# Sales Forecast For Sporting Goods

Bengi Koseoglu, Na Gong, Qian Xia, Sanjita Suresh

# AGENDA

Problem Statement

Exploratory Data Analysis

- Items
- Price
- Train

Pre-processing

- Data Cleaning
- Feature Engineering
- Combining

Feature Selection

Predictive Analytics

- Clustering
- Predictive Modeling
- Ensemble Learning

# PROBLEM STATEMENT

- E-commerce sporting goods company

- Goal: Predict the sold out date of the products for February
  - Stock at the begining of the month
  - Sales unit of each day
  - Sales data between october  2017 and January 2018 that covers 12824 unique products

- Solution : Predict the daily sales of each products and substract it from the stock.
- Tools: Python, R, RapidMiner

| Product | Day | Pred | Remaing stock |
|---|---|---|---|
| Id1 | 01.02.2019 | 0 | 4 |
| Id1 | 02.02.2019 | 1 | 3 |
| id1 | 02.03.2019 | 3 | 0 |

# EXPLORATORY DATA ANALYSIS

We have three datasets

- **Items**: serves as master data
- **Train**: daily sales of products
- **Price**: historical pricing information between october and february

Items

| | pid | size | color | brand | rrp | mainCategory | category | subCategory | stock | releaseDate |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10000 | XL ( 158-170 ) | gruen | Nike | 25.33 | 1 | 7 | 25.0 | 1 | 2017-10-01 |
| 1 | 10001 | L | schwarz | Jako | 38.03 | 1 | 7 | 16.0 | 1 | 2017-10-01 |
| 2 | 10003 | 3 (35-38 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |
| 3 | 10003 | 4 ( 39-42 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |
| 4 | 10003 | 5 ( 43-46 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |

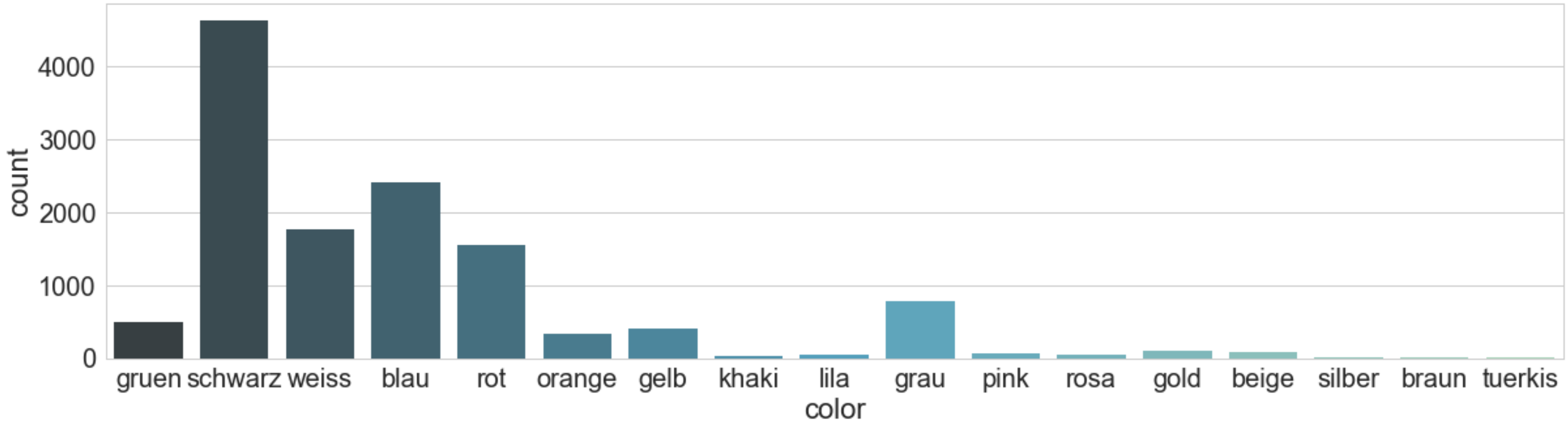
Train

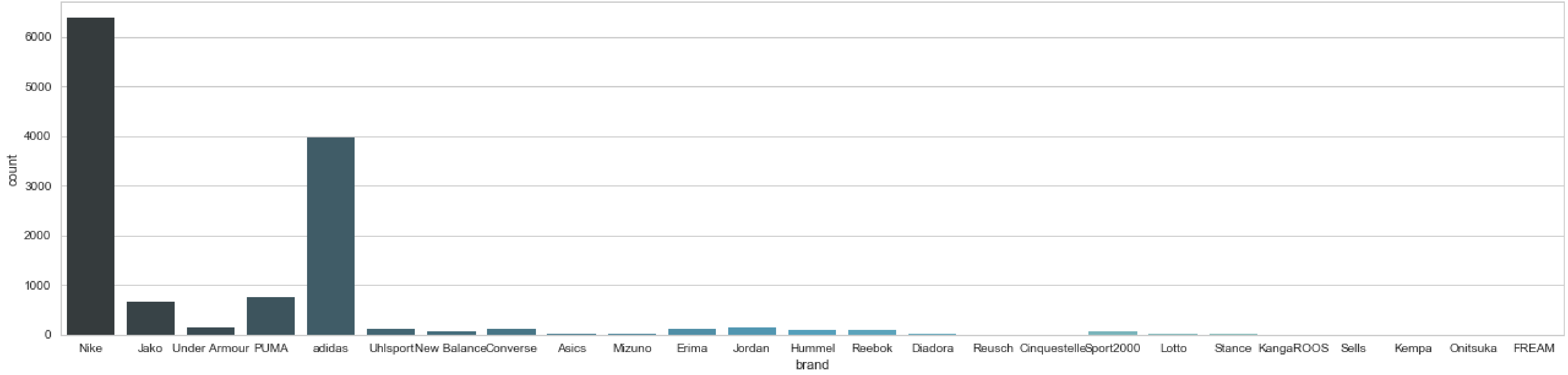
| | date | pid | size | units |
|---|---|---|---|---|
| 0 | 2017-10-01 | 14393 | 2 ( 37-39 ) | 1 |
| 1 | 2017-10-01 | 10069 | 36 | 2 |
| 2 | 2017-10-01 | 10069 | 35 | 1 |
| 3 | 2017-10-01 | 16221 | L | 1 |
| 4 | 2017-10-01 | 11317 | L | 1 |

Price

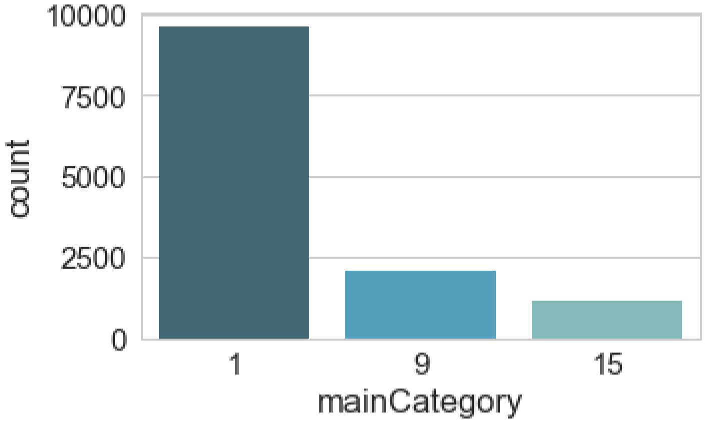
| | pid | size | 2017-10-01 | 2017-10-02 | 2017-10-03 | 2017-10-04 | 2017-10-05 | 2017-10-06 | 2017-10-07 | 2017-10-08 | ... | 2018-02-19 | 2018-02-20 | 2018-02-21 | 2018-02-22 | 2018-02-23 | 2018-02-24 | 2018-02-25 | 2018-02-26 | 2018-02-27 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 19671 | 39 1/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 |
| 1 | 19671 | 40 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 |
| 2 | 19671 | 41 1/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 |
| 3 | 19671 | 42 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 |
| 4 | 19671 | 42 2/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 |

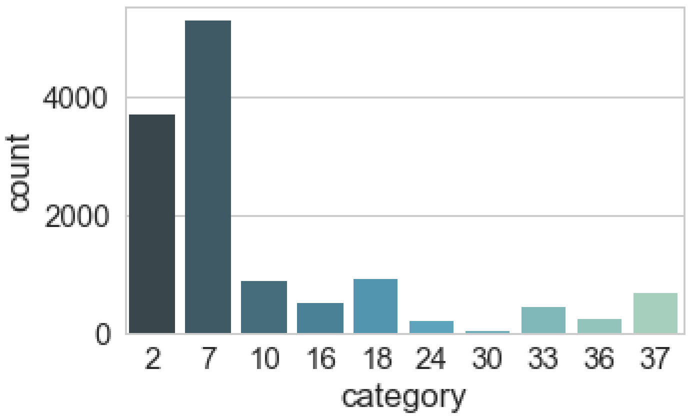
# EXPLORATORY DATA ANALYSIS

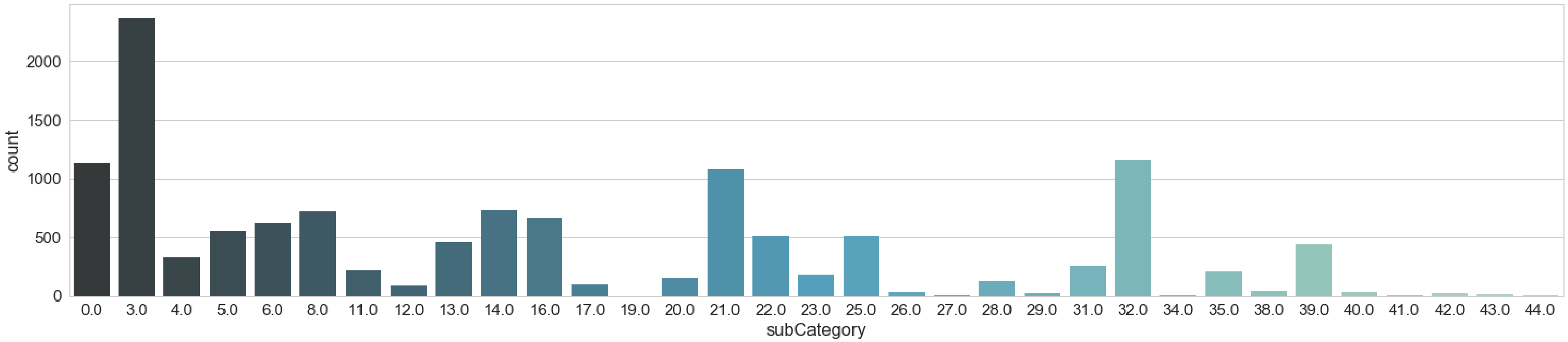


# EXPLORATORY DATA ANALYSIS

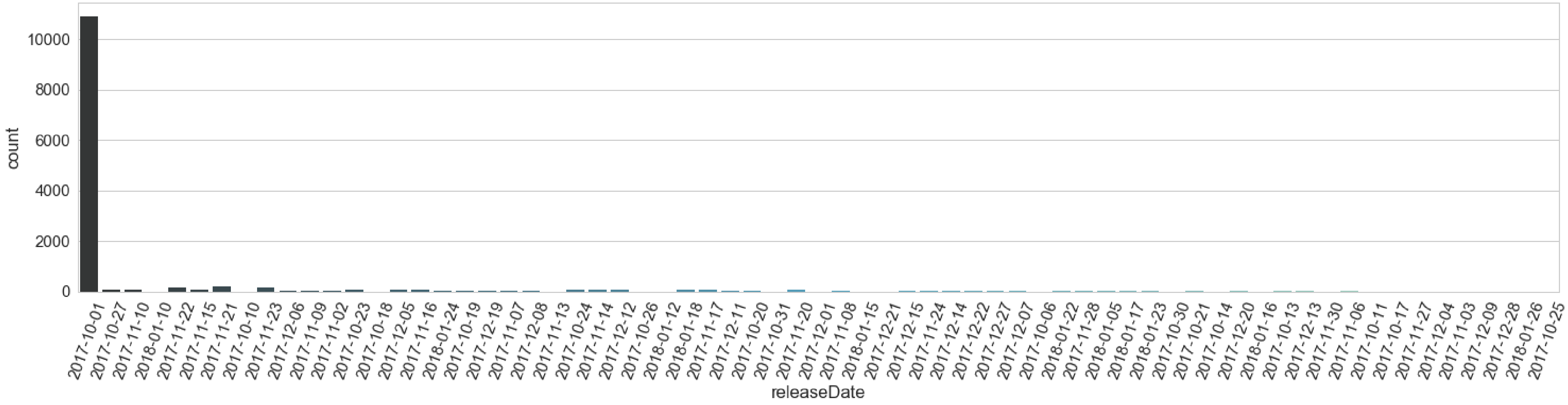


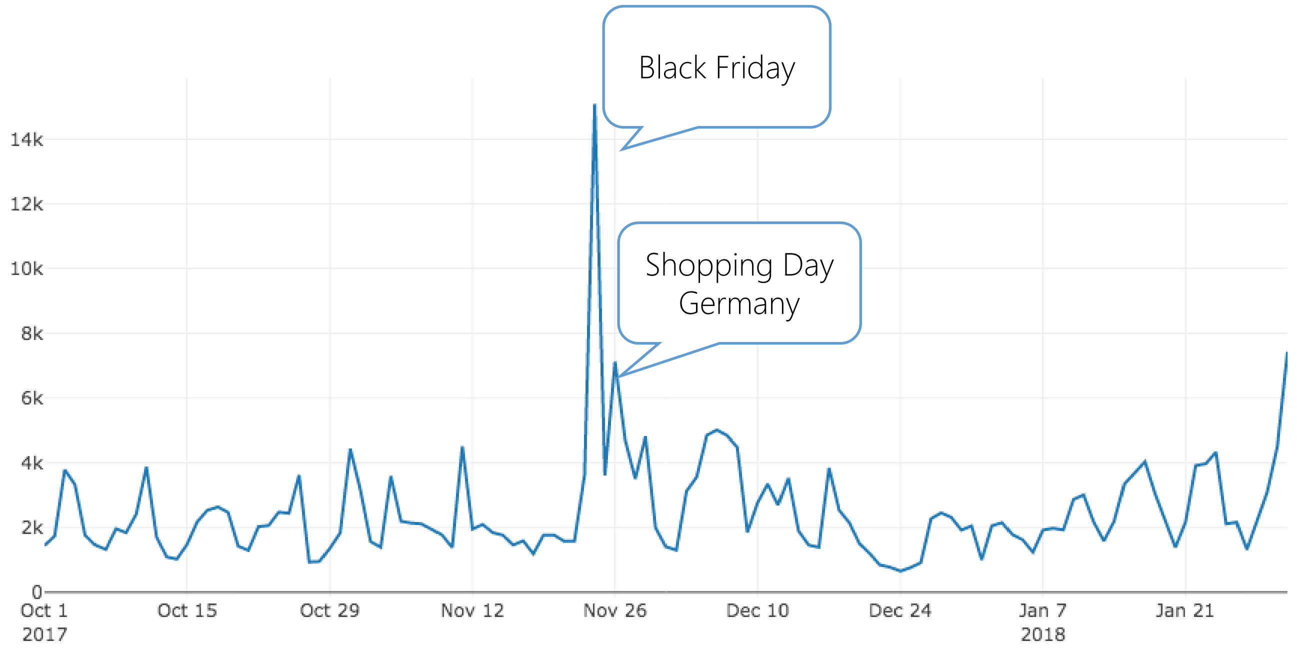



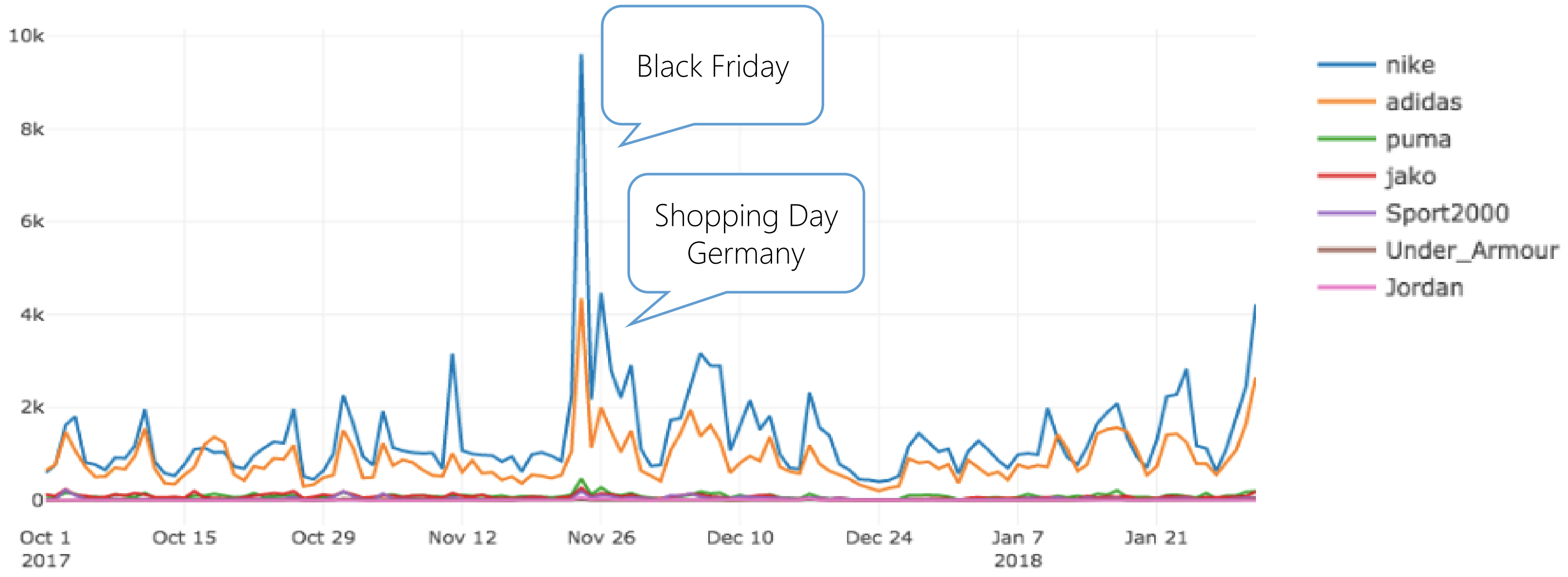
# EXPLORATORY DATA ANALYSIS

| | rrp | stock |
|---|---|---|
| **count** | 12824.000000 | 12824.000000 |
| **mean** | 98.526149 | 3.532829 |
| **std** | 90.787734 | 11.034285 |
| **min** | 2.470000 | 1.000000 |
| **25%** | 38.030000 | 1.000000 |
| **50%** | 69.780000 | 1.000000 |
| **75%** | 114.230000 | 2.000000 |
| **max** | 463.480000 | 459.000000 |



# EXPLORATORY DATA ANALYSIS



# PRE-PROCESSING (Data Cleaning)

- A unique column id is created
- Missing variables are handled
  - Price attributes in *price* dataset: mean of the product's average price
  - Subcategory attribute in *items* dataset: created another category
  - For size attribute in *items* dataset: filled with most frequent size of each brand
- Size information translated into a unified format

```
#size
print('n unique values=%s'%len(items['size'].unique()))
items.groupby('size').pid.nunique()
```

```
n unique values=179
```

```
#groupped_size
print('n unique values=%s'%len(items['groupped_size'].unique()))
items.groupby('groupped_size').pid.nunique()
```

```
n unique values=28
```

```
array(['XL ( 158-170 )', 'L', '3 (35-38 )', '4 ( 39-42 )', '5 ( 43-46 )',
       'XL', 'M', 'S', '140', '43', '44', '45', 'L ( 152-158 )',
       'XS ( 116-128 )', '46', '37,5', '42', 'M ( 140-152 )', '176',
       '39 1/3', '41 1/3', '44 2/3', '46 2/3', '48', '2 ( 37-39 )',
       '4 ( 43-45 )', '33', '34', '35', '36', '37 1/3', '45,5',
       'L ( 40/42 )', 'XL ( 44/46 )', '36,5', '41', '38', '39', '2XL',
       '7 ( L )', '43 1/3', '40', '40 2/3', '45 1/3', '40,5', '44,5',
       '152', '164', 'S ( 128-140 )', '3 ( 40-42 )', '5 ( 46-48 )',
       'L ( 42-46 )', 'M ( 38-42 )', 'S ( 34-38 )', 'XL (46-50 )',
       'XS ( 30-34 )', '36 2/3', '38,5', '38 2/3', '38/40 ( M / L )',
       '42 2/3', 'M ( 38/40 )', '33,5', '2 ( 35-38 )', '3 ( 39-42 )',
       '4 ( 43-46 )', '5 ( 47-49 )', '42,5', '164/176', '1 ( Junior)',
       '35,5', '128', '39/42', '43/46', '47', '47 1/3', 'XL (46-48,5)',
       'XS', '2 ( Senior )', nan, '116', '30', '32', '3XL', '41 - 44',
       '47,5', 'S ( 34/36 )', '6', '48 2/3', '37', '12 (41-45)', '39,5',
       '9', '31', '35 - 38', '39 - 42', '43 - 46', '1 ( 31-34 )', '41,5',
       '3', 'YLG 147,5-157,5', 'XS ( 32/34 )', '31,5', '8 ( XL )',
       '0 ( 31-33 )', '1 ( 34-36 )', '3 ( 41-43 )', 'M ( 40 )', '2XL/T',
       '43,5', '4XL', '116/128', '140/152', '2', 'XS ( 32 )',
       '0 ( Bambini )', '46,5', 'YXL 157,5-167,5', '35/38', '10 (36-40)',
       '29', '10 (140)', 'L (43 - 46)', '45 - 47', '14/16 (164-176)',
       '14 (46-48)', '00 ( 27-30 )', '102 (M)', '37 - 40', '6 ( 47-50 )',
       'L/XL ( 39-47 )', 'S ( 36 )', 'M (38 - 42)', '1 ( 140 )',
       '47 - 50', '47/49', '48,5', '0 ( 128 )', '11', '5', '7', '8', '4',
       'L ( 42-47 )', 'M/L', '2 ( 152 )', '3 ( 164 )', '1 ( 33-36 )',
       'YM 135-147,5', '1 ( 25-30 )', '2 ( 31-34 )', '10', '43-46',
       '6/8 (116-128)', '30 (5XL)', '134', '146', '158', '2 ( 37-40 )',
       '45-48', 'XS/S', '39-42', '3XL/T', 'XL/T', '4 ( 44-46 )', 'L/K',
       '24 (M)', '28 (3XL)', 'L/T', '19 (38)', 'YSM 125-135', 'L ( 44 )',
       '01 Junior', '02 Senior', '104', '116-122', '10/12 (140-152)',
       '14 (164)', '16 (176)'], dtype=object)
```

# PRE-PROCESSING(Feature Engineering)

- Price_daily_change : Price change of product compared to previous day
- New_product: Binary variable, based on release day
- Day: Day of the month
- Month: Month as categorical variable
- Weekday: Monday, Tuesday etc. as numerical variable

- Holiday: Binary variable (Christmas, school holiday)
- Avg_temp / Med_temp : average and median weather information of Germany

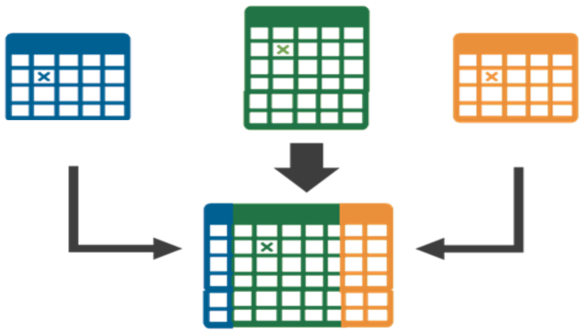
| | pid | size | color | brand | rrp | mainCategory | category | subCategory | stock | releaseDate |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10000 | XL ( 158-170 ) | gruen | Nike | 25.33 | 1 | 7 | 25.0 | 1 | 2017-10-01 |
| 1 | 10001 | L | schwarz | Jako | 38.03 | 1 | 7 | 16.0 | 1 | 2017-10-01 |
| 2 | 10003 | 3 (35-38 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |
| 3 | 10003 | 4 ( 39-42 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |
| 4 | 10003 | 5 ( 43-46 ) | weiss | Jako | 12.63 | 1 | 7 | 13.0 | 1 | 2017-10-01 |

| | date | pid | size | units |
|---|---|---|---|---|
| 0 | 2017-10-01 | 14393 | 2 ( 37-39 ) | 1 |
| 1 | 2017-10-01 | 10069 | 36 | 2 |
| 2 | 2017-10-01 | 10069 | 35 | 1 |
| 3 | 2017-10-01 | 16221 | L | 1 |
| 4 | 2017-10-01 | 11317 | L | 1 |

| | pid | size | 2017-10-01 | 2017-10-02 | 2017-10-03 | 2017-10-04 | 2017-10-05 | 2017-10-06 | 2017-10-07 | 2017-10-08 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 19671 | 39 1/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... |
| 1 | 19671 | 40 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... |
| 2 | 19671 | 41 1/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... |
| 3 | 19671 | 42 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... |
| 4 | 19671 | 42 2/3 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | 133.31 | ... |

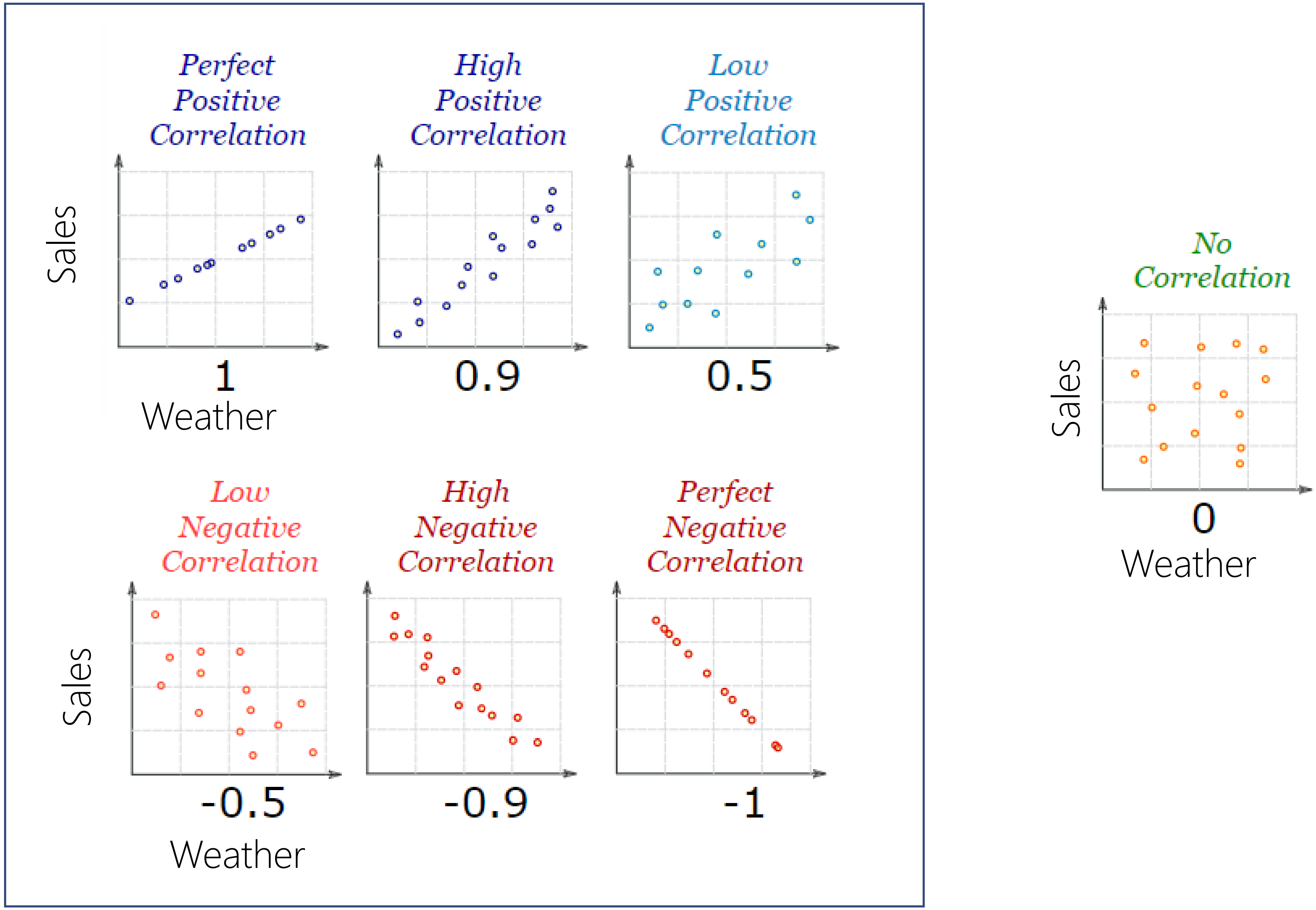
# PRE-PROCESSING (Combining)

- Categorical variables are convered to dummy variables
- Black Friday is removed from the dataset
- All datasets are combined -> 1.564.528 rows and 210 columns



| key | weekday | day | month | date | rrp | new size_L | new size_M | new size_S | ... | new size_44 | new size_43 | units | avg_temp | median_temp | company_offer | holiday | sum_unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19671 39 1/3 | 6 | 1 | 10 | 2017-10-01 | 190.43 | 0 | 0 | 0 | ... | 0 | 0 | 0.0 | 12.5625 | 12.50 | 0 | 0 | 0.0 |
| 19671 39 1/3 | 0 | 2 | 10 | 2017-10-02 | 190.43 | 0 | 0 | 0 | ... | 0 | 0 | 0.0 | 13.3125 | 13.75 | 0 | 0 | 0.0 |
| 19671 39 1/3 | 1 | 3 | 10 | 2017-10-03 | 190.43 | 0 | 0 | 0 | ... | 0 | 0 | 0.0 | 12.1875 | 12.50 | 0 | 1 | 0.0 |
| 19671 39 1/3 | 2 | 4 | 10 | 2017-10-04 | 190.43 | 0 | 0 | 0 | ... | 0 | 0 | 0.0 | 10.7500 | 10.75 | 0 | 0 | 0.0 |
| 19671 39 1/3 | 3 | 5 | 10 | 2017-10-05 | 190.43 | 0 | 0 | 0 | ... | 0 | 0 | 1.0 | 11.7500 | 11.50 | 0 | 0 | 1.0 |

# FEATURE SELECTION



# FEATURE SELECTION

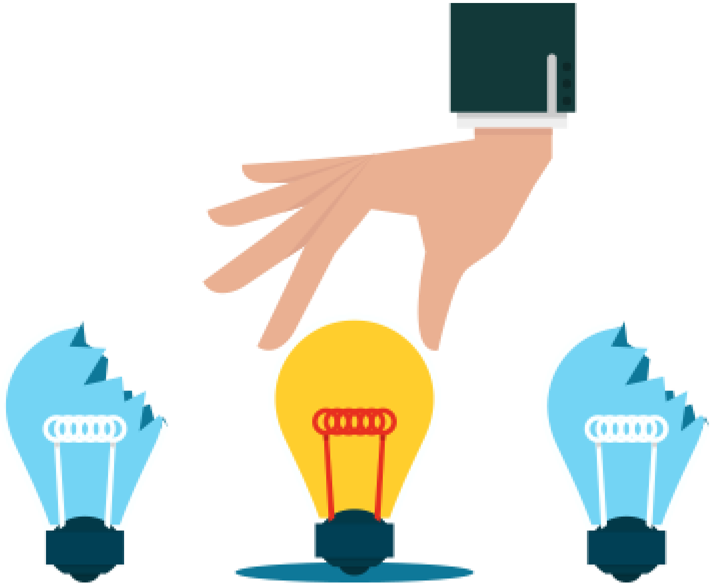
- New Product
- New size_m
- New size_l
- Brand: nike
- Brand: adidas
- Brand: sport2000
- Color: blau
- Color: grau
- Color: schwarz
- Color: weiss
- Temp: avg
- Temp: med
- Day: 5
- Day: 6

........

- Weekday_1
- Weekday_2
- Weekday_3
- Weekday_4
- Weekday_5
- Weekday_6
- Weekday_7
- Price_daily_change
- Holiday

........

34 VARIABLES
ARE SELECTED



# AGENDA

Problem Statement

Exploratory Data Analysis

- Items
- Price
- Train

Pre-processing

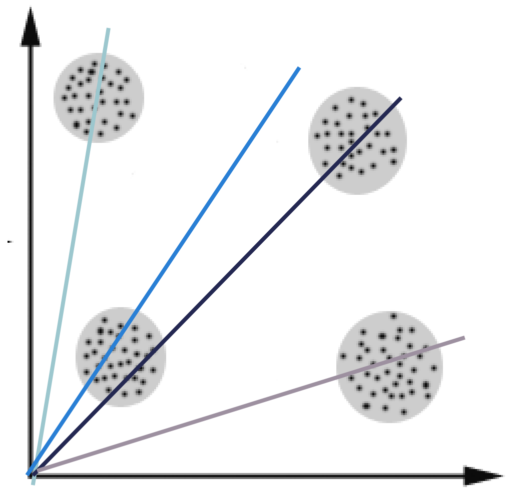
- Data Cleaning
- Feature Engineering
- Combining

Feature Selection

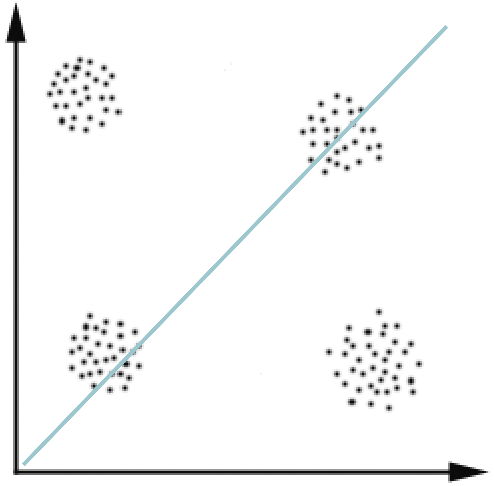
Predictive Analytics

- Clustering
- Predictive Modeling
- Ensemble Learning

# PREDICTIVE ANALYTICS



Clustering
+
Predictive Modeling



Predictive
Modeling on the
whole data

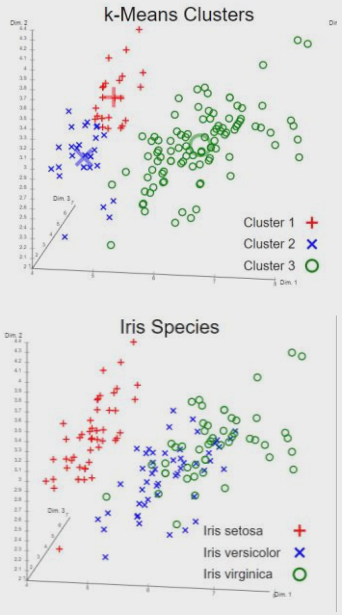
# PREDICTIVE ANALYTICS (Clustering)

## Brand Clustering

- Nike
- Adidas
- PUMA
- Jako
- Other brands

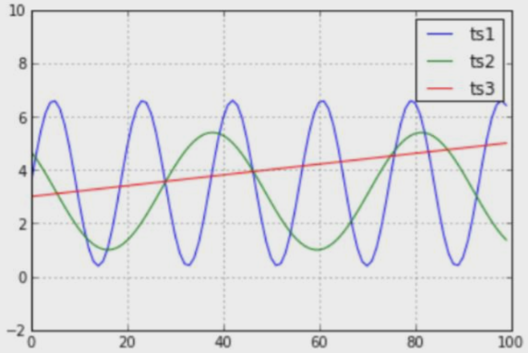
## Category Clustering

- mainCategory_1
- mainCategory_9
- mainCategory_15




## K- Means Clustering

- Input: items (!no sale info)
- Result: items that are similar to each other




d(ts1,ts2)=26.9
d(ts1,ts3)=23.2

## Dynamic Time Warping Clustering

- Input: Sale trend for each product
- Result: items that have similar sales units and trend across time

# PREDICTIVE ANALYTICS (Clustering)

Actual Day

Prediction Day

$$E = \sqrt{\sum_i |d_i - \hat{d}_i|}.$$

Actual Sold out date = 24th of June
Predicted Sold out date = 16 th of June

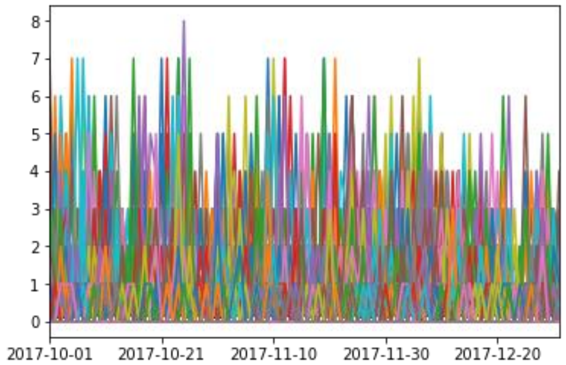
Difference= 8 days

| Clustering | Performance |
|---|---|
| K-means | 280 |
| Main Category | 276.6 |
| Brand | 276.1 |
| Time Series | 261 |

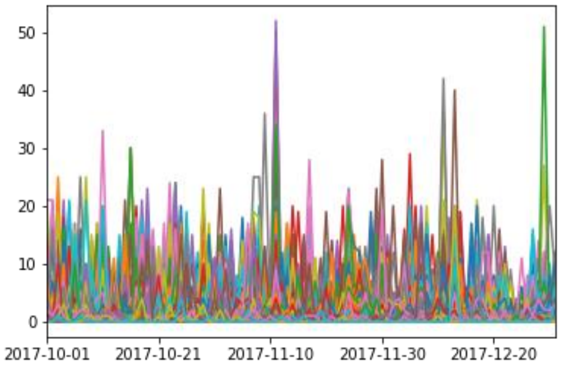
* Gboost regression without parameter tunning

# PREDICTIVE ANALYTICS (Clustering)

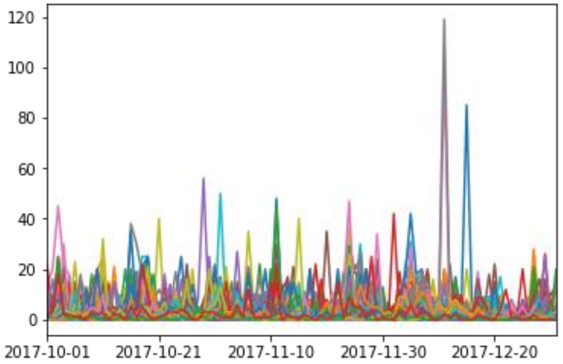
Cluster 1



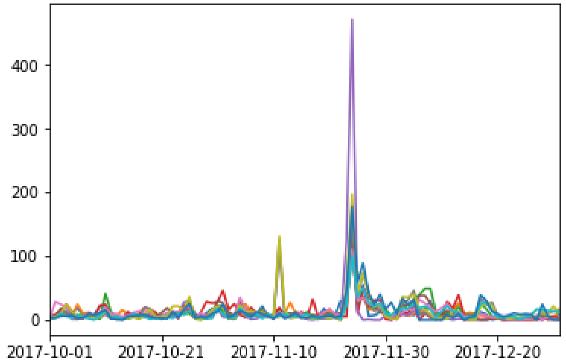

Cluster 2



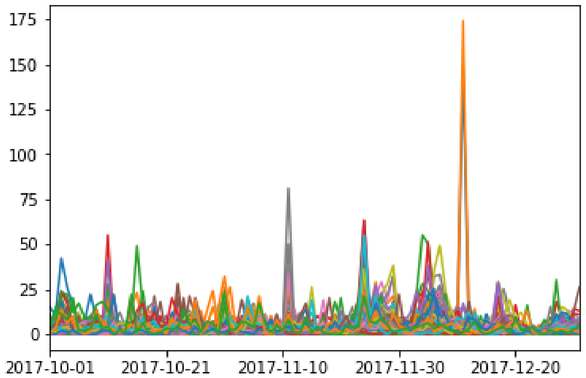

Cluster 3



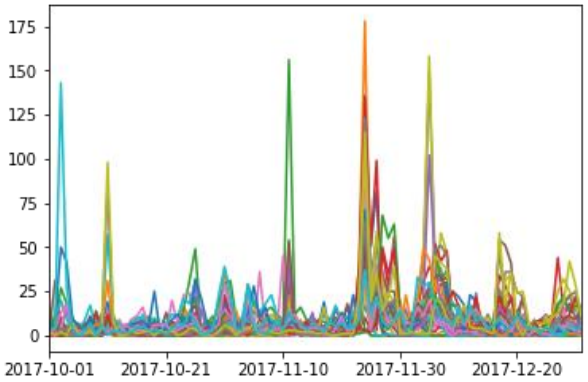

Cluster 4



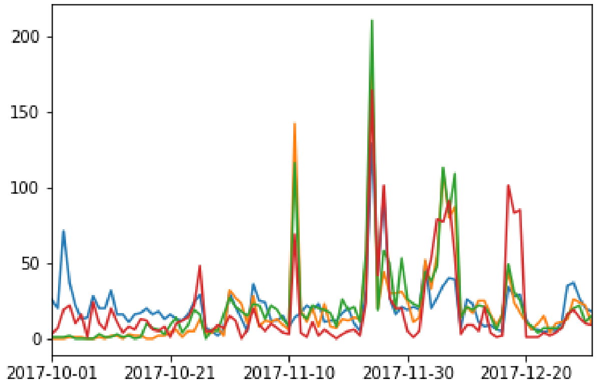

Cluster 5




Cluster 6




Cluster 7



# PREDICTIVE ANALYTICS (Modeling)

## SALES FORECASTING

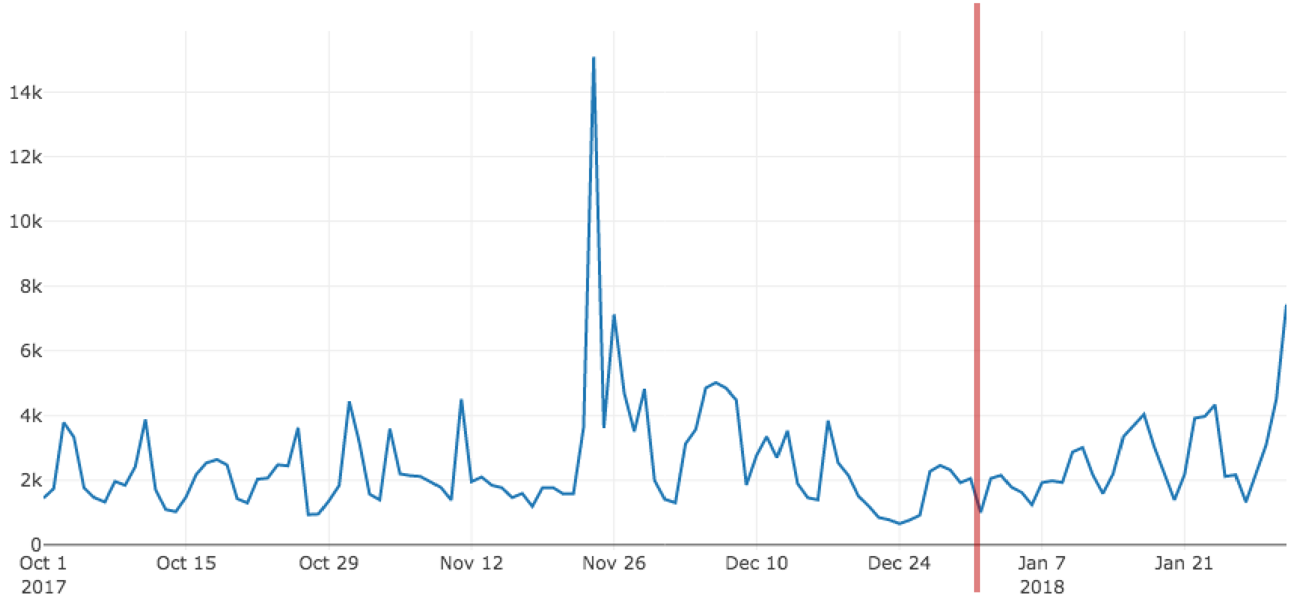
01 **ARIMA Model**
Pure time series model

02 **Windowing Approach**
Treating time series forecast a regression problem

03 **Regression Model**
Predict daily unit sales for each day by using regression

04 **Combined Model**
Train different models for each cluster

# PREDICTIVE ANALYTICS (Modeling)

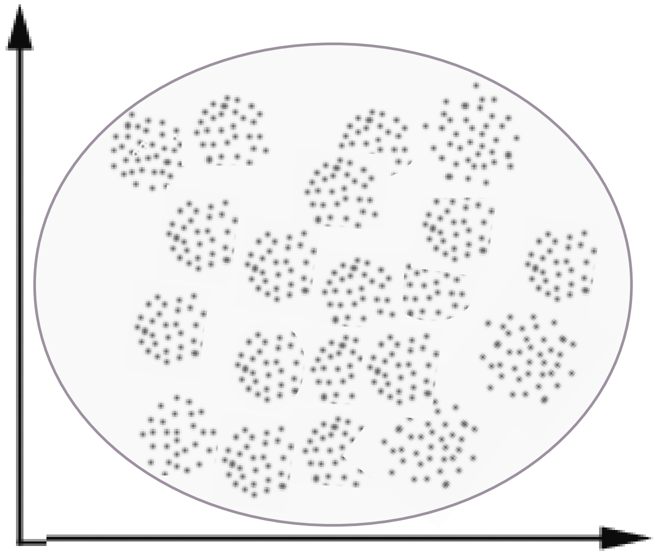



Train

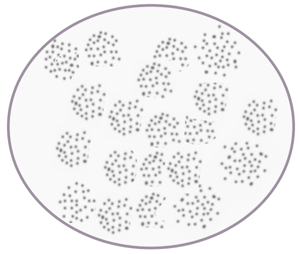
Test

# PREDICTIVE ANALYTICS (ARIMA)

$$E = \sqrt{\Sigma_i |d_i - \hat{d}_i|}.$$

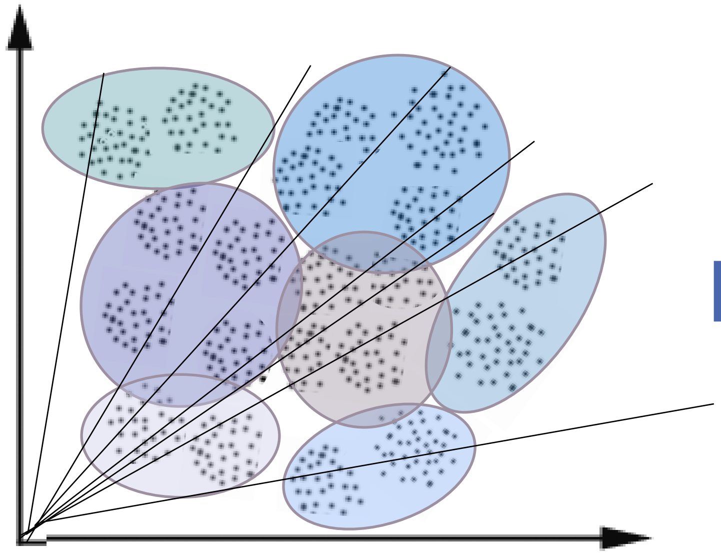


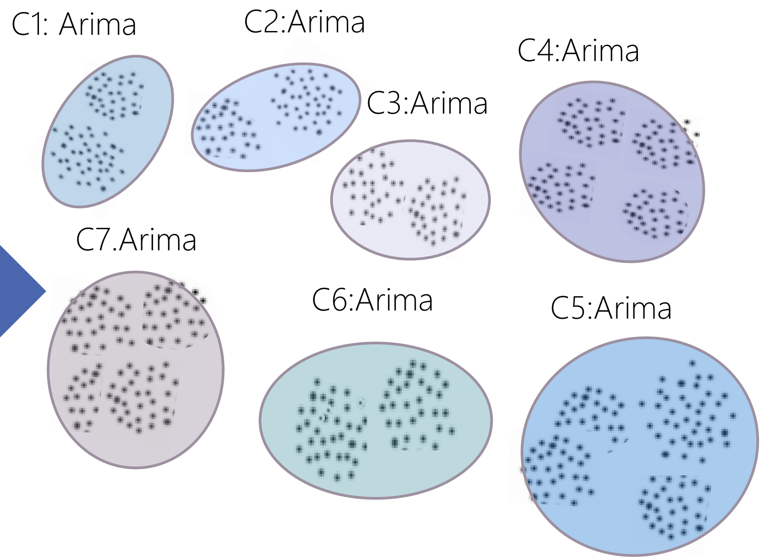
Arima All



Direct Result

| Model | Performance |
|---|---|
| ARIMA All | 280 |
| ARIMA Time Clustering | 276.6 |






Average

$$*Arima = \frac{C1: Arima + C2: Arima + C3: Arima + C4: Arima + C5: Arima + C6: Arima + C7: Arima}{7}$$

# PREDICTIVE ANALYTICS (Windowing)

| Key | Day | Sales |
|---|---|---|
| 19671 39 1/3 | 2017-10-01 | 3 |
| 19671 39 1/3 | 2017-10-02 | 0 |
| 19671 39 1/3 | 2017-10-03 | 1 |
| 19671 39 1/3 | 2017-10-04 | 2 |
| 19671 39 1/3 | 2017-10-05 | 4 |
| 19671 39 1/3 | 2017-10-06 | 5 |
| | .. | |
| 19671 39 1/3 | 2017-12-30 | 7 |
| 19671 39 1/3 | 2017-12-31 | 0 |

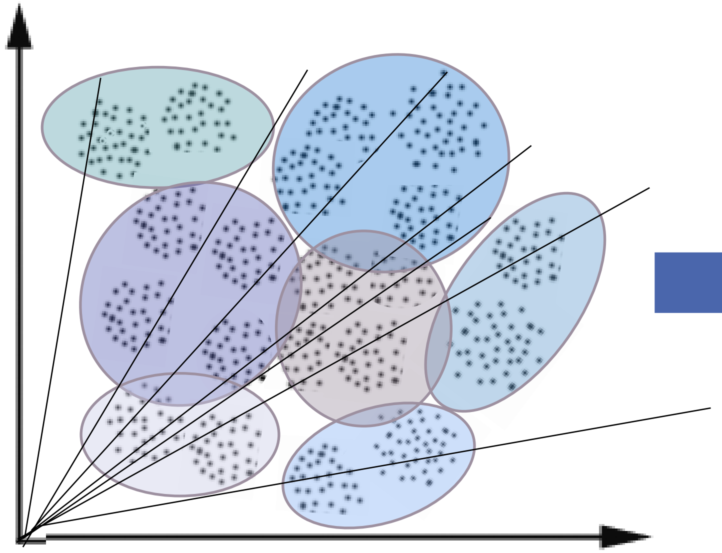
Lag 1

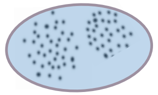
| Key | Sales(t-1) | Sales(t) |
|---|---|---|
| 19671 39 1/3 | 3 | 0 |
| 19671 39 1/3 | 0 | 1 |
| 19671 39 1/3 | 1 | 2 |
| 19671 39 1/3 | 2 | 4 |
| 19671 39 1/3 | 4 | 5 |
| | .. | |
| 19671 39 1/3 | 6 | 7 |
| 19671 39 1/3 | 7 | 0 |

| Key | Sales (t-4) | Sales (t-3) | Sales (t-2) | Sales (t-1) | Sales (t) |
|---|---|---|---|---|---|
| 19671 39 1/3 | 3 | 0 | 1 | 2 | 4 |
| 19671 39 1/3 | 0 | 1 | 2 | 4 | 5 |
| 19671 39 1/3 | 1 | 2 | 4 | 5 | 6 |
| | .. | .. | .. | .. | .. |
| 19671 39 1/3 | 9 | 0 | 2 | 7 | 0 |

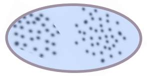
Lag 4

# PREDICTIVE ANALYTICS (Windowing)

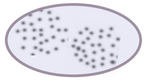




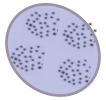
C1: W1 (1) Reg
C1: W2 (20) Ann
C1: W3 (20) Reg
C1: W4 (30) Reg



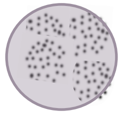
C2: W1 (1) Reg
C2: W2 (20) Ann
C2: W3 (20) Reg
C2: W4 (30) Reg



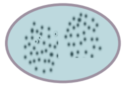
C3: W1 (1) Reg
C3: W2 (20) Ann
C3: W3 (20) Reg
C3: W4 (30) Reg



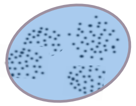
C4: W1 (1) Reg
C4: W2 (20) Ann
C4: W3 (20) Reg
C4: W4 (30) Reg



C5:W1 (1) Reg
C5 W2 (20) Ann
C5: W3 (20) Reg
C5 W4 (30) Reg



C6: W1 (1) Reg
C6 W2 (20) Ann
C6: W3 (20) Reg
C6 W4 (30) Reg



C7: W1 (1) Reg
C7 W2 (20) Ann
C7: W3 (20) Reg
C7 W4 (30) Reg

Average

$$*W1 = \frac{\text{C1: W1 (1) Reg + C2: W1 (1) Reg + C3: W1 (1) Reg + C4: W1 (1) Reg + C5: W1 (1) Reg + C6: W1 (1) Reg + C7: W1 (1) Reg}}{7}$$

$$*W2 = \frac{\text{C1: W2 (20) ANN + C2: W2 (20) ANN + C3: W2 (20) ANN + C4: W2(20) ANN + C5: W2 (20) ANN + C6: W2(20) ANN + C7: W2(20) ANN}}{7}$$

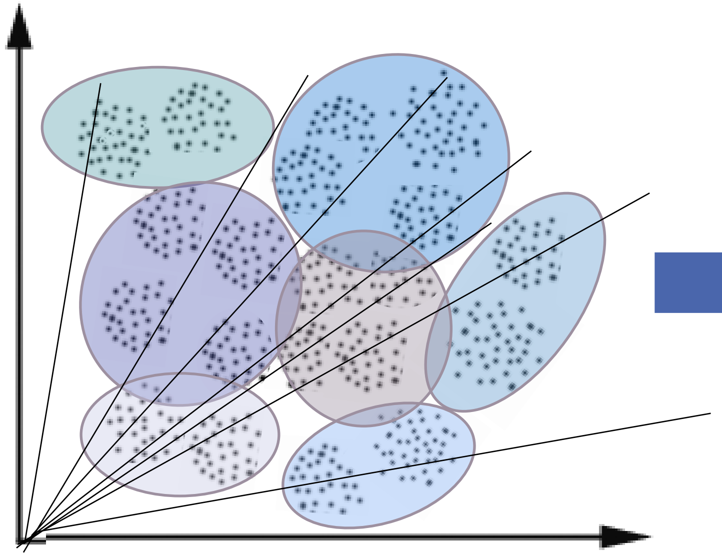
$$*W3 = \frac{\text{C1: W3 (20) Reg + C2: W3 (20) Reg + C3: W3 (20) Reg + C4: W3(20) Reg + C5: W3 (20) Reg + C6: W3(20) Reg + C7: W3(20) Reg}}{7}$$

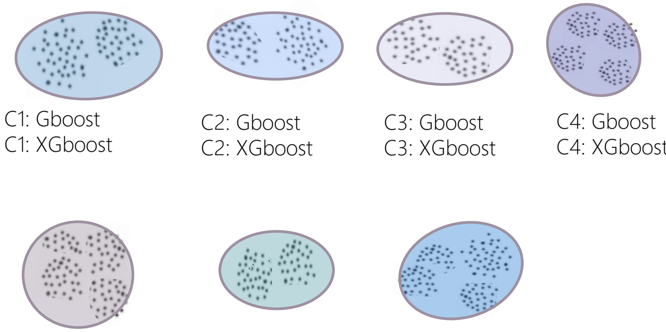
$$*W4 = \frac{\text{C1: W4 (30) Reg + C2: W4 (30) Reg + C3: W4 (30) Reg + C4: W4 (30) Reg + C5: W4 (30) Reg + C6: W4 (30) Reg + C7: W4 (30) Reg}}{7}$$

| Model | Performance |
|---|---|
| W1: Time Cluster (1) Reg | 254 |
| W2: Time Cluster (20) ANN | 250 |
| W3: Time Cluster (20) Reg | 252 |
| W4. Time Cluster (30) Reg | 266 |

$$E = \sqrt{\Sigma_i \left| d_i - \hat{d}_i \right|}.$$

# PREDICTIVE ANALYTICS (Regression)





C1: Gboost
C1: XGboost

C2: Gboost
C2: XGboost

C3: Gboost
C3: XGboost

C4: Gboost
C4: XGboost

C5: Gboost
C5 XGboost

C6: Gboost
C6: XGboost

C7: Gboost
C7: XGboost

Average

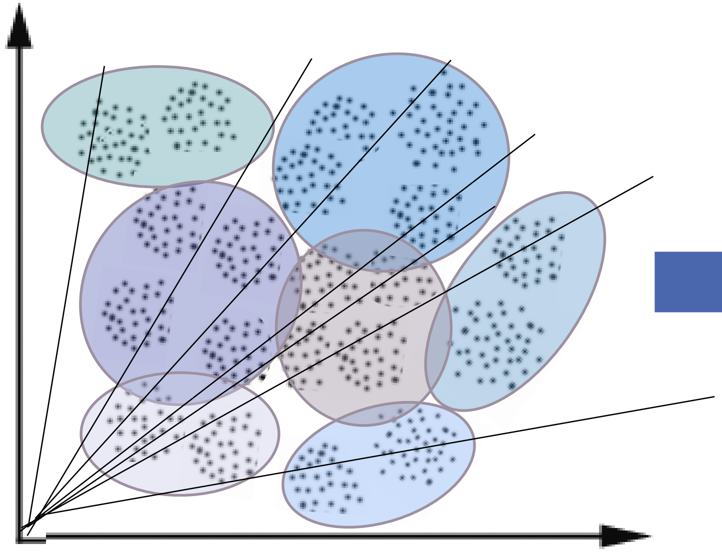
*Gboost = $\frac{\text{C1: Gboost + C2: Gboost + C3: Gboost + C4: Gboost + C5: Gboost + C6: Gboost + C7: Gboost}}{7}$

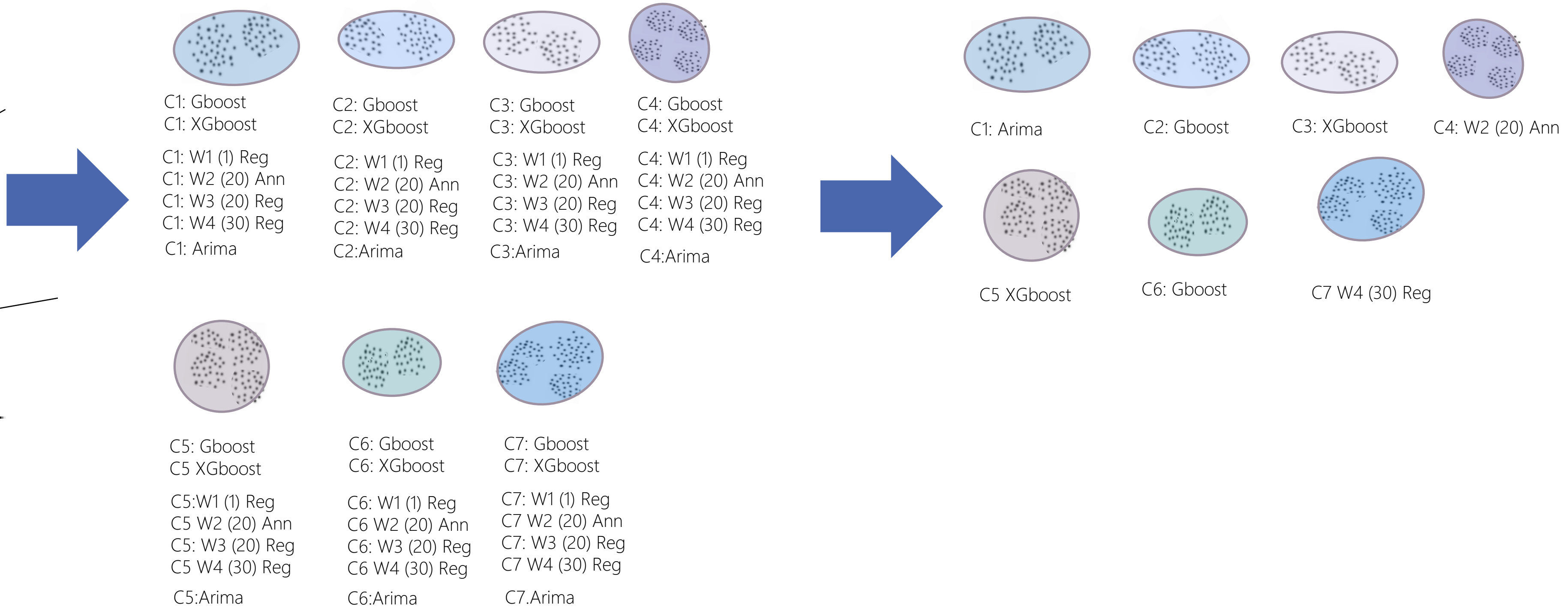
*XGboost = $\frac{\text{C1: XGboost + C2: XGboost + C3: XGboost + C4: XGboost + C5: XGboost + C6: XGboost + C7: XGboost}}{7}$

| Model | Performance |
|---|---|
| Time Cluster + Gboost Regressíon | 253 |
| Time Cluster + Xgboost Regression | 254 |

$$E = \sqrt{\Sigma_i |d_i - \hat{d}_i|}.$$

# PREDICTIVE ANALYTICS (Combined)

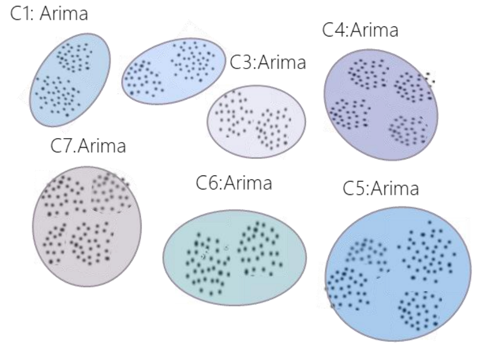





| Model | Performance |
|---|---|
| Combine Model-1 | 248.7 |
| Combine Model-2 | 247.0 |
| Combine Model-3 | 246.8 |
| Combine Model-4 | 245.8 |

$$E = \sqrt{\sum_i |d_i - \hat{d}_i|}.$$

# PREDICTIVE ANALYTICS (Modeling)

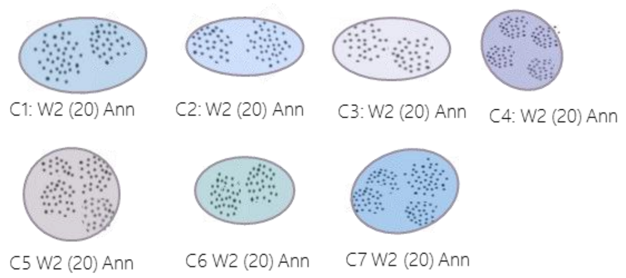



276.6

01

## ARIMA Model

Pure time series model

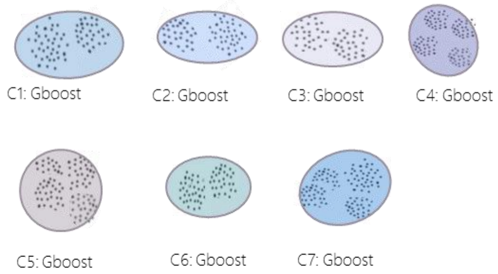



250

02

## Windowing Approach

Treating time series forecast a regression problem

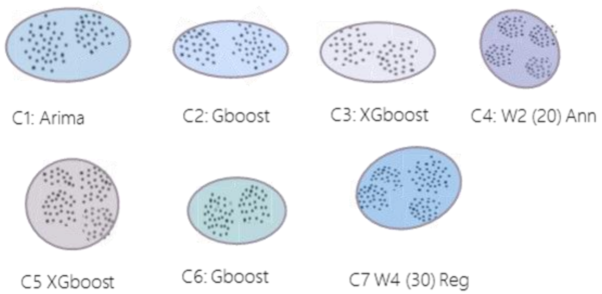



253

03

## Regression Model

Predict daily unit sales for each day by using regression




245

04

## Combined Model

Train different models for each cluster

# PREDICTIVE ANALYTICS (Lesson Learnt)

- Important variables that have an impact on sales:
  - Weather
  - Color of the product
  - Brand of the product
  - Product sold date (5th day of the month)
  - Category of the product
  - Product is new or not
  - Size of the product

- Variables that doesn‘t have a significant impact on sales:
  - Price daily change (a.k.a. discount)
  - Weekday of the month (Monday, Tuesday, etc.)

- For time series problems, time series clustering that takes sales trend into account yields the best resullts

- Clustering + Individual models for each clusters is the best technique

# THANK YOU