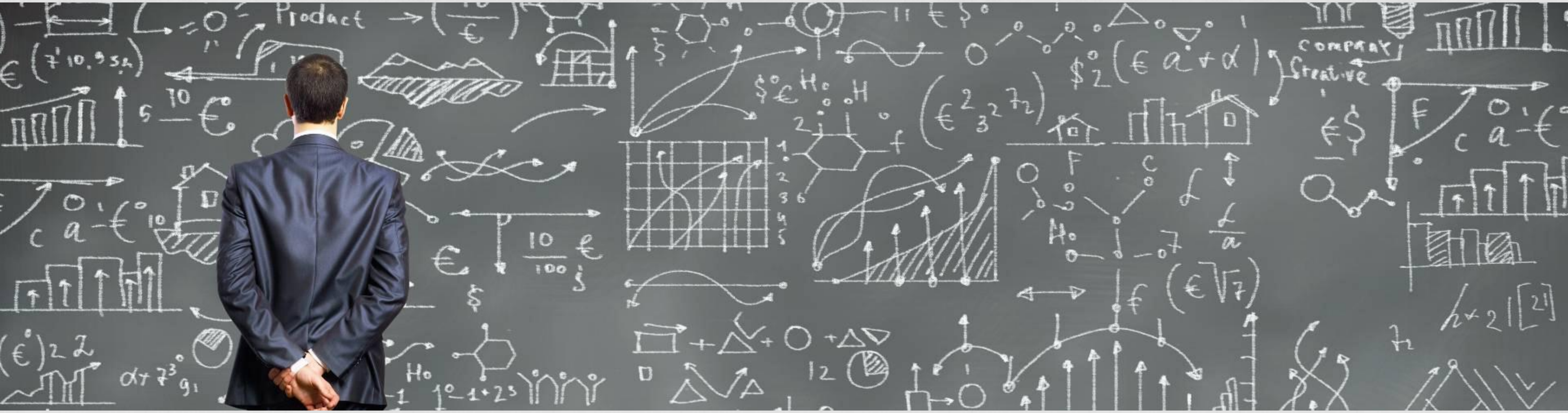
Khanh Ho



DATA SCIENCE PROJECT

FOOD DEMAND FORECAST

[05/07/2020]

AGENDA

FOOD DEMAND FORECAST

BACKGROUND / BUSINESS PROBLEM

OVERALL

Support the plan for food demand in the next 10 weeks

Situation	• Food demand forecast is a sample to track the business operation and control the inventory in Food and Beverage industry, particularly in online sales. Using the past data to train models aims to figure out the next 10 weeks food demand. • Based on the analysis and model, the organization has overview about upcoming numbers of meal orders. Further, with the feature importance, the corporate can manage the raw materials, change the sales and marketing activities or maybe prepare the shipping and staffs according to the regions or cities
Complication	• The amount of meal orders after aggregation is huge and has lots of outliers. Thus, need to consider how to reduce the violation and what types of models to support the predictions • The collecting and manipulating approaches has an impact on the evaluation metrics for each model and the final solution. Need some skills in Machine learning to train and test the set and to improve the model with tuning models

EXECUTIVE SUMMARY / KEY TAKEAWAYS

Approach & Solution

- Smooth the high fluctuation by taking log of the target variable 'num_orders' – number of meal orders in log
- Apply supervised ML to split the available data randomly from week 1 – 145 into the train and test set with the ratio 0.7: 0.3
- Conduct Linear Regression model and Random Forest model, choose the best model with lowest 100*RMSE (Root of mean squared logarithmic error) and interpret the coefficients in order to figure out the feature importance
- To do out-of-sample forecast in the time-series, the first step is tuning ARIMA model with optimal (p,d,q). Then, conduct the Dickey-Fuller test to examine the stationarity before doing forward predictions for the next 10 weeks based on the lags. After all, evaluate the efficiency of model

Evaluation metrics: 100*RMSE with supervised ML
Feature importance: Linear Regression and Random Forecast
Out-of-sample forecast: ARIMA model in the time-series

DATA SET CHARACTERISTICS

INFORMATION AND VISUALIZATION

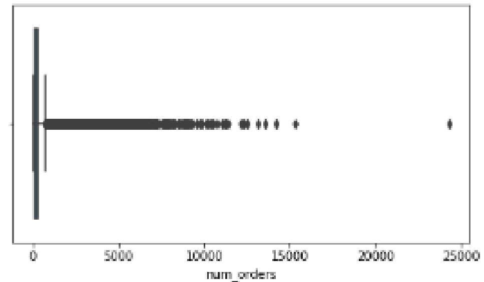
Dataset Information

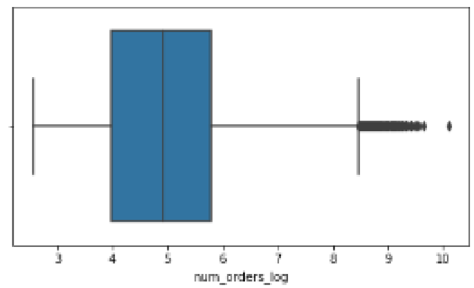
Dataset
- Size: 70.1MB
- Shape: Table with 3 separate dataset
- Observations: 456548
- Features: 15 in total
    - ➢ Categories: 4
    - ➢ Float64: 3
    - ➢ Int64: 9
- Label: num_orders or num_orders_log (float64)
- Missing values: no
- Duplicates: in weeks

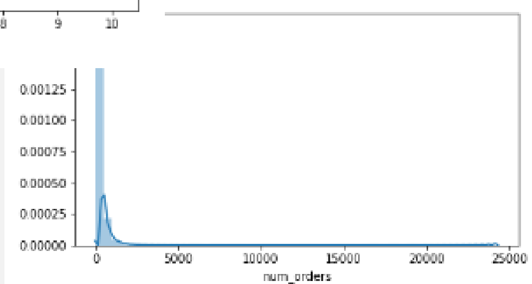
```
train_df.info()
```

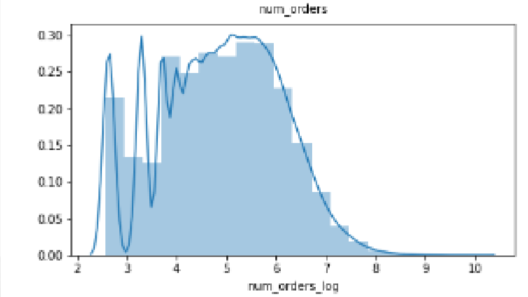
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 456548 entries, 0 to 456547
Data columns (total 16 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   id                     456548 non-null  int64
 1   week                   456548 non-null  int64
 2   center_id              456548 non-null  int64
 3   meal_id                456548 non-null  int64
 4   checkout_price         456548 non-null  float64
 5   base_price             456548 non-null  float64
 6   emailer_for_promotion  456548 non-null  int64
 7   homepage_featured      456548 non-null  int64
 8   num_orders             456548 non-null  int64
 9   city_code              456548 non-null  int64
 10  region_code            456548 non-null  int64
 11  center_type            456548 non-null  int8
 12  op_area                456548 non-null  float64
 13  category               456548 non-null  int8
 14  cuisine                456548 non-null  int8
 15  num_orders_log         456548 non-null  float64
dtypes: float64(4), int64(9), int8(3)
memory usage: 70.1 MB
```

Dataset Visualizations



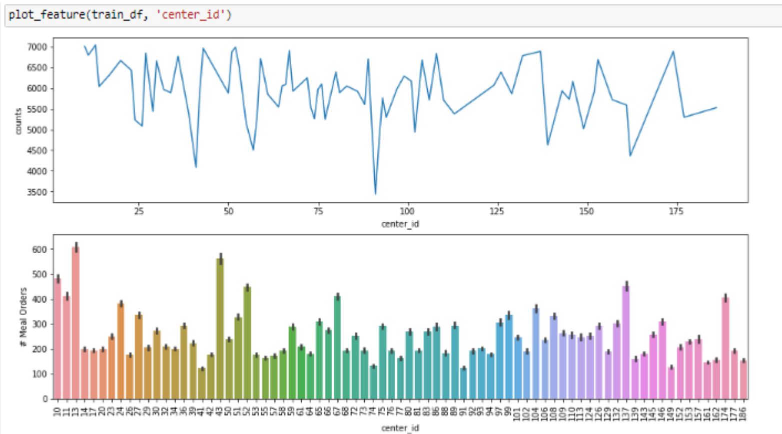






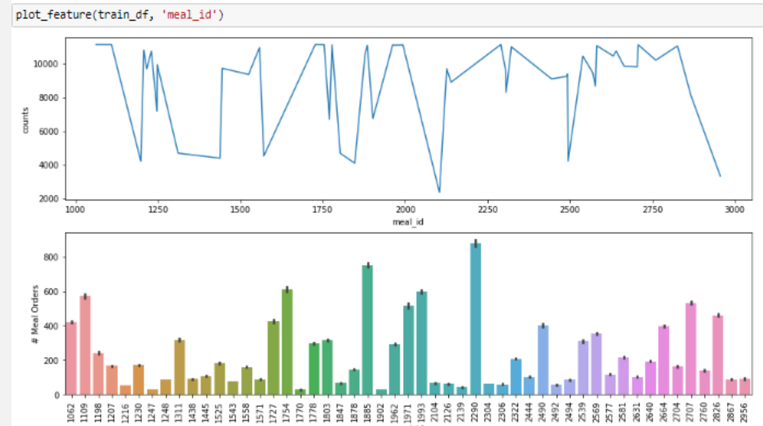
EDA – EXPLORATORY DATA ANALYSIS

Visualization EDA 1



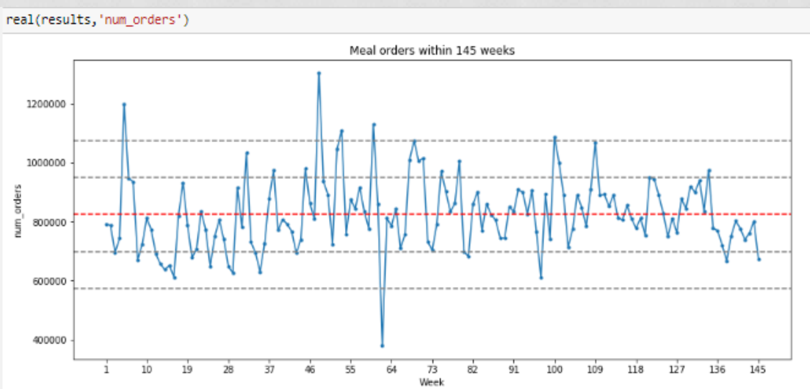
Correlation between num_orders and center_id

Visualization EDA 3



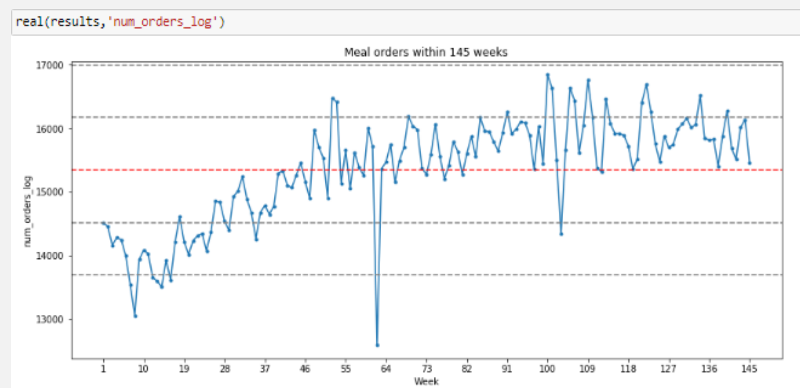
Correlation between num_orders and meal_id

Visualization EDA 2



Actual num_orders with many outliers

Visualization EDA 4



Log of num_orders with less outliers shows the trend

DATA CLEANSING & PRE-PROCESSING

[SLIDE TAG LINE]

Categorical Features

- 3 categories: 'center_type', 'category', 'cuisine'
- Encode them into 'int' to show the correlation in the heatmap

```
## Summarize categorical data
train_df.describe(include = ['O'])
```

	center_type	category	cuisine
count	456548	456548	456548
unique	3	14	4
top	TYPE_A	Beverages	Italian
freq	262881	127890	122925

```
## Encode the category by mean of num_orders_log
def encode_label(df, col):
    cat_dict = {}
    cats = df[col].cat.categories.tolist()
    for cat in cats:
        cat_dict[cat] = train_df[train_df[col] == cat]['num_orders'].mean()
    df[col] = df[col].map(cat_dict)
```

```
for col in train_df.columns:
    if train_df[col].dtype.name == 'category':
        encode_label(train_df, col)
```

Numerical Features

- Include: 'id', 'week', 'center_id', 'meal_id', 'checkout_price', 'base_price', 'emailer_for_promotion', 'homepage_featured', 'num_orders', 'city_code', 'region_code', 'op_area'

```
## Summarize numerical data
round(train_df.describe(include = [np.number]),2)
```

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	city_code	region_c
count	456548.00	456548.00	456548.00	456548.00	456548.00	456548.00	456548.00	456548.00	456548.00	456548.00	45654
mean	1250096.31	74.77	82.11	2024.34	332.24	354.16	0.08	0.11	261.87	601.55	56
std	144354.82	41.52	45.98	547.42	152.94	160.72	0.27	0.31	395.92	66.20	17
min	1000000.00	1.00	10.00	1062.00	2.97	55.35	0.00	0.00	13.00	456.00	23
25%	1124998.75	39.00	43.00	1558.00	228.95	243.50	0.00	0.00	54.00	553.00	34
50%	1250183.50	76.00	76.00	1993.00	296.82	310.46	0.00	0.00	136.00	596.00	56
75%	1375140.25	111.00	110.00	2539.00	445.23	458.87	0.00	0.00	324.00	651.00	77
max	1499999.00	145.00	186.00	2956.00	866.27	866.27	1.00	1.00	24299.00	713.00	93

Feature Engineering / Dimension Reduction

- Aggregate sum, group by week
- Reduce auto-correlated features
- Target variable/label: num_orders_log

Weekly meal orders

```
results = train_df.groupby('week').sum()
weeks = train_df.week.unique()
```

Feature engineering

```
# Remove some variables
results.drop(['id', 'base_price', 'num_orders'], axis = 1, inplace = True)
results
```

MODELLING, TUNING & EVALUATION

TARGET 1: FEATURE IMPORTANCE

Model Selection

model selection process:
- Supervised ML to split dataset into train and test set with ratio 0.7 : 0.3 randomly
- Regression vs. classification
- Model: Linear Regression and Random Forest
- Visualize the models to compare with reality: distribution and residual errors

Model Evaluation

model evaluation metrics
- Mainly: 100*RMSE: the lower, the better
- Score: the higher, the better

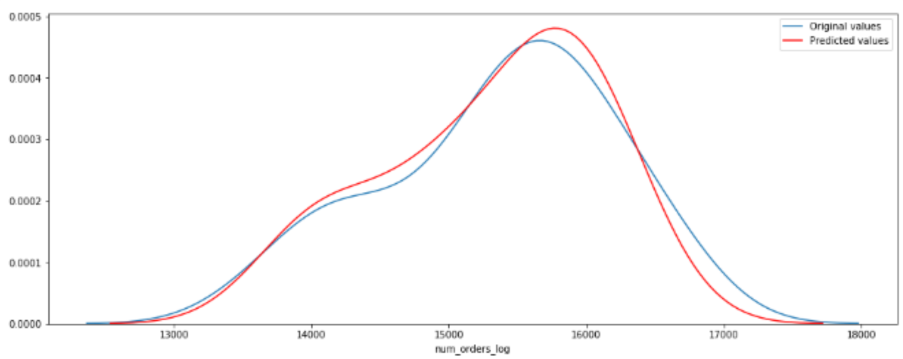
With A and B model, if A has smaller RMSE and score than B
Decision – making: still select model A

Model Performance Results

```
Linear Regression 100*RMSE =  0.0123049136812459
```

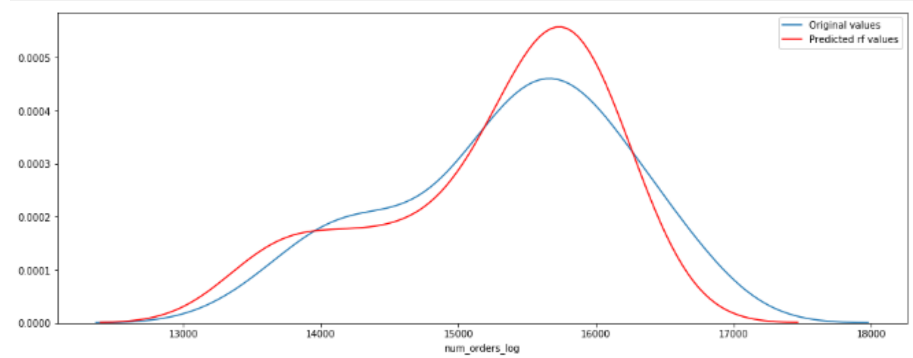
The 100*RMSLE is very small. This model is significant

```
plt.figure(figsize = (16,6))
ax1 = sns.distplot(y_test, hist = False, label = 'Original values')
ax2 = sns.distplot(y_pred, hist = False, color = 'r', label = 'Predicted values')
```




```
Random Forest 100*RMSE =  0.03181972168252317
```

```
plt.figure(figsize = (16,6))
ax1 = sns.distplot(y_test, hist = False, label = 'Original values')
ax2 = sns.distplot(y_pred_rf, hist = False, color = 'r', label = 'Predicted rf values')
```



The predicted data is much steeper in the center and more mass in the left tail, compared to the original ones

MODELLING, TUNING & EVALUATION

TARGET 2: OUT-OF-SAMPLE FORECAST

Model Selection

model selection process:
- Regression vs. Time series: ARIMA model
- Hyper parameter tuning: p, d, q
- Cross validation
- Show 10 week forward predictions

Model Evaluation

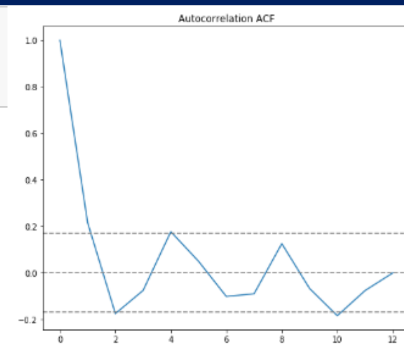
model evaluation metrics
- Mainly: 100*RMSE: the lower, the better

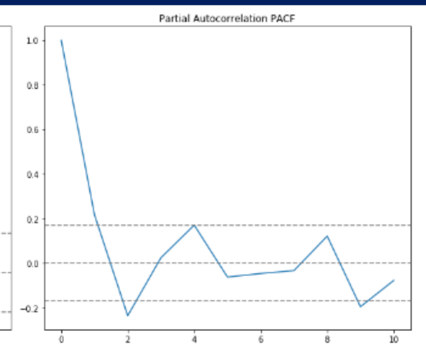
Model Performance Results

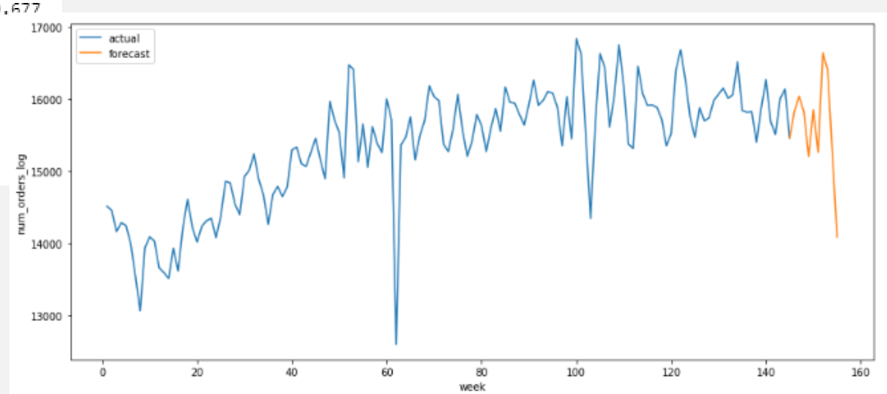
```
# Best the model ARIMA(2, 1, 1)
model = ARIMA(y, order = (2,1,1))
arima = model.fit(method = 'css', trend = 'c', disc = 0)
print(arima.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:       D.num_orders_log   No. Observations:                  144
Model:                 ARIMA(2, 1, 1)   Log Likelihood               -1074.264
Method:                           css   S.D. of innovations            466.985
Date:                Wed, 06 May 2020   AIC                           2158.527
Time:                        22:41:13   BIC                           2173.306
Sample:                             3   HQIC                          2164.533

==========================================================================================
                             coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                     12.7205      6.920      1.838      0.066      -0.842      26.283
ar.L1.D.num_orders_log     0.1950      0.098      1.997      0.046       0.004       0.386
ar.L2.D.num_orders_log    -0.2825      0.092     -3.085      0.002      -0.462      -0.103
ma.L1.D.num_orders_log    -0.8135      0.070    -11.668      0.000      -0.950      -0.677
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.3451           -1.8497j            1.8816           -0.2206
AR.2            0.3451           +1.8497j            1.8816            0.2206
MA.1            1.2292           +0.0000j            1.2292            0.0000
-----------------------------------------------------------------------------
```







- 100*RMSE: 0.0773

ANALYSIS RESULTS & RECOMMENDATIONS

INTERPRETATION

The key results and recommendations of the Analysis
- List and strength of the key predictors
- Performance of the model
- Business recommendation and outcomes

Result #1

- Feature importance: Linear Regression overweighs Random Forest model with lower 100*RMSE, even its score is also lower than the latter
- The feature importance: op-area (operation area) with negative impact around 5.2 and homepage_featured with positive high coefficient approximately 0.75 on the increase of meal orders

Result #2

- To ensure the stationarity, the forecast is based on the difference of target variable in t and t-1
- The result is reliable with 100*RMSE = 0.0773

Result #3

- Recommendations: combine more features with label to explore more insights
- For example, num_order_log with meal_id or with center_id

NEXT STEPS & IMPROVEMENTS

FURTHER RESEARCH

Lessons learned and possible project improvements or next steps if more time was permitted
- Source other data sets
- Research another data science technique
- Try other ideas

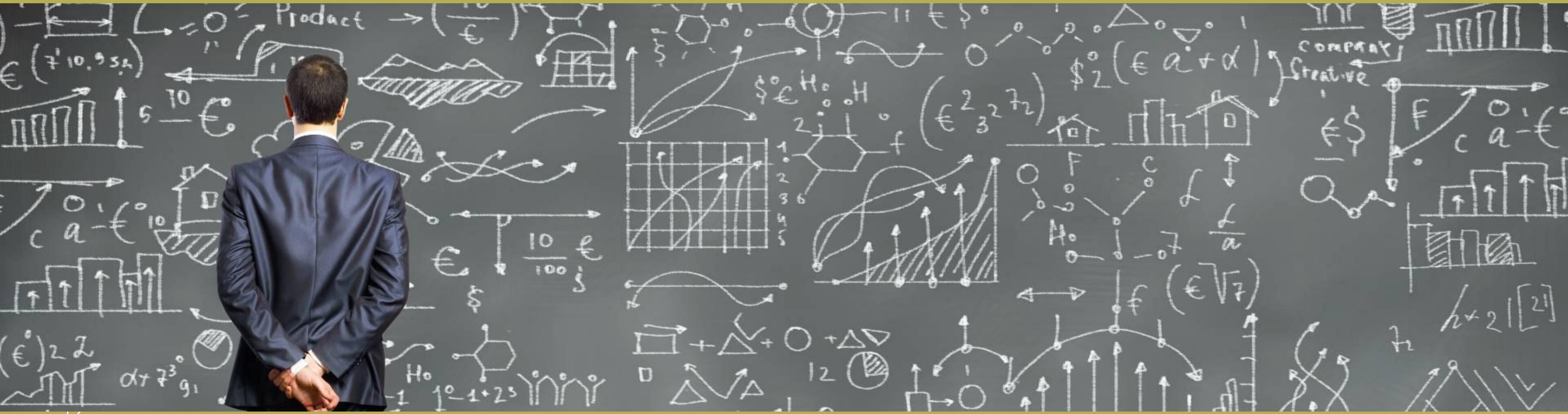
Project/Approach Improvements

- Conducting the product-center combination in the past to do prediction can show the food demand and the center related. This helps corporate plan well the shipping and the staffing issues
- The outliers in the dataset are large; however, to prevent removing the dataset, other investigations related to risk management and tail control should be proceeded
- Further, to improve the model, neuro-network models can be conducted to optimize the predictions

Lessons learned

1. Take time to identify clearly the goals before collecting and manipulating data
2. Take time to understand the dataset and plan the outlines is really helpful to do relational data, EDA and deploy solutions
3. Clarify every target to do suitable modelling in regards of data types and structures
4. Continue thinking about tuning models to get the optimal solutions
5. Keep track on the project and keep going to do improvements

APPENDIX



ASSUMPTIONS

Assumptions used for the analysis and obtaining results

1. **Linear Regression:**
   - Linearity between independent x (features) and dependent y (outcome/target/label)
   - Homoscedasticity: residual variance is similar among x
   - Independence: no autocorrelation
   - Normality

2. **Arima:** stationarity

3. Apply mainly the 100*RMSE in evaluation metrics to select the best model