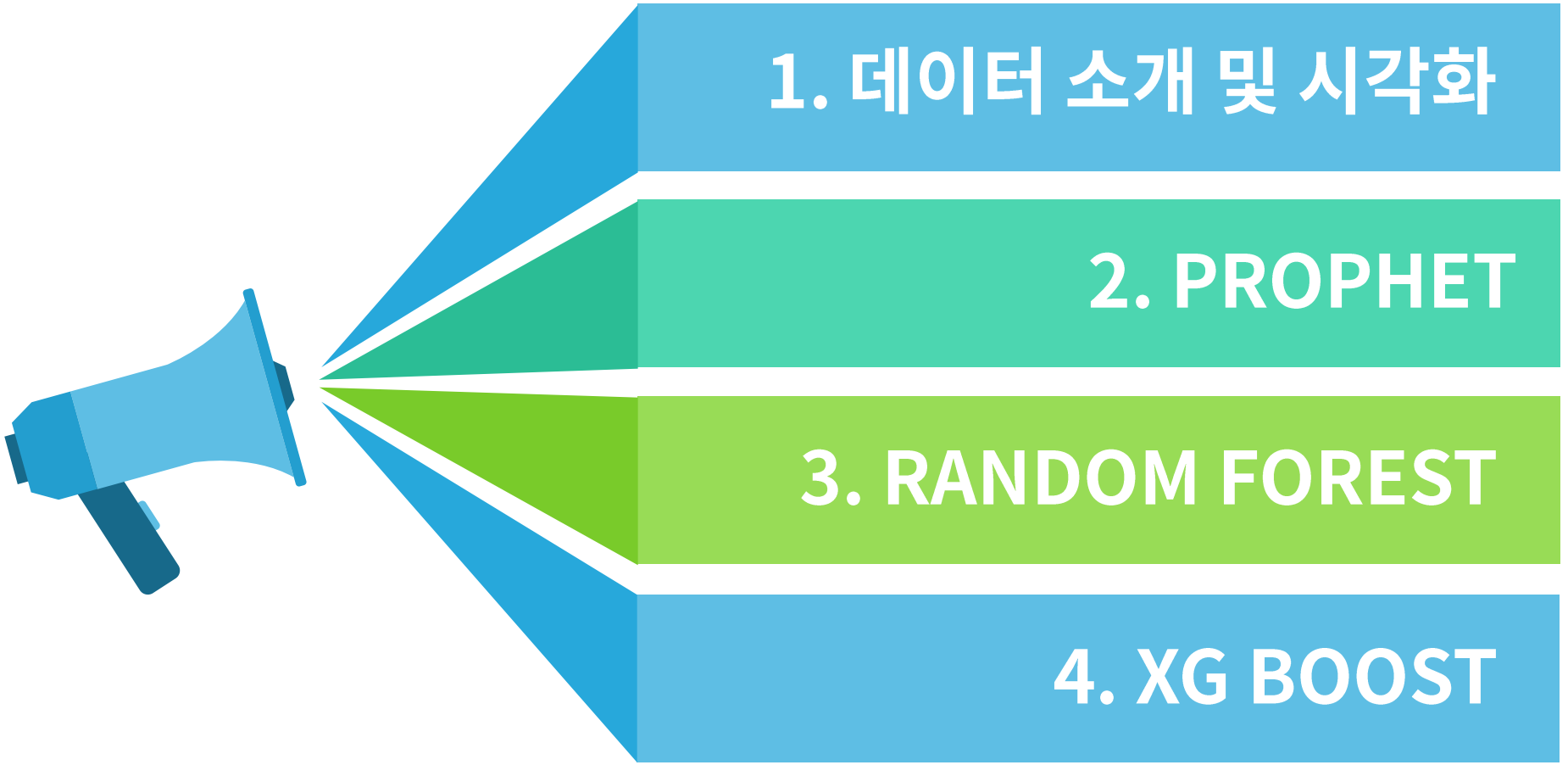
Drugstore Data

1팀
안주영 정재훈 유현조
구동현 강지원 이지윤

INDEX



(1) 데이터 소개 및 시각화

Purpose of Analysis

약국 회사들의 재고 관리 효율성 증대를 위한

일별 **판매량** 예측

(1) 데이터 소개 및 시각화

Table 1 Store.csv (1115 obs)

No.	변수 이름	변수 설명	비고
1	Store	가게를 구별해주는 Key	String
2	StoreType	가게 타입 구분 (a/b/c/d)	Categorical
3	Assortment	가게 종류 구분 (basic/extra/extended)	Categorical
4	CompetitionDistance	경쟁사와의 거리	Numeric
5	CompetitionOpenSinceMonth	경쟁사의 오픈 날짜	Numeric
6	CompetitionOpenSinceYear	경쟁사의 오픈 날짜	Numeric
7	Promo2	가게의 프로모션 유무	Categorical
8	Promo2SinceWeek	가게의 프로모션 시작 날짜	Numeric
9	Promo2SinceYear	가게의 프로모션 시작 날짜	Numeric
10	PromoInterval	가게의 프로모션 시작 날짜 (Jan,Apr,Jul,Oct/Feb,May,Aug,Nov/Mar,Jun,Sept,Dec)	Categorical

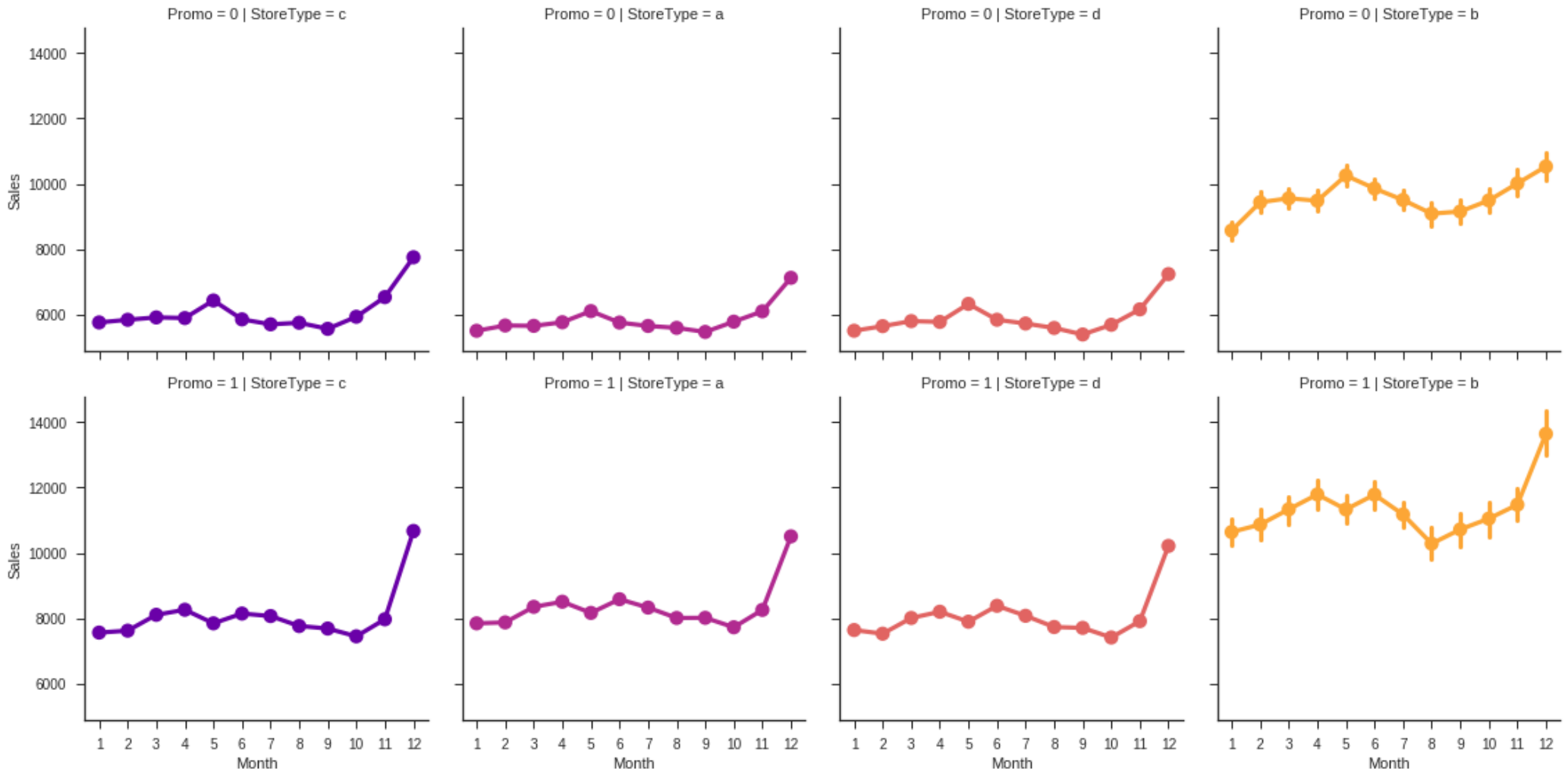
(1) 데이터 소개 및 시각화

Table 2 Train.csv (949194 obs)

No.	변수 이름	변수 설명	비고
1	Store	가게를 구별해주는 Key	String
2	Sales	매장 일별 판매량	Numeric
3	Customers	매장 일별 방문자 수	Numeric
4	DayOfWeek	요일	Categorical
5	Date	날짜 (2013-01-01~2015-05-31)	Date
6	Open	일별 오픈 유무	Categorical
7	StateHoliday	국가 휴무일 (Public holiday / Easter holiday / Christmas / None)	Categorical
8	SchoolHoliday	학교 휴무일 유무	Categorical
9	Promo	일별 프로모션 유무	Categorical

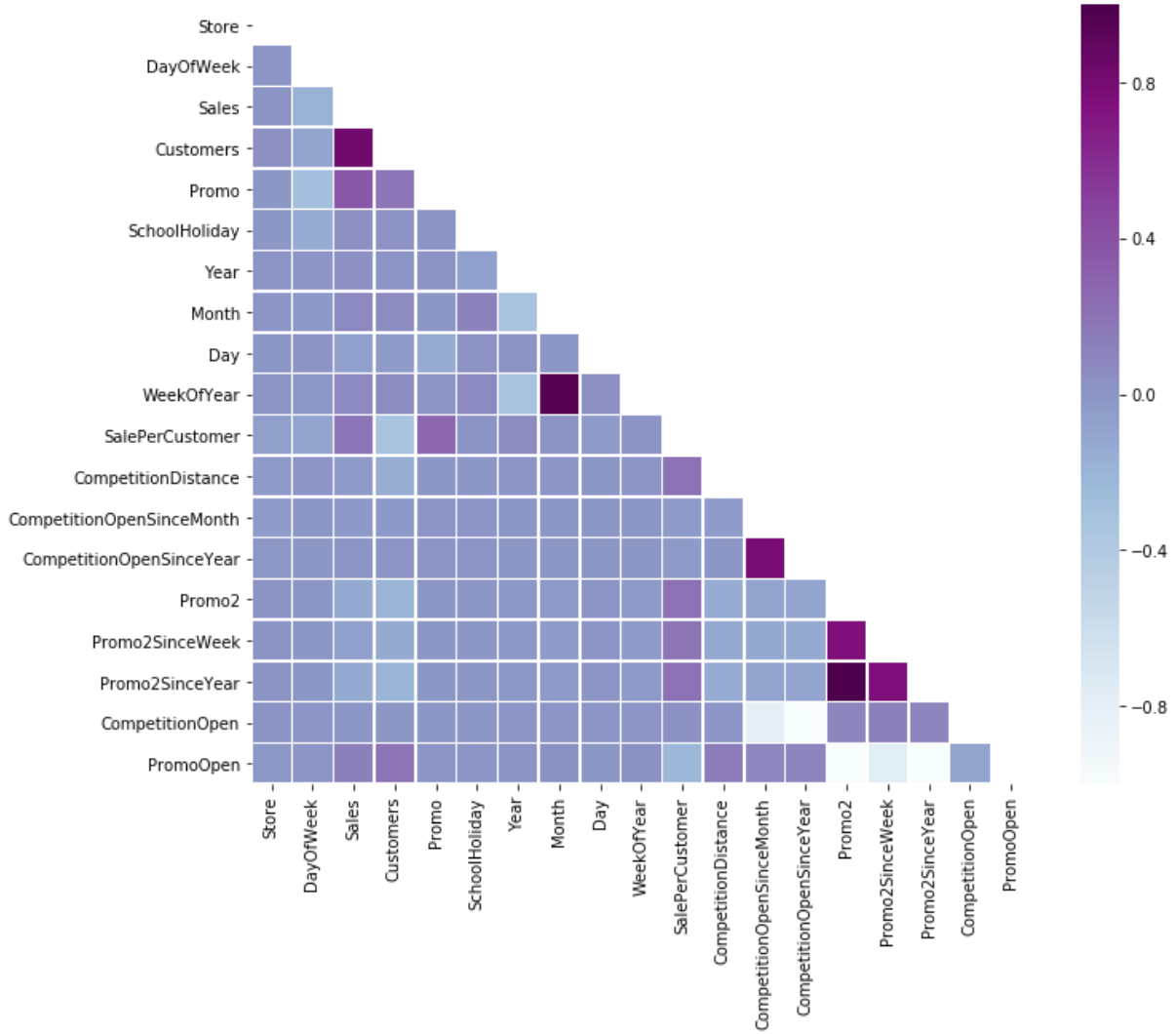
Our Goal !

(1) 데이터 소개 및 시각화



월별 판매 추이 (Store type / Promotion)

(1) 데이터 소개 및 시각화



변수 간의 상관관계

- 판매량과 소비자 수 → 관계 있음
- 일시적 promotion → 소비자 수 상승
- 정기적인 promotion →

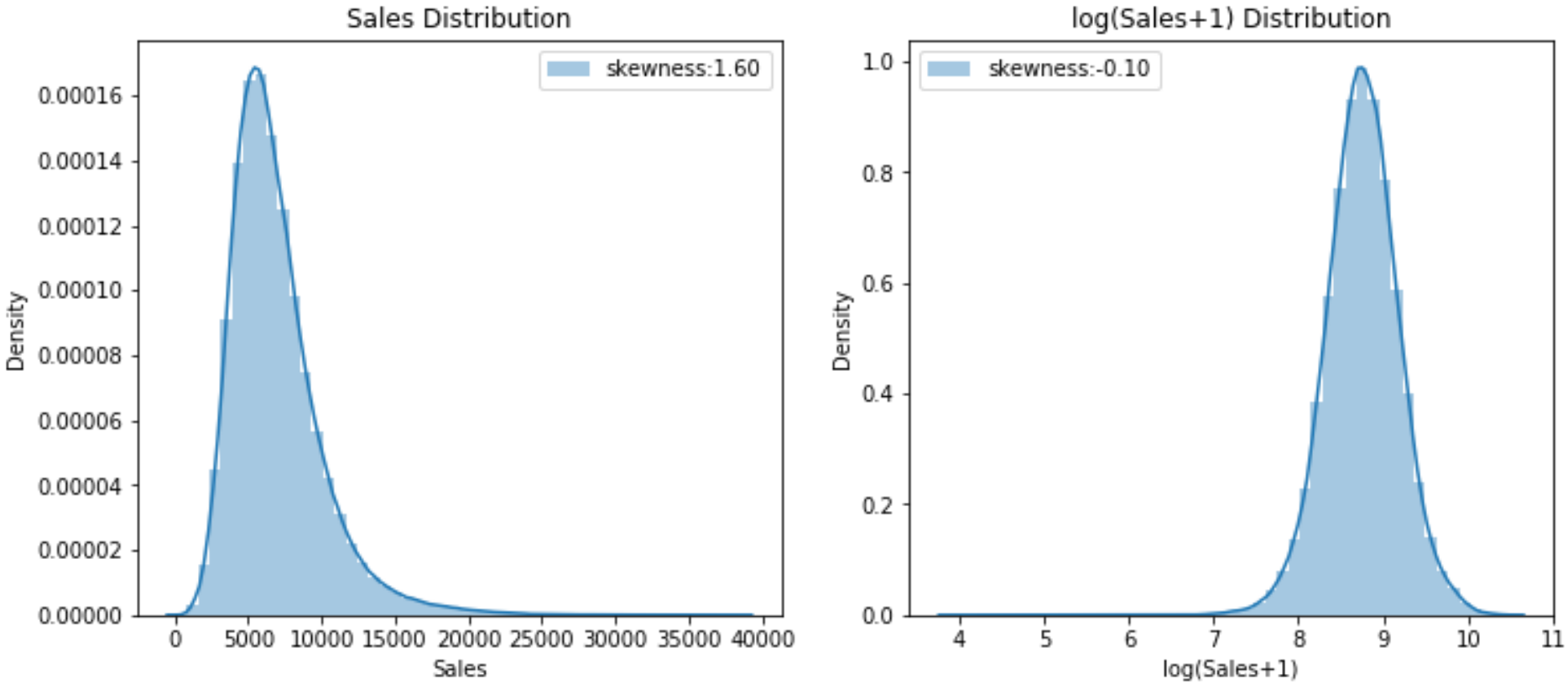
  판매량이나 소비자 수 변화 없음 or 낮게 함

결측치 처리

칼럼명	결측치 row count
Store	0
StoreType	0
Assortment	0
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2	0
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	544

(1) 데이터 소개 및 시각화

전처리



Sales 의 분포를 log변환을 통해 skewness를 줄여줌

(1) 데이터 소개 및 시각화

전처리

train과 test 테이블을 store를 기준으로 store테이블과 병합

(1) 데이터 소개 및 시각화

Date
2013/01/01
2013/01/02
2013/01/03
⋮
2014/12/30
2014/12/31

전처리

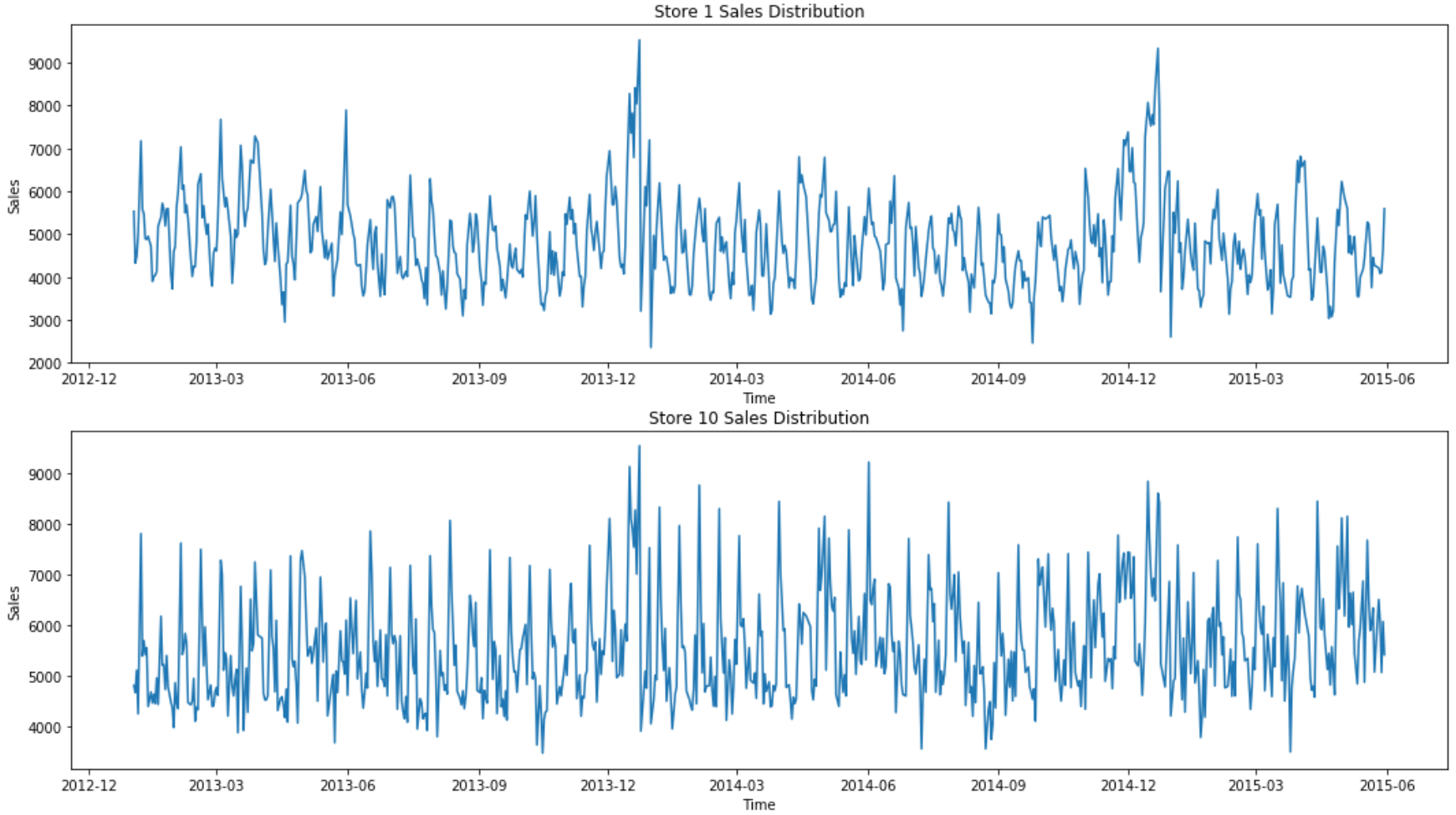
1. Date값을 pd.to_numeric을 이용해 변환시키면
1970년 1월1일 이후 몇 마이크로초가 지났는지를 기록하는 방법

2. 연, 월, 일, 요일 등의 필요한 정보들을
Date칼럼으로부터 추출해 내는 방법

→ 요일이나 월 등의 변수들이 매출에 영향을 주는 것을
고려해 두번째 방법을 사용하기로 함

(1) 데이터 소개 및 시각화

전처리



(1) 데이터 소개 및 시각화

전처리

- **State Holiday** — Test 데이터셋 내에서 모든 값이 0을 갖는 constant column
- **Competition Open** — 경쟁업체가 Open한 날짜 이후로 얼마만큼의 시간이 경과했는지
- **Promo Open** — 정기프로모션을 실행한지 얼마나 시간이 흘렀는지
- **Is Promo Month** — 현재 프로모션을 진행하는 월인지

(2) PROPHET

(2) PROPHET

SARIMA 모형 (Seasonal Autoregressive Integrated Moving Average)

→ 과거의 시계열 데이터가 현재의 데이터에 주는 영향에 주목하여서 어느 시점까지의 과거 자료를 이용
가중치는 어느 정도로 해야하는 지를 고려함 (Autoregressive & Moving Average)

→ 시계열 자료에서 계절성의 효과를 고려함 (Seasonal)
Ex) 매년 크리스마스 시즌에 Drugstore의 Sales가 늘어나는 경향

→ 비정상 시계열은 정상화하여 모형을 만듬 (Integrated)

→ R에서 'auto.arima'를 통해 자동으로 파악가능!

(2) PROPHET

Facebook Prophet?

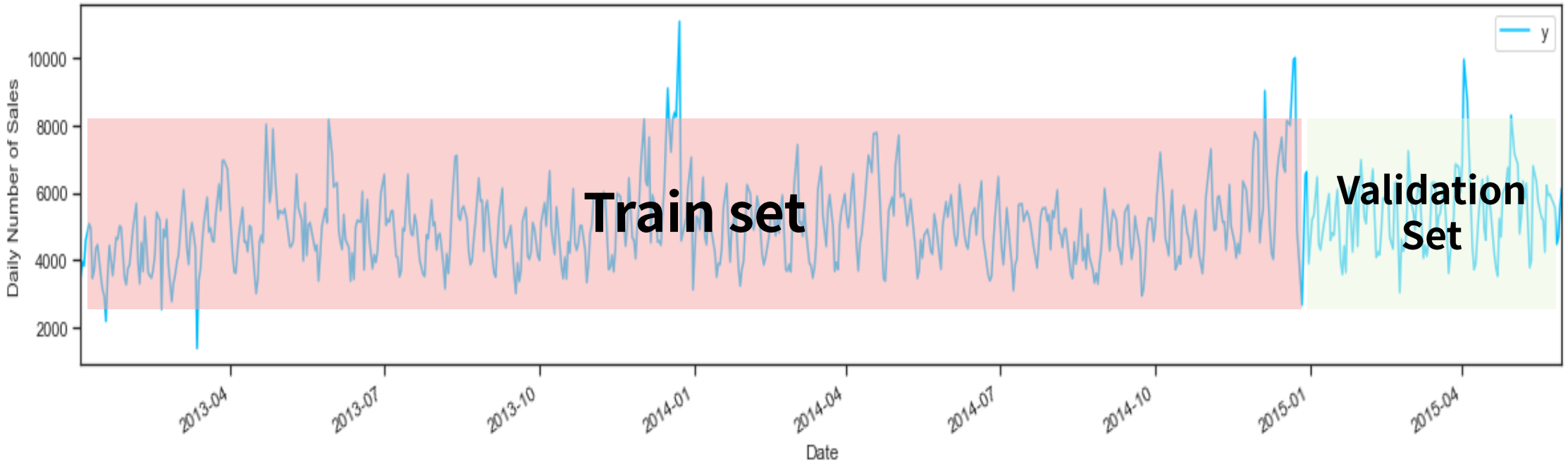
→ SARIMA의 경우보다 더 복잡한 알고리즘 적용(푸리에 급수 등…)

→ 휴일, 결측치 등의 요인들도 잘 고려
시계열 간의 관계성 파악보다는 예측에 초점을 두어 SARIMA보다 예측력이 좋다고함
(Sean J. Taylor, Benjamin Letham(2017): '**Forecasting at scale**')

→ 'pip install fbprophet'으로 설치가능'

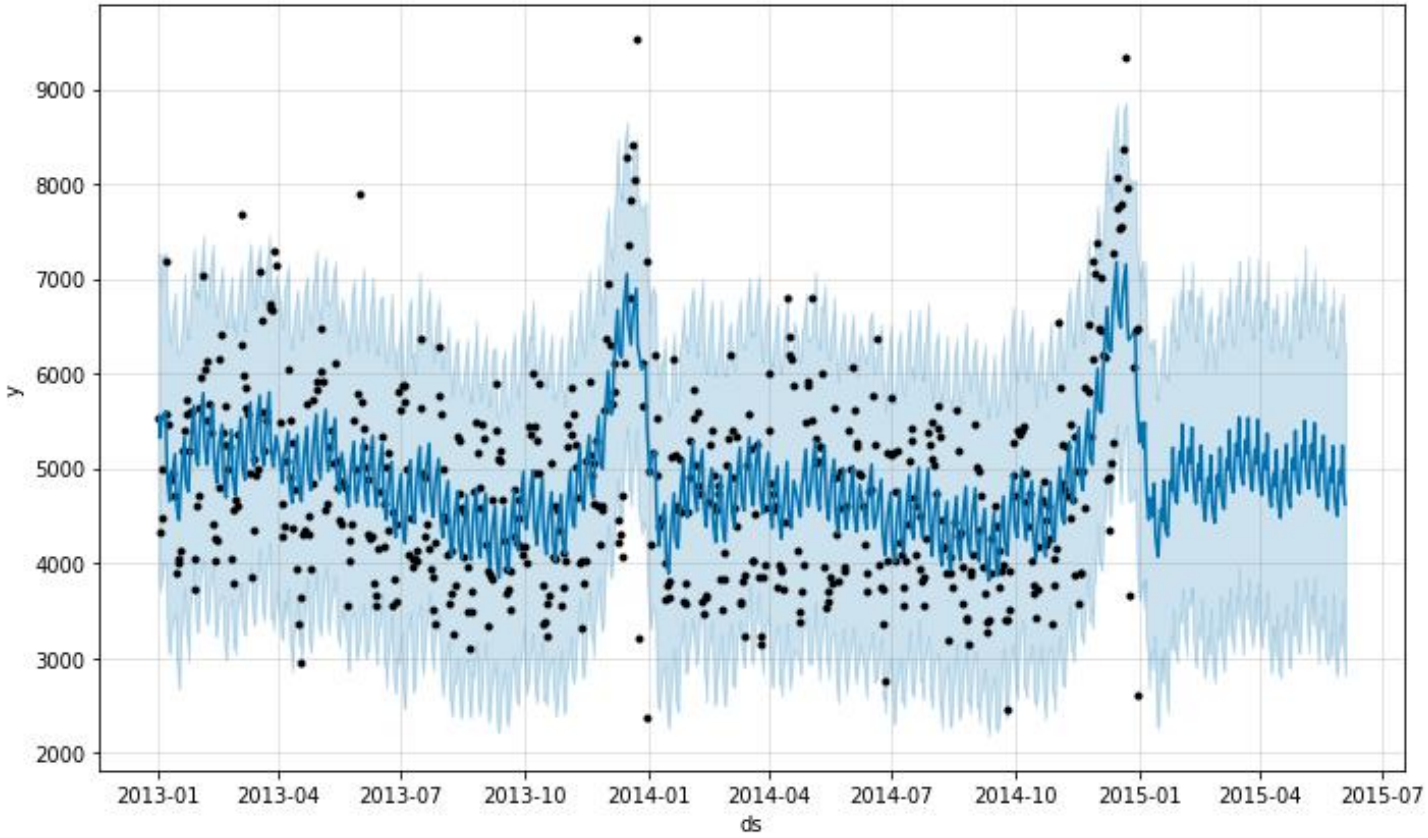
(2) PROPHET

모델 평가



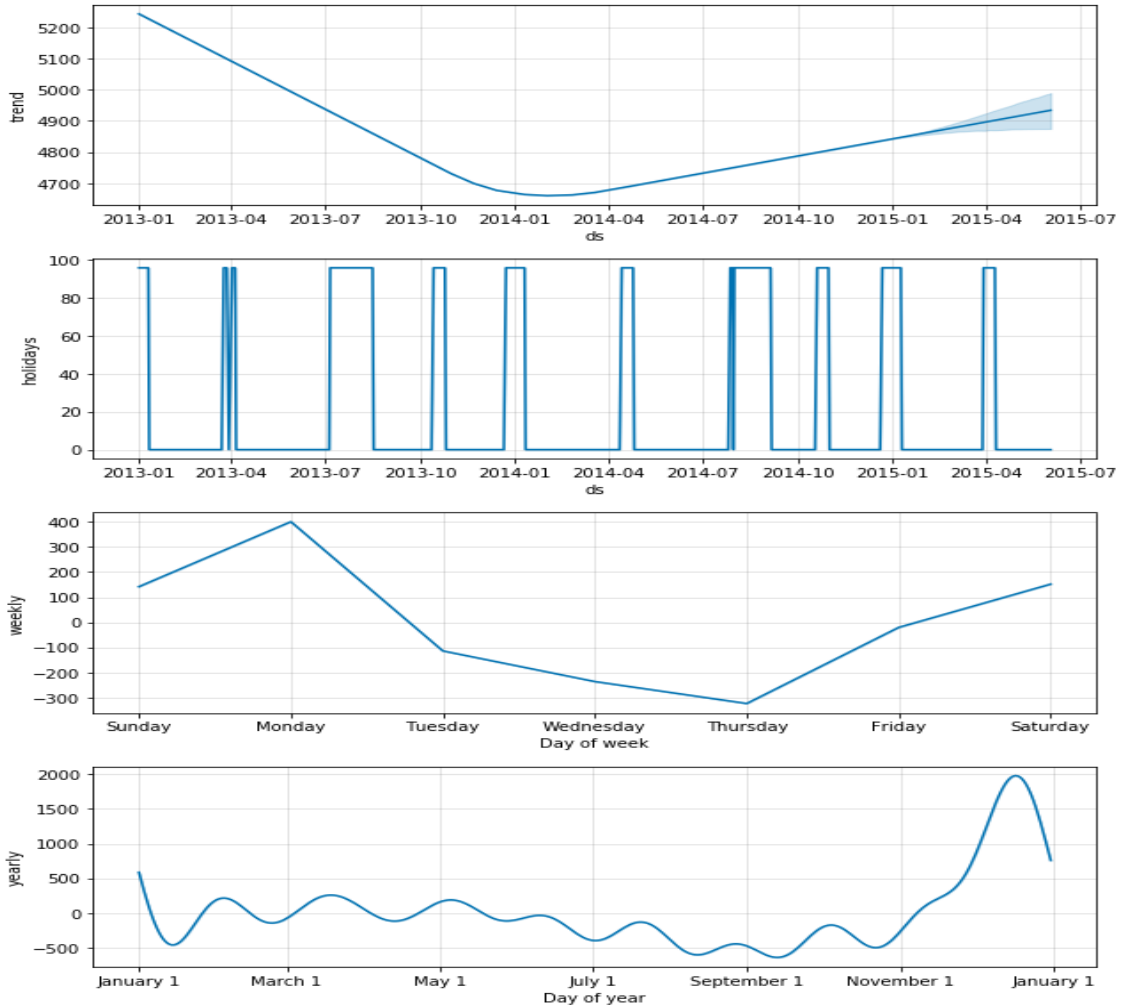
Train set (2013/01~2014/12)을 이용하여 적합 후,
Validation set (2015/01~2015/05)을 이용해 자체적으로 RMSPE를 구함

Store 337(Type D) Forecast (단변량)



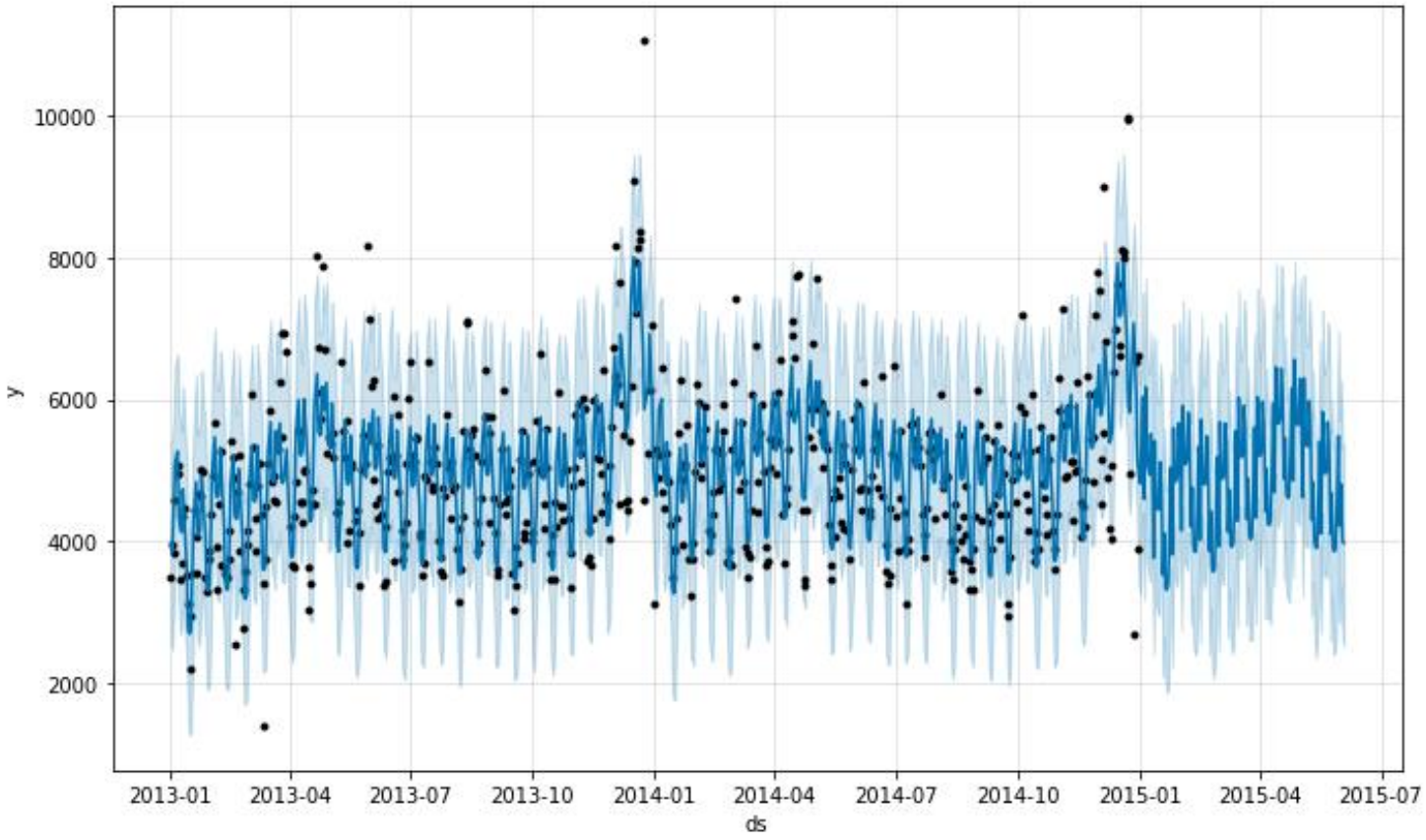
RMSPE: 0.1955

(2) PROPHET



각 변수들이 Sales에 미치는 영향

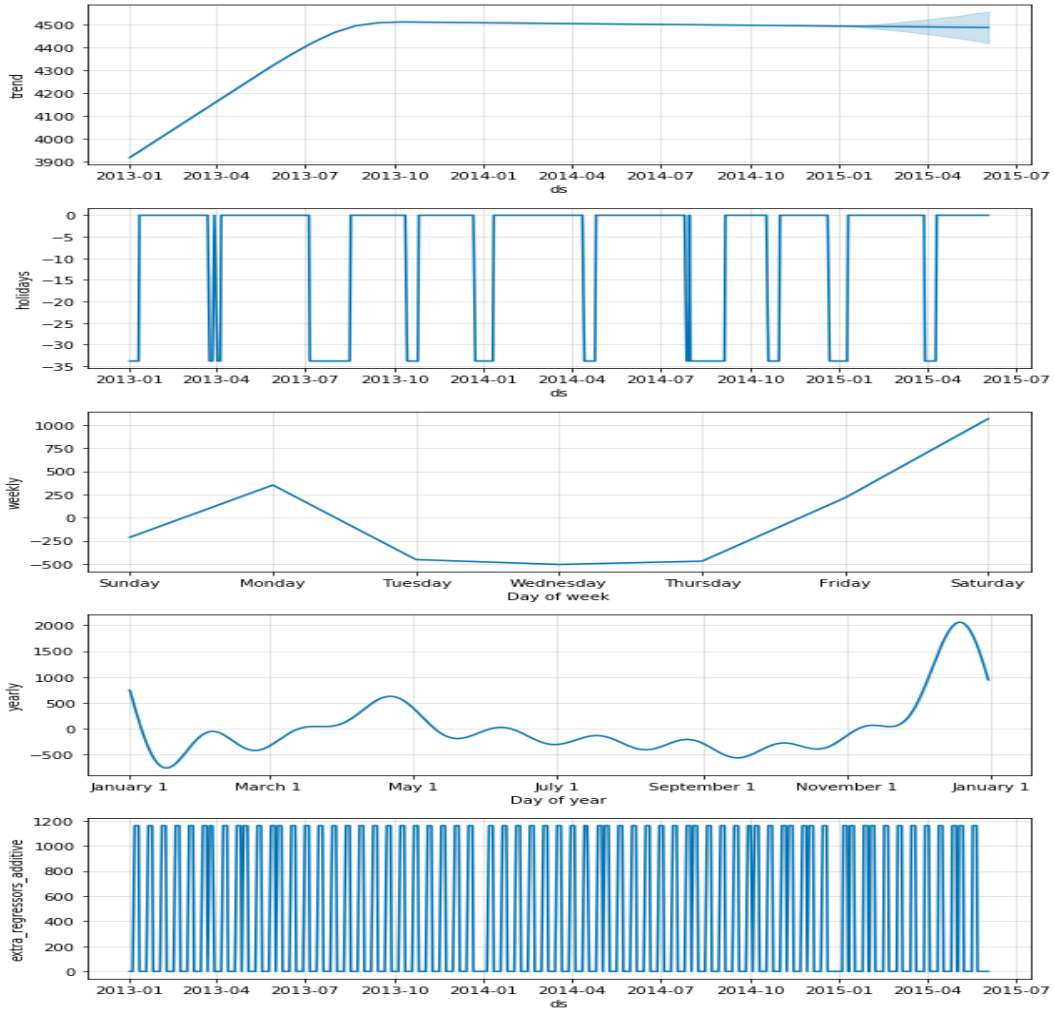
Store 337(Type D) Forecast (다변량)



예측 값의 분산 또한 줄어듦

RMSPE: 0.1603

(2) PROPHET



각 변수들이 Sales에 미치는 영향

(2) PROPHET

BEFORE

RMSPE:
약 0.2263

▲ 최대: 0.7305(store 837)

▼ 최소: 0.0928(store 498)

AFTER

RMSPE:
약 0.1724

▲ 최대: 0.7061(store 837)

▼ 최소: 0.0763(store 104)

(2) PROPHET

Rmspe값을 더 낮출 수 있을까?

Prophet은 다변량 분석 방법이지만, for 구문을 통해 가게 각각의 데이터만을 사용하여 예측을 진행한다. 따라서, 예측을 사용할 때 다른 가게의 판매량 추이 등은 고려하지 못한다(ARIMAX(전이함수모형) 등의 시계열 모델도 마찬 가지이다.)

예측에 있어서 자기 가게의 다양한 변수는 물론 다른 가게의 데이터까지도 고려하는 Random Forest와 XGBOOST등의 방법을 사용할 필요가 있다!

(3) RANDOM FOREST

(3) RANDOM FOREST

(3) RANDOM FOREST

h2o?

→ 자바 기반 머신러닝/AI 플랫폼으로 GLM, Random forest, Gradient boost 등 다양한 모델들을 제공한다.

→ 독자적인 플랫폼을 제공하지만, R, Python, JSON 등으로 가져다 쓸 수도 있다. (R studio, Tableau에서도 가능!)

→ 확장성이 좋으며 모델의 직관적인 해석이 쉽다. (In memory map reduce, Nano Second Predictions, Columnar Compression, Query Processor)

For more information
http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html

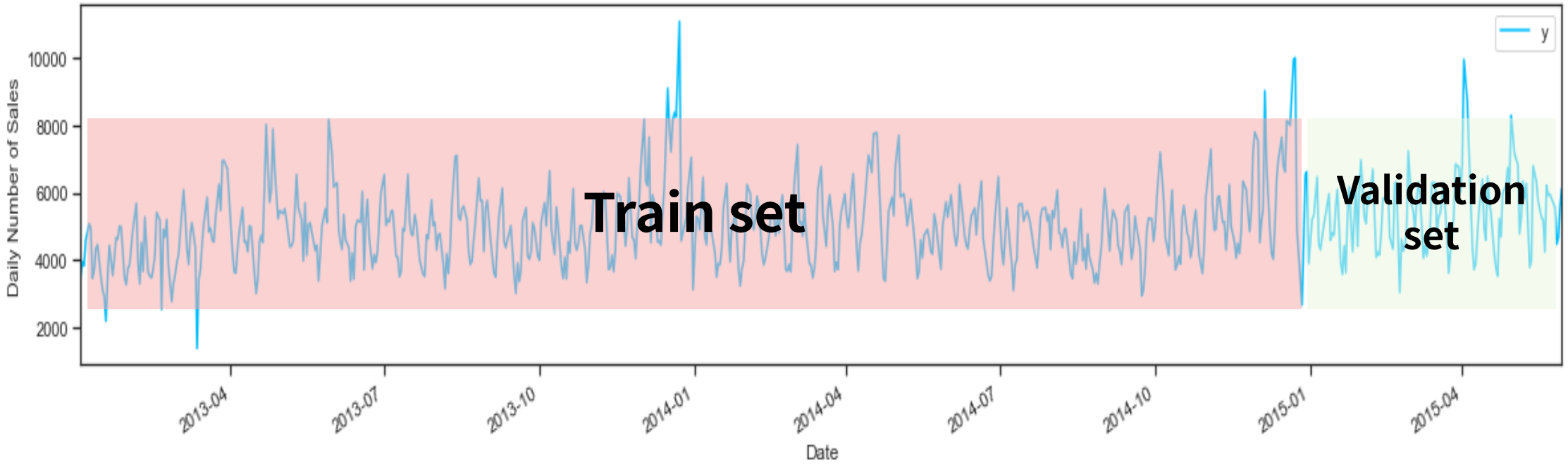
Parameter Tuning

칼럼명	결측치 row count
Store	0
StoreType	0
Assortment	0
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2	0
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	544

(4) XG BOOST

(2) PROPHET

모델 평가



다른 모델링들의 지표와 마찬가지로

Train set을 이용하여 적합 후, Validation set으로 평가

(3) XG BOOST

Hyper Parameter

Parameter	Values
objective	reg:linear
Loss function = RMSE Early stop → RMSPE	
booster	gbtree
gblinear < gbtree	
eta	0.01
max_depth	10
subsample	0.9
colsample_bytree	0.7
n_estimator	5000
early_stopping_rounds	100

(3) XG BOOST

Prediction

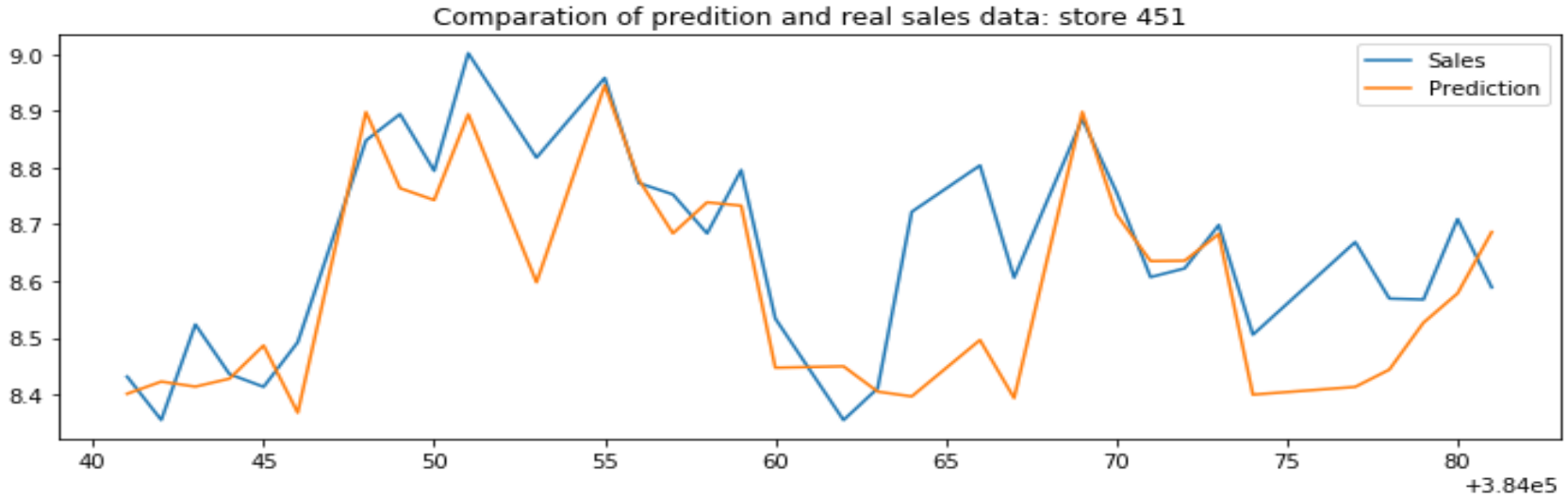
```
Stopping. Best iteration:
[4568]  train-rmse:0.080685     eval-rmse:0.132838      train-rmspe:0.091834    eval-rmspe:0.135185

Training time is 4489.085442 s.
validating

RMSPE: 0.135283
```

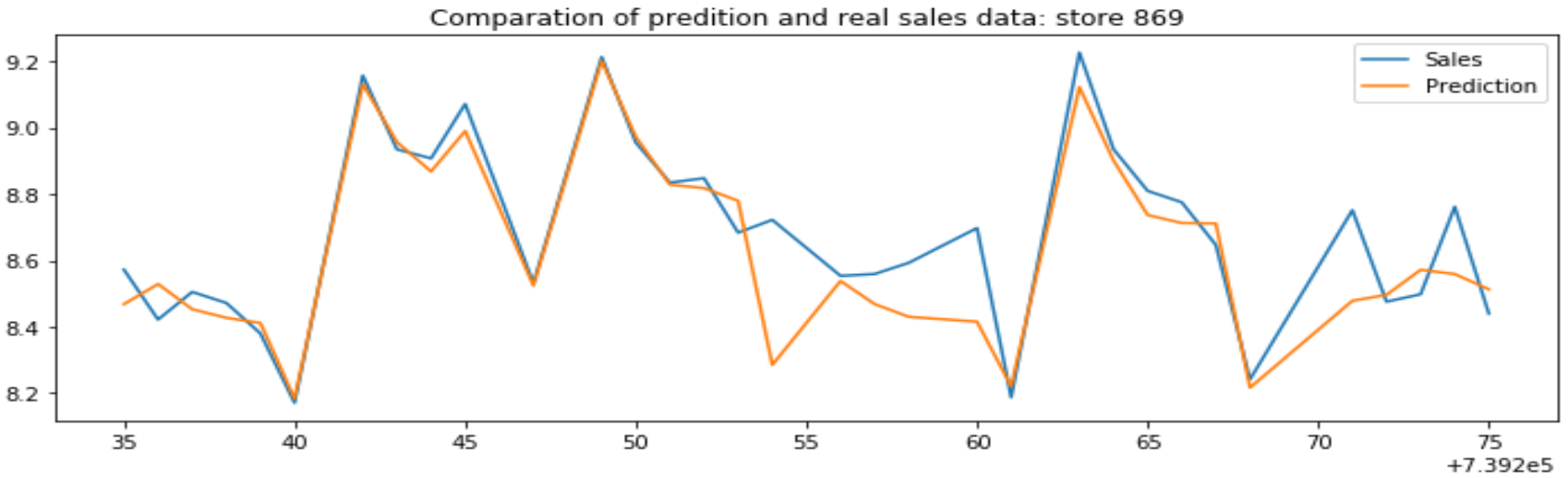
Validation RMSPE =

0.135283

(3) XG BOOST

Prediction





Thank you