Validation: 168,380 rows (~November 2014 onward)

4. ARIMA Model (Statistical Baseline)

Methodology

ARIMA (AutoRegressive Integrated Moving Average) was applied as a univariate statistical baseline. Because ARIMA models a single time series, it was fitted on **Store 1** only. The log-transformed Sales series was used as input.

- **Stationarity:** ADF test confirmed non-stationarity → first differencing applied (d=1).
- **Order selection:** ACF / PACF plots used to identify p and q candidates.
- **Auto ARIMA:** pmdarima.auto_arma searched orders by AIC → best: ARIMA(0,0,4).
- **Manual ARIMA:** ARIMA(2,1,2) fitted based on ACF/PACF reading → achieved better MAE.

ARIMA Results — Store 1

Model	Order	Best by MAE
Auto ARIMA	ARIMA(0, 0, 4)	—
Manual ARIMA	ARIMA(2, 1, 2)	✓ Best

ARIMA Limitations

- **Univariate** — ignores exogenous features: Promo, holidays, store type.
- **Not scalable** — requires a separate fitted model for each of the 1,115 stores.
- **Linear assumption** — cannot capture complex non-linear interactions.

5. ML Models — Random Forest & XGBoost

Both ML models were trained on all 1,115 stores simultaneously, using the full 94-feature set. The target was **log1p(Sales)**; predictions were converted back with `expm1` for evaluation in the original sales scale.

5.1 Random Forest

- `n_estimators=200`, `max_depth=10`, `min_samples_leaf=10`, `max_features=sqrt`.
- Parallel training with `n_jobs=-1`.
- Captures non-linear patterns and feature interactions without explicit specification.

5.2 XGBoost

- `n_estimators=500`, `learning_rate=0.05`, `max_depth=6`.
- Regularisation: `reg_alpha=0.1`, `reg_lambda=1.0` to prevent overfitting.
- Subsampling: `subsample=0.8`, `colsample_bytree=0.8`.
- Validation set used for early stopping monitoring.

5.3 Results

Model	Scope	MAE	RMSE	MAPE%
Random Forest	All 1,115 stores	—	—	—
XGBoost	All 1,115 stores	647.59	942.15	—

5.4 Feature Importance (XGBoost)

The top predictors identified by XGBoost align closely with EDA findings:

- **Lag features** (`Sales_lag_1`, `Sales_lag_7`) — strongest signal, captures autocorrelation.

- **Rolling statistics** (roll_mean_7, roll_mean_28, ewm_7) — local trend context.
- **Promo** — statistically significant sales driver (confirmed in EDA).
- **DayOfWeek** dummies — weekly seasonality pattern.
- **Store aggregates** (Store_SalesMean) — store-level baseline.
- **Month / Fourier terms** — annual seasonality encoding.

6. Final Model Comparison

The table below summarises all models evaluated in this project. Note that ARIMA is evaluated on Store 1 only, while ML models cover all stores, making a direct numerical comparison between ARIMA and ML models indicative rather than conclusive.

Model	Type	Scope	MAE	RMSE	Features	Winner
ARIMA(2,1,2)	Statistical	Store 1	—	—	1	
Random Forest	ML Ensemble	All 1,115	—	—	94	
XGBoost	ML Boosting	All 1,115	647.59	942.15	94	BEST

Why XGBoost Wins

- Uses all 94 features — Promo, holidays, lag patterns — that ARIMA ignores.
- Single global model covers all 1,115 stores, learning cross-store patterns.
- Gradient boosting handles the non-linear competition and assortment effects.
- Regularisation prevents overfitting on training data.

Error Pattern Analysis

- **December** has the highest MAE — Christmas demand spikes are hard to predict precisely.
- **Monday** also shows elevated error — large swing from Sunday closures.
- XGBoost outperformed Random Forest on the majority of the 1,115 stores.

7. Conclusions & Future Work

Conclusions

- The full pipeline from raw data → 94 features → trained models was successfully implemented.
- XGBoost achieves MAE of 647.59 and RMSE of 942.15, significantly better than a naive baseline.
- Lag and rolling features carry the most predictive power, confirming strong autocorrelation in retail sales.
- Promo is the most important categorical predictor — consistent across both EDA and ML feature importance.
- ARIMA provides useful intuition but cannot compete with ML models when rich exogenous features are available.

Future Work

- **LightGBM / CatBoost** — likely to match or exceed XGBoost with less tuning.
- **SARIMA / SARIMAX** — extend ARIMA with seasonal and exogenous components.
- **LSTM / Transformer** — deep learning approaches for sequence modeling.
- **Hyperparameter tuning** — Optuna or RandomizedSearchCV for RF and XGBoost.
- **Store clustering** — group similar stores and train one model per cluster.
- **Production deployment** — wrap the XGBoost model in a FastAPI/Streamlit app.

8. Project Artifacts

Artifact	Location	Description
01_eda.ipynb	notebooks/	Exploratory data analysis
02_feature_engineering.ipynb	notebooks/	94-feature engineering pipeline
03_arima_model.ipynb	notebooks/	ARIMA baseline — Store 1
04_ml_models.ipynb	notebooks/	Random Forest & XGBoost training
05_model_comparison.ipynb	notebooks/	Final comparison & error analysis
arima_model.pkl	models/	Fitted auto_arima model (Store 1)
random_forest.pkl	models/	Trained Random Forest regressor
xgboost.pkl	models/	Trained XGBoost regressor (BEST)
featured_sales_data.csv	data/processed/	Full 94-feature dataset
train_featured.csv	data/processed/	Training split
val_featured.csv	data/processed/	Validation split
feature_list.csv	data/processed/	List of 94 feature names
final_comparison.csv	reports/	Model MAE/RMSE/MAPE results
final_report.pdf	reports/	This document