



ISM6136 : Data Mining - Final Project

Under Guidance of:

Dr. Balaji Padmanabhan (Dept. Chair, ISDS Department)

Group Member(s):

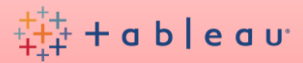
Abhinav Singh (U142741172)

Maurya Hosahalli Nagaraja (U23000754)

Radhika Viswakarma (U26251590)

Sumant Sharma (U29150574)

Powered by:



Data Provider:



[Watch Video](#)

About Rossmann Stores:

2nd

Largest
Drug Store chain
In Germany

3000

Stores
in

7

European Countries

Founded in

1972

by

**Dirk
Rossmann**



What Rossmann wants ?



Why ?

(Current Scenario)

Store managers predicting sales individually with their own logic & intuition

Varying accuracy of results

(Solution)

Create a Robust – Generalized prediction model

Help store managers stay focused on what's most important to the business

(Their customers and teams)

How Rossmann will evaluate the model:

Based on

Root Mean Square Percentage Error (RMSPE)

Calculated by:

$$\text{RMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2}$$

Where:

y_i

= Sales of a single store on a single day.

\hat{y}_i

= The corresponding Sales prediction.

Note: Any day and store with '**0**' Sales is ignored while scoring.

Scope of our project:

Forecasting the **"Sales"** column for the hold-out data set.

What does the data set mainly contain ?

Store info

Promotion details

Competitor data

School holidays

State holidays

Seasonality

Data Dictionary - 1

Id

An Id that represents a (Store, Date) duple within the test set.

Store

A unique Id for each store

Sales

The turnover for any given day

Customers

The number of customers on a given day

Open

An indicator for whether the store was open
(0 = closed, 1 = open)

Data Dictionary - 2

StateHoliday

Indicates a state holiday

Normally all stores, with few exceptions, are closed on state holidays

(a = public holiday, b = Easter holiday, c = Christmas, 0 = None)

SchoolHoliday

Indicates if the (Store, Date) was affected by the closure of public schools

All schools are closed on public holidays and weekends

StoreType

Differentiates between 4 different store models/types

(Store types = a, b, c or d)

Assortment

Describes assortment level of the store

(Assortment types: a = basic, b = extra, c = extended)

CompetitionDistance

Distance in meters to the nearest competitor store

CompetitionOpenSince

Indicates approximate [Month/Year] of time when the nearest competitor was opened

Data Dictionary - 3

Promo

Indicates whether a store is running a promo on that day

Promo2

Indicates continuing & consecutive promotions for stores
(0 = store is not participating, 1 = store is participating)

Promo2Since

Indicates [Year/Calendar Week] when the store started participating in Promo2.

PromoInterval

Describes the consecutive intervals Promo2 is started
Value = Start month of the Promo2

(Ex. Value = 'Feb, May' indicated that new promo rounds started in February & May anew)

Data set snapshot – Train.csv

(Contains historical data including Sales)

Store	Date [^]	Day Of Week	Customers	Open	Promo	School Holiday	State Holiday	Sales
1	1/1/2013	2	0	0	0	1	Null	0
2	1/1/2013	2	0	0	0	1	Null	0
3	1/1/2013	2	0	0	0	1	Null	0
4	1/1/2013	2	0	0	0	1	Null	0
5	1/1/2013	2	0	0	0	1	Null	0
~~~~~								
1,110	1/1/2013	2	0	0	0	1	Null	0
1,111	1/1/2013	2	0	0	0	1	Null	0
1,112	1/1/2013	2	0	0	0	1	Null	0
1,113	1/1/2013	2	0	0	0	1	Null	0
1,114	1/1/2013	2	0	0	0	1	Null	0
1,115	1/1/2013	2	0	0	0	1	Null	0
~~~~~								
1	7/31/2015	5	555	1	1	1	0	5,263
2	7/31/2015	5	625	1	1	1	0	6,064
3	7/31/2015	5	821	1	1	1	0	8,314
4	7/31/2015	5	1,498	1	1	1	0	13,995
5	7/31/2015	5	559	1	1	1	0	4,822
~~~~~								
1,110	7/31/2015	5	642	1	1	1	0	6,198
1,111	7/31/2015	5	422	1	1	1	0	5,723
1,112	7/31/2015	5	767	1	1	1	0	9,626
1,113	7/31/2015	5	720	1	1	1	0	7,289
1,114	7/31/2015	5	3,745	1	1	1	0	27,508
1,115	7/31/2015	5	538	1	1	1	0	8,680

# Data set snapshot – Store.csv

(Contains supplemental information about the stores)

Store	Store Type	Assortment	Competition Distance	Competition Open Since Month	Competition Open Since Year	Promo2	Promo Interval	Promo2SinceWeek	Promo2SinceYear
1	c	a	1,270	9	2008	0	Null	Null	Null
2	a	a	570	11	2007	1	Jan,Apr,Jul,Oct	13	2010
3	a	a	14,130	12	2006	1	Jan,Apr,Jul,Oct	14	2011
4	c	c	620	9	2009	0	Null	Null	Null
5	a	a	29,910	4	2015	0	Null	Null	Null
1,110	c	c	900	9	2010	0	Null	Null	Null
1,111	a	a	1,900	6	2014	1	Jan,Apr,Jul,Oct	31	2013
1,112	c	c	1,880	4	2006	0	Null	Null	Null
1,113	a	c	9,260	Null	Null	0	Null	Null	Null
1,114	a	c	870	Null	Null	0	Null	Null	Null
1,115	d	c	5,350	Null	Null	1	Mar,Jun,Sept...	22	2012

# Data set snapshot

## Test.csv

(Contains historical data excluding Sales)

Store	Id	Date	Day Of Week	Open	Promo	School Holiday	State Holiday
1	40233	8/1/2015	6	1	0	1	0
3	40234	8/1/2015	6	1	0	0	0
7	40235	8/1/2015	6	1	0	0	0
8	40236	8/1/2015	6	1	0	0	0
9	40237	8/1/2015	6	1	0	0	0
~~~~~							
1,111	41084	8/1/2015	6	1	0	0	0
1,112	41085	8/1/2015	6	1	0	0	0
1,113	41086	8/1/2015	6	1	0	0	0
1,114	41087	8/1/2015	6	1	0	0	0
1,115	41088	8/1/2015	6	1	0	1	0
~~~~~							
1	1	9/17/2015	4	1	1	0	0
3	2	9/17/2015	4	1	1	0	0
7	3	9/17/2015	4	1	1	0	0
8	4	9/17/2015	4	1	1	0	0
9	5	9/17/2015	4	1	1	0	0
~~~~~							
1,111	852	9/17/2015	4	1	1	0	0
1,112	853	9/17/2015	4	1	1	0	0
1,113	854	9/17/2015	4	1	1	0	0
1,114	855	9/17/2015	4	1	1	0	0
1,115	856	9/17/2015	4	1	1	0	0

Sample_Submission.csv

(Contains a sample submission file in the correct format)

Id	Sales
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
~~~~~	
41079	0
41080	0
41081	0
41082	0
41083	0
41084	0
41085	0
41086	0
41087	0
41088	0

# Exploratory Data Analysis - 1

The training data set consists of sales data for **1,115 stores** across **942 days**.

---

Data Expectation:  $1,115 \times 942 =$   
**1,050,330 data rows**

---

Data Actuality:  
**1,017,209 data rows**

---

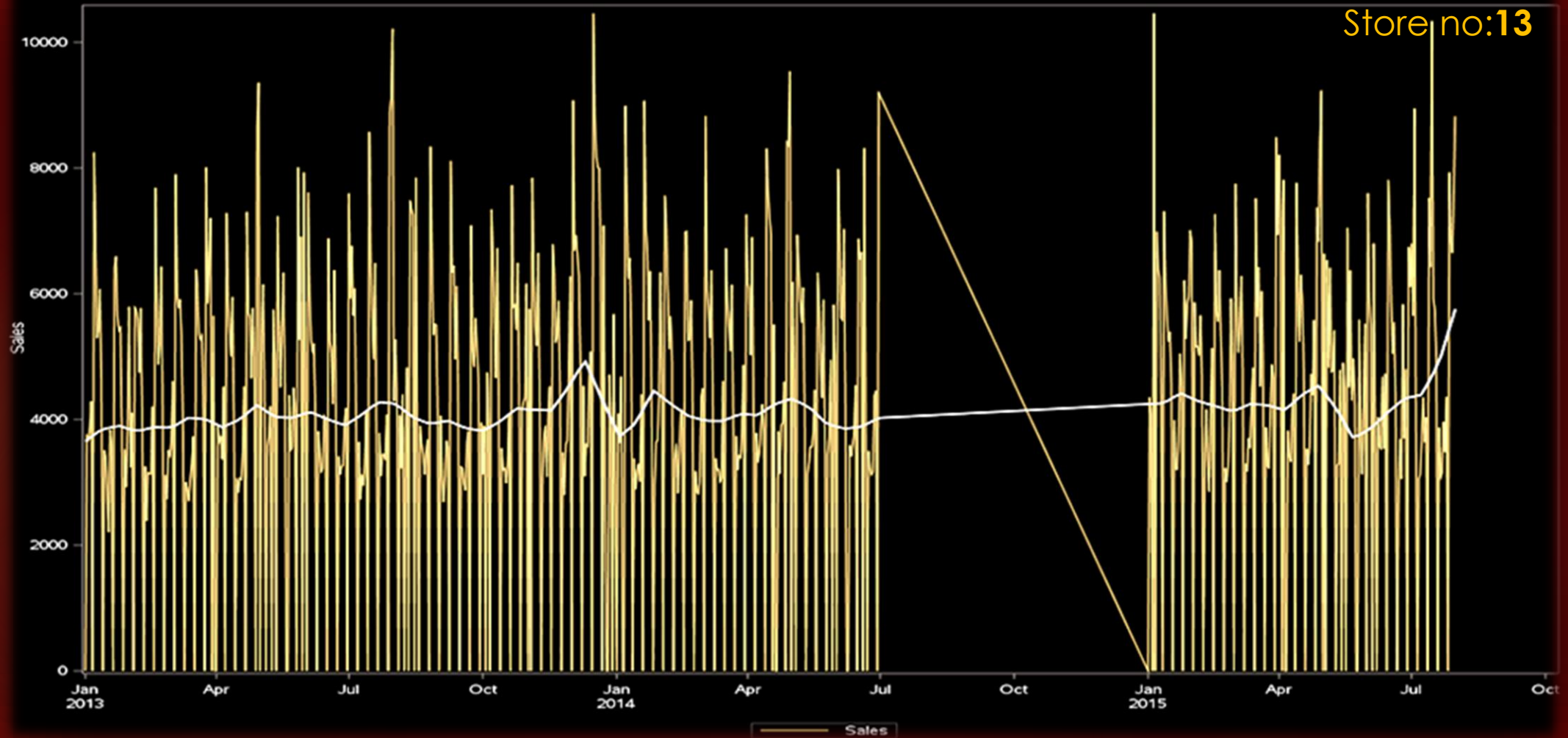
Missing Instance count:  
**33,121 data rows**

---

Reasoning:  
**Some stores in the dataset were temporarily closed for refurbishment.**



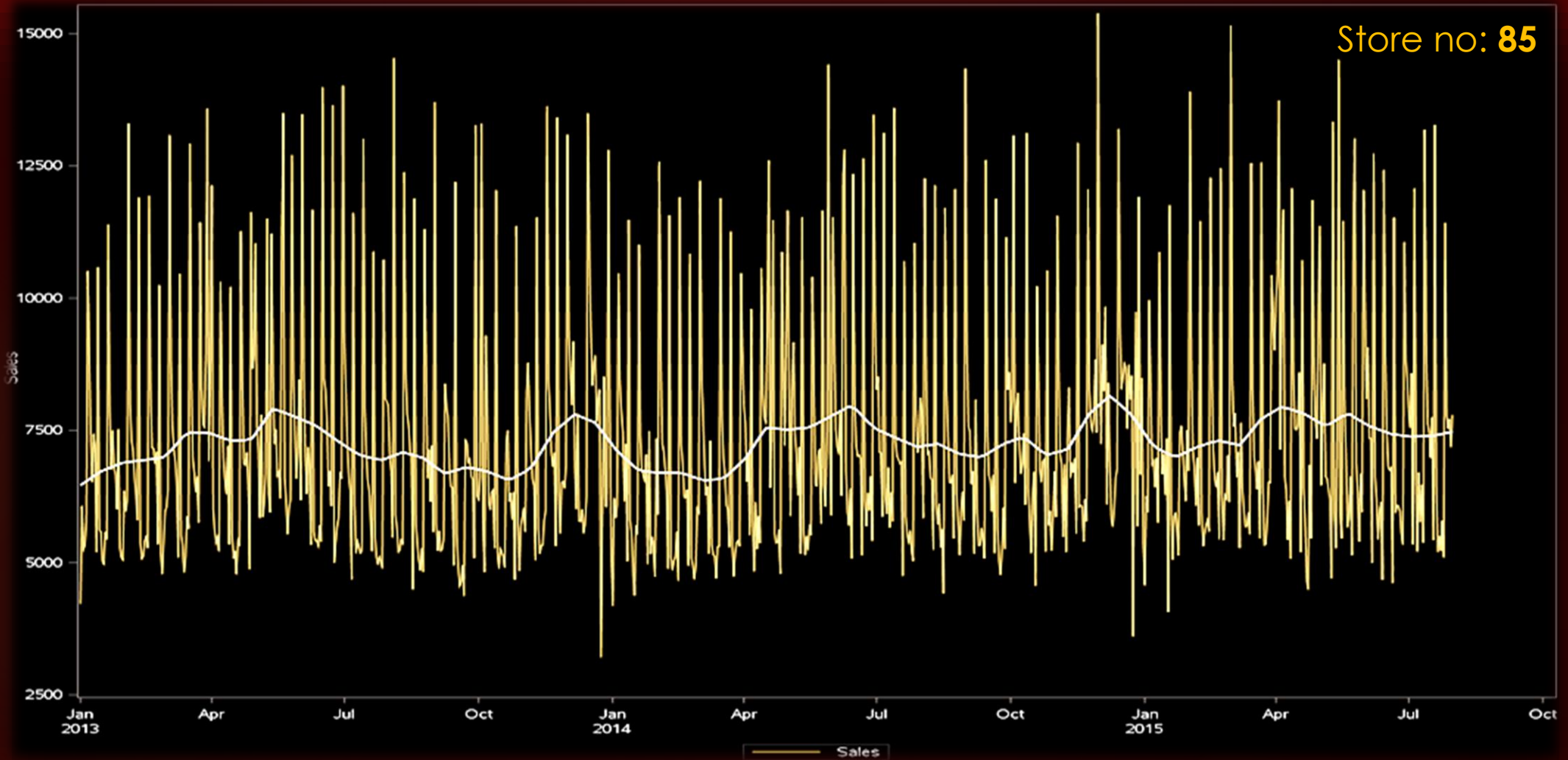
# Exploratory Data Analysis – 2 (Missing Instances)



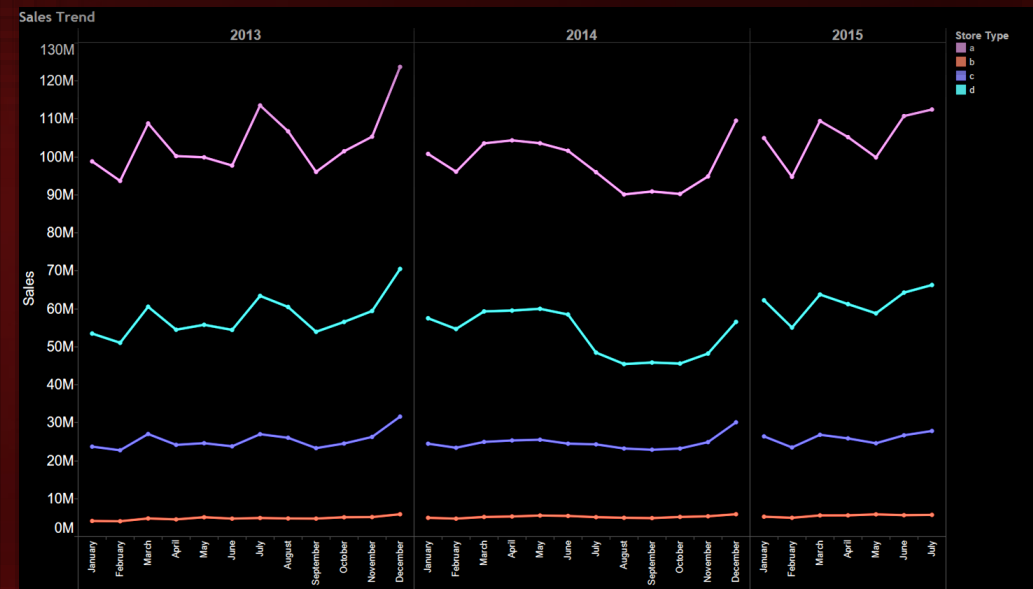
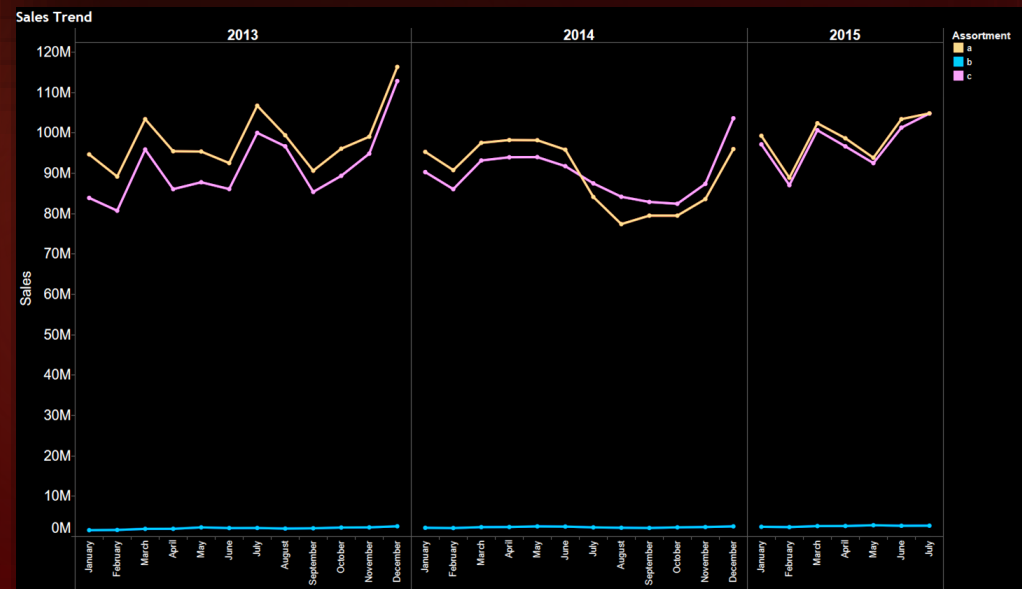
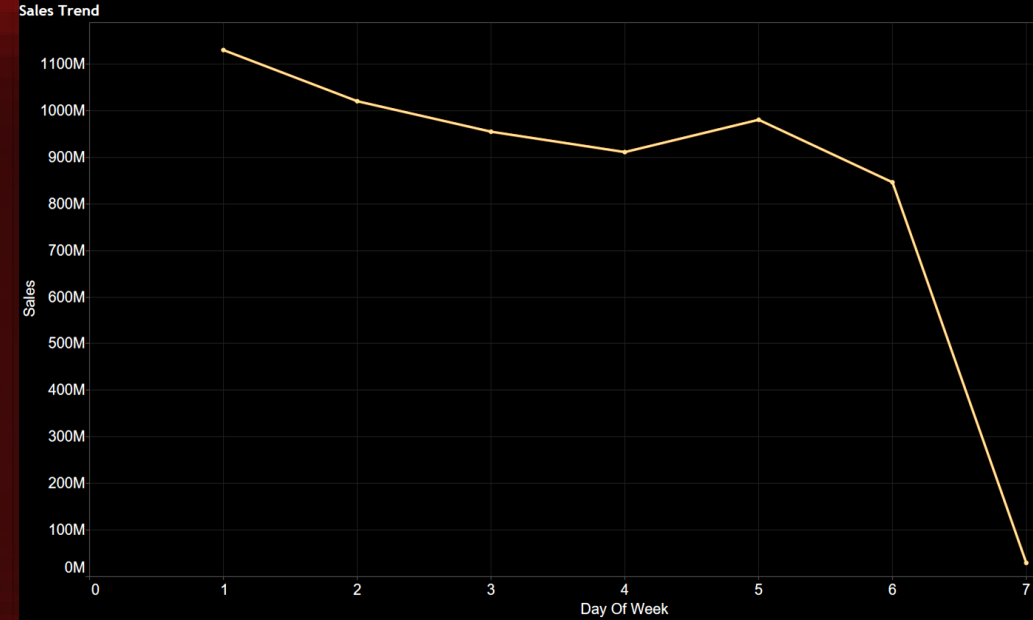
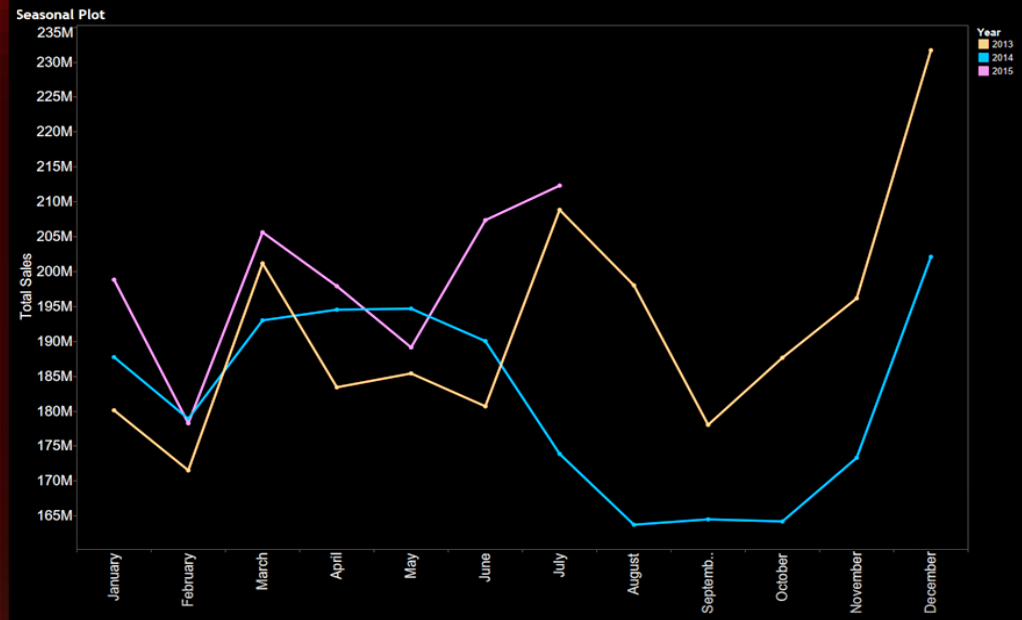


# Exploratory Data Analysis – 3

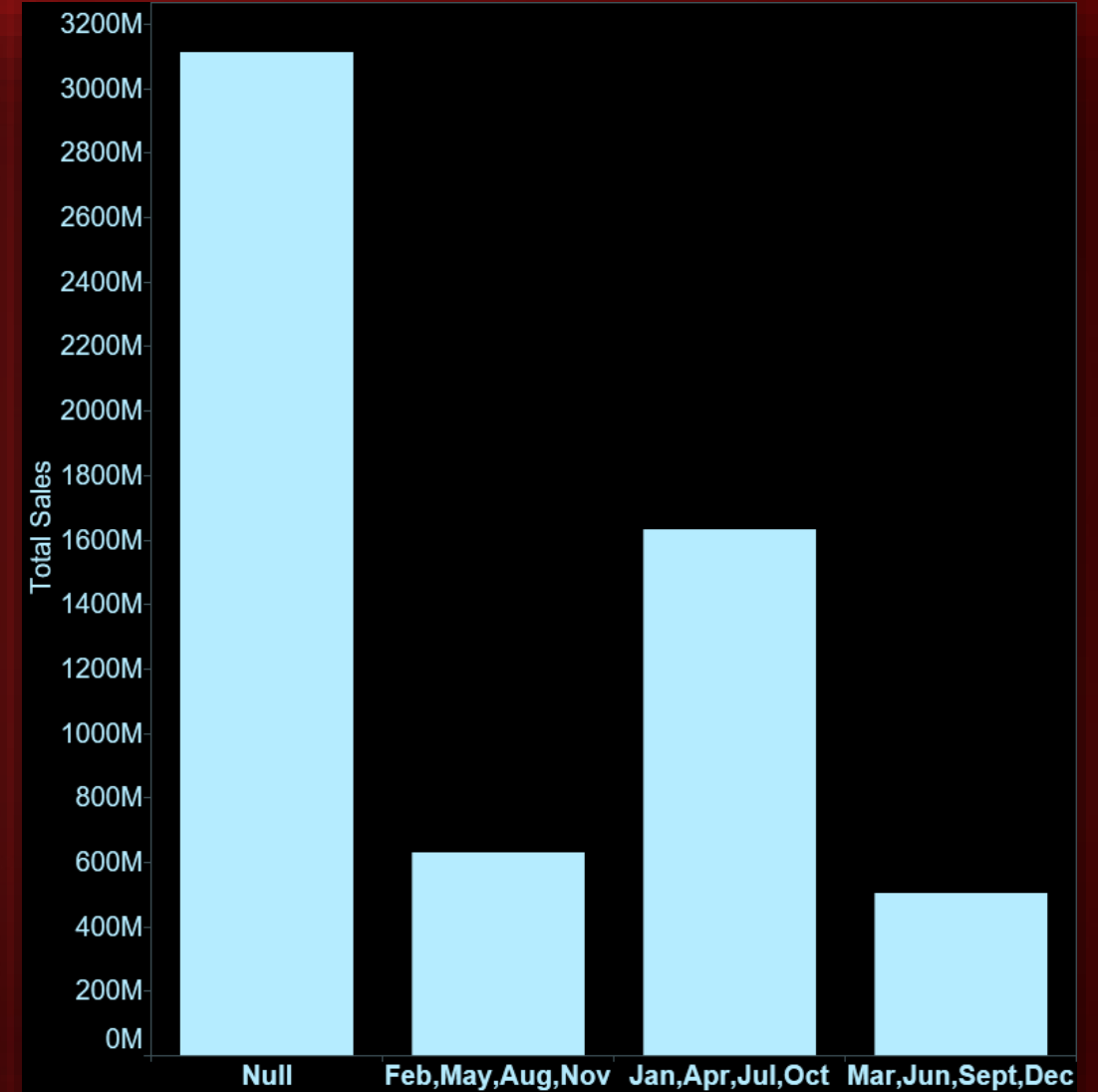
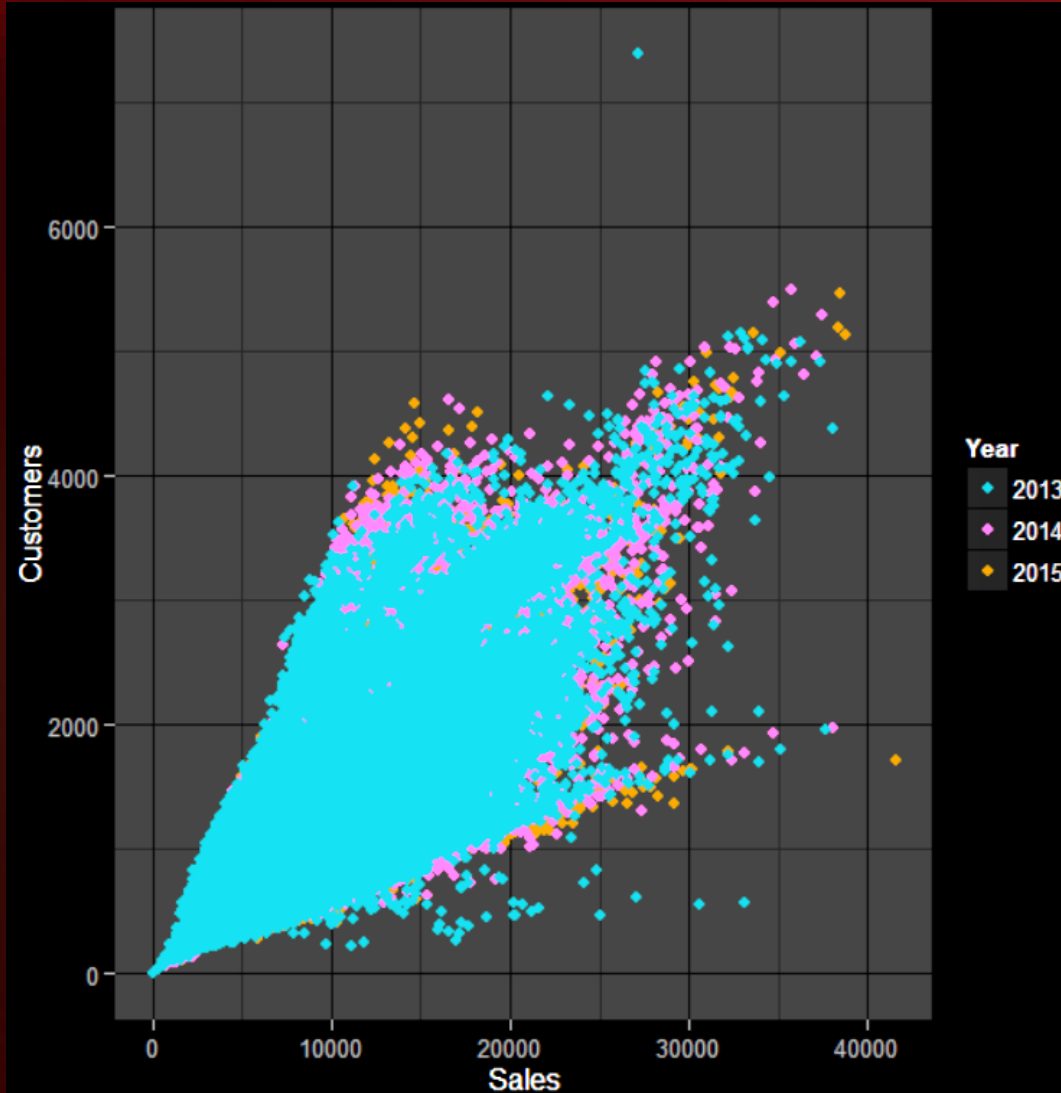
(Stores that were open on Sundays/holidays)



# Exploratory Data Analysis – 4



# Exploratory Data Analysis – 5



# Data Preprocessing - 1

## Step 1:

Merging the CompetitionOpenSinceMonth and CompetitionOpenSinceYear into CompetitionOpenSince

---

## Step 2:

Merging the Promo2SinceWeek and Promo2SinceYear into Promo2Since

---

## Step 3:

Converting the date predictor variables CompetitionOpenSince and Promo2Since to a relative measure  
(i.e. CompetitionOpenSince_Days and Promo2Since_Days respectively)

---

## Step 4:

Creating two subsets of the train dataset based on stores with and without missing sales.

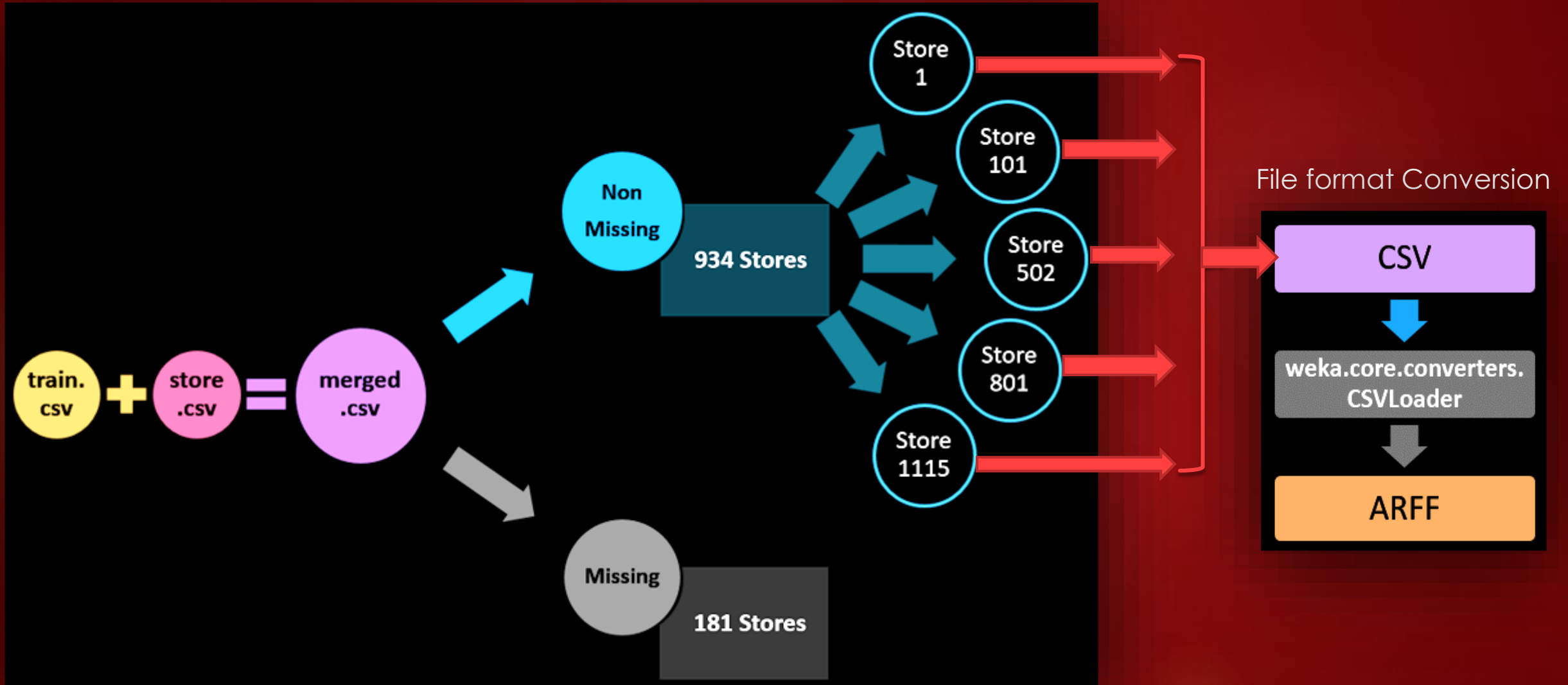
---

## Step 5:

Merging train and store data according by Store Id.

# Data Preprocessing – 2

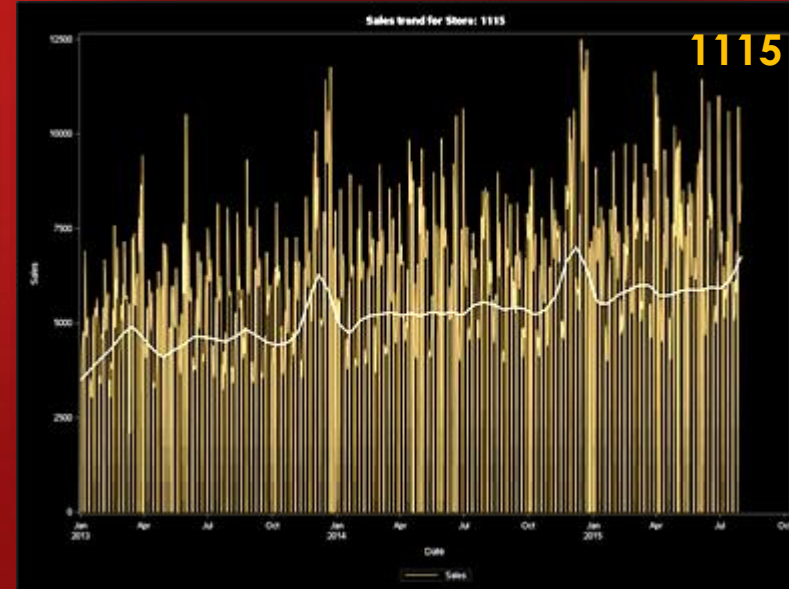
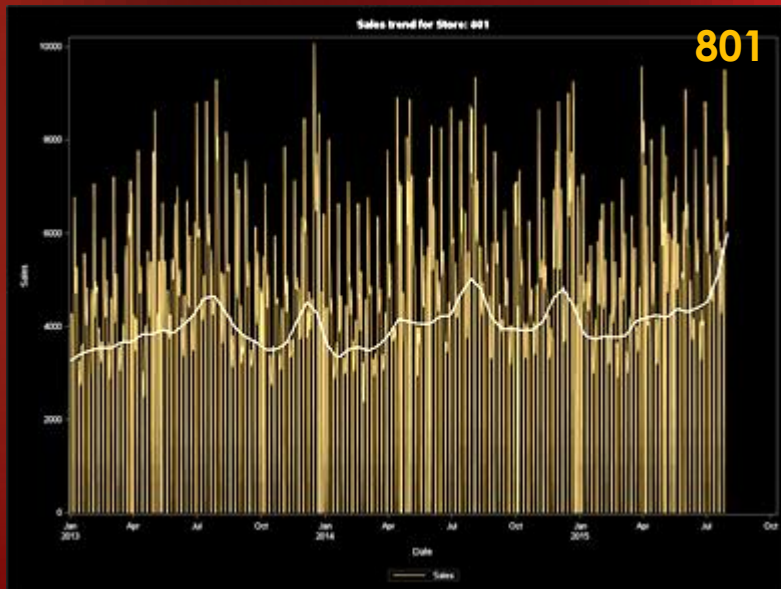
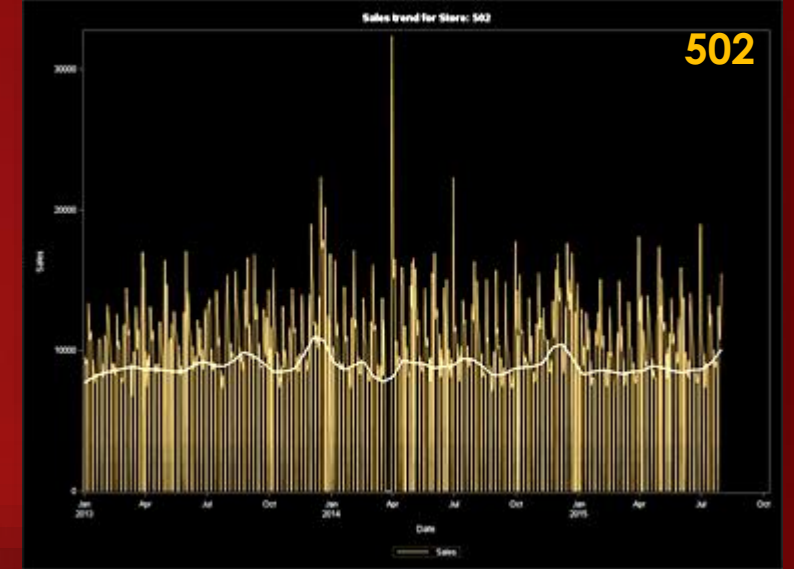
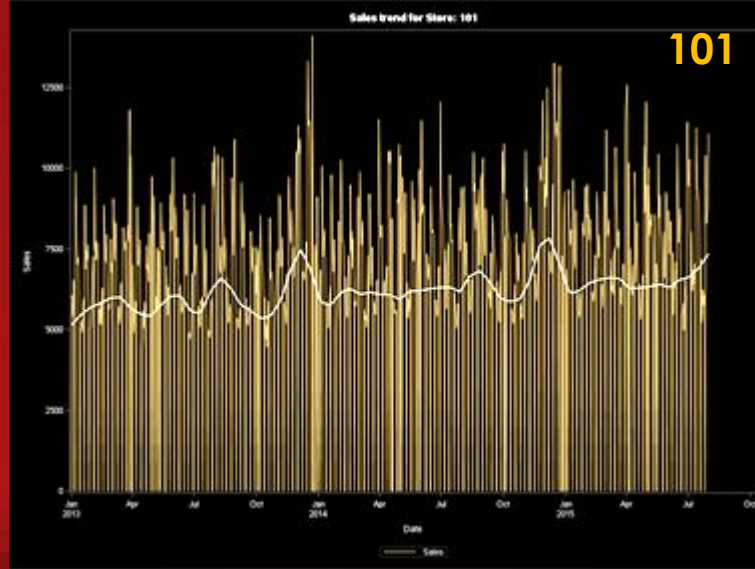
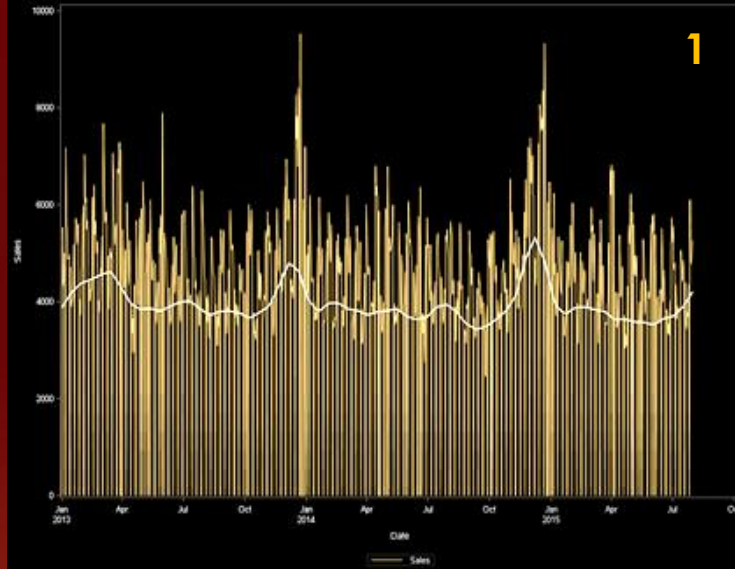
(Visual Representation)





# Training Data (By Store)

(Store Numbers: 1, 101, 502, 801, 1115)



# Training Data (Store 1)

## (Merged Data Table)

View Data: Rossmann_Store1

942 rows

Store	Date	Day Of Week	Open	Customers	State Holiday	School Holiday	Assortment	Store Type	Competition Distance	CompetitionOpenSince Days	Promo	Promo2	Promo Interval	Promo2Since Days	Sales
1	1/1/2013	2	0	0	Null	1	a	c	1,270	2,525	0	0	Null	Null	0
1	1/2/2013	3	1	668	0	1	a	c	1,270	2,525	0	0	Null	Null	5,530
1	1/3/2013	4	1	578	0	1	a	c	1,270	2,525	0	0	Null	Null	4,327
1	1/4/2013	5	1	619	0	1	a	c	1,270	2,525	0	0	Null	Null	4,486
1	1/5/2013	6	1	635	0	1	a	c	1,270	2,525	0	0	Null	Null	4,997
~~~~~															
1	7/25/2015	6	1	500	0	0	a	c	1,270	2,525	0	0	Null	Null	4,364
1	7/26/2015	7	0	0	0	0	a	c	1,270	2,525	0	0	Null	Null	0
1	7/27/2015	1	1	612	0	1	a	c	1,270	2,525	1	0	Null	Null	6,102
1	7/28/2015	2	1	560	0	1	a	c	1,270	2,525	1	0	Null	Null	5,011
1	7/29/2015	3	1	523	0	1	a	c	1,270	2,525	1	0	Null	Null	4,782
1	7/30/2015	4	1	546	0	1	a	c	1,270	2,525	1	0	Null	Null	5,020
1	7/31/2015	5	1	555	0	1	a	c	1,270	2,525	1	0	Null	Null	5,263

1-Jan-2013

13-Jun-2015

14-Jun-2015

31-Jul-2015

1-Aug-2015

17-Sep-2015

Training Data

Holdout Data

Test Data

894 Days

48 Days

48 Days

Data Modeling - 1

(Common Configuration Settings)

Weka Version Used:
v3.7.13

Weka Plugin Used:
timeseriesForecasting
V1.0.16

Basic Configuration:

Number of time units
to forecast: **48**

Time Stamp: **Date**

Periodicity: **Daily**

Target Selection: **Sales**

Overlay Data:

Customers

Open

Promo

StateHoliday

SchoolHoliday

StoreType

Assortment

CompetitionDistance

CompetitionOpenSince_Days

Promo2

PromoInterval

Promo2Since_Days

Evaluation:

MAPE: **Mean absolute
Percentage error**

Evaluate on training:
FALSE

Evaluate on hold out
training: **TRUE**

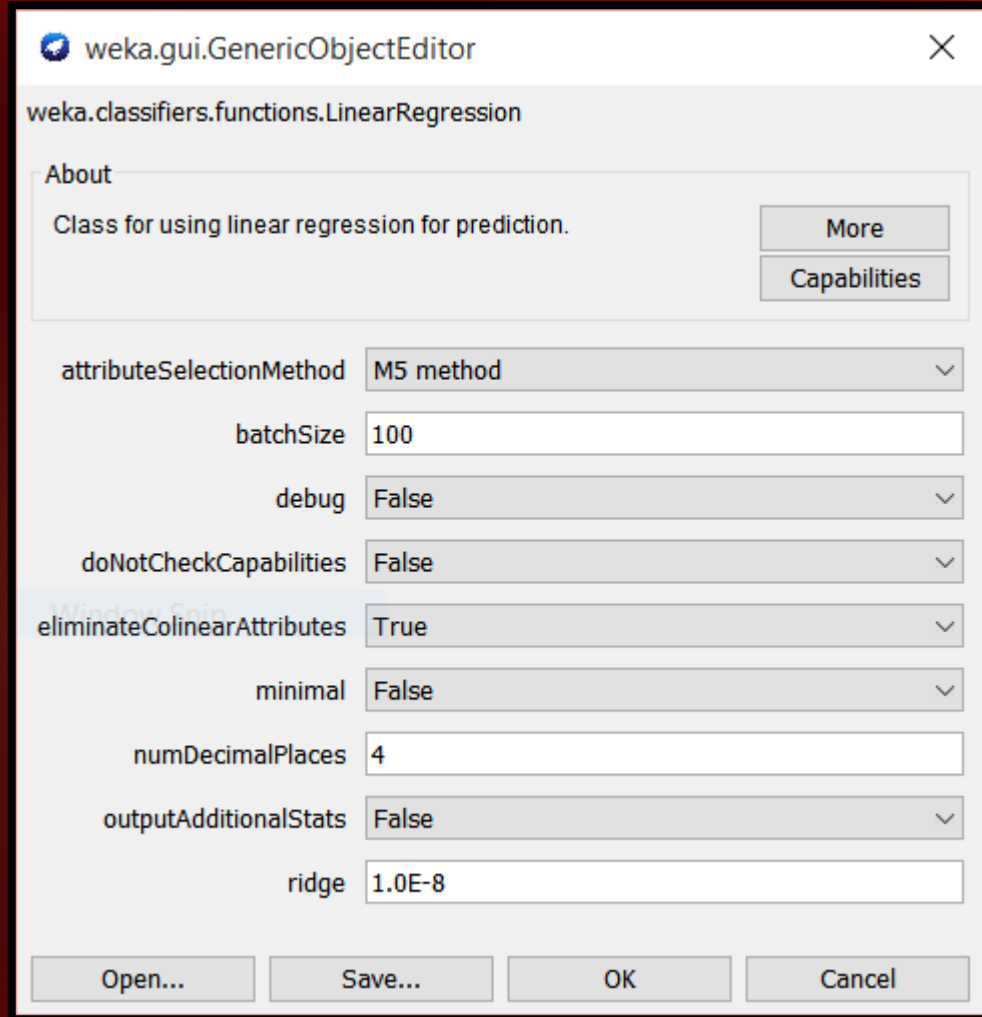
Lag creation:

Minimum: **1**

Maximum: **21**

Data Modeling - 2

(Linear Regression)

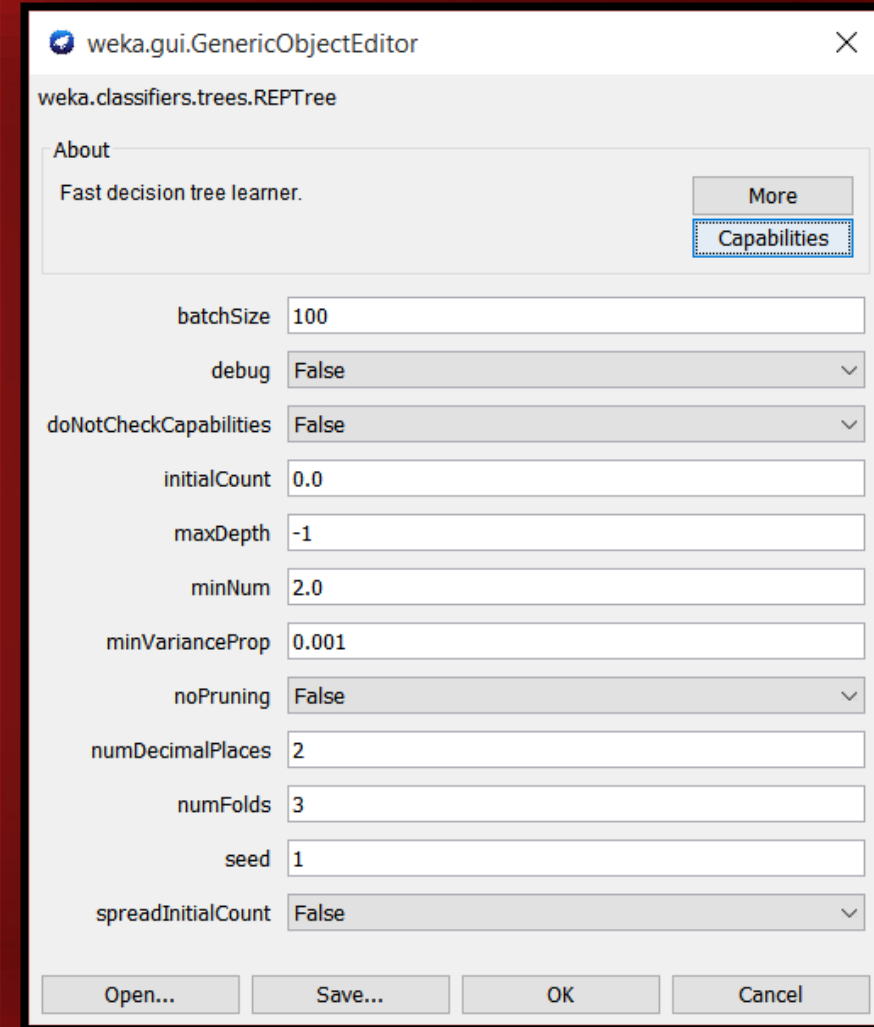


The screenshot shows the 'weka.gui.GenericObjectEditor' window for the 'weka.classifiers.functions.LinearRegression' class. The 'About' tab is active, displaying a description: 'Class for using linear regression for prediction.' Below this, there are two buttons: 'More' and 'Capabilities'. The main area contains several configuration options, each with a label and a value in a text box or dropdown menu:

- attributeSelectionMethod: M5 method (dropdown)
- batchSize: 100 (text box)
- debug: False (dropdown)
- doNotCheckCapabilities: False (dropdown)
- eliminateColinearAttributes: True (dropdown)
- minimal: False (dropdown)
- numDecimalPlaces: 4 (text box)
- outputAdditionalStats: False (dropdown)
- ridge: 1.0E-8 (text box)

At the bottom, there are four buttons: 'Open...', 'Save...', 'OK', and 'Cancel'.

(Regression Tree)



The screenshot shows the 'weka.gui.GenericObjectEditor' window for the 'weka.classifiers.trees.REPTree' class. The 'About' tab is active, displaying a description: 'Fast decision tree learner.' Below this, there are two buttons: 'More' and 'Capabilities'. The main area contains several configuration options, each with a label and a value in a text box or dropdown menu:

- batchSize: 100 (text box)
- debug: False (dropdown)
- doNotCheckCapabilities: False (dropdown)
- initialCount: 0.0 (text box)
- maxDepth: -1 (text box)
- minNum: 2.0 (text box)
- minVarianceProp: 0.001 (text box)
- noPruning: False (dropdown)
- numDecimalPlaces: 2 (text box)
- numFolds: 3 (text box)
- seed: 1 (text box)
- spreadInitialCount: False (dropdown)

At the bottom, there are four buttons: 'Open...', 'Save...', 'OK', and 'Cancel'.

Data Modeling - 3

(Random Forest)

(SMO Regression)

weka.gui.GenericObjectEditor

weka.classifiers.trees.RandomForest

About

Class for constructing a forest of random trees.

More

Capabilities

batchSize 100

breakTiesRandomly False

debug False

doNotCheckCapabilities False

dontCalculateOutOfBagError False

maxDepth 0

numDecimalPlaces 2

numExecutionSlots 1

numFeatures 0

numTrees 100

printTrees False

seed 1

Open... Save... OK Cancel

weka.gui.GenericObjectEditor

weka.classifiers.functions.SMOreg

About

SMOreg implements the support vector machine for regression.

More

Capabilities

batchSize 100

c 1.0

debug False

doNotCheckCapabilities False

filterType Normalize training data

kernel Choose PolyKernel -E 1.0 -C 250007

numDecimalPlaces 2

regOptimizer Choose RegSMOImproved -T 0.001 -V -P 1.0E-12 -L 0.0

Open... Save... OK Cancel

Accuracy Comparison (By Model)

Mean absolute percentage error (For holdout data)				
	Linear Regression	REP Tree	Random Forest	SMO Regression
Store 1	3.8191	4.471	5.2619	4.329
Store 101	6.05	9.6617	9.8814	7.111
Store 502	5.9151	6.1163	7.2902	5.64
Store 801	3.9062	5.2266	8.9934	4.021
Store 1115	3.5507	5.4983	6.805	4.083
MAPE	4.64822	6.19478	7.64638	5.0368
Accuracy	91.35178	93.80522	92.35362	94.9632

Data Modeling – Algorithm (SMOReg)

SMOReg implements the support vector machine for regression.

Polynomial Kernel:

A transformation of the input data in a feature space over polynomials of the original variables that enables them to operate in a higher dimension (i.e. hyperplane)

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^p$$

Learning Algorithm: RegSMOImproved:

Remarkable improvement in efficiency by using two threshold parameters

β_{low} and β_{up} instead of one.

Reference:

Shevade, S., Keerthi, S., Bhattacharyya, C., & Murthy, K. (2000). Improvements to the SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks*, 11(5), 1188-1193.

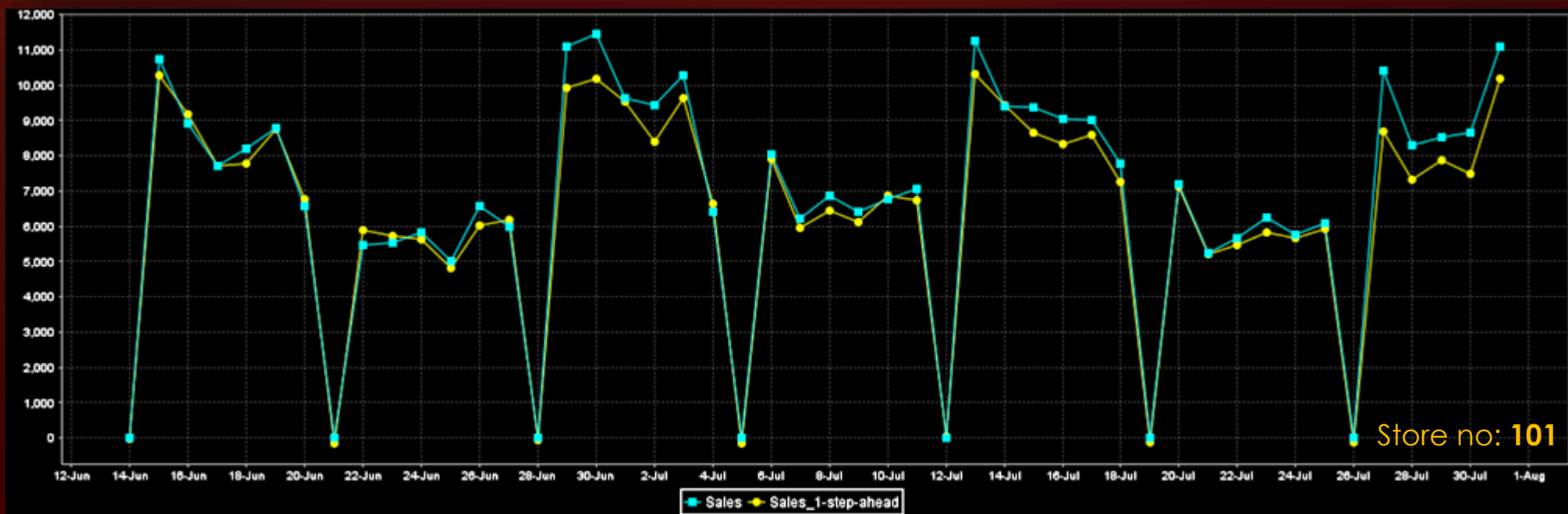
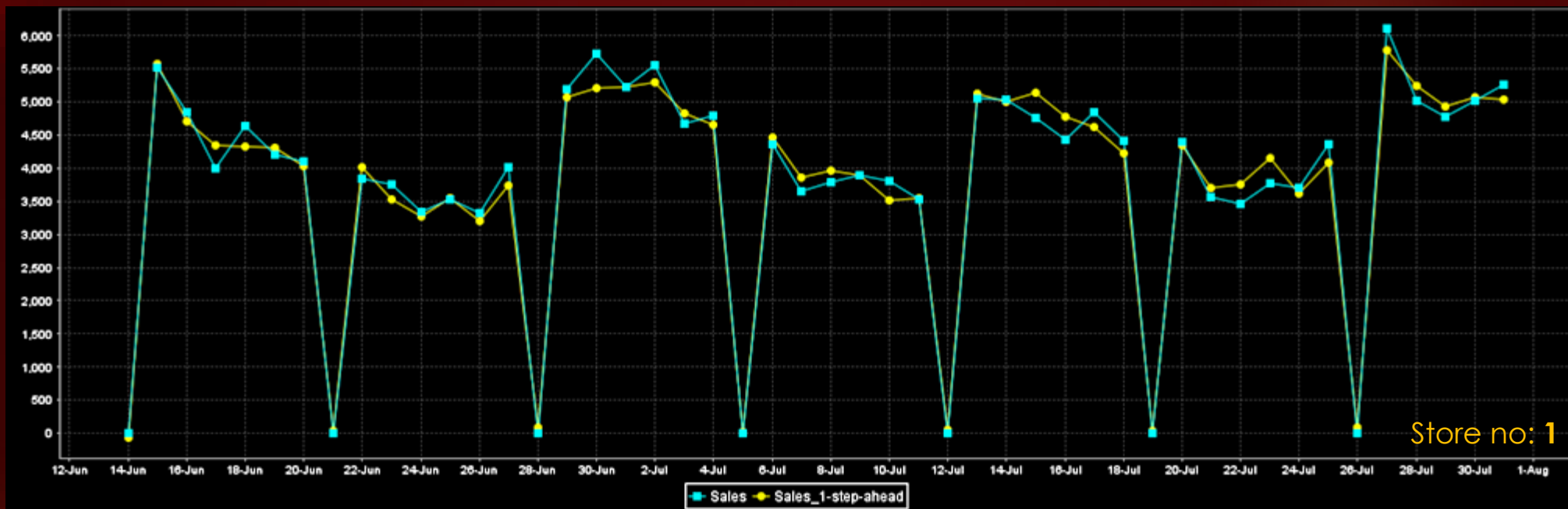
Data Modeling - 5

Sales

- + 1.1259 * (normalized) Customers
- 0.0451 * (normalized) Open=1
- + 0.0444 * (normalized) Promo=1
- 0.0015 * (normalized) StateHoliday=a
- 0.025 * (normalized) StateHoliday=0
- + 0 * (normalized) StateHoliday=b
- + 0.0265 * (normalized) StateHoliday=c
- + 0.0057 * (normalized) SchoolHoliday=1
- + 0 * (normalized) StoreType=c
- + 0 * (normalized) StoreType=a
- + 0 * (normalized) StoreType=d
- + 0 * (normalized) StoreType=b
- + 0 * (normalized) Assortment=a
- + 0 * (normalized) Assortment=c
- + 0 * (normalized) Assortment=b
- + 0 * (normalized) CompetitionDistance
- + 0 * (normalized) CompetitionOpenSince_Days
- + 0 * (normalized) Promo2=1
- + 0 * (normalized) PromoInterval=Jan, Apr, Jul, Oct
- + 0 * (normalized) PromoInterval=Mar, Jun, Sept, Dec
- + 0 * (normalized) PromoInterval=Feb, May, Aug, Nov
- + 0 * (normalized) Promo2Since_Days
- + 0.0201 * (normalized) DayOfWeek=sun
- + 0.017 * (normalized) DayOfWeek=mon
- 0.0018 * (normalized) DayOfWeek=tue

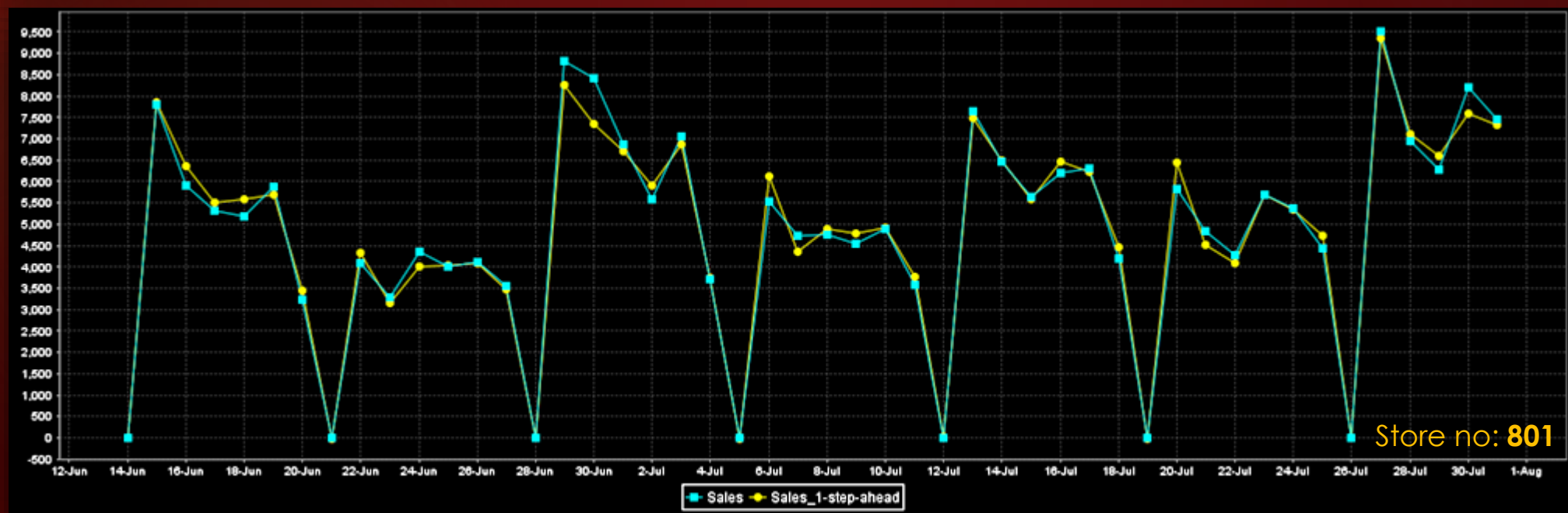
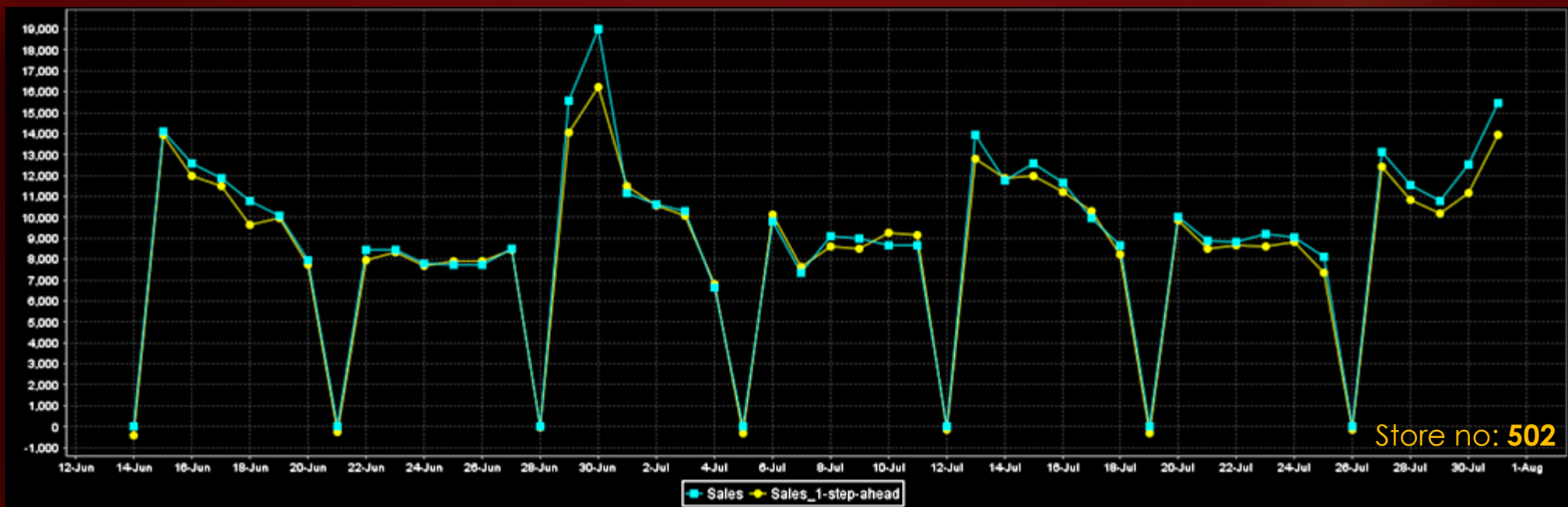
- + 0.0073 * (normalized) DayOfWeek=wed
- 0.007 * (normalized) DayOfWeek=thu
- 0.0292 * (normalized) DayOfWeek=fri
- 0.0063 * (normalized) DayOfWeek=sat
- + 0.0138 * (normalized) Weekend
- + 0.0678 * (normalized) Date-remapped
- 0.0284 * (normalized) Lag_Sales-1
- + 0.011 * (normalized) Lag_Sales-2
- + 0.0075 * (normalized) Lag_Sales-3
- + 0.0126 * (normalized) Lag_Sales-4
- 0.0092 * (normalized) Lag_Sales-5
- + 0.0013 * (normalized) Lag_Sales-6
- 0.033 * (normalized) Lag_Sales-7
- + 0.0258 * (normalized) Lag_Sales-8
- 0.0158 * (normalized) Lag_Sales-9
- + 0.0225 * (normalized) Lag_Sales-10
- 0.0002 * (normalized) Lag_Sales-11
- 0.0283 * (normalized) Lag_Sales-12
- + 0.0049 * (normalized) Lag_Sales-13
- + 0.0181 * (normalized) Lag_Sales-14
- + 0.0164 * (normalized) Lag_Sales-15
- 0.0134 * (normalized) Lag_Sales-16
- 0.0038 * (normalized) Lag_Sales-17
- + 0.0053 * (normalized) Lag_Sales-18
- 0.014 * (normalized) Lag_Sales-19

- 0.0094 * (normalized) Lag_Sales-20
- 0.0187 * (normalized) Lag_Sales-21
- 0.1007 * (normalized) Date-remapped^2
- + 0.0664 * (normalized) Date-remapped^3
- + 0.0414 * (normalized) Date-remapped*Lag_Sales-1
- 0.0039 * (normalized) Date-remapped*Lag_Sales-2
- 0.014 * (normalized) Date-remapped*Lag_Sales-3
- 0.0183 * (normalized) Date-remapped*Lag_Sales-4
- 0.0116 * (normalized) Date-remapped*Lag_Sales-5
- 0.0032 * (normalized) Date-remapped*Lag_Sales-6
- + 0.0162 * (normalized) Date-remapped*Lag_Sales-7
- 0.0245 * (normalized) Date-remapped*Lag_Sales-8
- + 0.0162 * (normalized) Date-remapped*Lag_Sales-9
- 0.0046 * (normalized) Date-remapped*Lag_Sales-10
- 0.0128 * (normalized) Date-remapped*Lag_Sales-11
- 0.0001 * (normalized) Date-remapped*Lag_Sales-12
- + 0.0004 * (normalized) Date-remapped*Lag_Sales-13
- + 0.015 * (normalized) Date-remapped*Lag_Sales-14
- 0.0147 * (normalized) Date-remapped*Lag_Sales-15
- 0.0011 * (normalized) Date-remapped*Lag_Sales-16
- 0.0044 * (normalized) Date-remapped*Lag_Sales-17
- 0.0075 * (normalized) Date-remapped*Lag_Sales-18
- + 0.0091 * (normalized) Date-remapped*Lag_Sales-19
- + 0.0075 * (normalized) Date-remapped*Lag_Sales-20
- 0.0008 * (normalized) Date-remapped*Lag_Sales-21
- 0.0059

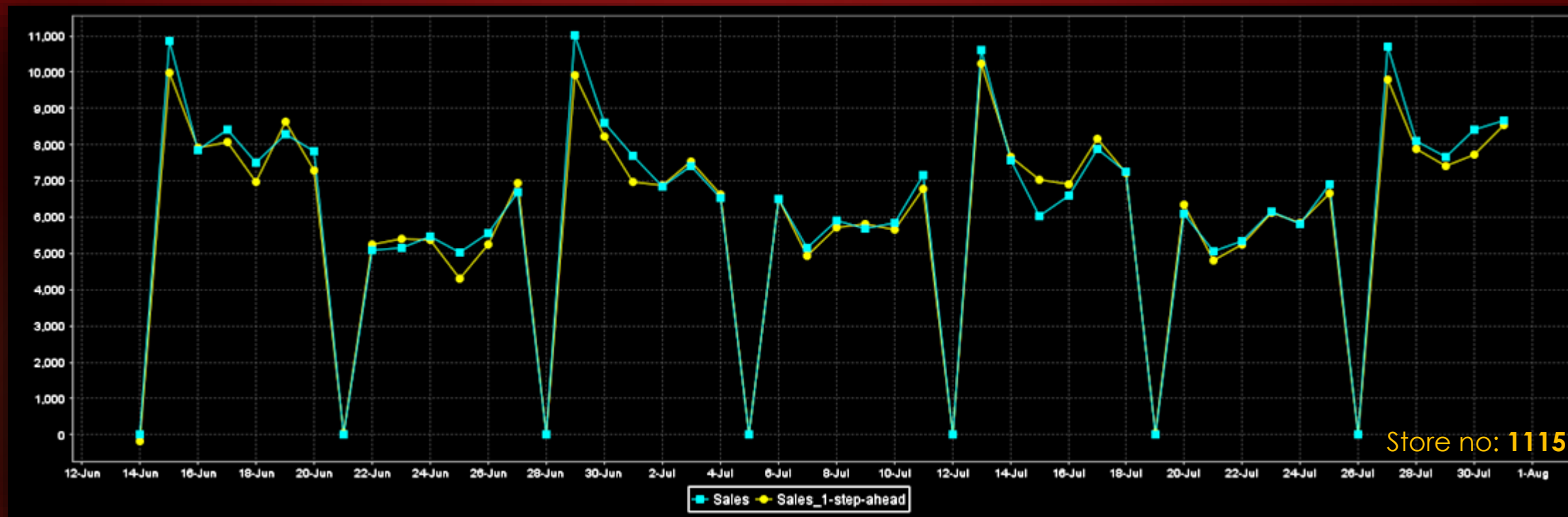


Actual v/s Predicted Sales - 1

Actual v/s Predicted Sales - 2



Actual v/s Predicted Sales - 3



So What's Next?

1. Imputing gaps in sales for 181 stores in the training data followed by training model on these stores to forecast sales and finally evaluating the model on the holdout dataset.
2. Use SAS to create test and train CSV files for each store and then converting those csv files to ARFF in java using the Weka API.
3. Use the timeseries API provided by [Pentaho](#) to build models for each store and then predict sales from 08/01/2015 to 09/17/2015 on the test data using the model built on each store.
4. Forecast sales for all stores and upload the sample_submission.csv on Kaggle.
5. Experiment with more data preprocessing like log transformation, de-seasonalization, scaling, principal component analysis etc. Customers is a very important feature used in training but customer information is missing from test data.

Thank You