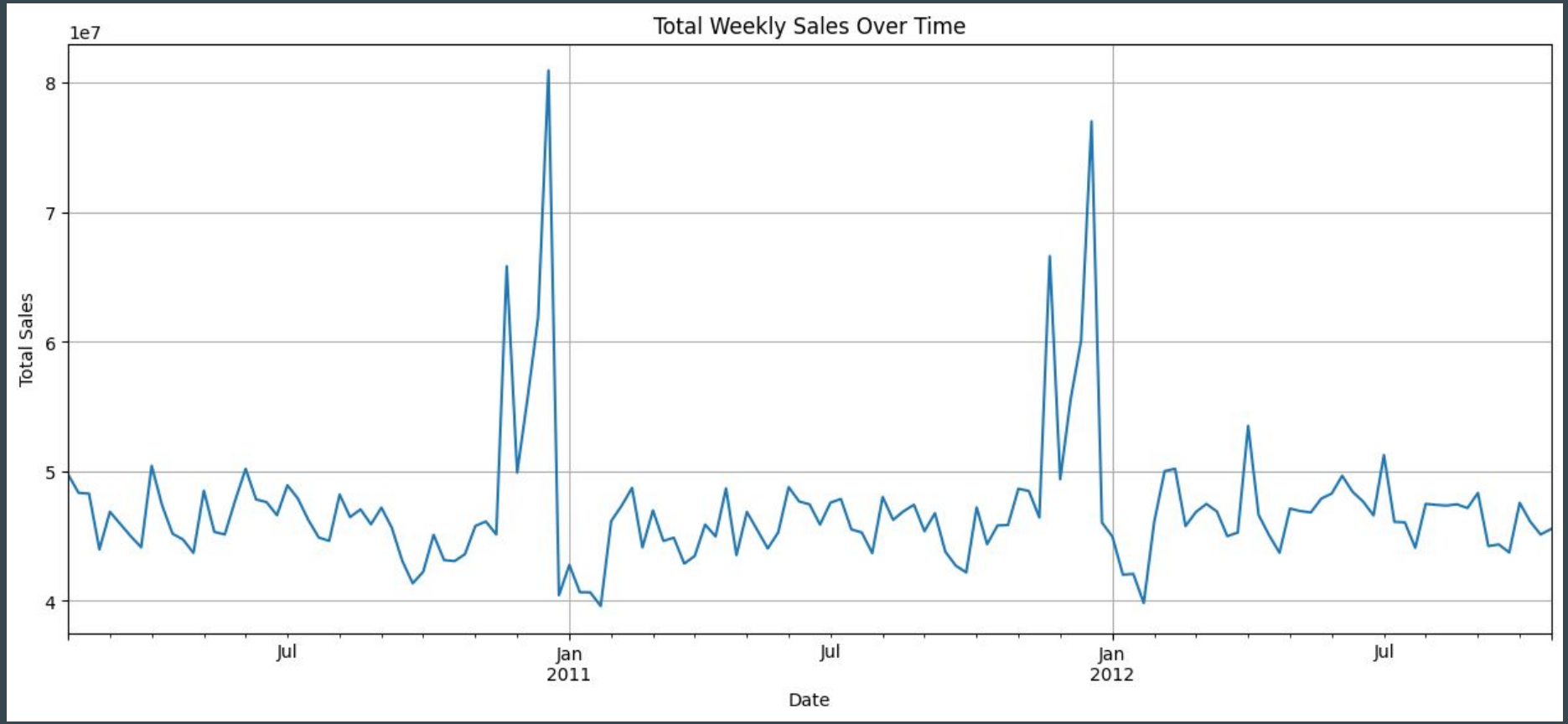
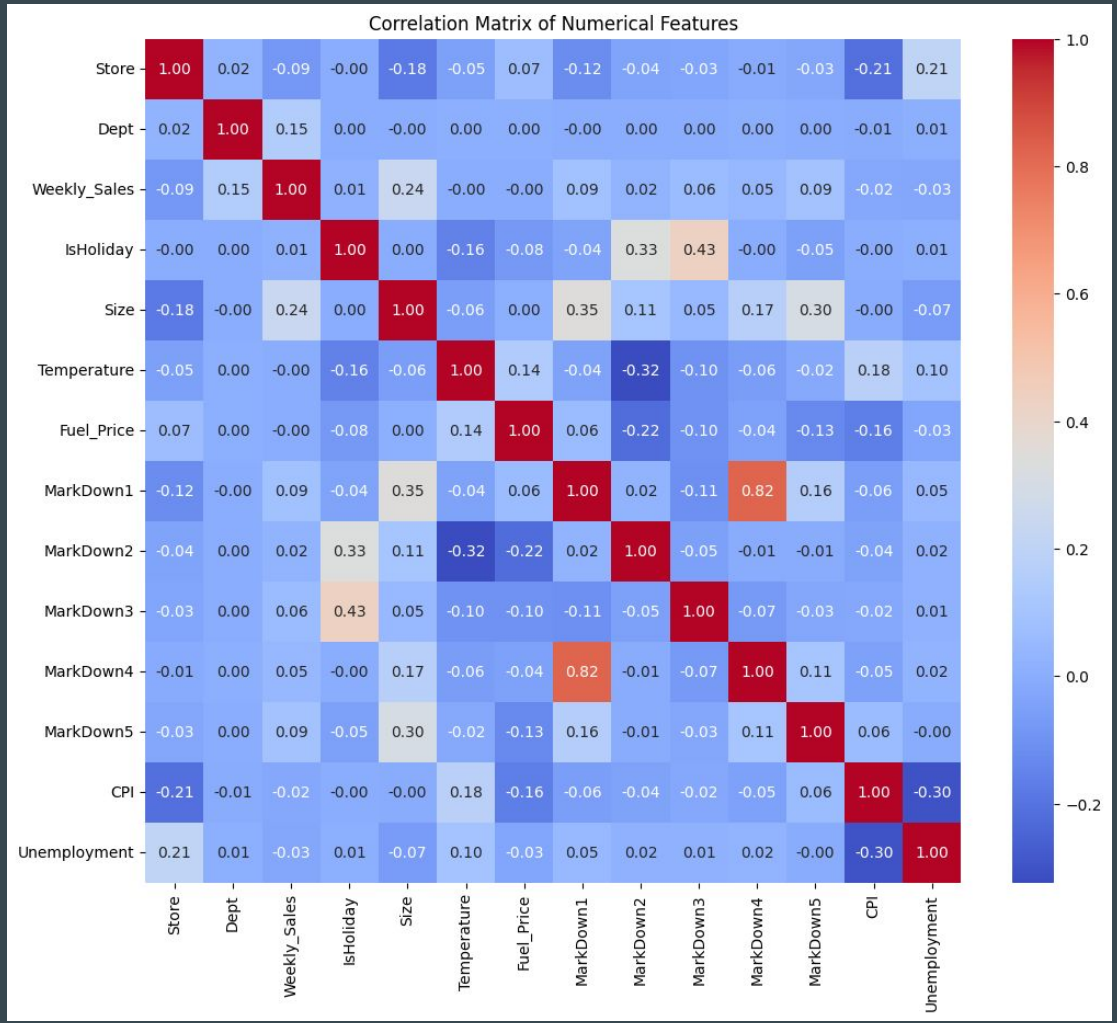
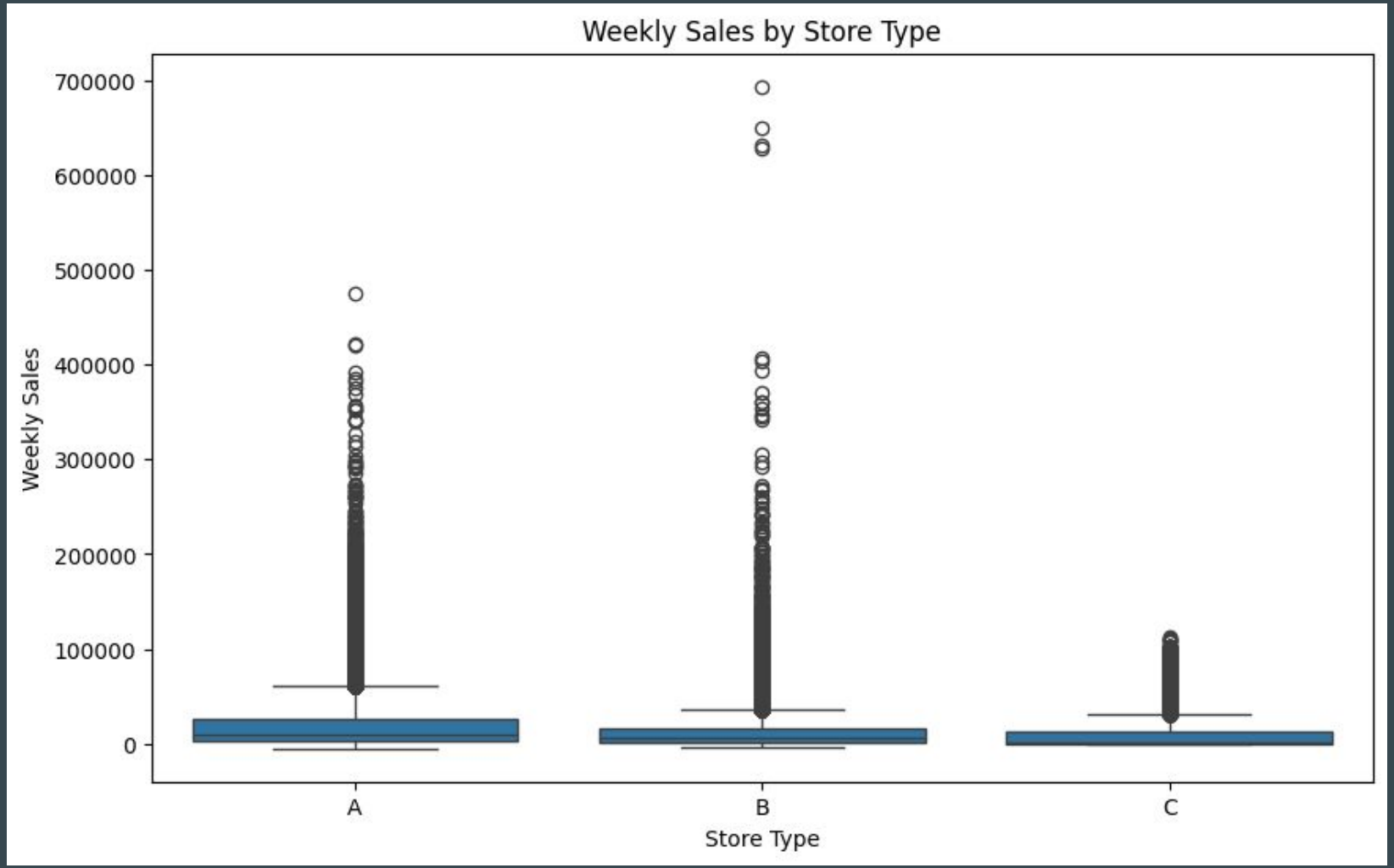
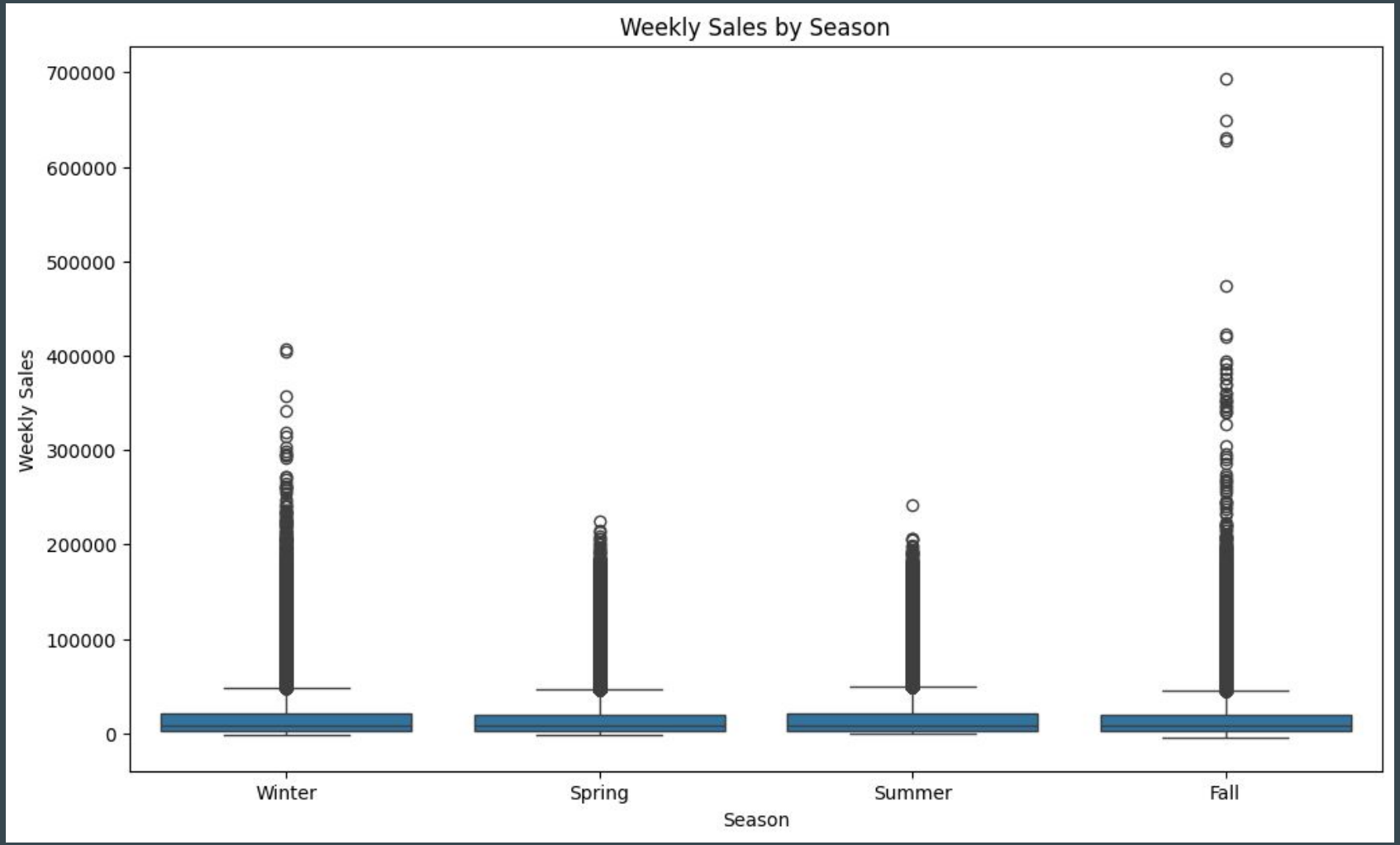

Total Weekly Sales Over Time
1e7
8
7
6
5
4
Total Sales
Jul
Jan
2011
Jul
Jan
2012
Jul
Date

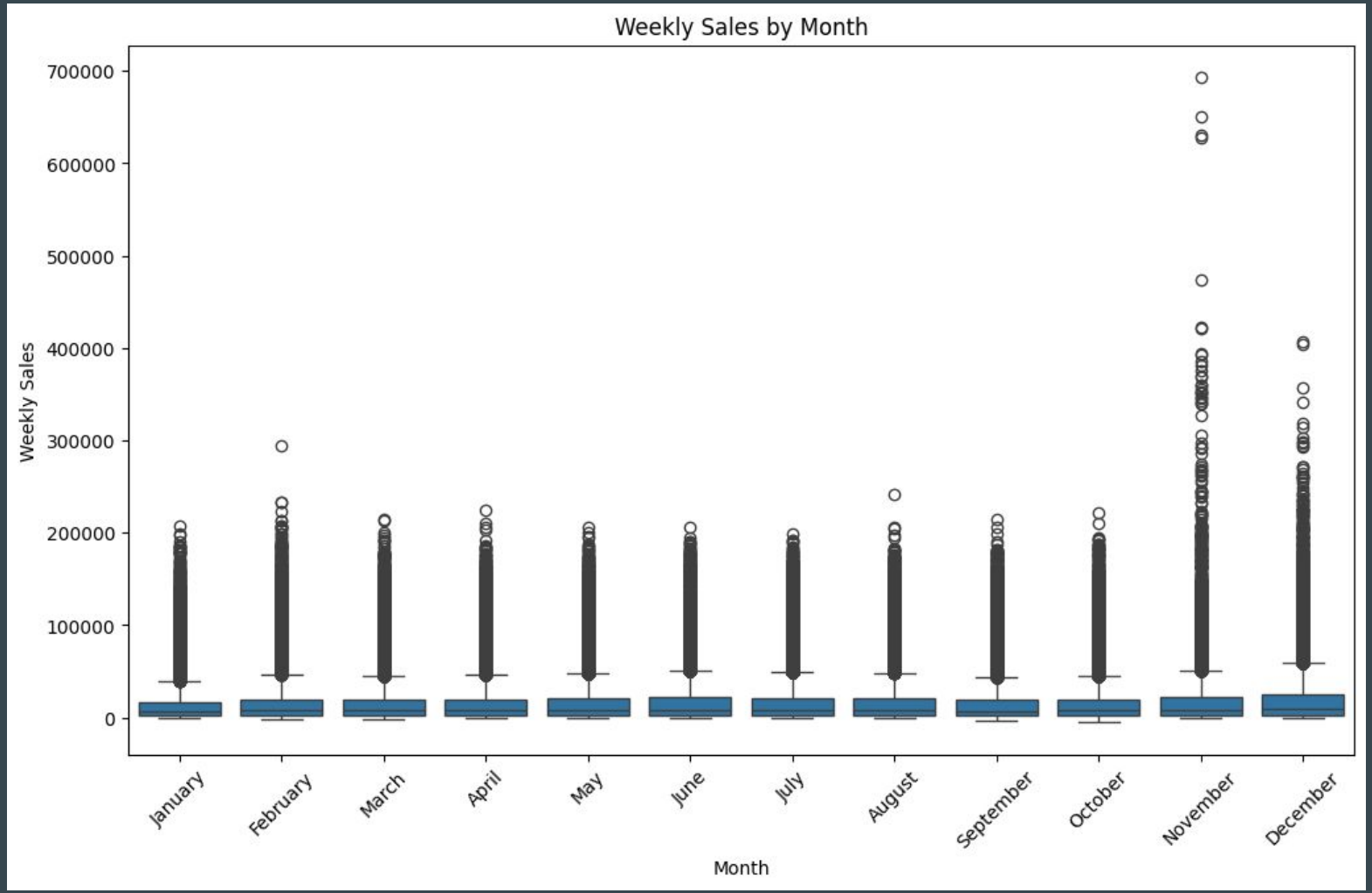
Walmart Recruiting - Store Sales Forecasting

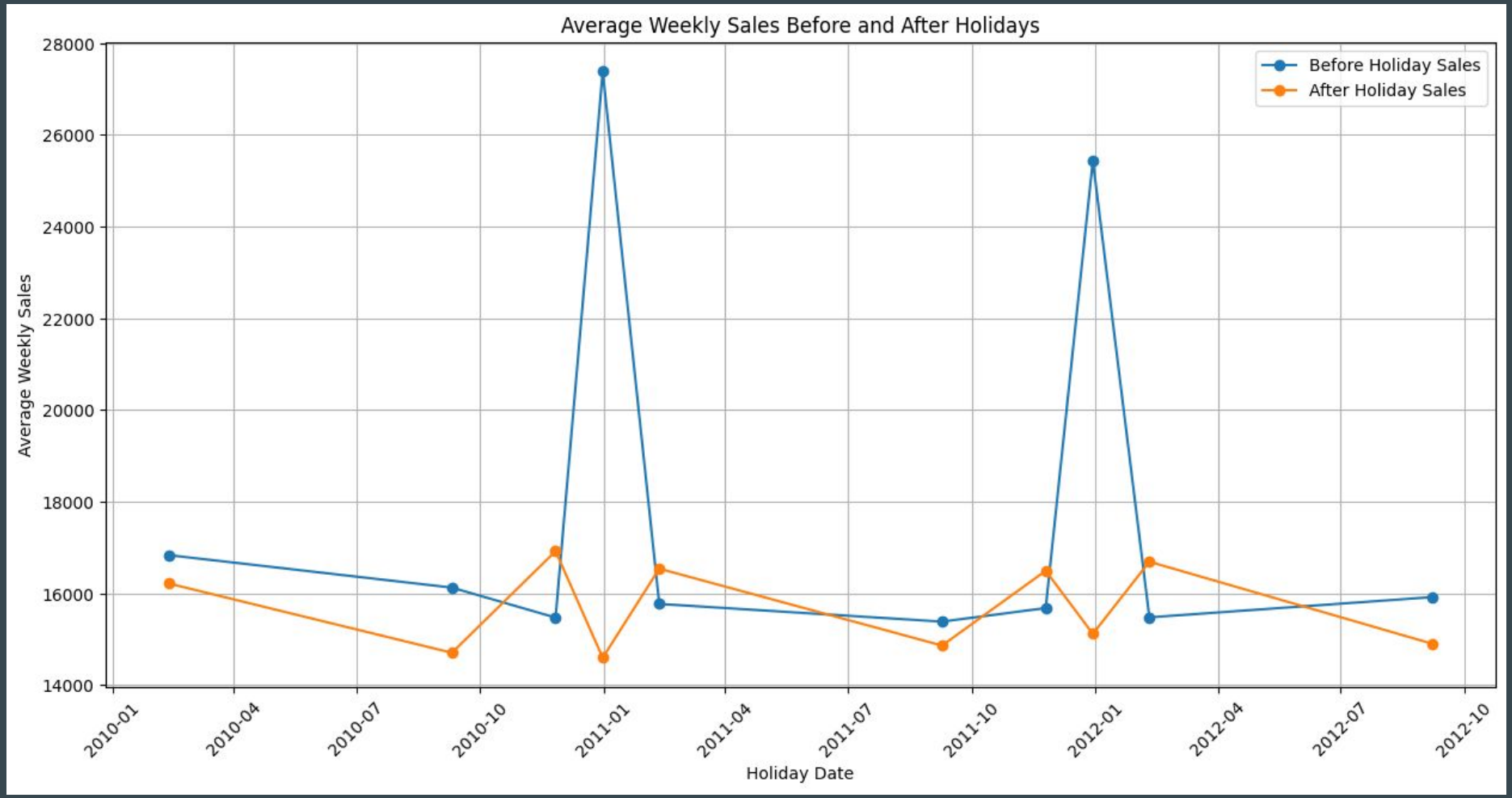
Mate Arevadze, Giorgi Toronjadze

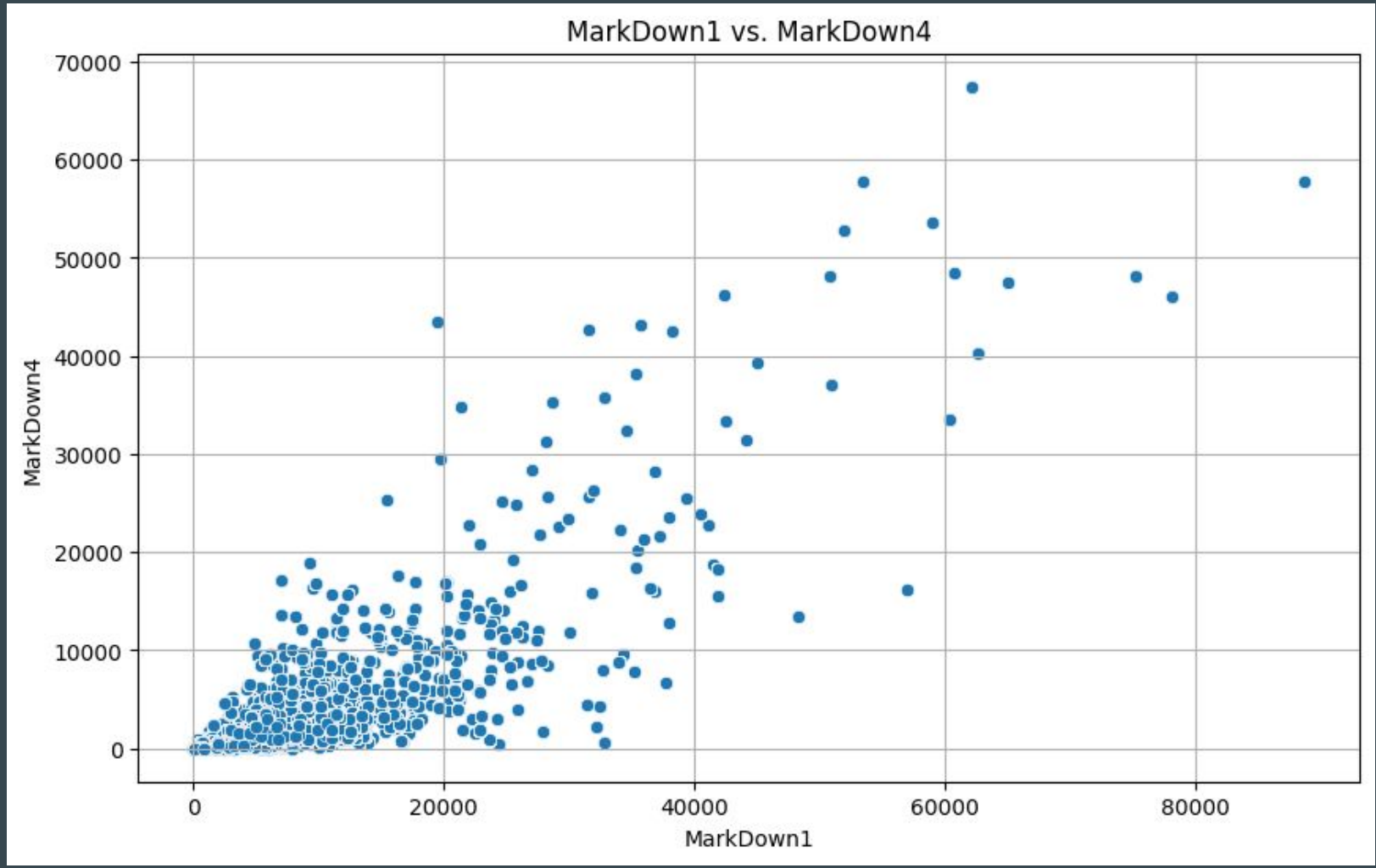

Correlation Matrix of Numerical Features
Store 1.00 0.02 -0.09 -0.00 -0.18 -0.05 0.07 -0.12 -0.04 -0.03 -0.01 -0.03 -0.21 0.21
Dept 0.02 1.00 0.15 0.00 -0.00 0.00 0.00 -0.00 0.00 0.00 0.00 0.00 -0.01 0.01
Weekly_Sales -0.09 0.15 1.00 0.01 0.24 -0.00 -0.00 0.09 0.02 0.06 0.05 0.09 -0.02 -0.03
IsHoliday -0.00 0.00 0.01 1.00 0.00 -0.16 -0.08 -0.04 0.33 0.43 -0.00 -0.05 -0.00 0.01
Size -0.18 -0.00 0.24 0.00 1.00 -0.06 0.00 0.35 0.11 0.05 0.17 0.30 -0.00 -0.07
Temperature -0.05 0.00 -0.00 -0.16 -0.06 1.00 0.14 -0.04 -0.32 -0.10 -0.06 -0.02 0.18 0.10
Fuel_Price 0.07 0.00 -0.00 -0.08 0.00 0.14 1.00 0.06 -0.22 -0.10 -0.04 -0.13 -0.16 -0.03
MarkDown1 -0.12 -0.00 0.09 -0.04 0.35 -0.04 0.06 1.00 0.02 -0.11 0.82 0.16 -0.06 0.05
MarkDown2 -0.04 0.00 0.02 0.33 0.11 -0.32 -0.22 0.02 1.00 -0.05 -0.01 -0.01 -0.04 0.02
MarkDown3 -0.03 0.00 0.06 0.43 0.05 -0.10 -0.10 -0.11 -0.05 1.00 -0.07 -0.03 -0.02 0.01
MarkDown4 -0.01 0.00 0.05 -0.00 0.17 -0.06 -0.04 0.82 -0.01 -0.07 1.00 0.11 -0.05 0.02
MarkDown5 -0.03 0.00 0.09 -0.05 0.30 -0.02 -0.13 0.16 -0.01 -0.03 0.11 1.00 0.06 -0.00
CPI -0.21 -0.01 -0.02 -0.00 -0.00 0.18 -0.16 -0.06 -0.04 -0.02 -0.05 0.06 1.00 -0.30
Unemployment 0.21 0.01 -0.03 0.01 -0.07 0.10 -0.03 0.05 0.02 0.01 0.02 -0.00 -0.30 1.00
Store
Dept
Weekly_Sales
IsHoliday
Size
Temperature
Fuel_Price
MarkDown1
MarkDown2
MarkDown3
MarkDown4
MarkDown5
CPI
Unemployment
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

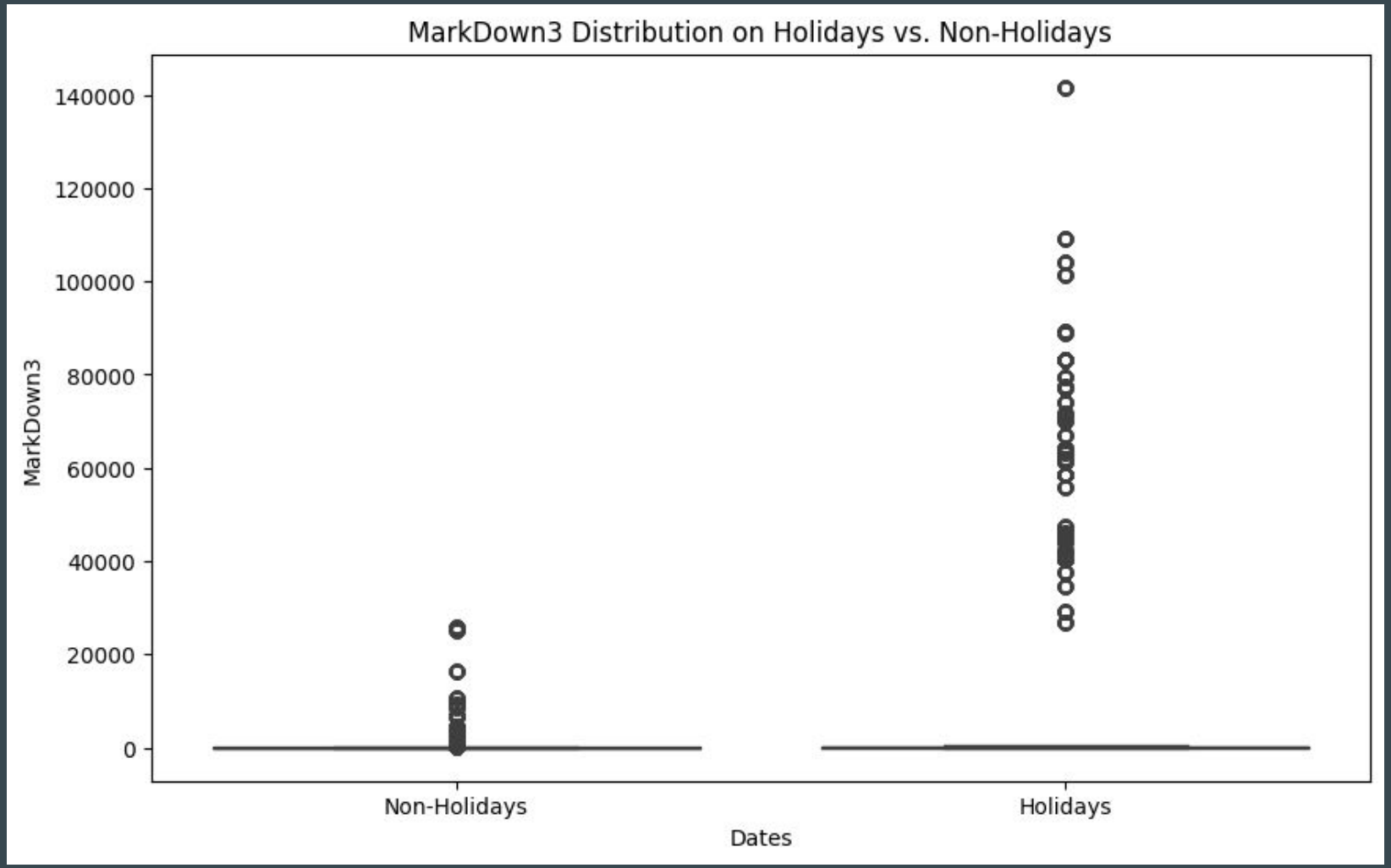

Weekly Sales by Store Type
Weekly Sales
700000
600000
500000
400000
300000
200000
100000
0
A
B
C
Store Type

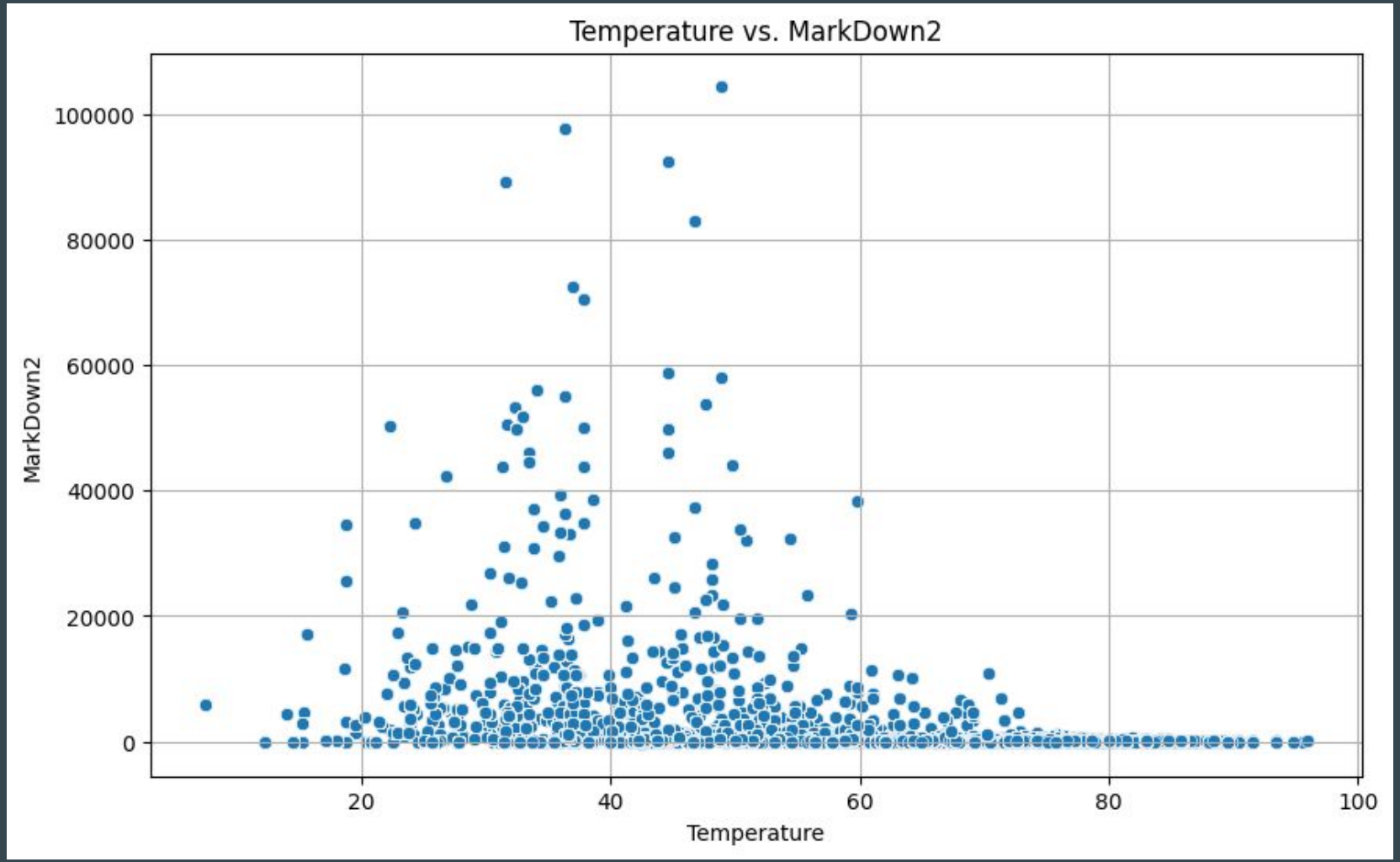

Weekly Sales by Season
700000
600000
500000
400000
300000
200000
100000
0
Weekly Sales
Winter
Spring
Summer
Fall
Season

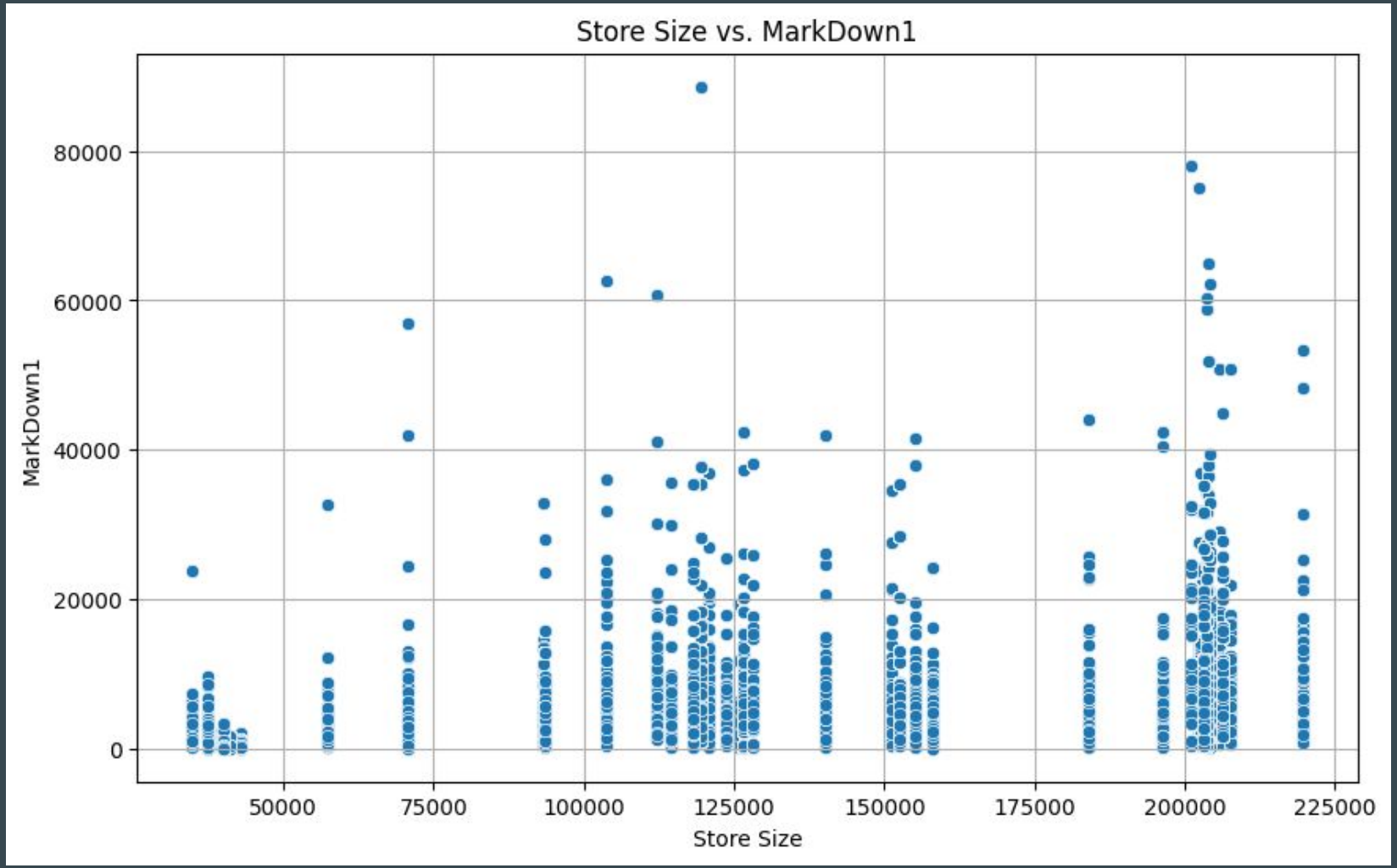

Weekly Sales by Month
Weekly Sales
700000
600000
500000
400000
300000
200000
100000
0
January
February
March
April
May
June
July
August
September
October
November
December
Month

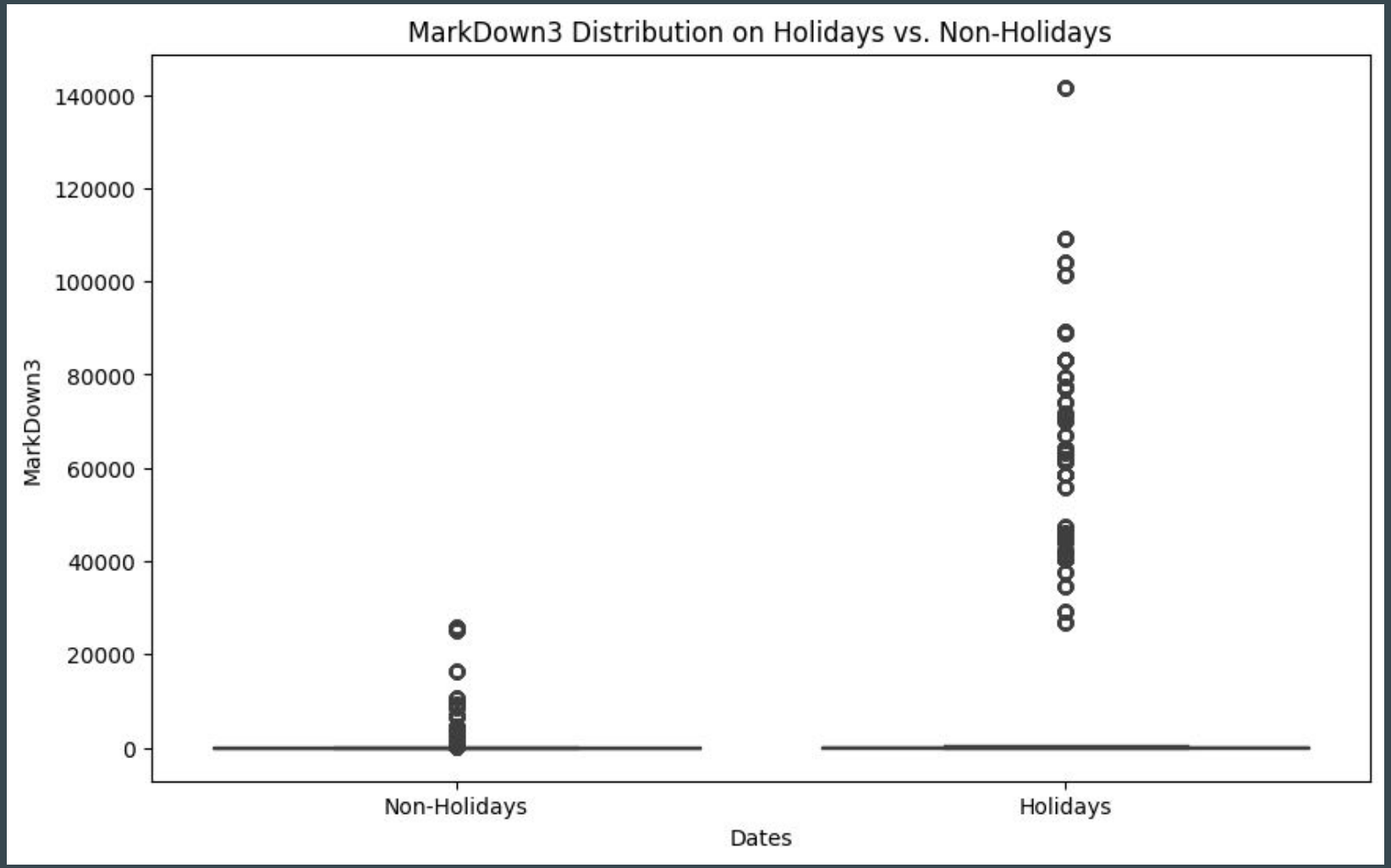

Average Weekly Sales Before and After Holidays
Before Holiday Sales
After Holiday Sales
Average Weekly Sales
Holiday Date
28000
26000
24000
22000
20000
18000
16000
14000
2010-01
2010-04
2010-07
2010-10
2011-01
2011-04
2011-07
2011-10
2012-01
2012-04
2012-07
2012-10

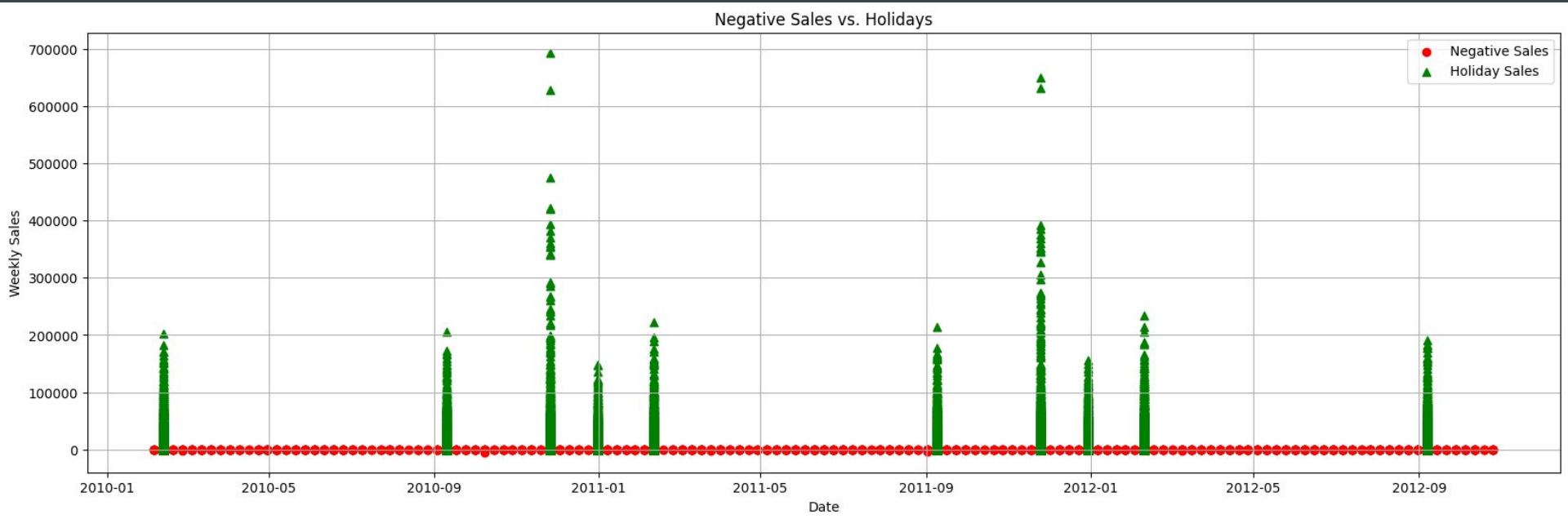

MarkDown1 vs. MarkDown4
MarkDown4
MarkDown1
70000
60000
50000
40000
30000
20000
10000
0
0
20000
40000
60000
80000


MarkDown3 Distribution on Holidays vs. Non-Holidays
140000
120000
100000
80000
60000
40000
20000
0
MarkDown3
Non-Holidays
Holidays
Dates


Temperature vs. MarkDown2
MarkDown2
Temperature
100000
80000
60000
40000
20000
0
20
40
60
80
100


Store Size vs. MarkDown1
MarkDown1
80000
60000
40000
20000
0
50000
75000
100000
125000
150000
175000
200000
225000
Store Size

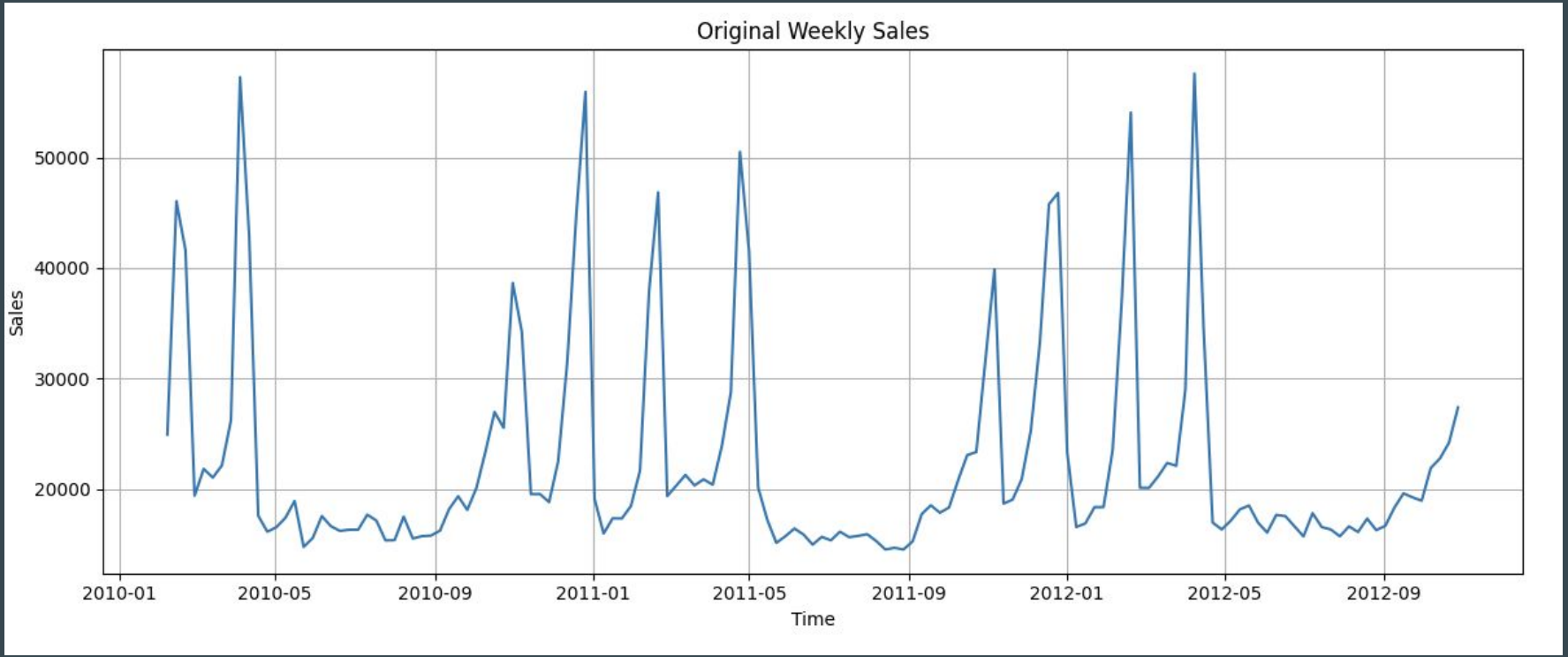

MarkDown3 Distribution on Holidays vs. Non-Holidays
140000
120000
100000
80000
60000
40000
20000
0
MarkDown3
Non-Holidays
Holidays
Dates


Negative Sales vs. Holidays
Negative Sales
Holiday Sales
Weekly Sales
700000
600000
500000
400000
300000
200000
100000
0
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
Date

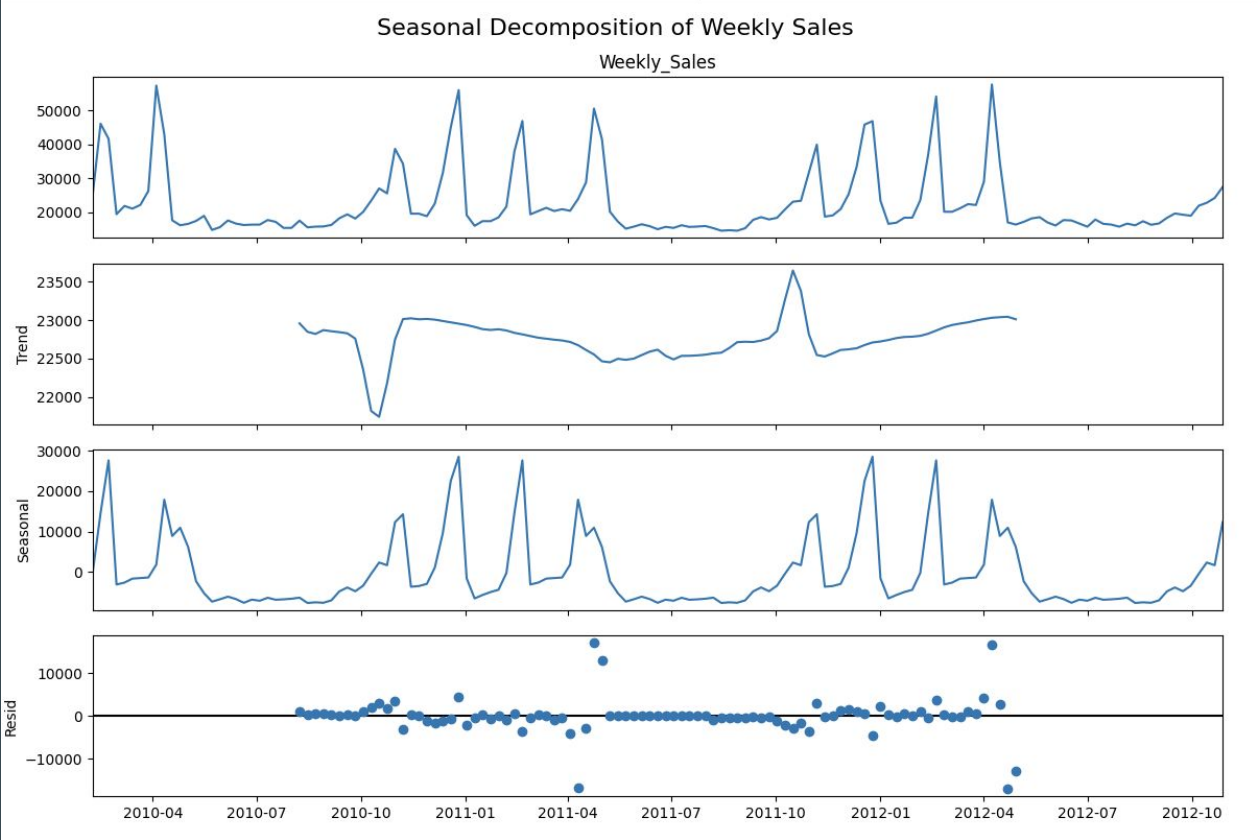
ARIMA/SARIMA/SARIMAX

- Used ARIMA (AutoRegressive Integrated Moving Average) for time series forecasting
- ARIMA(p,d,q): Models autoregressive (p), differencing (d), and moving average (q) components
- Challenges Encountered:
  - Non-stationarity: Sales data exhibited trends and non-constant variance.
  - Seasonality: Periodic patterns (e.g., holidays) not captured by basic ARIMA.
  - Errors: Inconsistent sample sizes, infinite values, and shape mismatches in some cases.

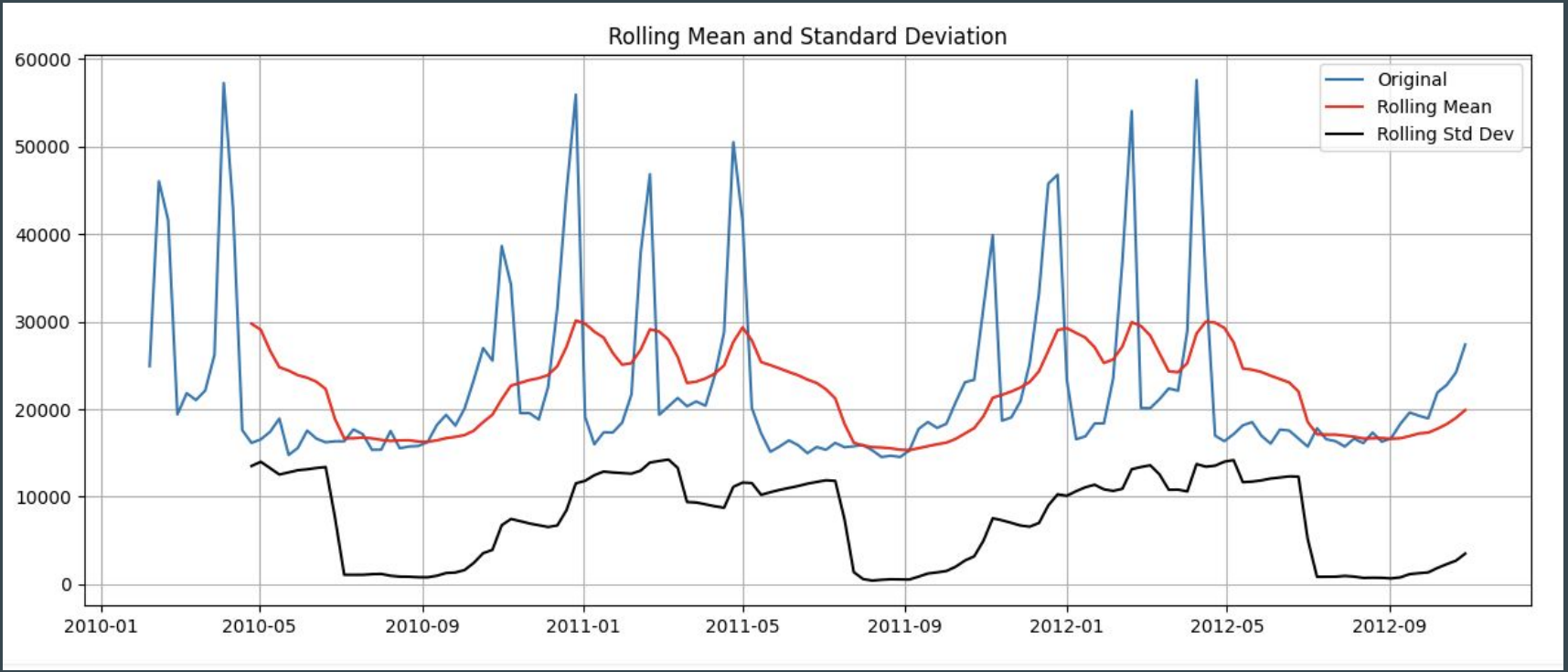
the sales data shows a clear upward/downward trend and fluctuating variance — meaning the data is non-stationary



Using seasonal decomposition, we can clearly observe repetitive seasonal patterns and a trend component



Both the rolling mean and standard deviation change over time



Exogenous Variables in SARIMAX Model

Exogenous variables (exogs) are external factors that impact sales.
Examples we use:

- **IsHoliday:** Whether the week includes a holiday (affects sales spikes).
- **Temperature:** Weather influences shopping behavior.
- **Fuel_Price:** Cost of fuel can affect store visits.
- **MarkDown1–5 :** Promotional discounts offered.
- **Season:** Season of the year.
- **Weeks_until_next_holiday:** Time until the next holiday.

Including these variables helps the model understand sales beyond historical patterns. This improves forecast accuracy.

PREPROCESSING

- MarkDown columns had many missing values (64%-74% in some)
- Fill with 0s missing MarkDown Columns
- Fill with mean other missing columns

- Got better result when filling with 0s

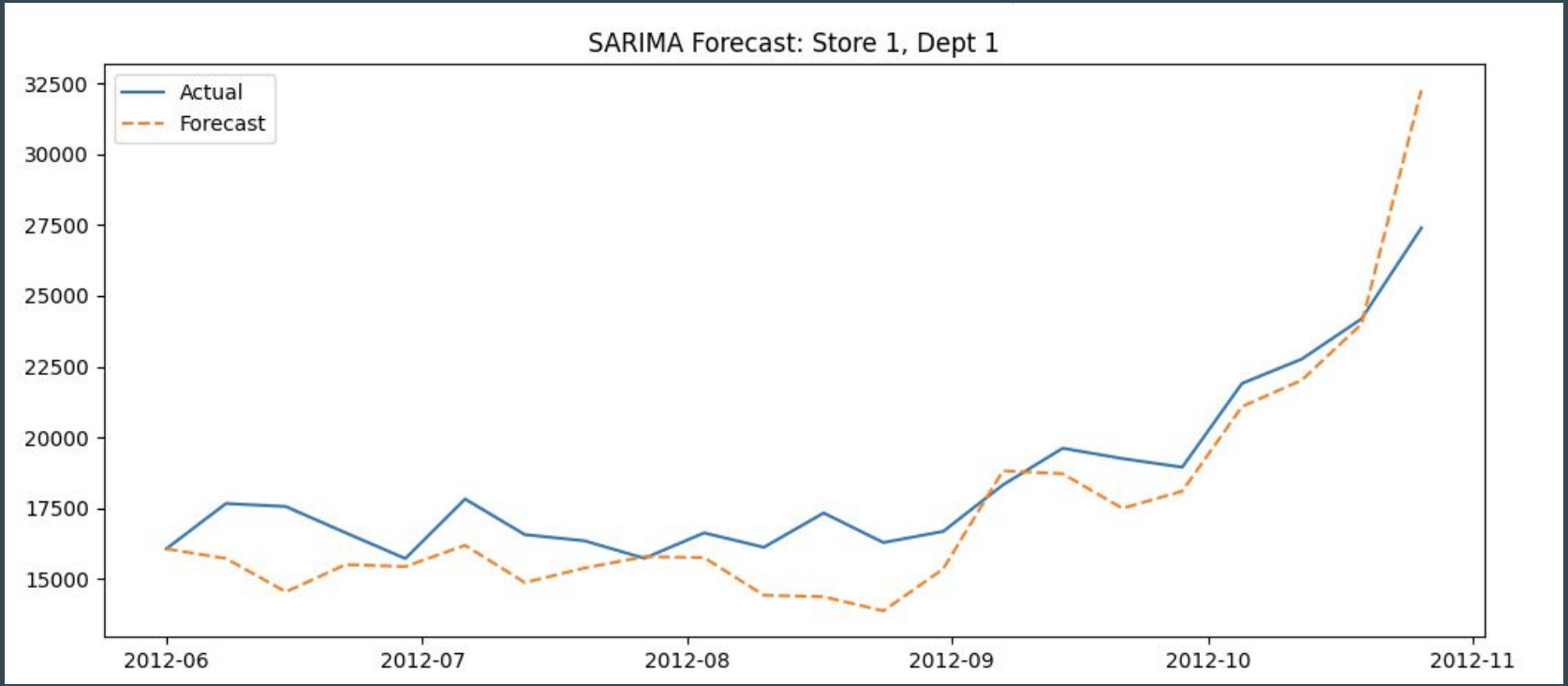
Hyperparameter Tuning with Grid Search

SARIMAX has several parameters:

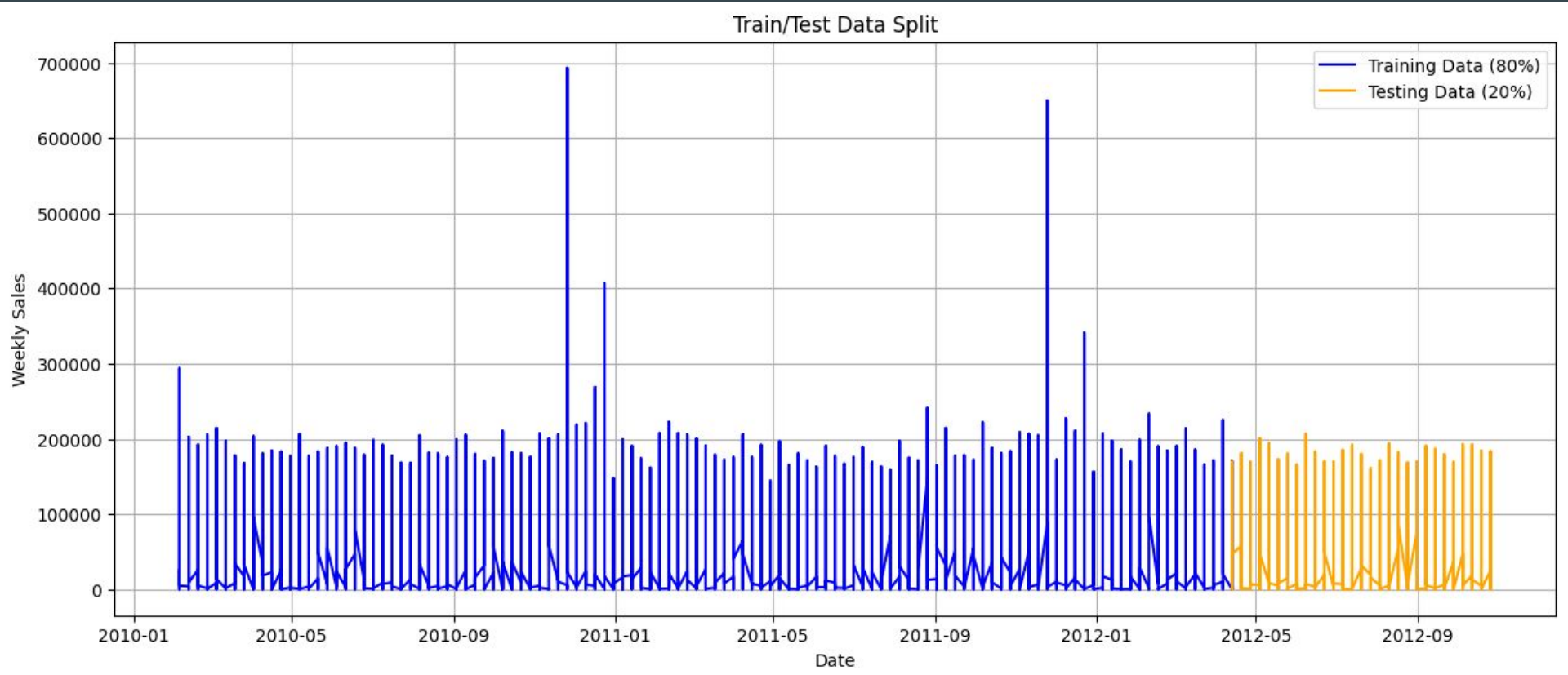
- Non-seasonal: **p, d, q**
- Seasonal: **P, D, Q, s** (with seasonal period s=52 weeks)

Choosing the right parameters is critical for model accuracy.
We automate this with **Grid Search**: trying all combinations in a defined range.

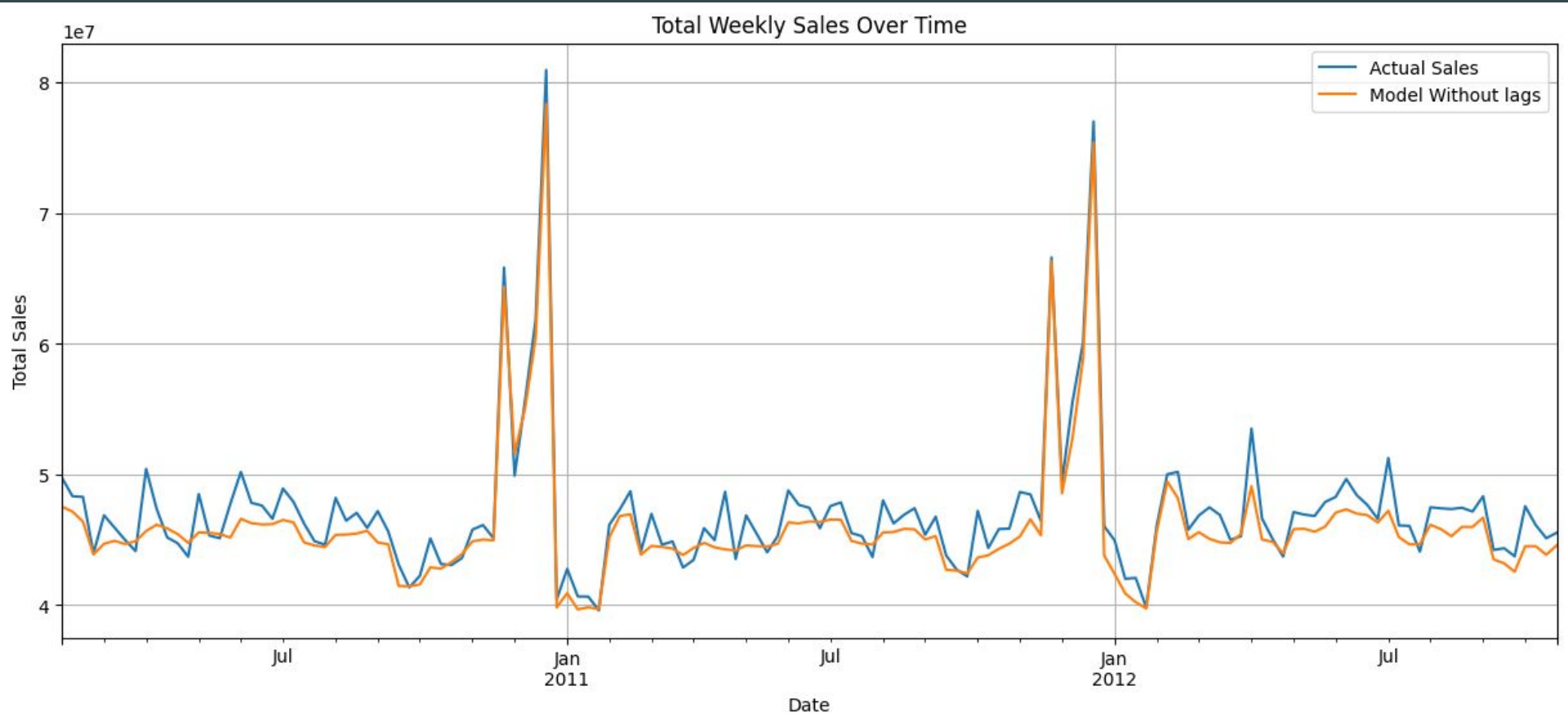
```
Best SARIMA: (1, 0, 1) x (0, 1, 0, 52) - AIC: 13.683230948927266
Best WMAE: 844.76
```


SARIMA Forecast: Store 1, Dept 1
Actual
Forecast
32500
30000
27500
25000
22500
20000
17500
15000
2012-06
2012-07
2012-08
2012-09
2012-10
2012-11

XGBOOOST

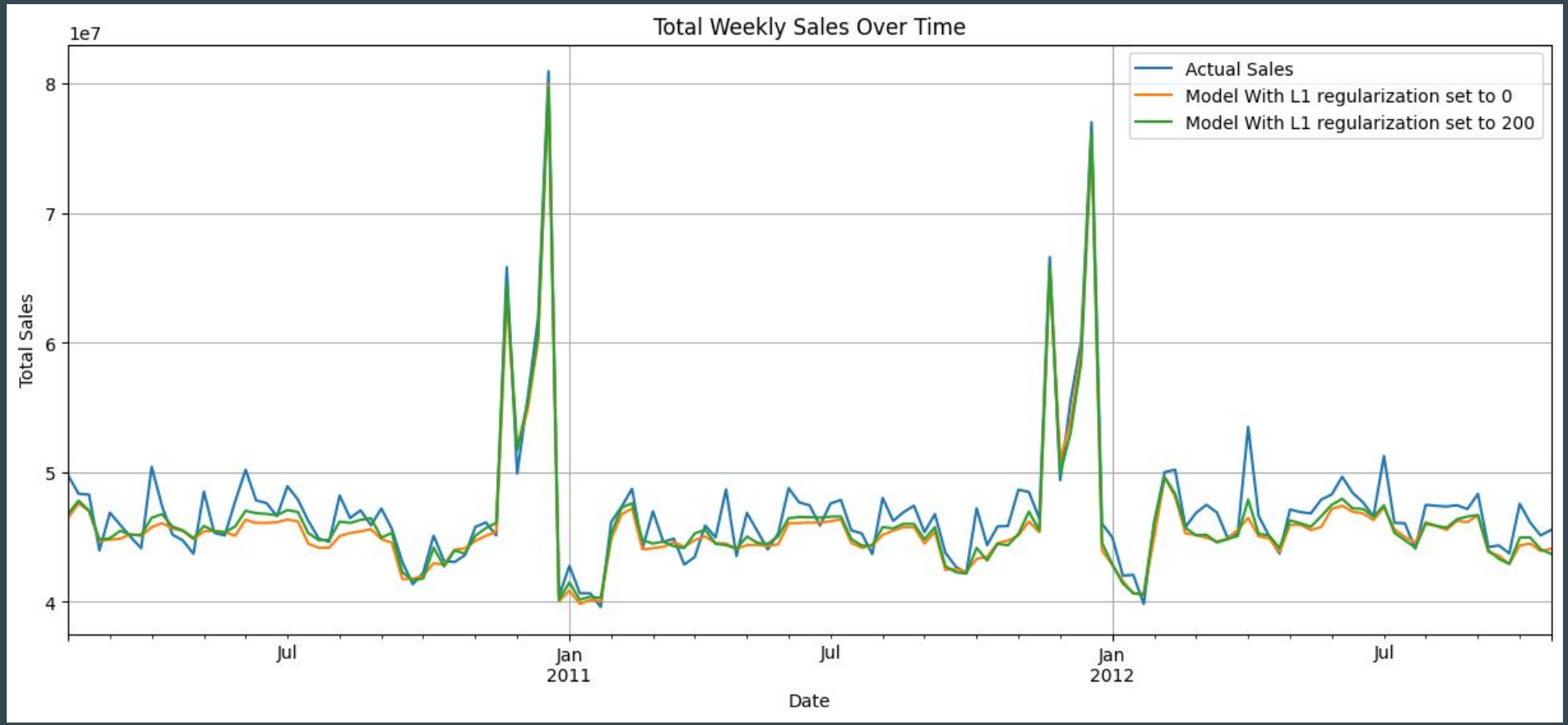

Train/Test Data Split
Training Data (80%)
Testing Data (20%)
Weekly Sales
700000
600000
500000
400000
300000
200000
100000
0
2010-01
2010-05
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
Date

Without Lagging Features, Validation WMAE: 6436, With Lagging Features: 6000

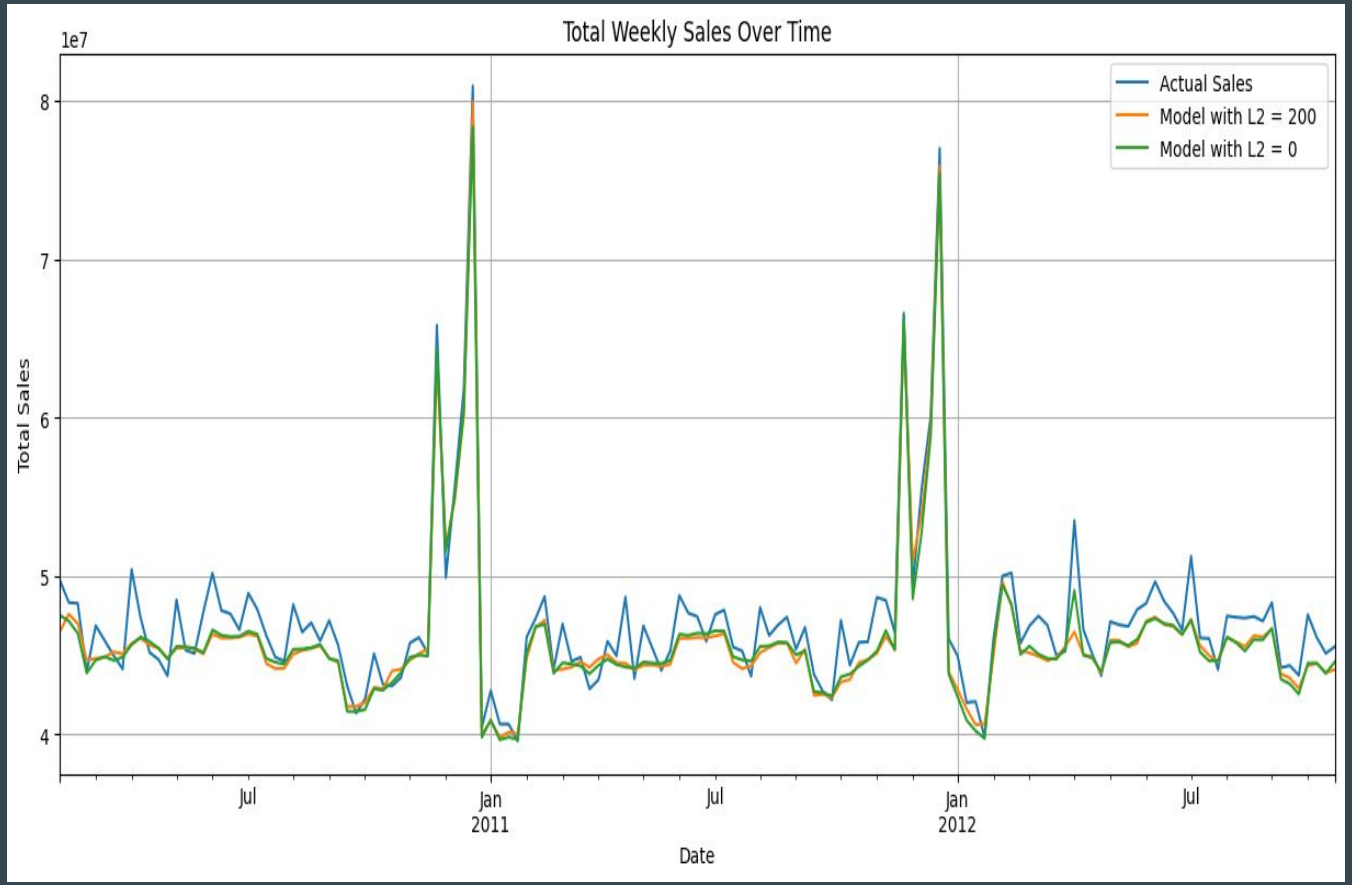


ALPHA REGULARIZATION EFFECTIVENESS

Validation WMAE for model with no L1 : 5700
Validation WMAE for model with L1 set to 200: 6400



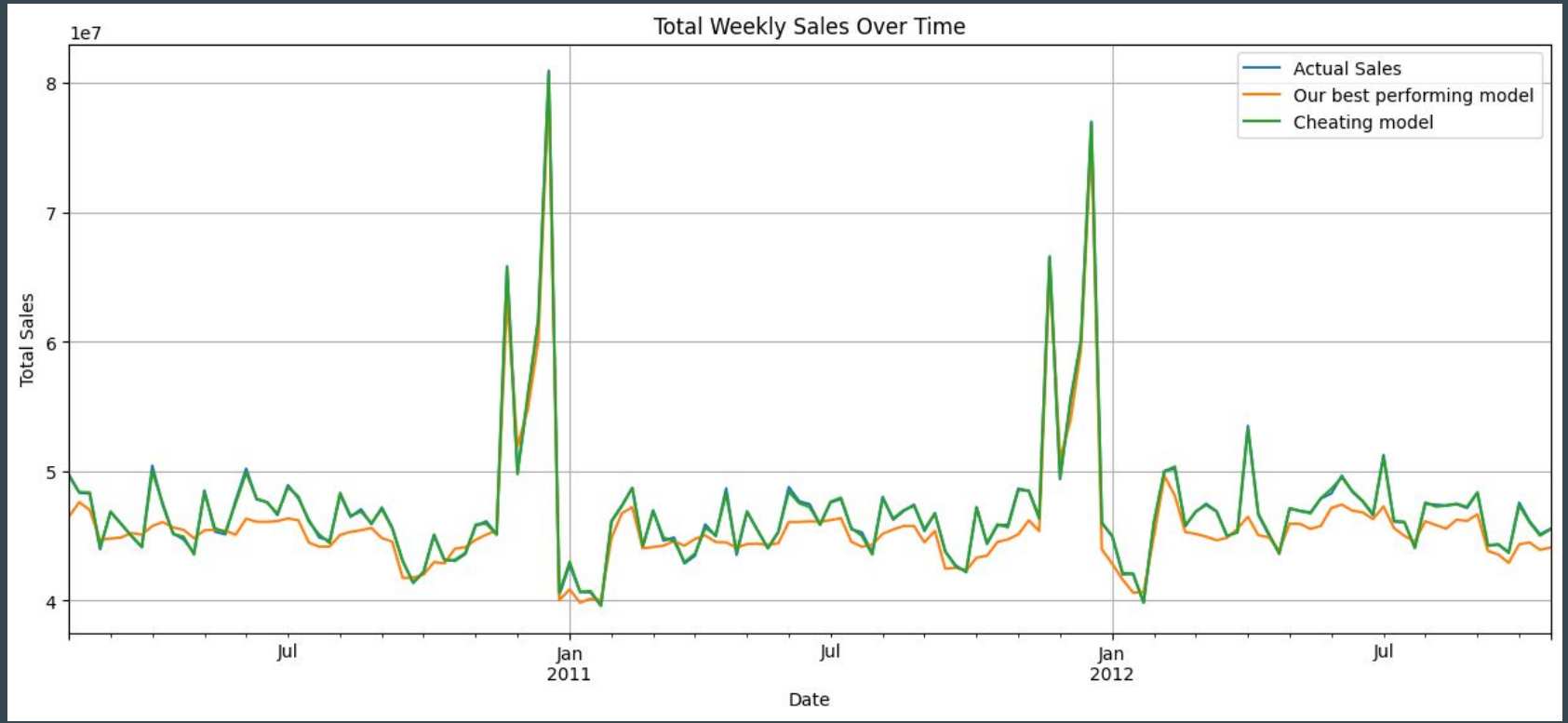
L2 (Lambda) Regularization effects



- Max_depth : 5
- n_estimators: 100
- reg_alpha: 0

- L2 Reg = 0: 6450 WMAE
- L2 Reg = 200 : 5800 WMAE

We Tried to see behavior of a “cheating” model that used random train/test split

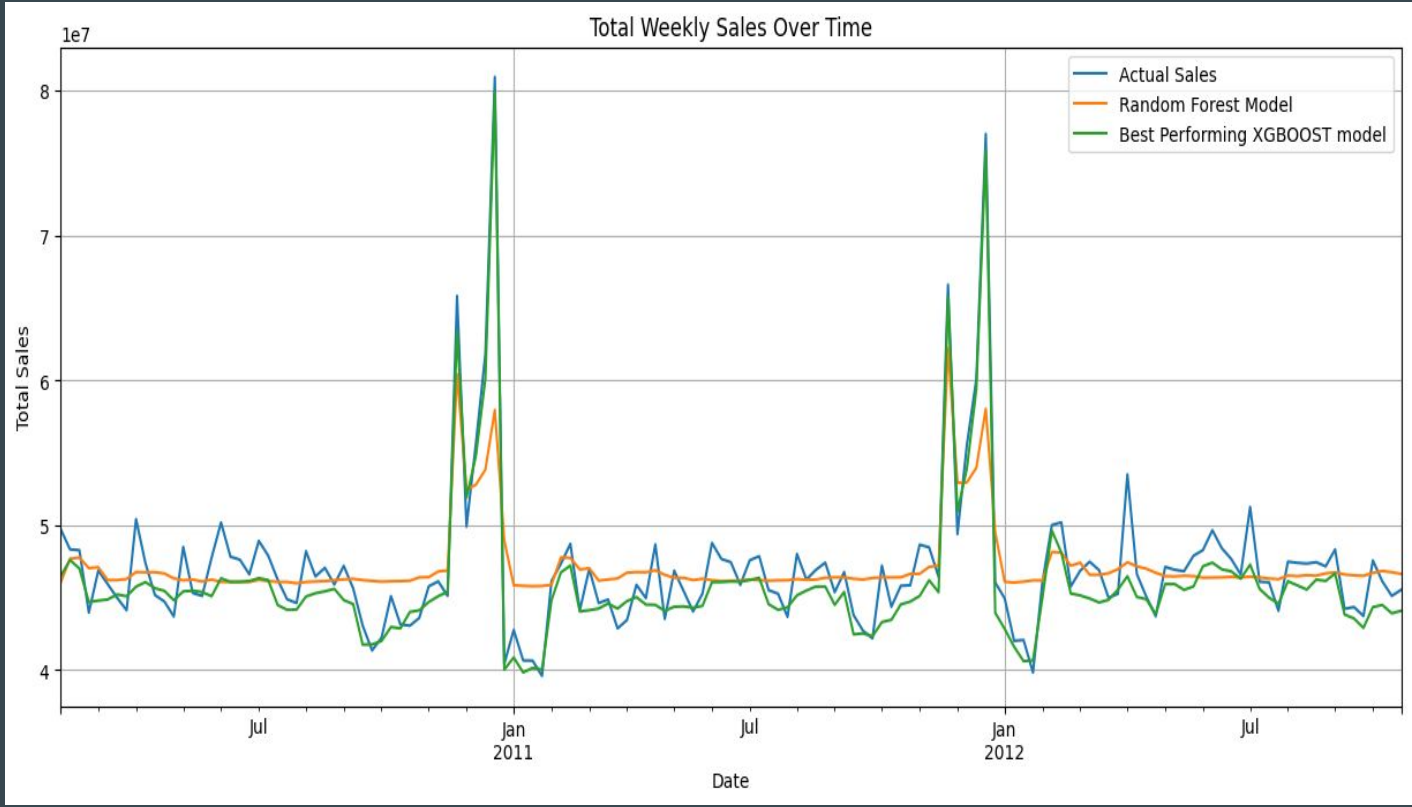


Other Features

- Max Depth was also an another important feature. setting it to high amounts (10,12) would cause overfitting, it would give us excellent results on train data (approx. 1800-2000), but it had bad results with validation data. even lambda regularization wouldn't work well if we didn't set max depth on conservative levels (5,6,7)
- Number of Estimators didn't affect the model that much, higher amount would improve the model, but not by significant margin
- we didn't experiment with learning rate that much, most of the time it was set to 0.1 or 0.3

RANDOM FOREST

Best Performing Random Forest Model compared to best performing XGBOOST model



Random Forest Model
WMAE: 5285

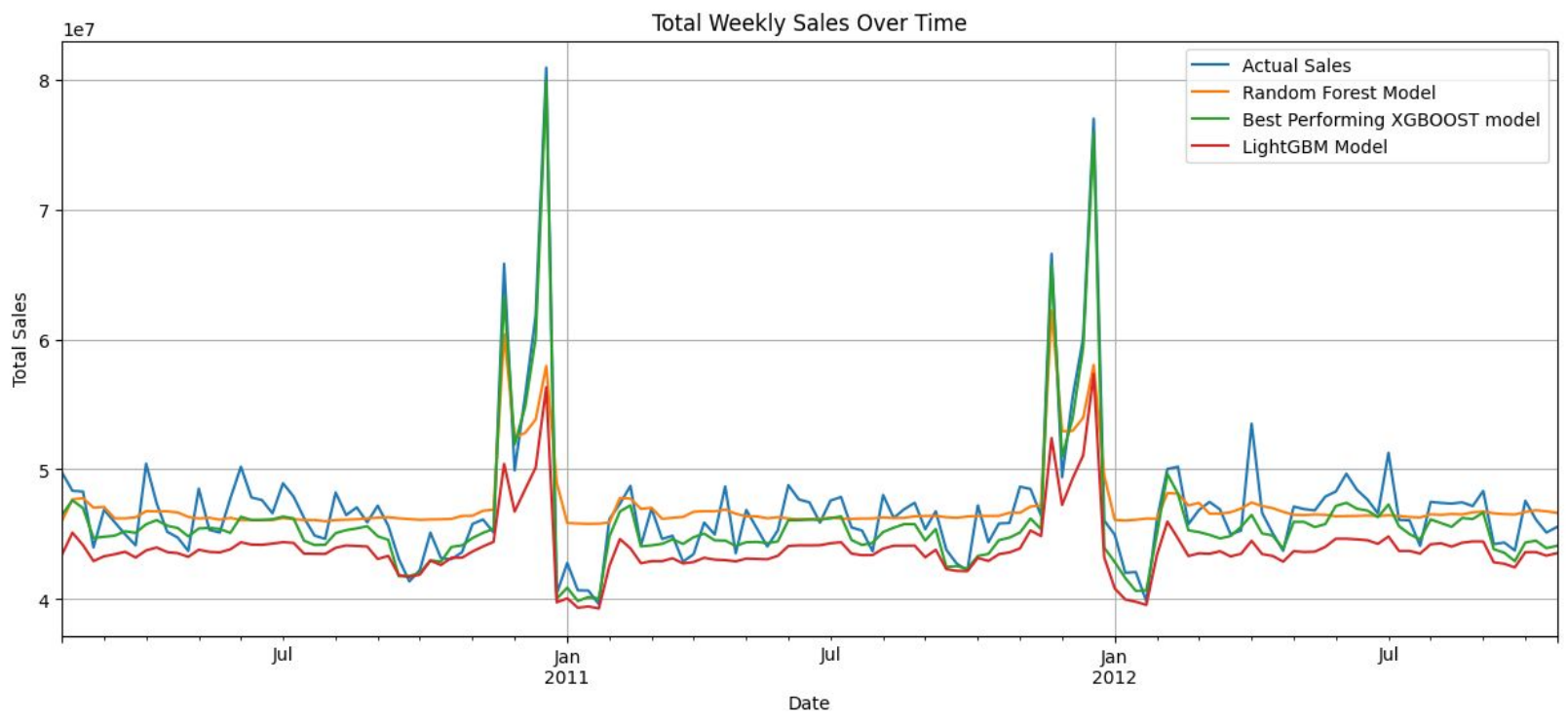
XGBOOST Model
WMAE: 5863

Despite this, XgBoost had a better score on kaggle and it was better at detecting seasonality and trends

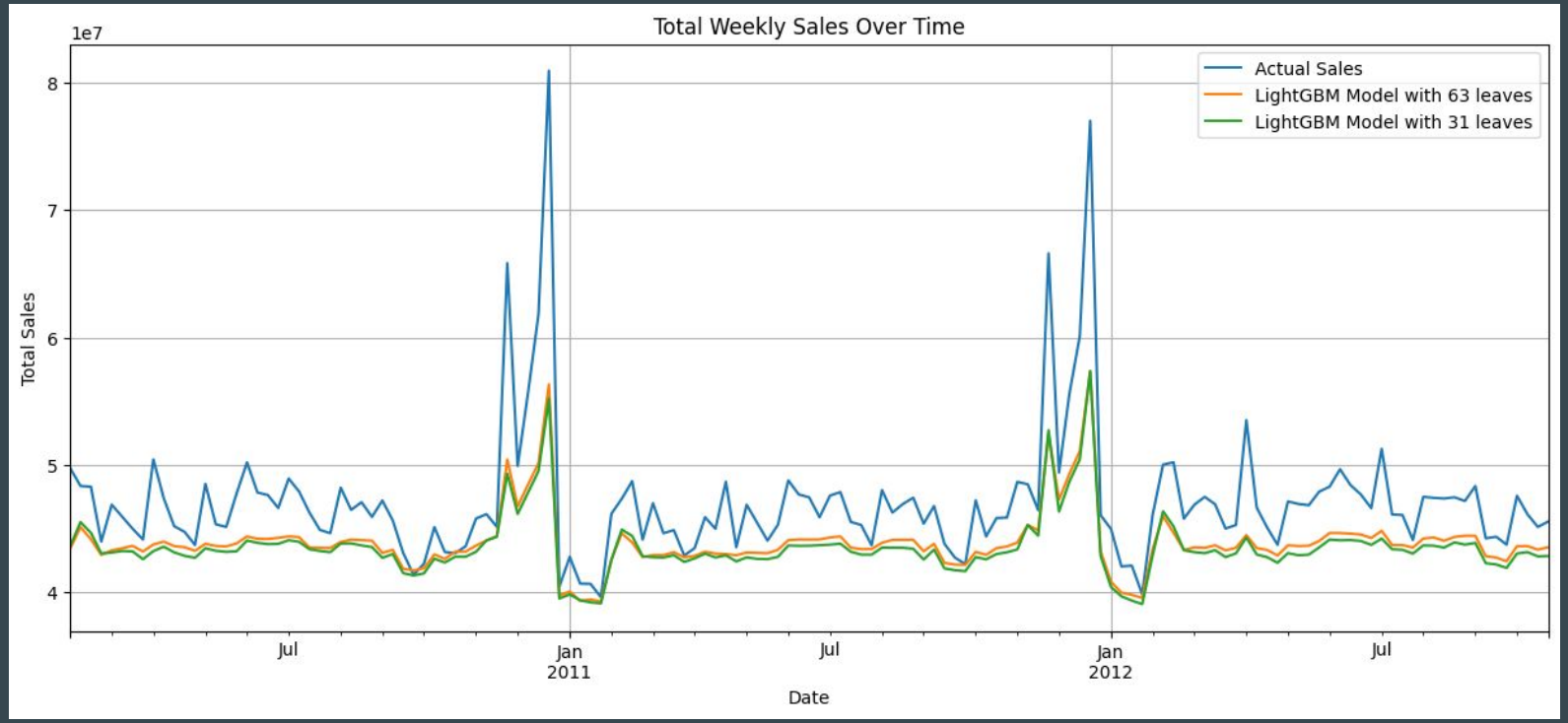
- Random Forests are not naturally suited for time series modeling because they do not account for the temporal order of data.
- While it's possible to add lag features or time-based variables through feature engineering, Random Forests treat observations as independent and identically distributed (i.i.d.), ignoring any sequential dependency between them.
- Moreover, since Random Forests randomly sample both features and data in parallel without preserving order, modeling temporal dynamics (like trends or seasonality) is very difficult. that is shown in the plots, even though its WMAE score is better than that of the xgboost, it can’t predict trends such as holiday sales

LIGHTGBM

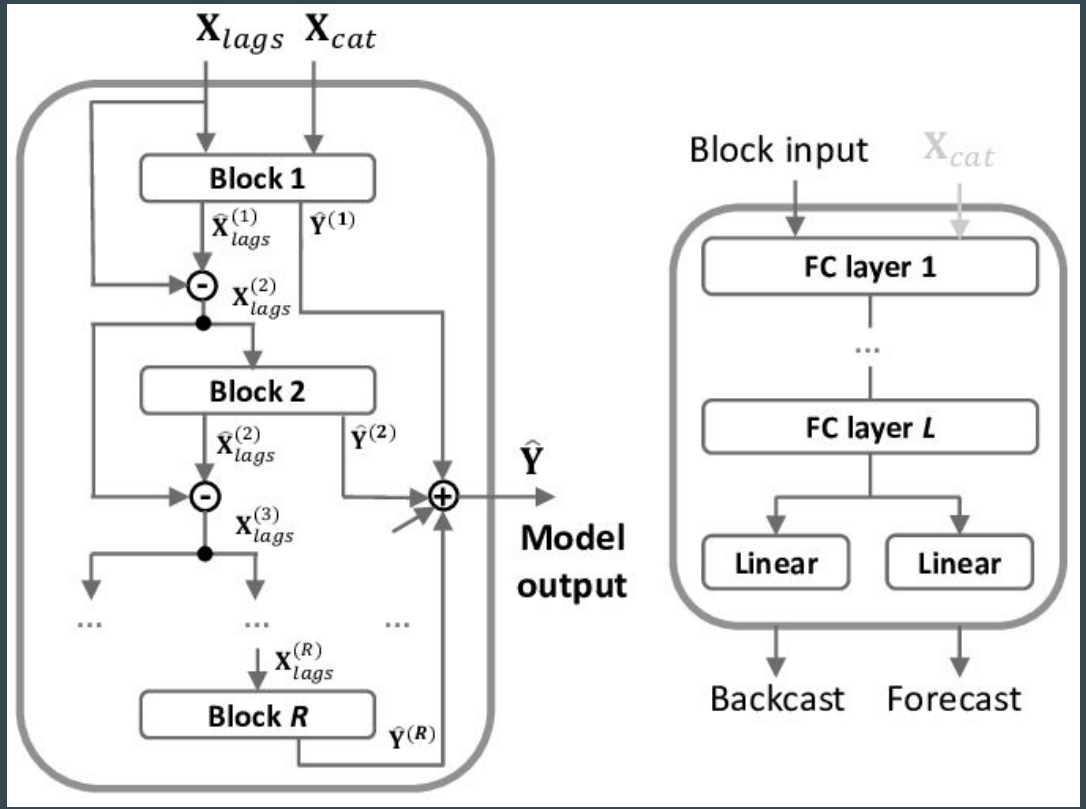
LightGBM was the best performing Decision Tree model for this problem, it had the highest train, validation and kaggle scores.



The Parameter which made the biggest improvement was the number of leaves, high amount of leaves (in our case, 63). number of estimators, reg_alpha and reg_lambda certainly affected the model, but not at the same magnitude

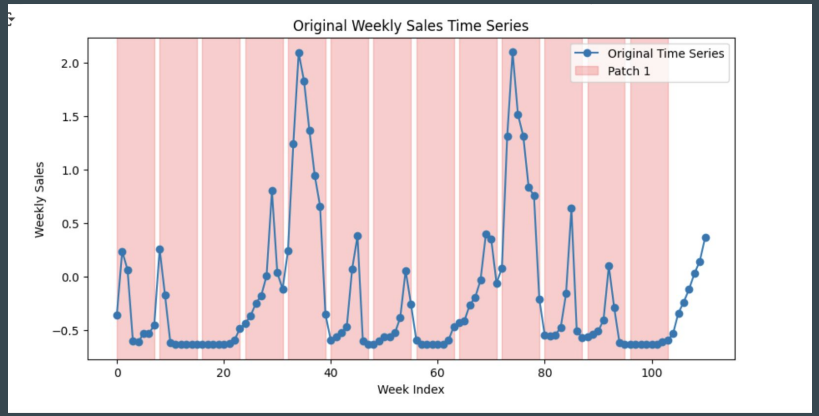


NBEATS


$\mathbf{X}_{lags}$ $\mathbf{X}_{cat}$
Block 1
$\hat{\mathbf{X}}^{(1)}_{lags}$
$\hat{\mathbf{Y}}^{(1)}$
$\mathbf{X}^{(2)}_{lags}$
Block 2
$\hat{\mathbf{X}}^{(2)}_{lags}$
$\hat{\mathbf{Y}}^{(2)}$
$\mathbf{X}^{(3)}_{lags}$
$\mathbf{X}^{(R)}_{lags}$
Block *R*
$\hat{\mathbf{Y}}^{(R)}$
$\hat{\mathbf{Y}}$
Model output
Block input
$\mathbf{X}_{cat}$
FC layer 1
FC layer *L*
Linear
Linear
Backcast
Forecast

PatchTST

- **PatchTST (Patch-based Time Series Transformer)** is a deep learning model tailored for **univariate and multivariate time series forecasting**.
- Unlike traditional transformers, PatchTST divides the input series into **non-overlapping patches** and learns dependencies between them, making it both efficient and accurate.
- It avoids complex encoder-decoder setups by using a **simplified encoder-only architecture**.
- Handles **seasonality and long-range dependencies**
- Efficiently models **multivariate time series**, which suits our rich feature set: temperature, holidays, markdowns, CPI, unemployment, etc.
- Is highly scalable and works well with large datasets



Grid Search

```
sweep_config = {
    'method': 'grid',
    'name': 'patchtst_grid_search',
    'metric': {'goal': 'minimize', 'name': 'val_wmae'},
    'parameters': {
        'seq_len': {'values': [26]},
        'pred_len': {'values': [1]},
        'patch_len': {'values': [4, 8, 12]},
        'stride': {'values': [4, 8, 12]},
        'd_model': {'values': [32, 64]},
        'learning_rate': {'values': [1e-4, 1e-3]},
        'n_layers': {'values': [1, 2]},
        'epochs': {'values': [10]}
    }
}
```

seq_len: Number of past time steps fed into the model (input window).
pred_len: Number of future steps to predict
patch_len: Length of each input patch – smaller patches capture finer patterns, longer ones capture broader trends. |
Stride: How much to shift between patches – controls overlap and diversity.
D_model: Dimensionality of the model embeddings (internal feature space).
 Learning_rate: Controls how quickly the model adapts weights during training.
N_layers: Depth of the transformer (more layers → more learning capacity, but risk of overfitting).
Epochs: Total training iterations through the dataset.

Best Configuration and Result

```
d_model: 32
epochs: 10
learning_rate: 0.0001
patch_len: 12
seq_len: 26
stride: 4
```

Val WMAE: 2957.1

Final Results on Kaggle

submission (6).csv Complete (after deadline) · 13h ago · XGBoost without Lag features	7116.07521	7014.83048
submission (5).csv Complete (after deadline) · 19h ago · XGBoost with lag features and best parameters	6037.21085	5860.71966
submission (4).csv Complete (after deadline) · 21h ago · Random Forest Model	7261.05578	7128.24506
submission (3).csv Complete (after deadline) · 21h ago · Light GBM with low leaves	5773.85751	5650.46062
submission (2).csv Complete (after deadline) · 1d ago · LightBGM model	5104.75333	4952.52175