Linear Models, Bayesian Inference, Machine Learning and Reinforcement Learning In Business Time Series Forecasting

Bohdan Pavlyshenko (Ph.D.)

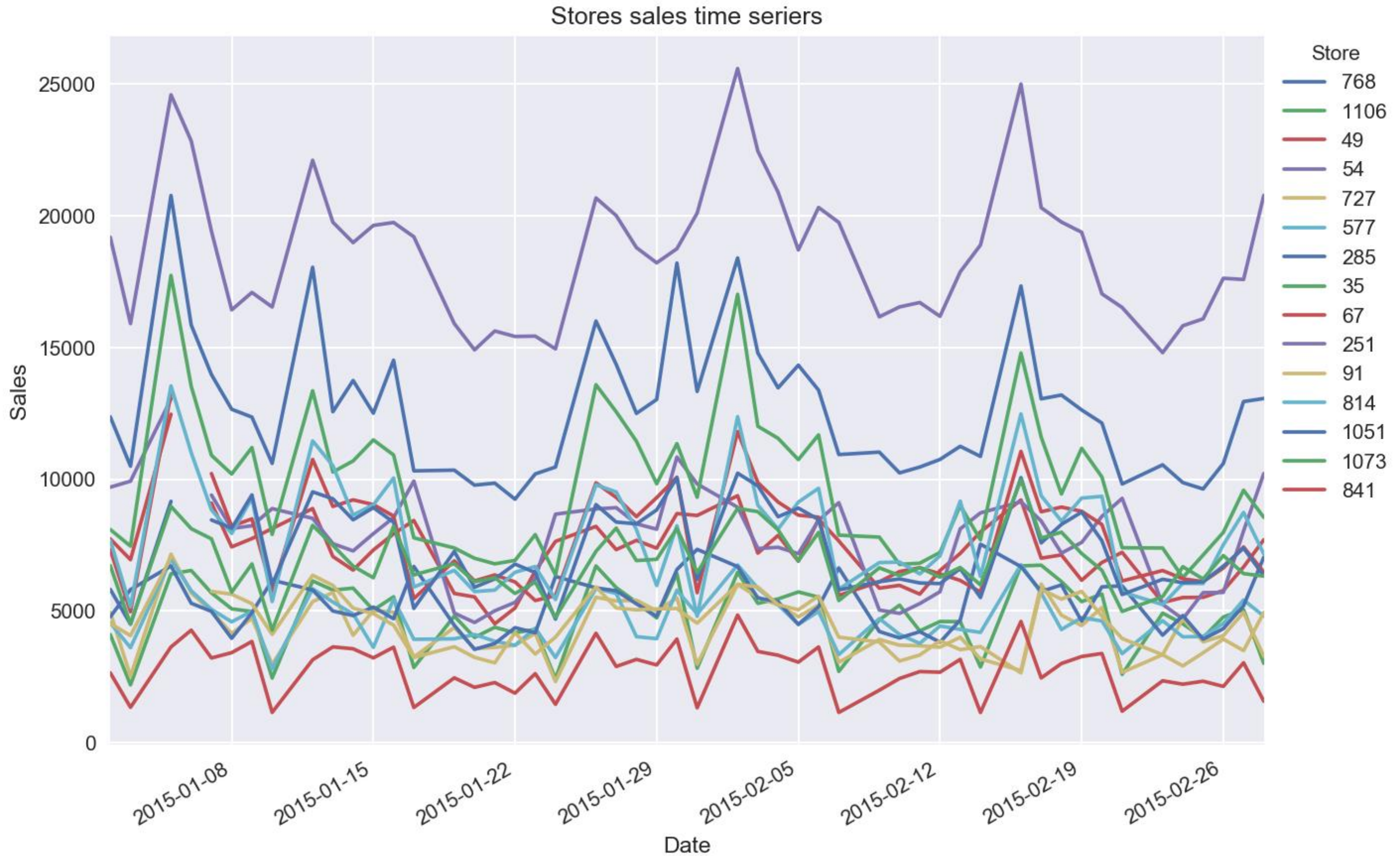
LinkedIn: https://www.linkedin.com/in/bpavlyshenko/

e-mail: b.pavlyshenko@gmail.com

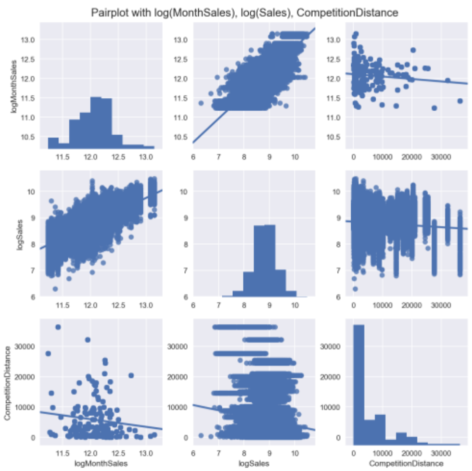
Time Series

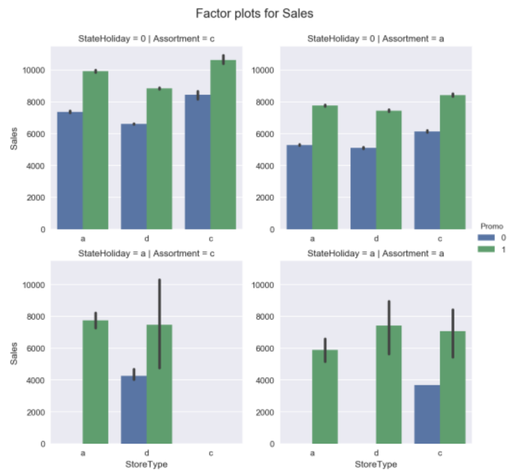
Effects:

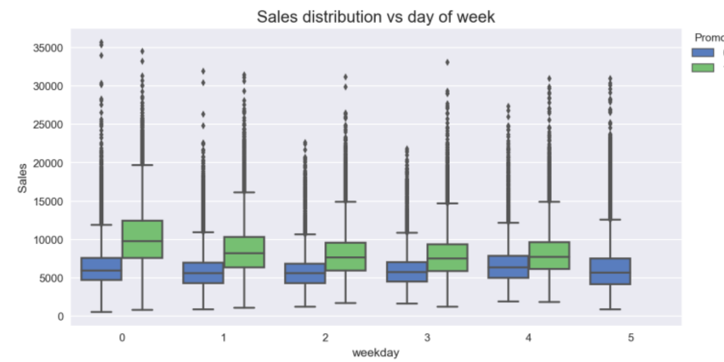
- ➢ Seasonality
- ➢ Trend
- ➢ Autoregression
- ➢ External factors influence



Descriptive Analytics







Univariate Time Series Forecasting

Popular models:

- ARIMA, SARIMA, SARIMAX
- Holt-Winters
- GARCH

Challenges of using univariate time series methods:

- It is needed to have historical data for long time period to capture seasonality.
- Sales can have complicated seasonality - intra-day, intra-week, intra-month, annual.
- Sales data can have a lot of outliers and missing data.
- It is problematic to take into account a lot of exogenous factors which have impact on sales.
- These methods do not explain sales movements under external factors.

Time Series Regression Approaches

Additive Regression Model:

Sales = Trend + Seasonality + Events_Impact + Pricing_Effects + Advertising_Effects+ Promo_Effects + Competitor_Factors + Social_Economic_Factors + Macro_Economic_Factors + Other_Factors

- Multiplicative model can be received by logarithmic transformation of target variable.

Predictive regression models can be split into three categories:

- Linear models
- Probabilistic models
- Machine learning models

Linear Regression

$$y(x, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M-1} w_j \emptyset_j(x)$$

$$\Phi = \begin{pmatrix} \emptyset_0(x_1) & \emptyset_1(x_1) & \cdots & \emptyset_{M-1}(x_1) \\ \emptyset_0(x_2) & \emptyset_1(x_2) & \cdots & \emptyset_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \emptyset_0(x_N) & \emptyset_1(x_N) & \cdots & \emptyset_{M-1}(x_N) \end{pmatrix}$$

$$t = y(x, \boldsymbol{w}) + \varepsilon$$

$$\boldsymbol{w} = (\Phi^T \Phi)^{-1} \Phi^T t$$

Lasso Regression

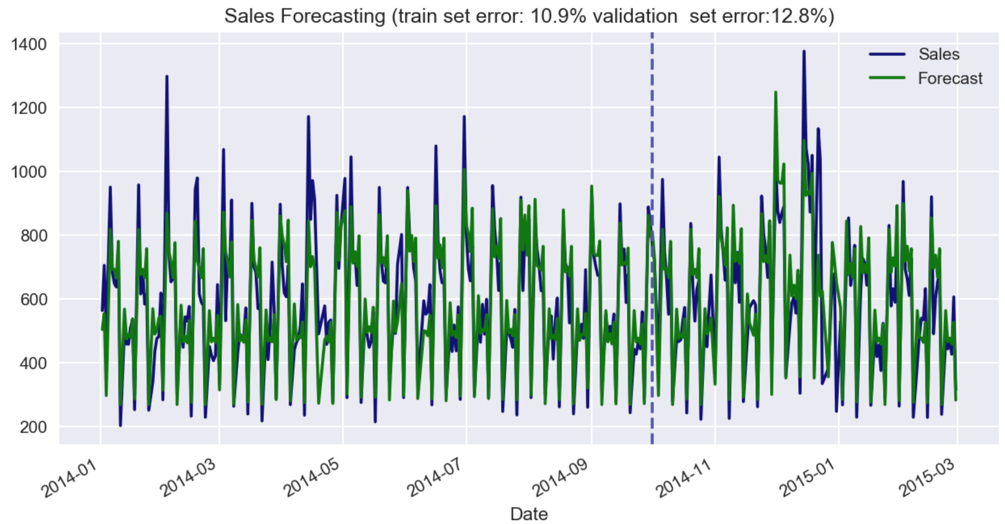
$$E = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{w^T} \emptyset(\boldsymbol{x})\}^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|$$

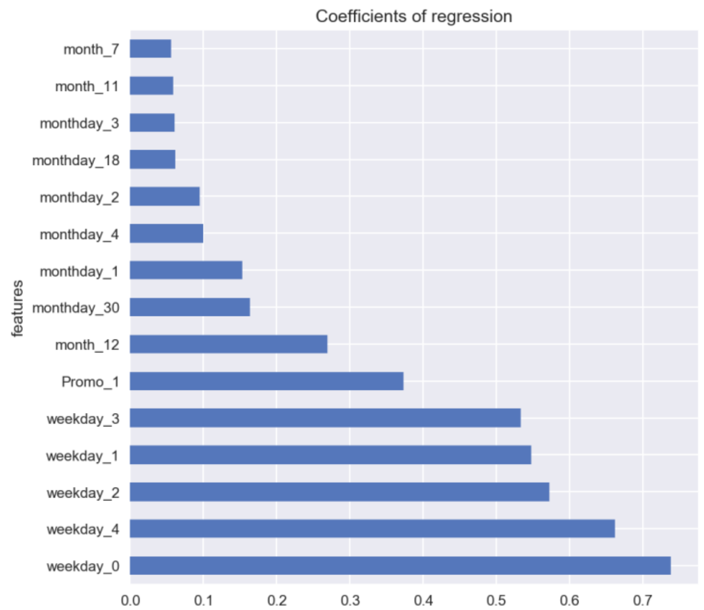
Ridge Regression

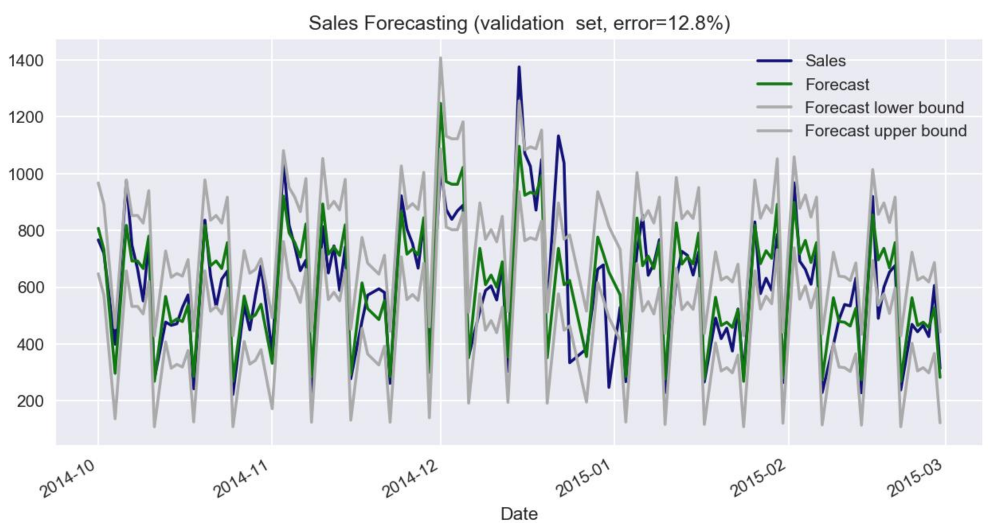
$$E = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{w^T} \emptyset(\boldsymbol{x})\}^2 + \frac{\lambda}{2} \boldsymbol{w^T w}$$

$$\boldsymbol{w} = (\lambda \mathbf{I} + \Phi^T \Phi)^{-1} \Phi^T t$$

Lasso Regression







Bayesian Inference

$$p(\boldsymbol{w}|D) = \frac{p(D|\boldsymbol{w})p(\boldsymbol{w})}{p(D)} \qquad p(D) = \int p(D|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w}$$

Sampling

$$E(f(z)) = \int f(z)p(z)dz$$

$$\{z_i|i = 1, \dots, L\}$$

$$\hat{f} = \frac{1}{L}\sum_{i=1}^{L} f(z_i)$$

$$t = 1,2, \dots, T$$

$$\boldsymbol{z} = \{z_i|i = 1, \dots, M\}$$

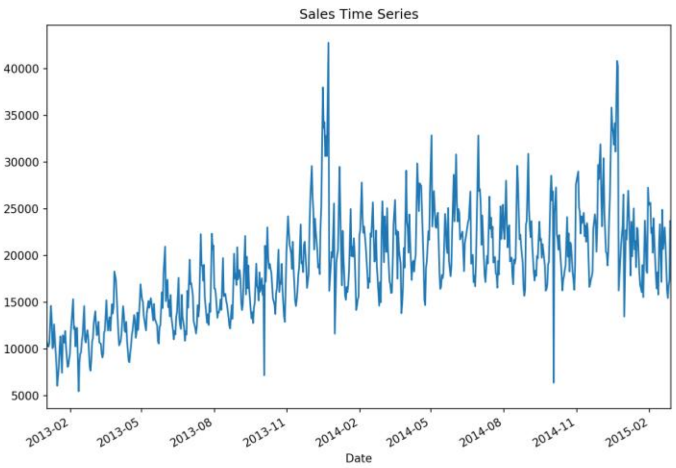
$$z_1^{(t+1)} \sim p\left(z_1|z_2^{(t)}, z_3^{(t)}, \dots, z_M^{(t)}\right)$$

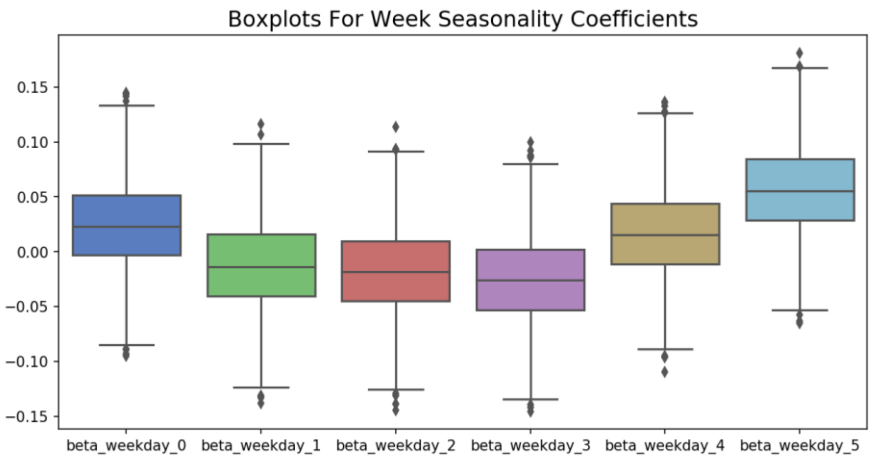
$$z_2^{(t+1)} \sim p\left(z_2|z_1^{(t+1)}, z_3^{(t)}, \dots, z_M^{(t)}\right)$$

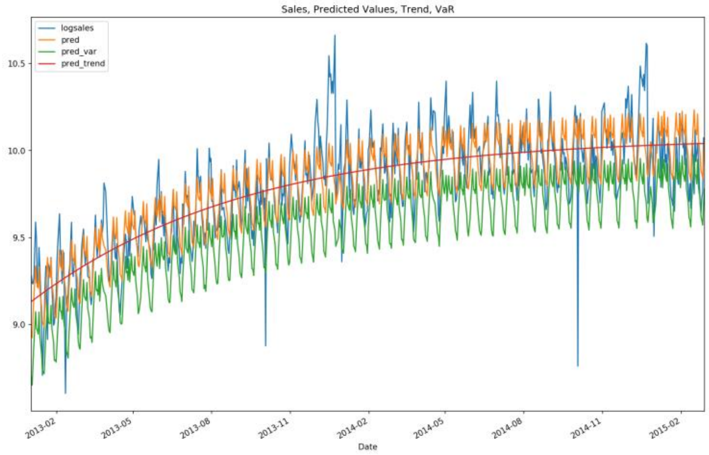
$$z_j^{(t+1)} \sim p\left(z_j|z_1^{(t+1)}, \dots, z_{j-1}^{(t+1)}, z_{j+1}^{(t)}, \dots, z_M^{(t)}\right)$$

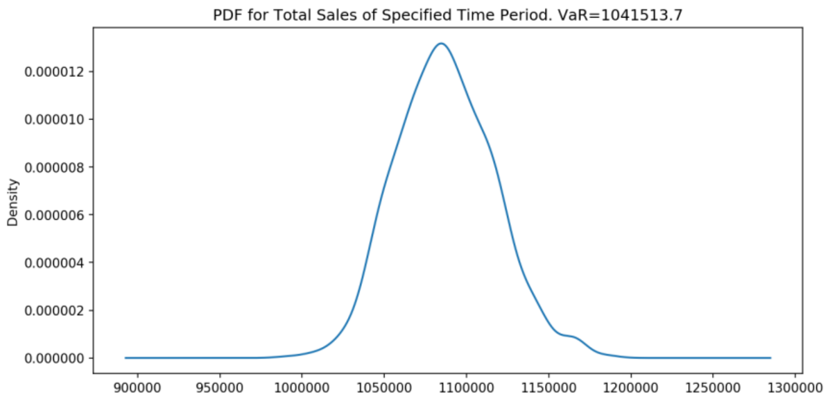
$$z_M^{(t+1)} \sim p\left(z_M|z_1^{(t+1)}, z_2^{(t+1)}, \dots, z_{M-1}^{(t+1)}\right)$$

Bayesian Inference for Sales Time Series Forecasting







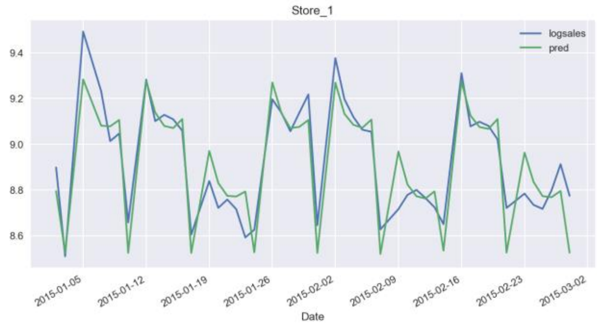


Bayesian Inference for Sales Time Series Forecasting

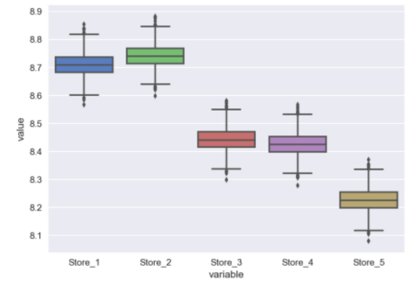
Hierarchical Models

Sales Time Series (log scale)

Sales Time Series Prediction (log scale)

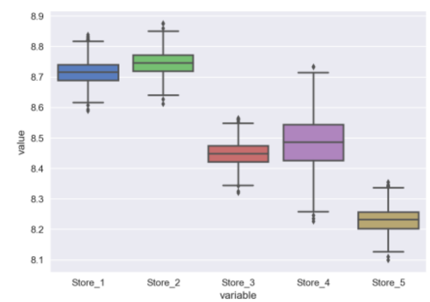


Box plots for intersect parameter



Sales Time Series Prediction for Store with Short Historical Data

Box plots for intersect parameter when one store has short historical data



Supervised Machine Learning

Popular Classifiers and Regressors:

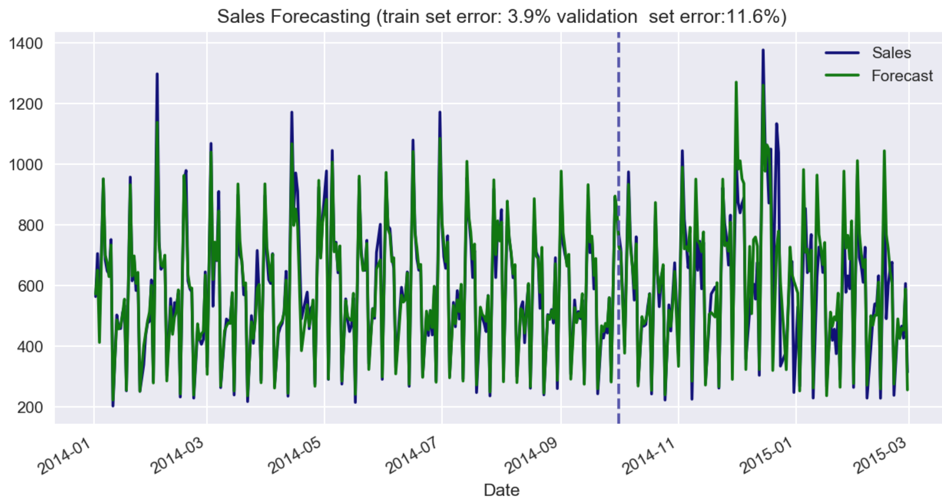
- SVM
- Random Forest
- XGBoost
- LightGBM
- Neural Networks (Keras)

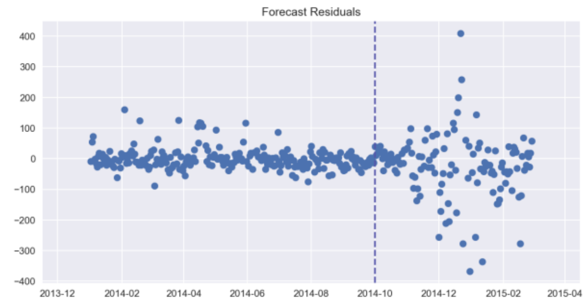
Ensemble methods:

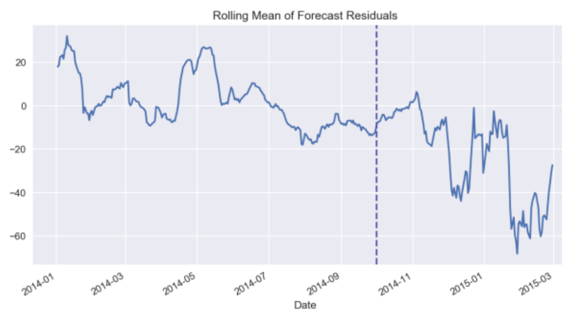
- Bagging
- Stacking

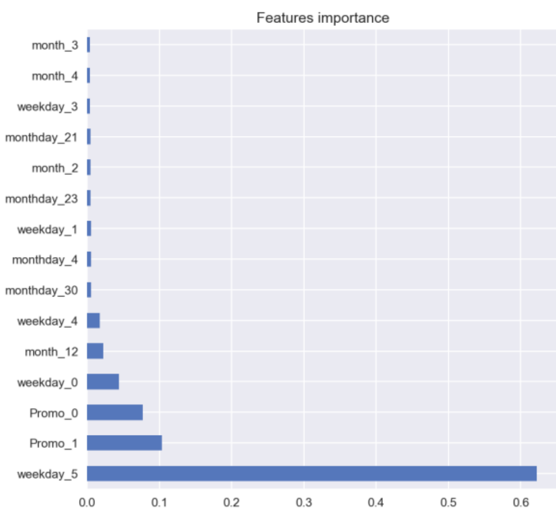
- Tree based ML algorithms are not sensitive to monotonic transformations of the features.
- Most machine learning methods can work with stationary data only.

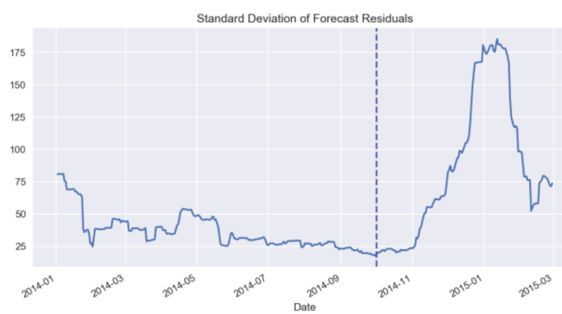
Random Forest Regression





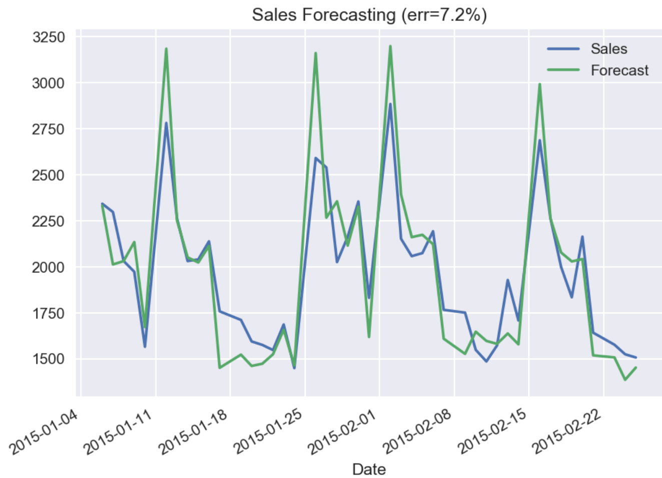


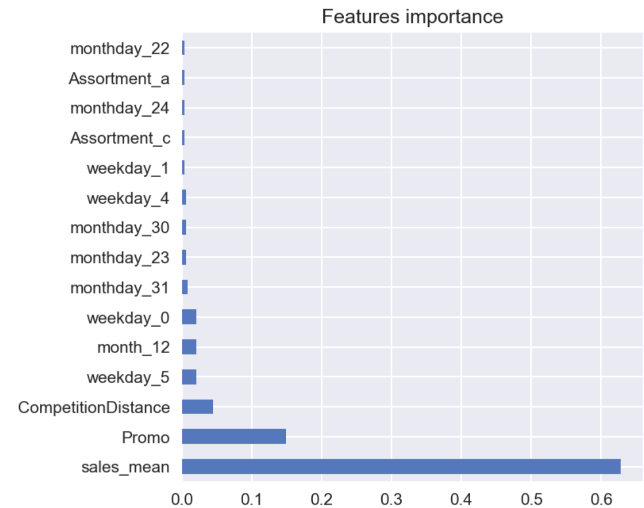




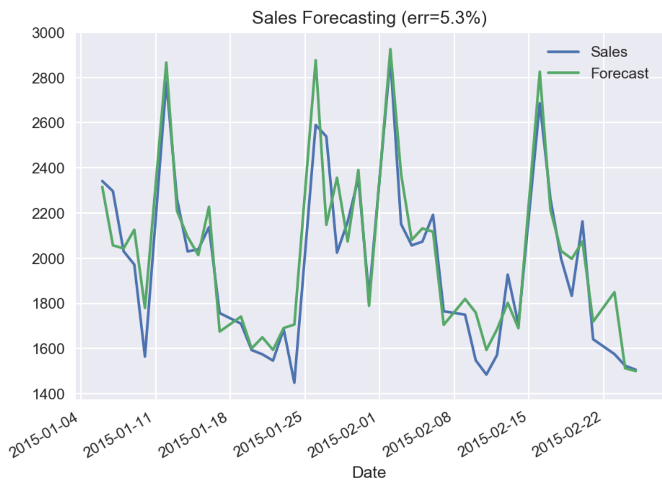
ML Generalization

Case with long time (2 years) historical data for specified store.



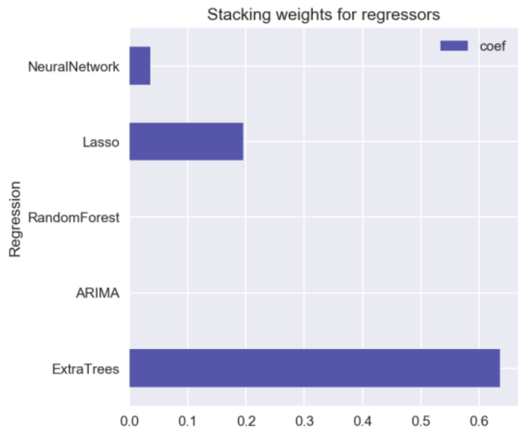


Case with short time (5 days) historical data for specified store.



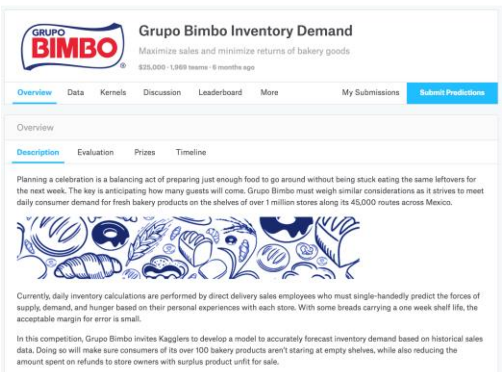
Ensemble of Classifiers Using Stacking Approach

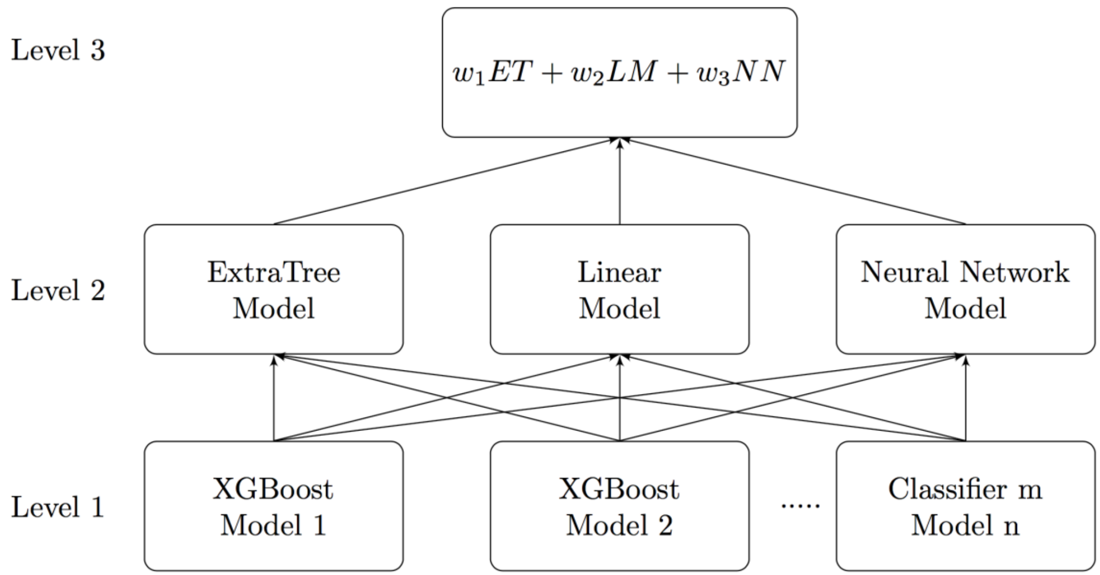
Model	Validation Error	Out-of-sample Error
ExtraTrees	14.7%	14.0%
ARIMA	13.8%	11.4%
RandomForest	13.8%	12.1%
Lasso	13.4%	11.5%
NeuralNetwork	13.0%	10.9%
Stacking	12.6%	10.2%



Winner Solution for Grupo Bimbo Inventory Demand Kaggle Competition

https://www.kaggle.com/c/grupo-bimbo-inventory-demand/discussion/23863





Reinforcement Learning

Main Notions:

- ✓ Environment
- ✓ Agent
- ✓ States (s)
- ✓ Action (a)
- ✓ Reward (R(s,a))
- ✓ Policy (π(s)→ a)
- ✓ Episode

Policy Gradient:

$$J(\theta) = \sum_{s} d(s) \sum_{a} \pi_{\theta}(\mathrm{s,a})\mathrm{R}(\mathrm{s,a})$$

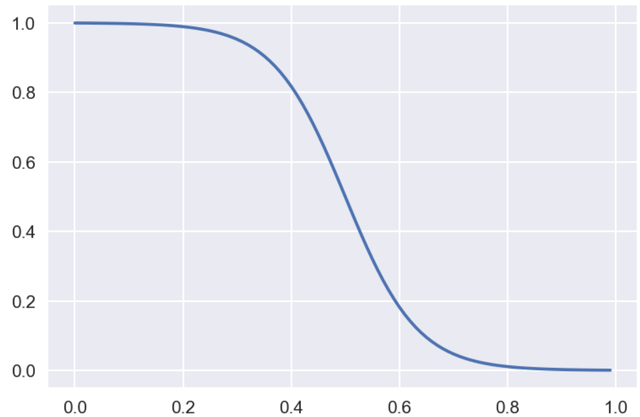
$$\Delta\theta = \alpha \nabla_{\theta} J(\theta)$$

Q-Learning:

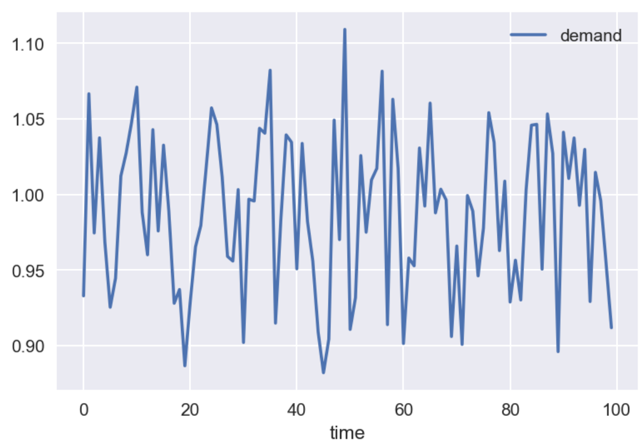
$$Q^{new}(s,a) = (1-\alpha)Q(s,a) + \alpha\left[R(s,a) + \gamma \max_{a} Q(s',a')\right]$$

Q-Learning for Pricing Strategy

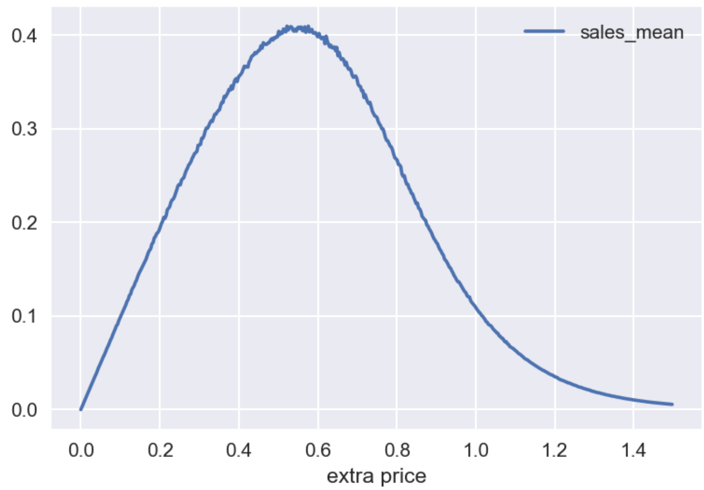
Sales vs Extra Price



Demand Time Series

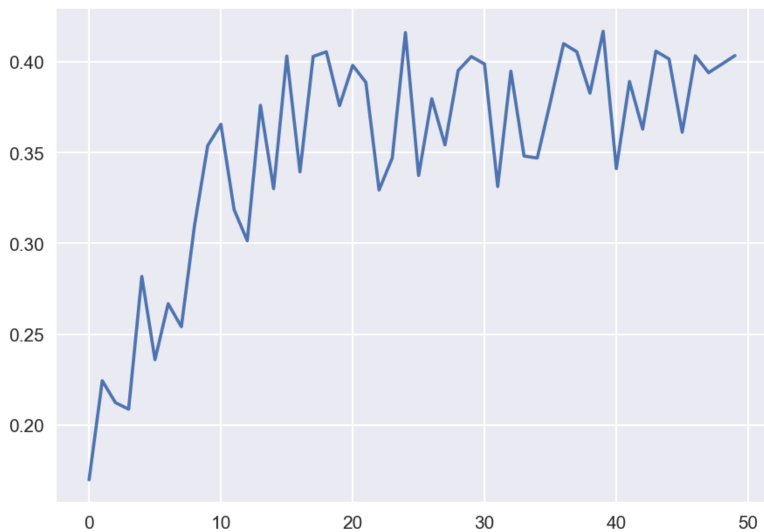


Profit vs Extra Price

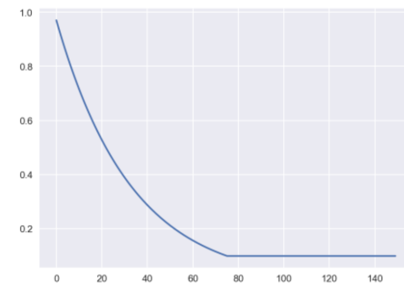


Q-Learning for Pricing Strategy

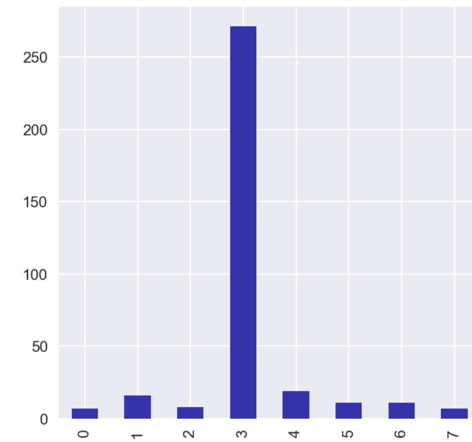
Mean Reward on Episodes



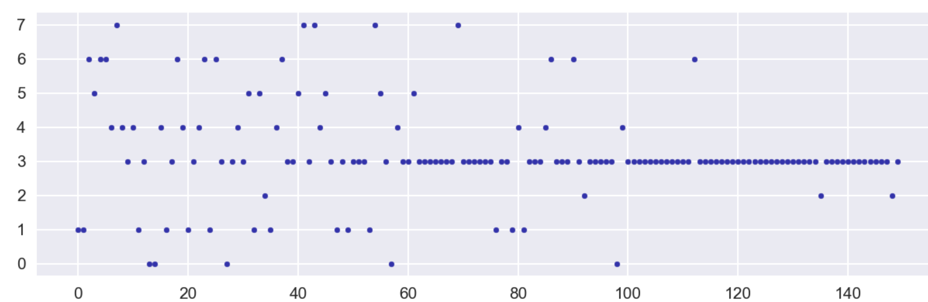
Epsilon vs Time



Action Frequency

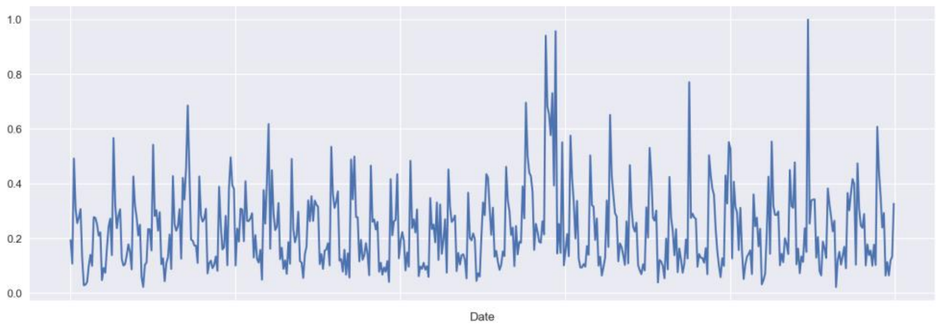


Actions vs Time

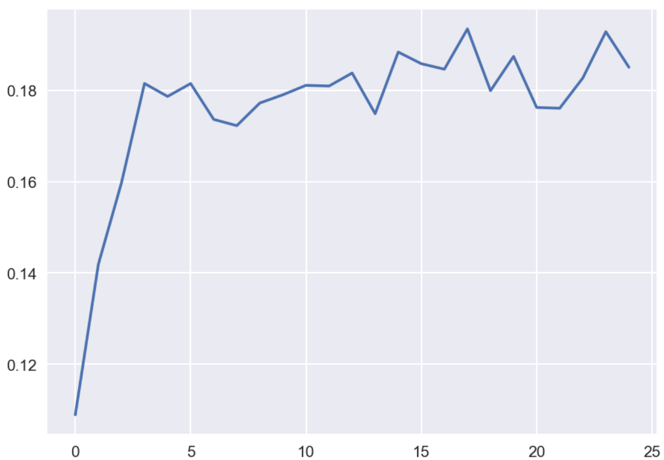


Q-Learning for Supply-Demand Problems

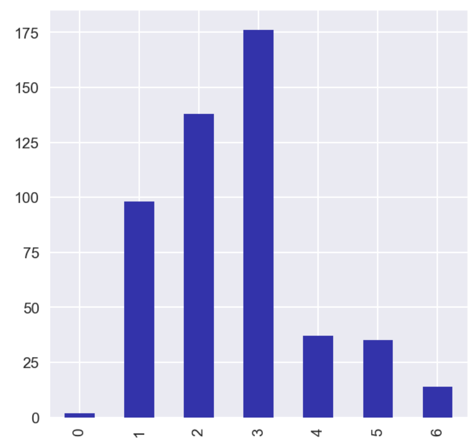
Demand Time Series



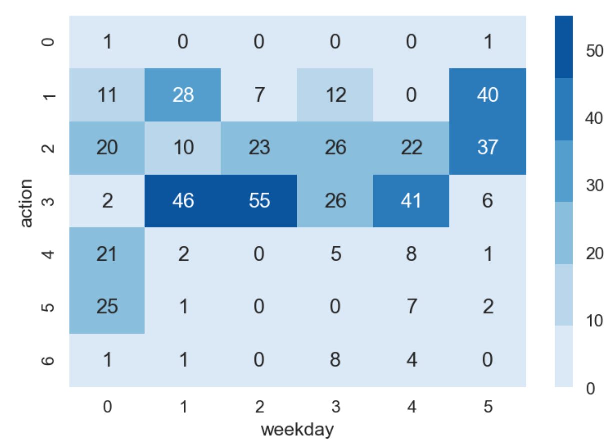
Mean Reward on Episodes



Action Frequency



Actions vs Week Day Heatmap



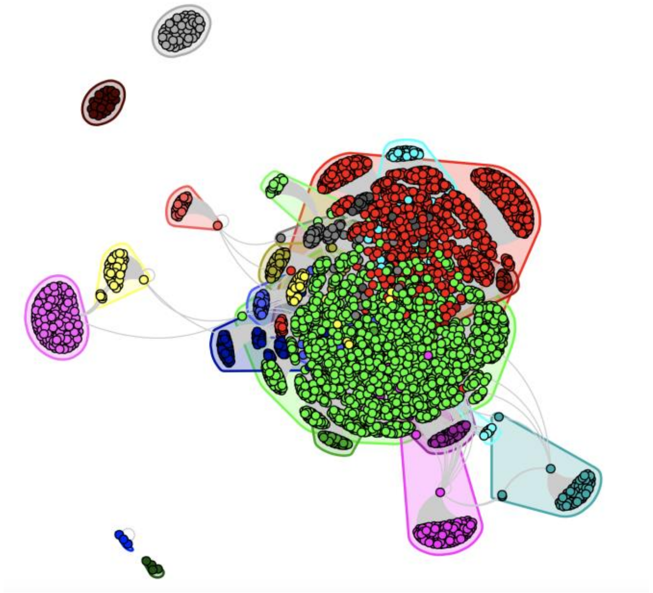
Using Alternative Data from Twitter

Community Detection in Graph of Users' Connections

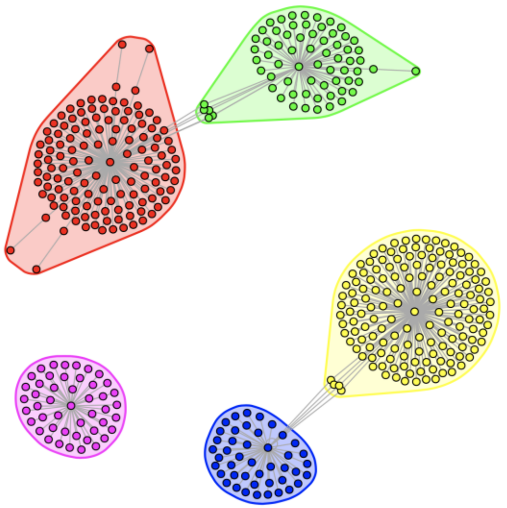
Community Walktrap Algorithm
This function tries to find densely connected subgraphs, also called communities in a graph via random walks. The idea is that short random walks tend to stay in the same community.

Fruchterman-Reingold layout
Place vertices on the plane using the force-directed layout algorithm by Fruchterman and Reingold.

Users' Communities



Subgraph of highly isolated users' communities



Using Alternative Data from Twitter

Graph Vertices Scores

Hub Score
The hub scores of the vertices are defined as the principal eigenvector of A*t(A), where A is the adjacency matrix of the graph.

Authority Score
The authority scores of the vertices are defined as the principal eigenvector of t(A)*A, where A is the adjacency matrix of the graph

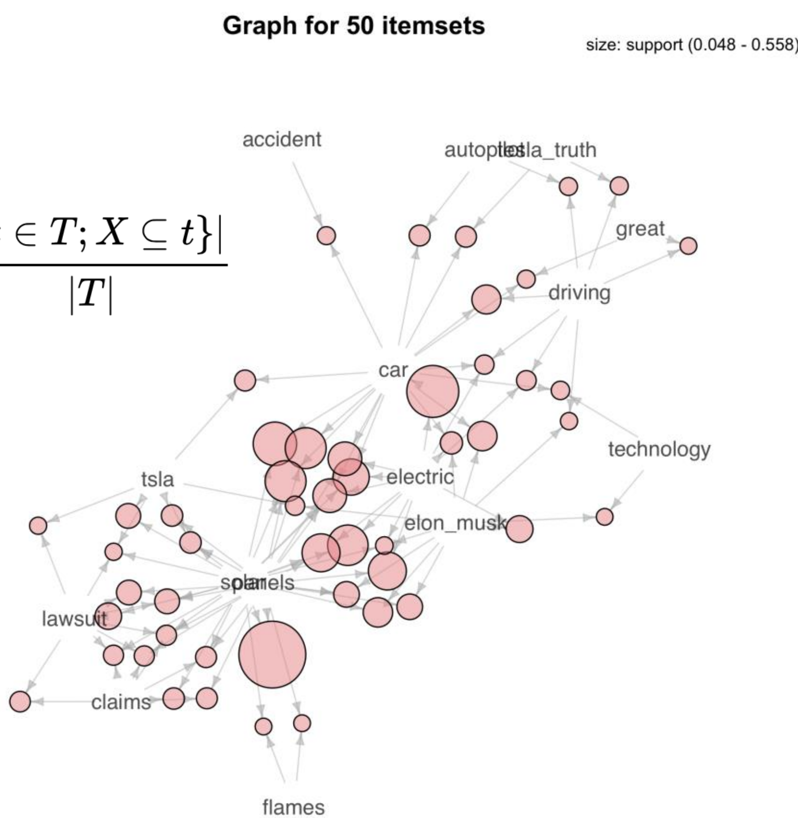
PageRank Score
Calculates the Google PageRank for the specified vertices.

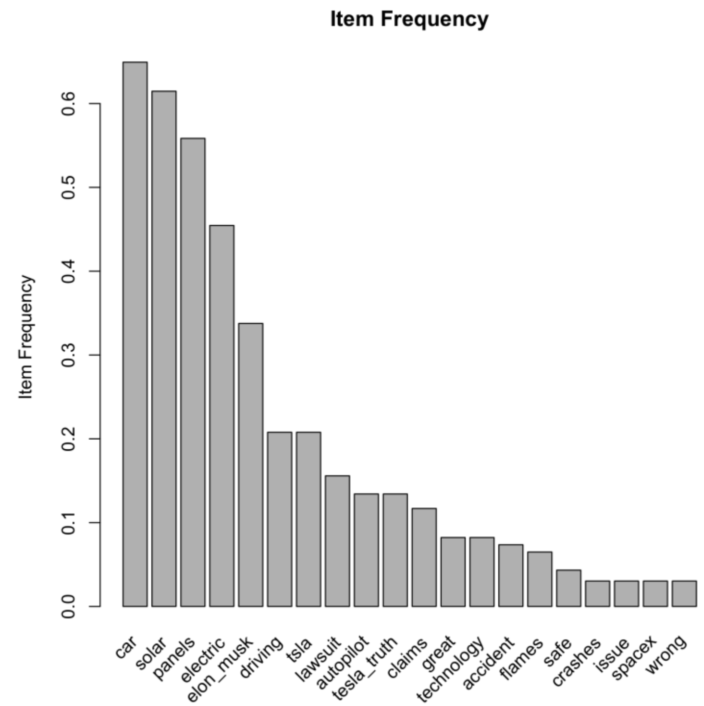
Betweenness Score
The vertex and edge betweenness are (roughly) defined by the number of geodesics (shortest paths) going through a vertex or an edge

Using Alternative Data from Twitter

Frequent Itemsets

$$\mathrm{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$



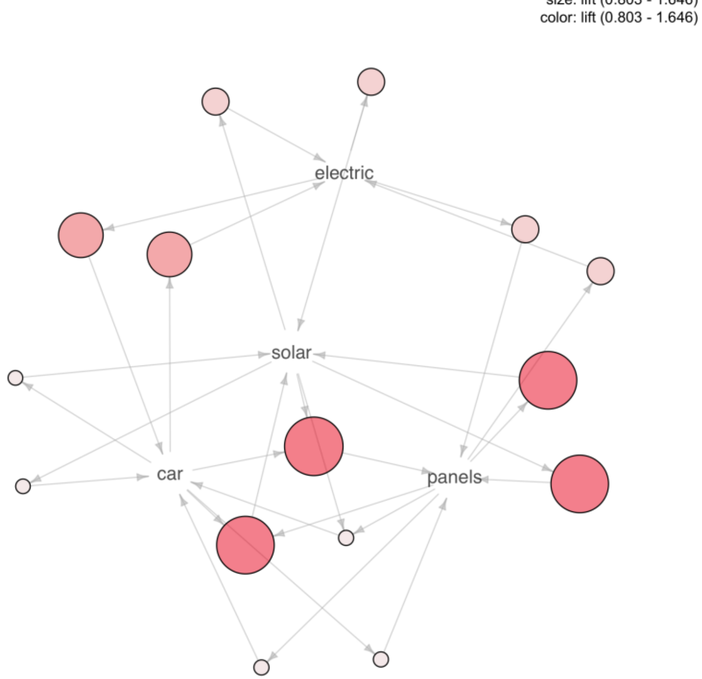


Using Alternative Data from Twitter

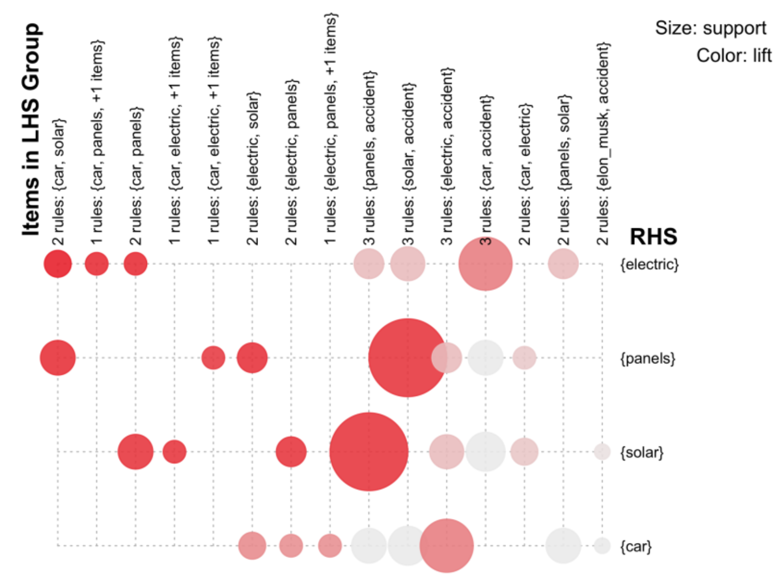
Association Rules

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X) \qquad \mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X) \times \mathrm{supp}(Y)}$$

Graph for 15 rules

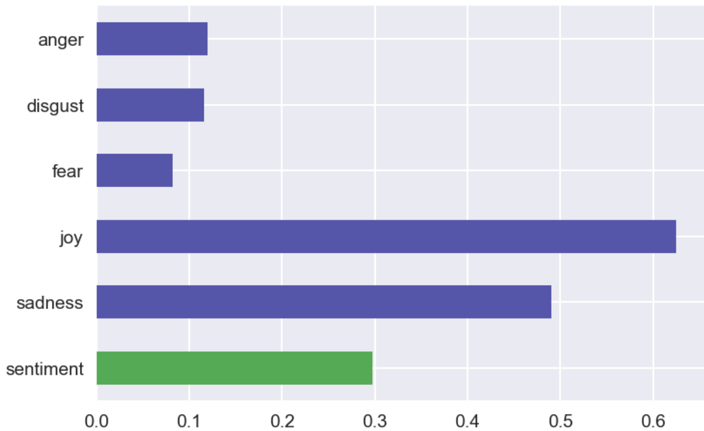


Grouped Matrix for 30 Rules

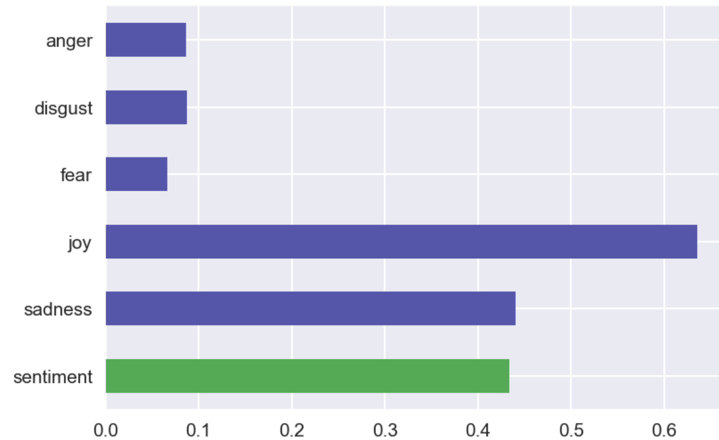


Sentiment & Personality Analytics with IBM Watson

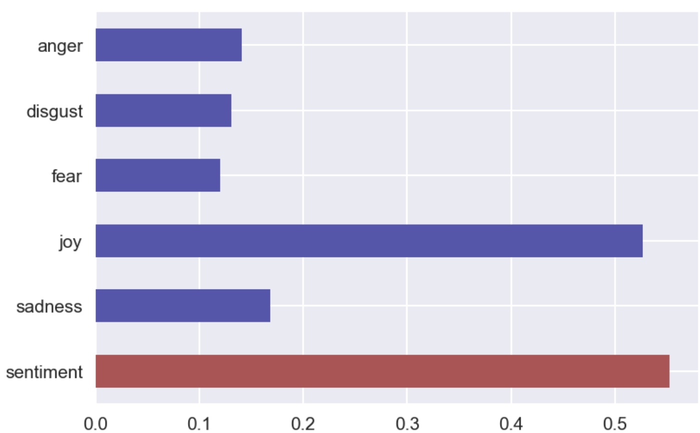
Random tweets subset



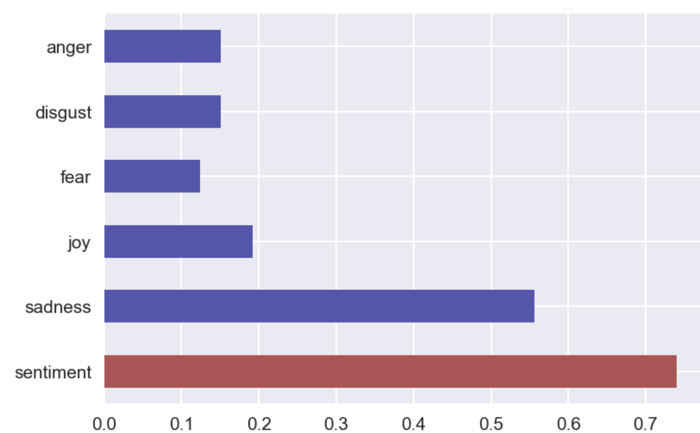
Hashtag ‘teslamodel3’



Keyword ‘walmart’

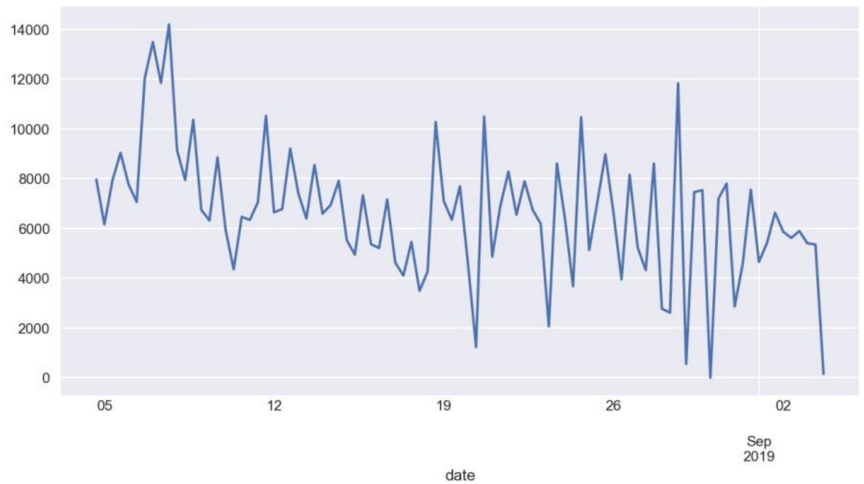


Hashtag ‘teslasolarissues’

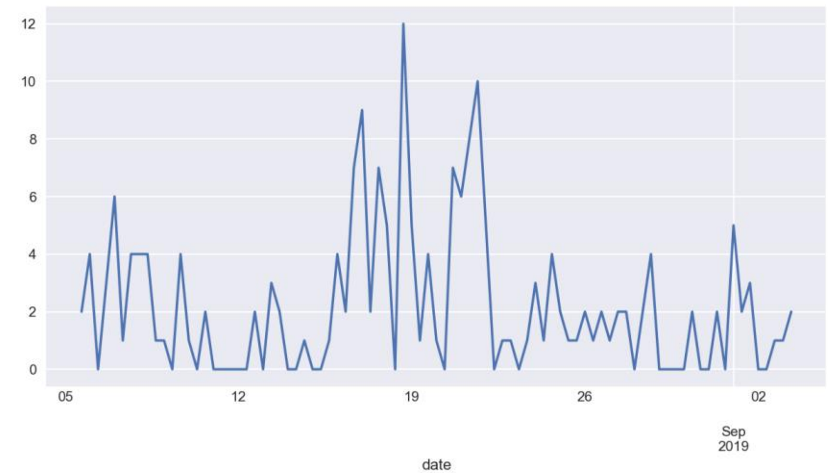


Keywords Time Series

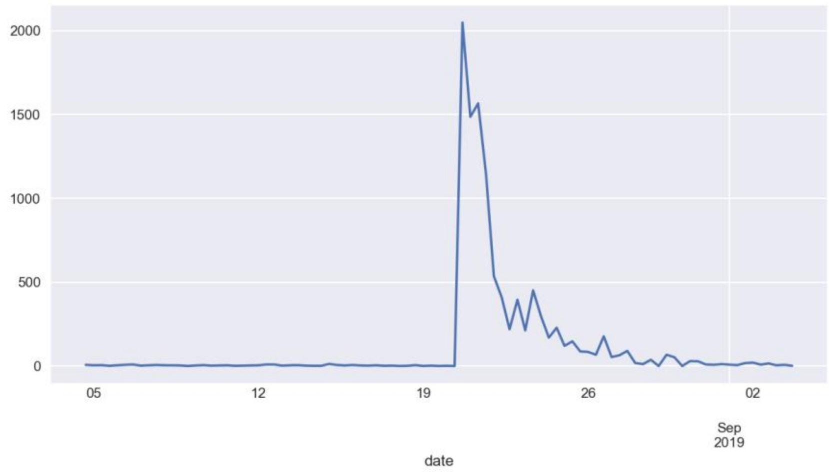
Tweets time series (keyword 'tesla')



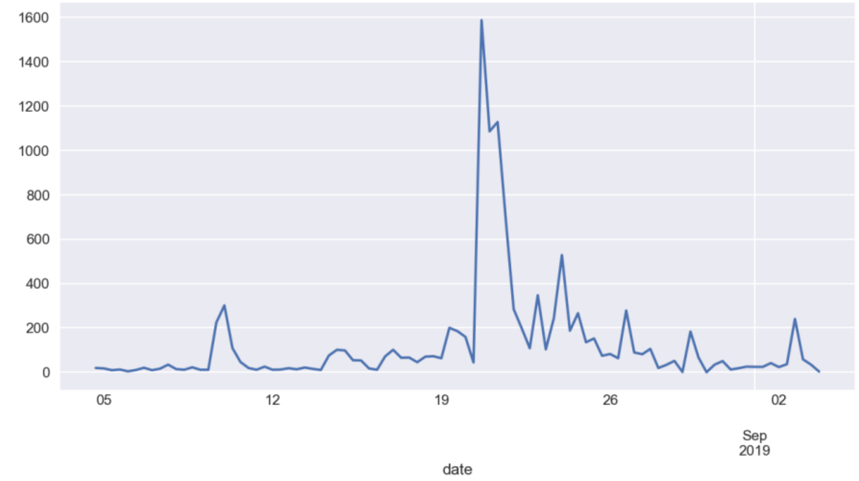
Tweets time series (hashtag 'teslasolarissues')



Tweets time series (keyword 'walmart')

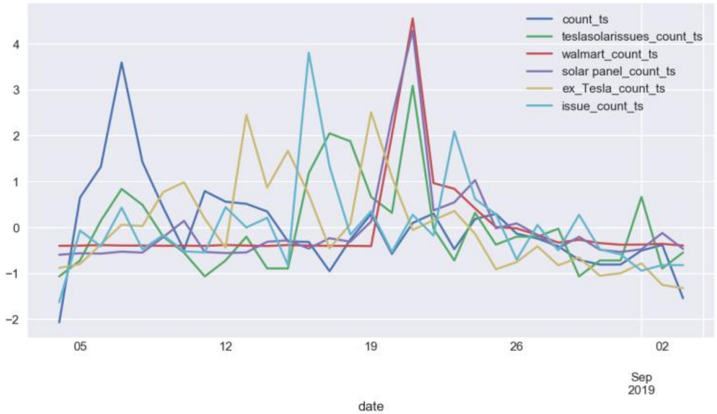


Tweets time series (keyword 'solar panel')



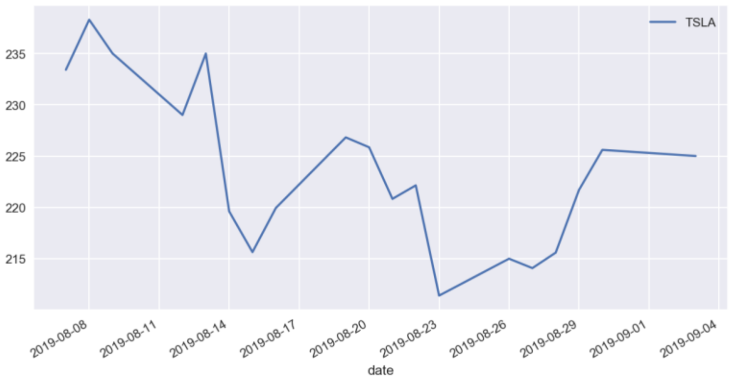
Stock Price Predictive Analytics (ticker TSLA)

Normalized time series for keywords



Modeling of stock price return using Lasso regression

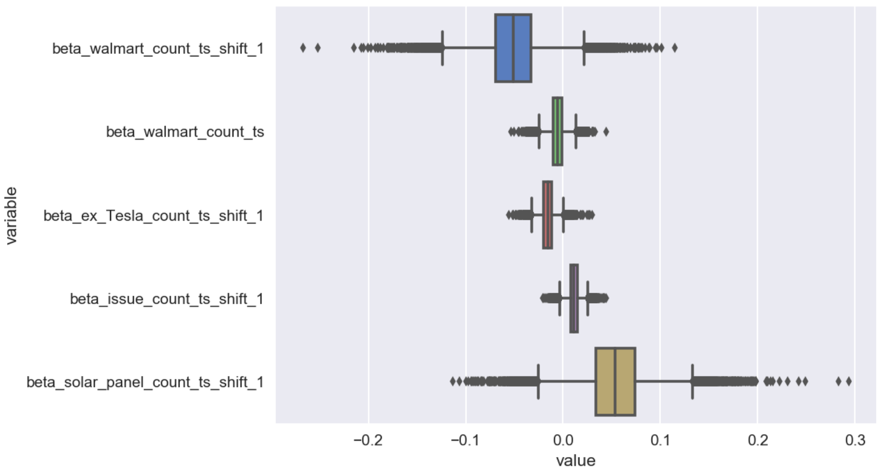
Time Series for stock price



Coefficients for tweets keywords features in Lasso regression model for price return

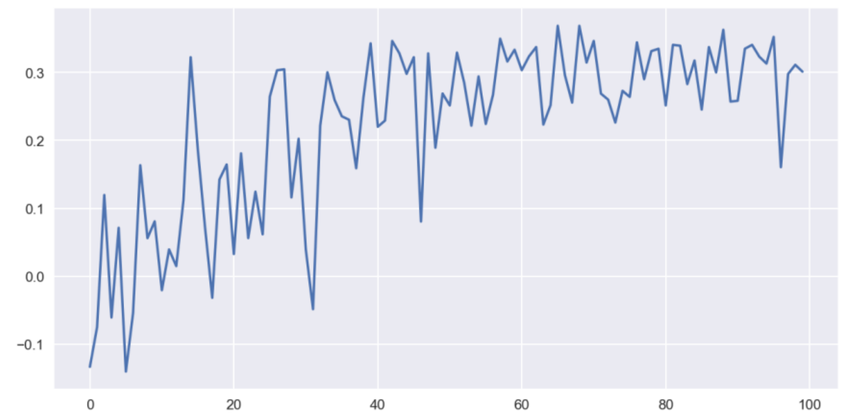
Stock Price Predictive Analytics (ticker TSLA)

Boxplots for coefficients in Bayesian regression model



Q-Learning for Stock Price Analytics (ticker TSLA)

Price return for the episodes



Thank you !