

---

# Walmart Recruiting - Store Sales Forecastingの振り返り

藤崎勇哉

# アジェンダ

---

- サマリ
- 実験の詳細
  - 結果の一覧
  - 作業の要点
  - データの俯瞰
  - 特徴量
  - 使用したモデル
- 参考文献

# サマリー

## 総括

もっとも誤差の小さい結果は14のモデルをアンサンブルした結果で、誤差(WMAE)はPrivateで2637だった(20位相当)。特徴量はWeekly\_Salesを特定の時間単位で集約・集計(aggregate)して作成したもの、Kmeansで算出したものが筋が良かった。欠損補間はトレンドが明白な特徴量は移動平均で補間し、その他は-9999で補間した[1]。モデルはアンサンブル系のモデル、時系列予測系のモデルを選択した。モデル選択の基準は前処理が簡単なこと、パラメータのチューニングが容易なことである。15時間程度の工数しか避けなかったため、妥当な選択だったと考えている。(機械工数は50時間程度割いた)

## 最終的な結果 ・モデル使用の根拠

複数のXGB、ExtraTreesRegressor、Prophetの試行とその他アンサンブル系のモデルの推定結果を平均した。

- ・ アンサンブル系のモデルは変数の数値変換をせずともロバストなため、積極的に使用した。(数値変換の方法は多種多様なため、数値変換方法がハイパラとなり、チューニングの手間となると考えた。そのため今回は数値変換の使用を避けるためのモデル選択をした。)
- ・ 時系列系の予測モデルにおける自己回帰モデルはハイパラチューニングが難しいため、Prophetを使用した。

## データの俯瞰 ・特徴量

各Storeにおける各DeptのWeekly\_Salesを定性的に確認すると時間的な傾向があった。XGBのfeature importanceによると重要度(gainにより算出)が高い特徴量は、Weekly\_Salesの時間的な傾向をもとに作成した、特定の時間単位で集約し作成した特徴量、またKmeansを用いて作成した各クラスターのセントロイドからの距離だった。その他、イベント日、ラグ特徴量、Kmeansを用いて作成したクラスター番号などを追加したがそれらの重要度は比較的低かった。

## 今後の改善

複雑な欠損値補間、NNの使用、ハイパラ探索方法の改善などが挙げられる。

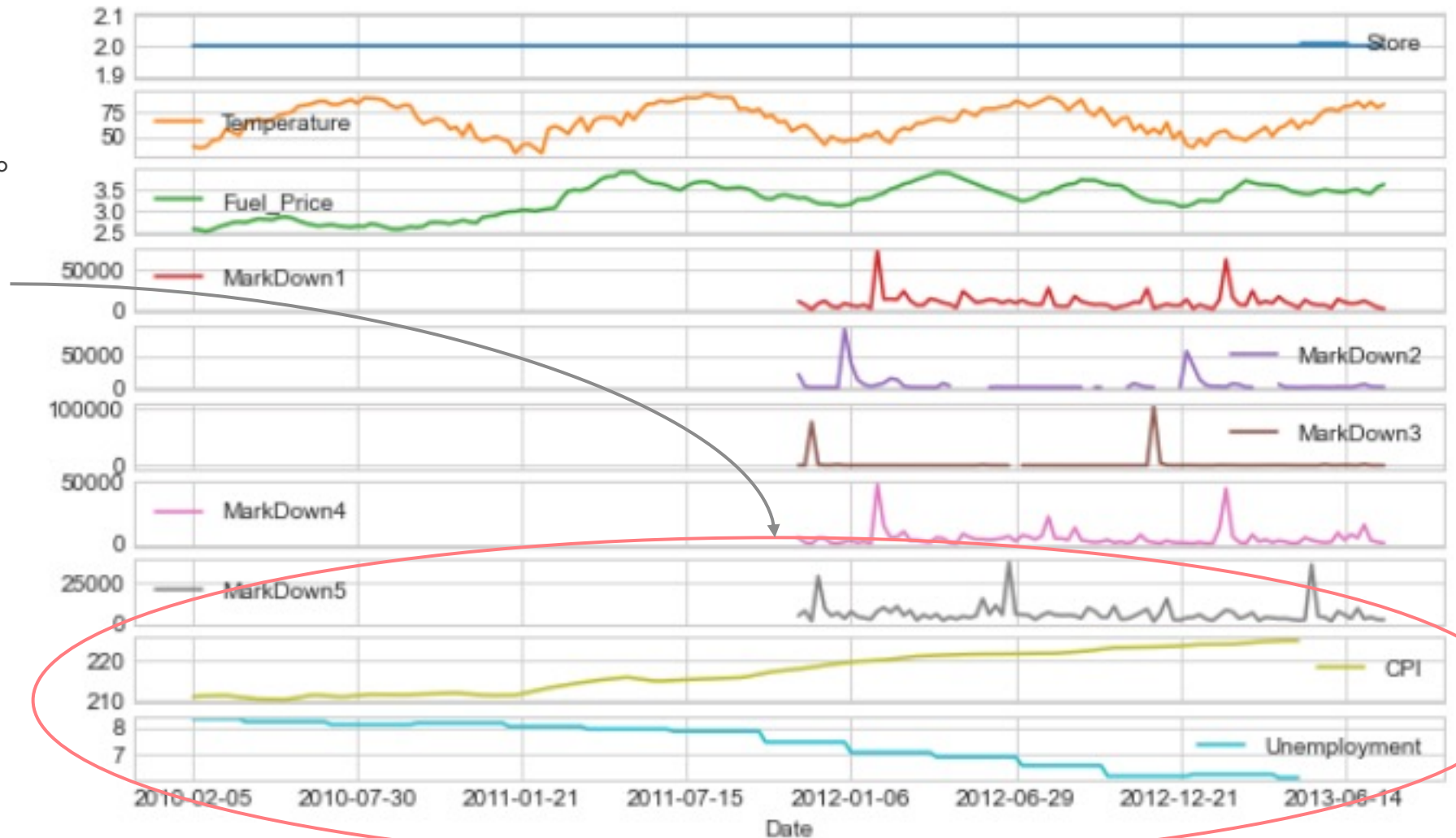
- ・ 欠損値補間：他の変数からの予測[1]
- ・ NNの使用：NNはチューニングの手間、計算時間がかかるが、非線形データや時系列データについても有効性が実証されている。(今回は計算機の制約で十分に使用できなかった。)
- ・ ハイパラ探索：時間の制約で探索回数がこなせず単純なRandomCVを使用した。あたりの良い初期値が判明し、また十分回数の探索が可能ならばOptunaの方が有効と考えられる。

## (参考)欠損の保管方法について

以下の図は feature.csv の Store1における各特徴量の Dateに沿った可視化である。

CPI, Unemployment は Store1に限らず他のStoreでもトレンドが確認された。

-> 各 Store の CPI, Unemploymentの補間には移動平均を用いることが可能と判断した。



# 実験の詳細

## 結果の一覧

以下は主な実験結果の一覧。使用した特徴量とCross Validationの有無、スコアについて表とした。これらのモデルらをアンサンブルし、**Private**で約2637(単独20位相当)となった。

#	モデル	集約	ラグ	Kmeans	休日	Cross Val	Private
1	ExtraTreesRegressor	-	-	-	○	-	約3389
2	ExtraTreesRegressor	○	○	-	○	○	約2978
3	ExtraTreesRegressor	○	-	-	○	○	約3024
4	ExtraTreesRegressor	○	-	○	○	○	約3089
5	ExtraTreesRegressor	○	-	○	○	-	約2977

# 実験の詳細

## 結果の一覧

以下は主な実験結果の一覧。使用した特徴量とCross Validationの有無、スコアについて表とした。これらのモデルらをアンサンブルし、**Private**で約2637(単独20位相当)となった。

#	モデル	集約	ラグ	Kmeans	休日	Cross Val	Private
7	XGB	-	-	-	○	-	約6380
8	XGB	○	○	-	○	○	約2971
9	XGB	○	-	○	○	○	約2821
10	BaggingRegressor	○	○	○	○	-	約3002
11	GradientBoostingRegressor	○	○	○	○	-	約2934
12	HistGradientBoostingRegressor	○	○	○	○	-	約3161
13	RandomForestRegressor	○	○	○	○	-	約3032
14	AdaBoostRegressor	○	○	○	○	-	約3053

# 実験の詳細

## 結果の一覧

以下は主な実験結果の一覧。使用した特徴量とCross Validationの有無、スコアについて表とした。これらのモデルらをアンサンブルし、**Private**で約2637(単独20位相当)となった。

Trainは基本的に2010/2/5 ~ 2012/10/26までであるが、一部週の欠損がある。(次ページ詳細) Prophet使用時はこの欠損補間は[2]を参考とし、0で補間した。その上で各Storeごとの各Deptごとの過去のWeekly\_Salesを特徴量としてtestのWeekly\_Salesを推定した。

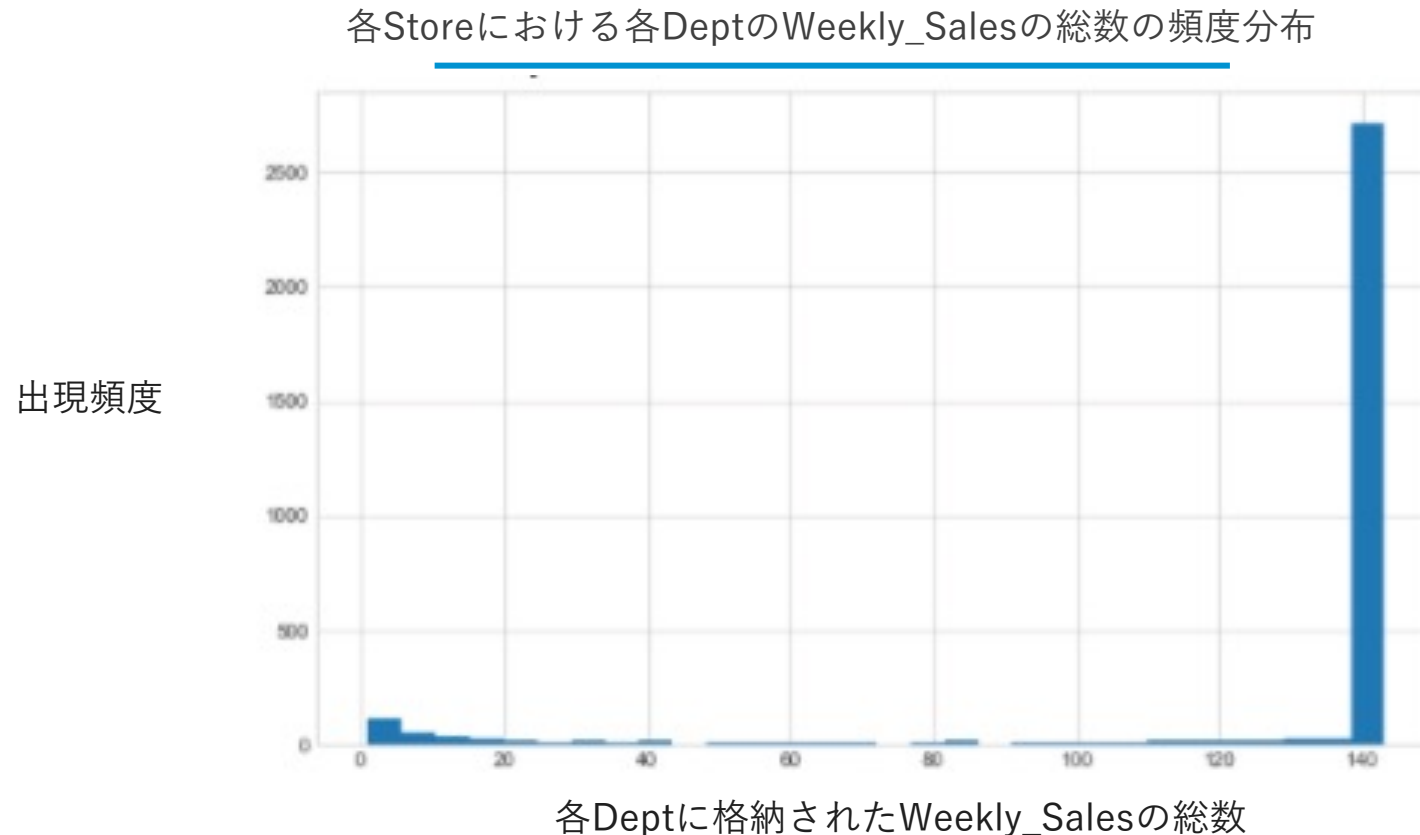
#	モデル	休日	周期	Private
15	Prophet	-	-	約3350
16	Prophet	○	-	約2999
17	Prophet	○	○	約2934

## (参考)Trainデータの欠損について

以下の図はtrain.csvの各Storeにおける各DeptのWeekly\_Salesの総数の可視化である。

概ね各Deptで143期のWeekly\_Salesが格納されているが、一部のDeptで欠損が確認された。

-> 自己回帰モデル等の使用時には欠損の補間が必要と考えられる。Propetの使用時には[2]を参考に欠損を補間した。





以下はベースラインから劇的な改善が見られた試行の主な作業内容である。  
Weekly\_Salesの特性を発見できたことで筋の良い特徴量の生成・良好な精度が得られたと考える。

### 良好な精度を得るまでに行ったこと

#### データ俯瞰

Weekly\_Salesに注目し、以下の示唆が得られた。

- (示唆1)例年売り上げが上昇する週数は似ていた。
- (示唆2)例年同じ月日はWeekly\_Salesが似ていた。

#### 特徴量

得られた示唆から主に以下の特徴量を使用した

- (示唆1)の考察により、予測対象の日付と同様の過去の月日の統計量が使用可能と考えた
- (示唆2)の考察により、過去の時点での値をそのまま特徴量とするラグ特徴量を使用した

#### モデル

ラグ特徴量を十分に活用するため、GBDTを活用した。  
(GBDTモデルは欠損値を扱うことができるため)

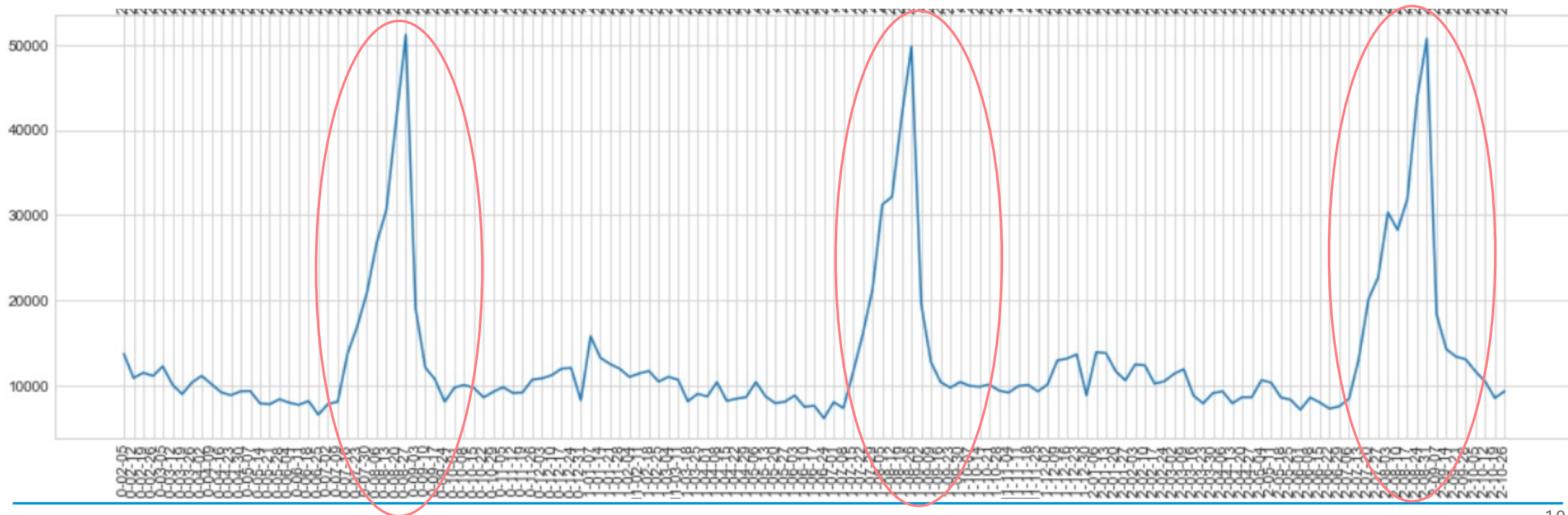
# 実験の詳細

## データの俯瞰

以下はStore1のDept3についてWeekly\_Salesを描画した例。2010年から2012年では8月の最終週に年間で最も高いWeekly\_Salesを記録していた。

-> 例年売り上げが上昇する週数は似ていた。またStore1のDept3以外にも同様の傾向は多く見られた。  
(示唆1)

Store1のDept3のWeekly\_Salesの描画例



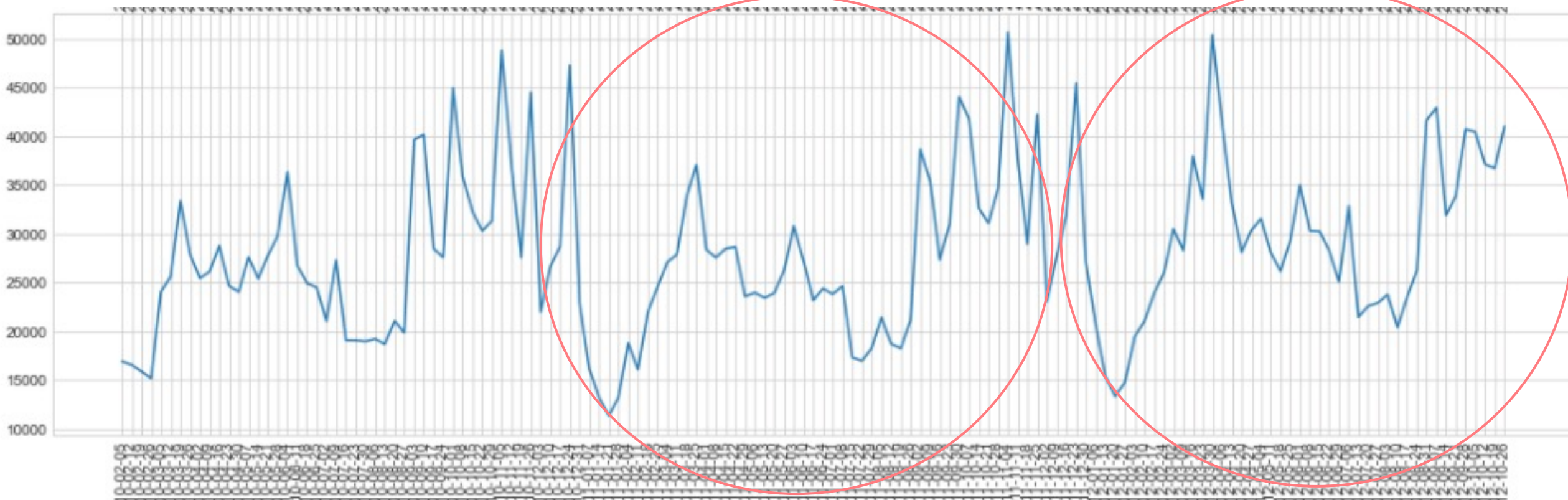
# 実験の詳細

## データの俯瞰

以下はStore1のDept9についてWeekly\_Salesを描画した例。2010年から2012年では1月中旬ごろに売り上げが低下し、3月ごろに売り上げが増加、その後8月ごろに低下、その後断続的に高い売り上げが見られた。

-> 例年同じ月日はWeekly\_Salesが似ていた。またStore1のDept9以外にも同様の傾向は多く見られた。  
(示唆2)

Store1のDept9のWeekly\_Salesの描画例

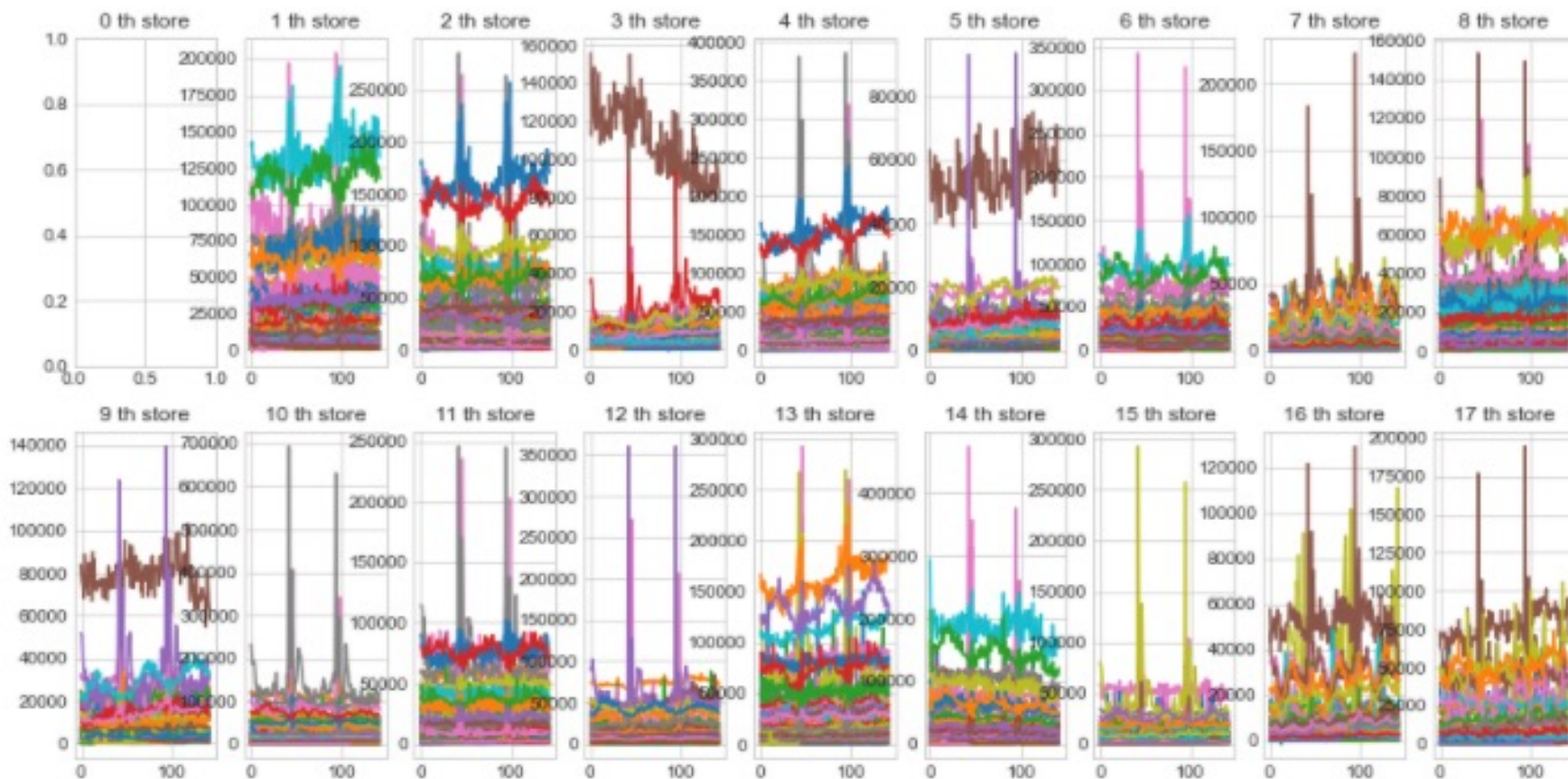


# 実験の詳細

## データの俯瞰

Storeごとの全てのDeptの可視化の一部(Store1~Store17)。

定性的な確認において**Weekly\_Sales**の上限がDeptごとに大きく異なっており、単一のモデルではこれらのバリエーションに対応した推定は難しいと判断した。-> 各Storeの各Deptごとにモデルを作成する必要性があると判断した。



Storeごとの全てのDeptの可視化の一部(Store1~Store17)



# 実験の詳細

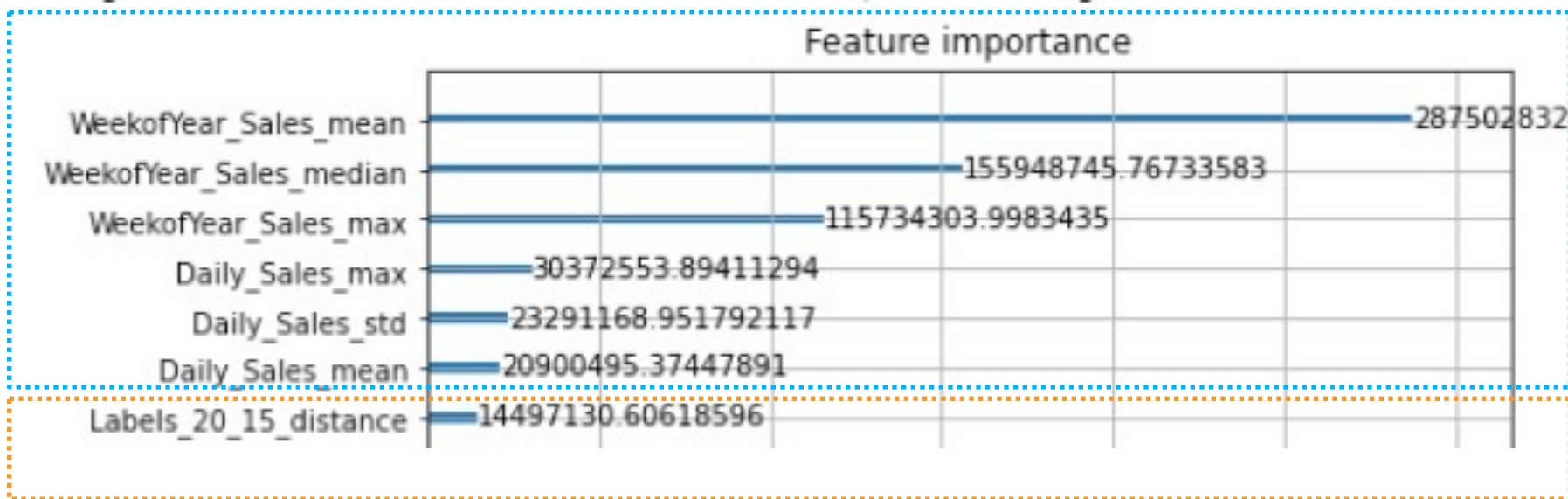
## 特徴量

(示唆1)の考察により、予測対象の日付と同様の過去の月日の統計量が使用可能と考えた

(示唆2)の考察により、過去の時点での値をそのまま特徴量とするラグ特徴量を使用した ラグ={52, 104}また、週番号は季節的な傾向をより捉えられるため、活用した。一年の週数は1~53(なので1年のラグは52)と想定した。

-> Store1, Dept1のモデルの上位のfeature\_importance。確認した範囲では全てのモデルで類似の特徴量の重要度が高かった。特に示唆1に基づいて作成した特徴量が効果的だったと考えられる。

集計に関する特徴量は各Storeの各Deptごとに算出した。



集計結果から算出した特徴量

Kmeansから算出した特徴量

# 実験の詳細

## 使用したモデル

使用したモデルと使用の理由は以下の通り

#	項目	使用した理由	使用した感想等
1	ExtraTreesRegressor	<ul style="list-style-type: none"><li>[3]で良い精度が報告されていたため初手として選択した。</li></ul>	-
2	XGB	<ul style="list-style-type: none"><li>GBDTはテーブルデータに対して有効[1][4]なため、選択した。</li></ul>	-
3	NN	<ul style="list-style-type: none"><li>時系列データに対しても従来の自己回帰モデルに近い精度が実験的に確認されているから。</li></ul>	パラメータ数の大きい全結合層のみのNNでは、精度が不十分だった。LSTM等の記憶セルがあるレイヤーを用いることで精度が向上する可能性があったが、より精度の良くチューニングが少ないensemble baseの手法での試行を優先した。
4	Prophet	<ul style="list-style-type: none"><li>計算速度が速いから。</li><li>また自己回帰モデルと異なり、モデリングの手間が少ないから。</li><li>欠損の補間が必ずしも必要ないから。</li></ul>	実装の手間が少ないにも関わらず、(特徴量はWeekly_Salesのみ)十分な精度だった。

- 参考にした資料
  - [1] : 門脇 大輔., 阪田 隆司., 保坂 桂佑., *Kaggleで勝つデータ分析の技術*
  - [2] : Abhishek ., *Walmart-Sales Forecasting.*, <https://medium.com/@abhishek.palle123/walmart-store-sales-forecasting-d15a205e3348>
  - [3] : Andre, D., Raissa, Dutra., *TP2 - Walmart Sales Forecast.*, <https://www.kaggle.com/andredornas/tp2-walmart-sales-forecast>
  - [4] : Ravid, Shwartz-Ziv., Amitai, Armon., *Tabular Data: Deep Learning is Not All You Need.*, arXiv: CS/ 2106.03253