CORPORACIÓN FAVORITA

Seafood weekly sales forecasting

Author:

YIFTACH BEINART

Introduction

Grocery stores are always in a delicate dance with purchasing and sales forecasting. Predict a little over, and grocers are stuck with overstocked, perishable goods. Guess a little under, and popular items quickly sell out, leaving money on the table and customers fuming. The problem becomes more complex as retailers add new locations with unique needs, new products, ever transitioning seasonal tastes, and unpredictable product marketing.

Corporación Favorita, a large Ecuadorian-based grocery retailer operates hundreds of supermarkets, with over 200,000 different products on their shelves. They challenged the Kaggle community to build a model that more accurately forecasts product sales, so they could improve the automation process for execute corporation plans using machine learning.

The purpose of this final project, is to practice and better understand the scientific method and process flow of a data science project. My chosen project is based on the FAVORITA challenge in Kaggle.

The objective is to choose and train a machine learning based model, which will eventually successfully forecast the weekly sales of SEAFOOD items in one type of stores of Corporación Favorita Ecuador.

Methodology (Project design)

Data

The data source in Kaggle contains 6 data sets shared by Corporación Favorita. It contains the items and their categories, geographical information about stores and much more.

Train (~125,000,000 rows) includes attributes such as store number, item number and the unit sale on a particular date.

Stores (54) includes attributes such as city, state, type and cluster of stores (cluster is a grouping of similar stores)

Items (4,100) includes attributes such as family and class, as well as if they are perishable or not

Transactions (~80,000 rows) includes details of transactions at a store on a particular date

Oil (~1,200 rows) includes daily oil prices, as Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices

Holidays (350 rows) includes holidays and events suspected to affect supermarket sales. It includes everything from Christmas to the World Soccer Cup in Brasil, from Black Friday to a 7.8 magnitude earthquake.

Dataset	variables	variable.count
train	id, date, store.nbr, item.nbr, unit.sales, onpromotion	6
oil	date, dcoilwtico	2
holidays	date, type, locale, locale.name, description, transferred	6
items	item.nbr, family, class, perishable	4
stores	store.nbr, city, state, type, cluster	5
transactions	date, store.nbr, transactions	3

Time frames periods

The raw data reflects the sales from Jan 2013 until Aug 15th 2017. Nevertheless, as the chosen objective is to predict the weekly sales over relative long period of time (1 year), the predictive model will not be Time Series based.

This project focus on stores of cluster 14, as seafood sales in those stores is significant compare to others. The stores are located at Ambato & Quito.

As can be seen, seafood sales distribution is pretty stable over the given years.

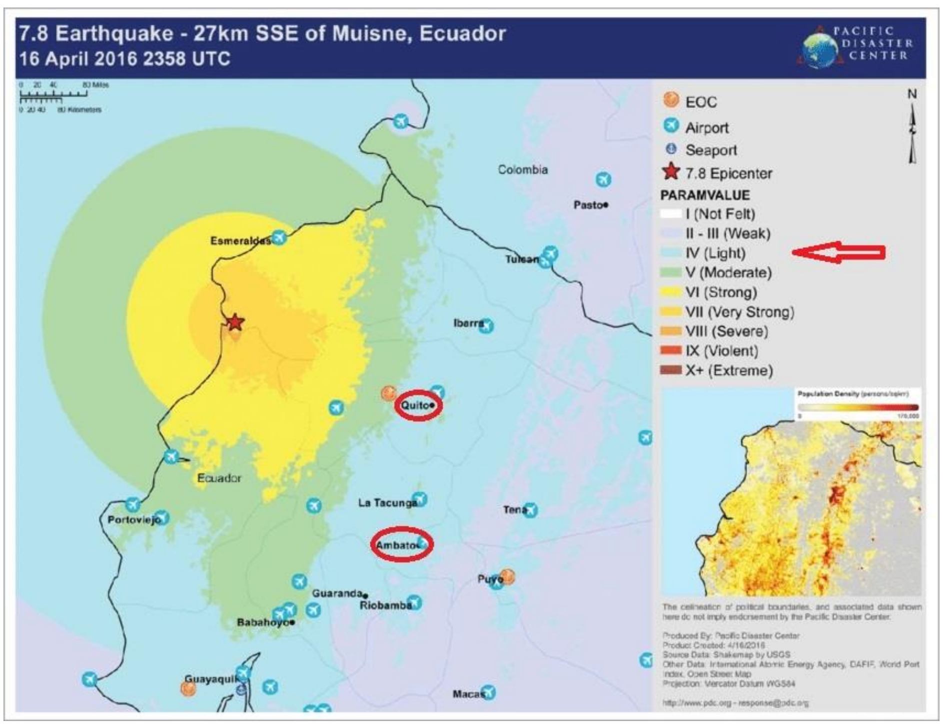
Year	Seafood #Sales	Seafood Distinct Items Count
2013	56,331	7
2014	66,280	7
2015	76,653	7
2016	70,881	7
2017	41,825	7

All Seafood product are perishable.

Additional information

- ✓ Wages in the public sector are paid every two weeks on the 15th and on the last day of the month. Supermarket sales could be affected by this.
- ✓ A magnitude 7.8 earthquake struck Ecuador on April 16, 2016. People rallied in relief efforts donating water and other first need products which greatly affected supermarket sales for several weeks after the earthquake.

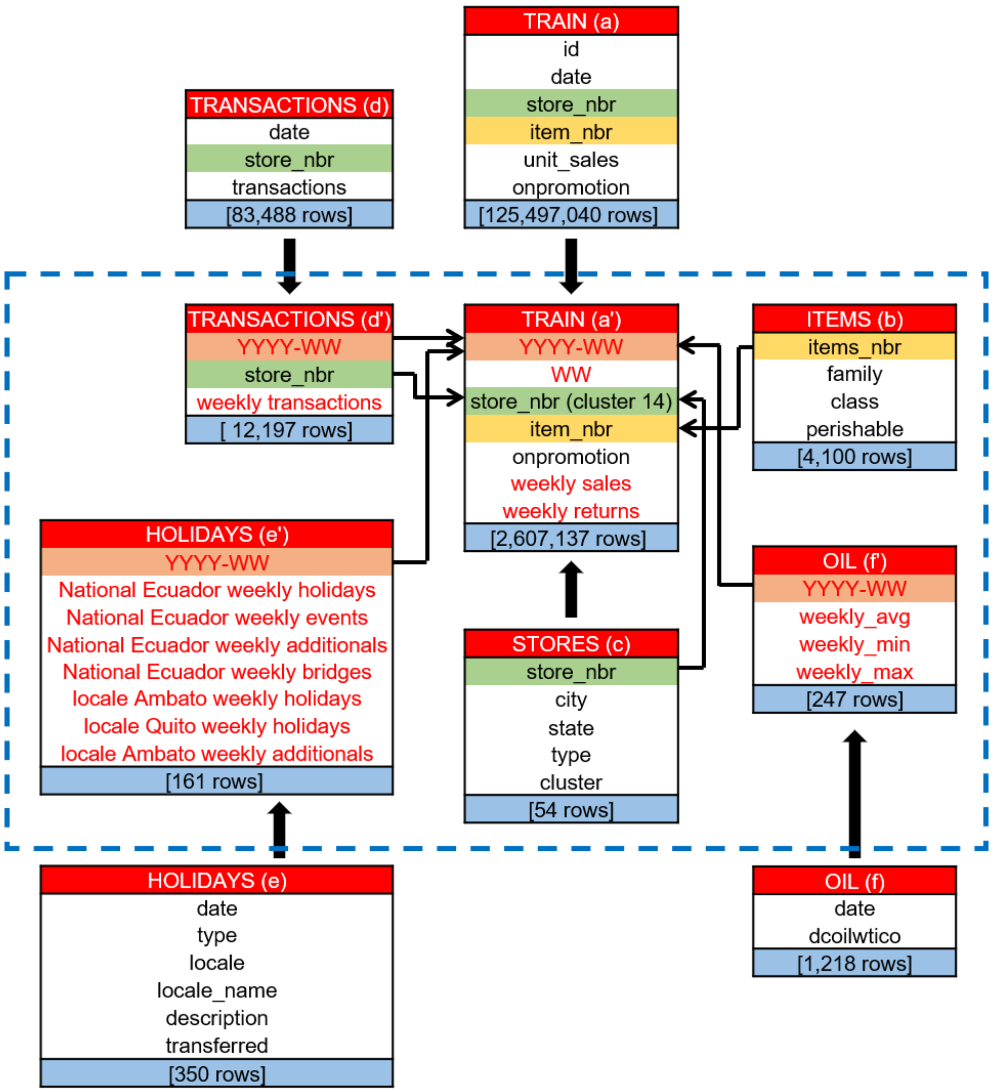
As shown in the map below, the physical effect over those regions was light. Nevertheless, the week of the earthquake and several weeks after will be indicated by an engineered feature. It will be interesting to see whether it had any effect over seafood items at these locations, and if so, will the feature be selected during feature selection voting as feature which effects the prediction.



Exploratory Data Analysis

Steps:

1. Explore each of the tables
2. Creating a flat file using SQL
   - ✓ generate new set of tables based on originals - add new variables & neglect others
   - ✓ Join tables to a single FF



3. EDA using R

   Data dimension - 22 X 9,951
   - ✓ Dealing with NA's and variables type
   - ✓ Data summary (attached in the appendix)
   - ✓ Check missing (14,223)
   - ✓ Imputation for the oil parameters
   - ✓ Logarithmic transform label (weekly sales) in order to normalize its distribution (appendix).

```
In [32]: summary(df$weekly_sales)
         summary(df$log_weekly_sales)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00    8.00   18.00   32.09   41.00  267.00 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6931  2.1972  2.9444  2.9548  3.7377  5.5910 
```

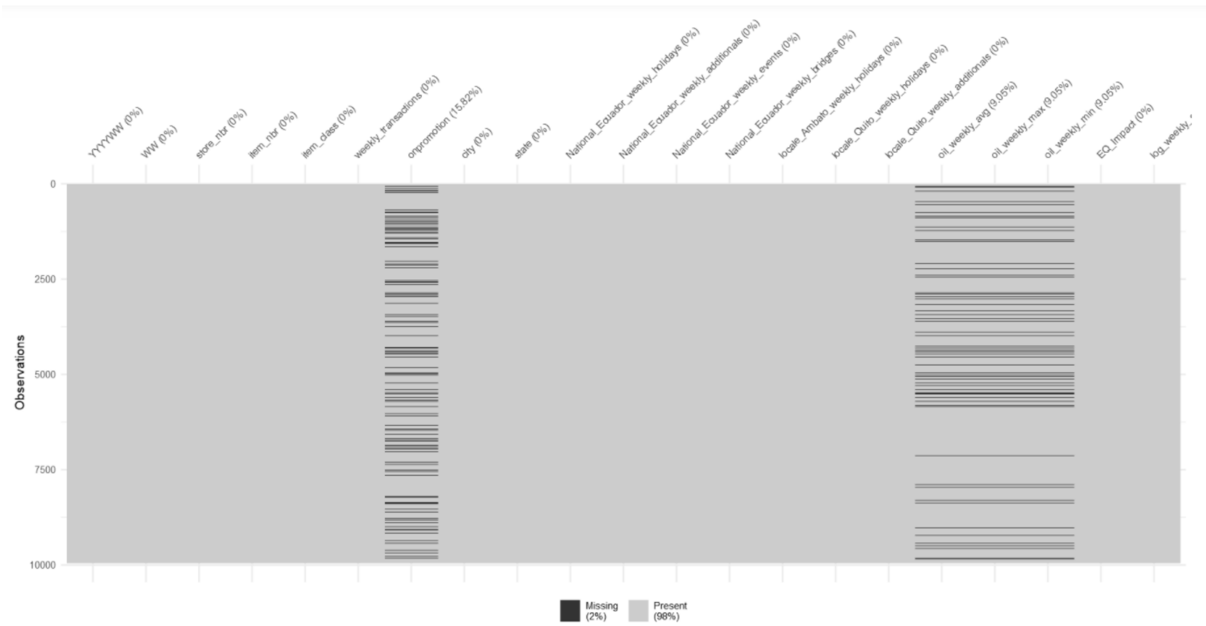
✓ Looking at the data using Table1 and Explore data functions in order to see variables distribution, missing and outliers (attached in the appendix)

<u>EDA Highlights:</u>

- No outliers
- Due to low exposure of 2 variables in higher levels, we chose to joined levels into one:
  Re categories National Ecuador weekly additionals > 1 as level '2'
  Re categories National Ecuador weekly events > 2 as level '3'

- Missing in 'Onpromotion' variable 1574 values (15.82%). Since it is a categorical variable we'll add a level: 0=False, 1=True, 2 =Null
- 9% (901 rows) missing values in the oil variables.
  We checked that no variable can explain the presence of missing values on any of the missing variables, thus we can assume that the missing mechanism is at least MAR. Since those missing values stands at 9% of the data, one should impute values. The preferred way to impute the data is with KNN. Nevertheless, we chose to impute the current oil variable with the average oil price of nearest previous weeks.

```
: mm <- getMissingness(data = df1)

[[1]]
             var na.count rate
1    onpromotion     1574 15.8
2 oil_weekly_avg      901  9.1
3 oil_weekly_max      901  9.1
4 oil_weekly_min      901  9.1

[[2]]
[1] "This dataset has 7888 (79.3%) complete rows. Original data has 9951 rows."
```



- Check the label variable vs. the factor variables.
  Different distribution between stores, item nbr, item class, onpromotion, cities, states.
  In the onpromotion variable for the TRUE level, there are outliers of the weekly log sales.

- Find Correlation between numeric variables (spearman) and factor variables (Cramer.V).
  Attached is the correlation matrix for the numeric variables which are significant (P<0.05). It is expected that the correlation between oil parameters will be high and significant (will be handled in the feature selection phase).
  The correlation between oil and transactions is weak (0.087)

A data.frame: 10 × 4

row	column	cor	p
<chr>	<chr>	<dbl>	<dbl>
weekly_transactions	oil_weekly_avg	0.087	0.000000e+00
weekly_transactions	oil_weekly_max	0.088	0.000000e+00
oil_weekly_avg	oil_weekly_max	0.996	0.000000e+00
weekly_transactions	oil_weekly_min	0.082	4.440892e-15
oil_weekly_avg	oil_weekly_min	0.997	0.000000e+00
oil_weekly_max	oil_weekly_min	0.988	0.000000e+00
weekly_transactions	log_weekly_sales	0.307	0.000000e+00
oil_weekly_avg	log_weekly_sales	0.122	0.000000e+00
oil_weekly_max	log_weekly_sales	0.122	0.000000e+00
oil_weekly_min	log_weekly_sales	0.121	0.000000e+00

4. Feature Engineering / Impact Coding / Data Extraction / Data Transformation
   - ✓ One-hot encoding was used in order to treat the following variables:
     Item_nbr
     Onpromotion
     Store_nbr
     Item_class
     City
     State
     YYYYWW
     WW
   - ✓ For each original categorical variable that we used one-hot-encoding on, we'll reduce one dummy that have the less frequent "1"
   - ✓ Add indicator variable EQ_impact that reflects the earthquake in Ecuador on April -May 2016 (weeks 16-19)
   - ✓ Check missing again

New data dimension - 82 x 9,951

5. Feature selection and voting
   - ✓ Since our label is continuous, Table1 in Python does not run. First, we will have the multivariate analysis, make the voting procedure and then proceed with univariate analysis and correlations to the label variable using R.
   - ✓ Using Lasso, RandomForest, GradientBoost and SVM end by voting for total_count >1 which reflects 23 selected variables.

```
In [46]: varSel.groupby('Sum')['Variable'].count()

Out[46]: Sum
         0    58
         1     3
         2    10
         3     3
         4     7
         Name: Variable, dtype: int64
```

Correlation > 0.9 between the selected variables:

```
In [14]: numcormatsel %>% filter(cor>0.9)
```

A data.frame: 6 × 4

row	column	cor	p
<chr>	<chr>	<dbl>	<dbl>
oil_weekly_avg	oil_weekly_max	0.997	0
oil_weekly_avg	oil_weekly_min	0.997	0
oil_weekly_max	oil_weekly_min	0.991	0
item_nbr_741201	item_class_2854	1.000	0
item_nbr_1247036	item_class_2864	1.000	0
city_Quito	state_Pichincha	1.000	0

   - ✓ Using Regression to calculate variable importance and removing weekly_avg & oil_weekly_min item_class_2854 & 2864 and state_Pichincha variables.

The final data dimension - 18 x 9,951

Note: the 'Data retrieval protocol' is attached in the Appendix.

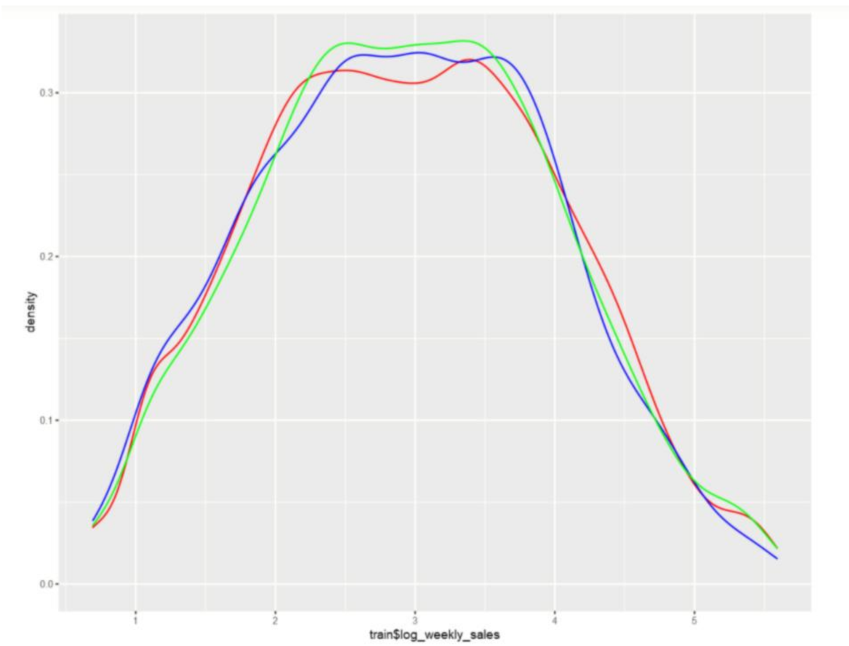
Models

Steps:

1. Preparing the data for modeling – use 'Table 1' and divide the data into 3 perfectly balanced datasets:

   Test (20%): 18 x 1,991 (green)
   Dev (16%): 18 x 1,592 (blue)
   Train (64%): 18 x 6,368 (red)

   Check the distribution of the outcome on the three subsets.



Note: Since the label variable (weekly_log_sales) is a continuous variable, we will use regression techniques for predicting.

2. Two metrics chosen for comparing the models (since outliers are absent):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$$

3. Run 10 different models

Attached are the metric results (Dev vs. Train results) in order by RMSE_dev.
It is noticed that the XGBoost model has an overfitting.

The selected model is RandomForest (RF-mod4)

A data.frame: 10 × 6

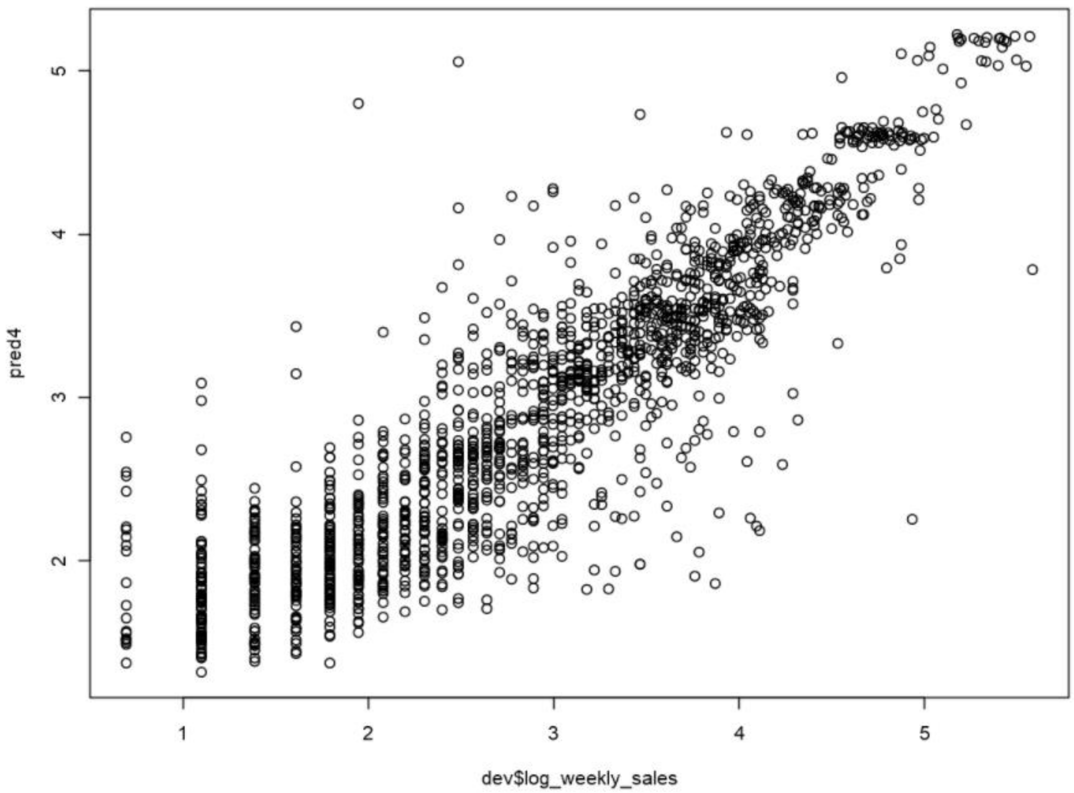
Name	Model	RMSE_Dev	RMSLE_Dev	RMSE_Train	RMSLE_Train
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
RandomForest (RF)	mod4	0.4957582	0.1490027	0.42626830	0.12802322
SVM	mod7	0.4977178	0.1493421	0.49341688	0.14622767
RandomForest (ranger)	mod5	0.5101627	0.1543961	0.46813395	0.14065353
GBM	mod10	0.5170288	0.1552648	0.52488079	0.15607587
XGBoost	mod8	0.5418664	0.1617597	0.08013069	0.02776829
Base Linear regression	mod1	0.5886486	0.1744881	0.60941196	0.17927886
GBM	mod9	0.6030014	0.1786473	0.62218536	0.18232785
Decision Trees-tree	mod2	0.6308729	0.1840876	0.63070361	0.18295666
Decision Trees-rpart	mod3	0.6308729	0.1840876	0.63070361	0.18295666
kNN	mod6	0.6843906	0.2046648	NA	NA

RandomForest (RF-mod4) received % Var explained of 77.29%

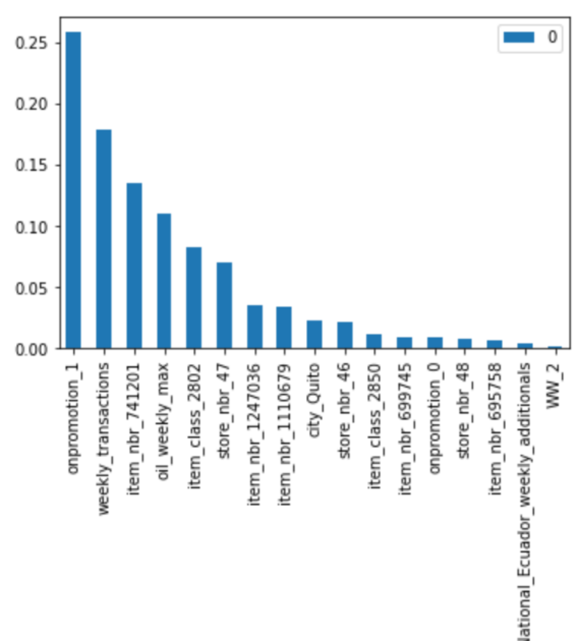
```
In [52]: set.seed(3)
         mod4 <- randomForest(log_weekly_sales ~., data=train)
         mod4
```

```
Call:
 randomForest(formula = log_weekly_sales ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5

          Mean of squared residuals: 0.2578529
                    % Var explained: 77.29
```

The plot graph of the predictive results vs log_weekly_sales is



4. Check variable importance



5. RandomForest Hyper parameter and fine tuning
   - ✓ All 3 partitioned data sets generated in R, where saved and imported to Python for the final phase of hyper parameter and fine tuning:
   - ✓ Re-generate base RF model in Python (since it was initially generated in R)
   - ✓ Perform random search
   - ✓ Perform nonrandom search (Fine tuning)
   - ✓ Set a grid space to search for the best hyper parameters (attached in appendix).

   The best parameters out of that grid are:

```
In [30]: rf_random.best_params_

Out[30]: {'n_estimators': 775,
          'min_samples_split': 5,
          'min_samples_leaf': 3,
          'max_features': 'auto',
          'max_depth': 10,
          'bootstrap': True}
```

The model was improved by 4.04% (RMSE ~ 0.510) in reference with the base model

The following is comparison table between the base model and the best parameters from the grid model:

Model	Date Set	RMSE	RMSLE
Base Model	Test	0.531	0.0237
	Train	0.197	0.0037
Grid Model	Test	0.510	0.0221
	Train	0.422	0.0164

Perform fine tune over the grid model by narrow the vector of each parameter around the best parameter from the previous step. The best parameters result after fine tuning are

```
In [37]: grid_search.best_params_

Out[37]: {'bootstrap': True,
          'max_depth': 10,
          'max_features': 'auto',
          'min_samples_leaf': 3,
          'min_samples_split': 4,
          'n_estimators': 600}
```

Final model

```
In [38]: Fine_Tuned_Model = grid_search.best_estimator_
         grid_accuracy = evaluate(Fine_Tuned_Model, X_test, y_test)
```

```
Model Performance
Root Mean Squared Error: 0.510
Root Mean Squared Log Error: 0.0221
```

```
In [39]: print('Improvement in reference to the base model {:0.2f}%.'.format( 100 * (base_accuracy - grid_accuracy) / base_accuracy))
         print('Improvement in reference to the best grid model {:0.2f}%.'.format( 100 * (random_accuracy - grid_accuracy) / random_accura
```

```
Improvement in reference to the base model 4.07%.
Improvement in reference to the best grid model 0.04%.
```

```
In [40]: Fine_Tuned_Model_train = rf_random.best_estimator_
         random_accuracy = evaluate(Fine_Tuned_Model_train, X_train_064, y_train_064)
```

```
Model Performance
Root Mean Squared Error: 0.422
Root Mean Squared Log Error: 0.0164
```

Check fine-tuned model results shows an improvement additional 0.04%

Weekly Sales Prediction

Eventually we compare the predictive weekly sales values (after re-transform by exponential) vs. the given original weekly sales column in the test dataset (was performed using EXCEL)

Week #	1	2	3	4	5	6	7	8	9	10
Test actual	690	1,970	1,533	1,181	1,417	916	897	1,075	1,730	1,105
Test predict	789	1,607	1,364	994	1,282	825	828	1,044	1,514	869

Week #	11	12	13	14	15	16	17	18	19	20
Test actual	1,220	1,325	1,733	1,935	1,038	2,121	1,906	1,537	1,509	1,586
Test predict	1,182	1,088	1,442	1,855	1,037	2,021	1,827	1,206	1,380	1,581

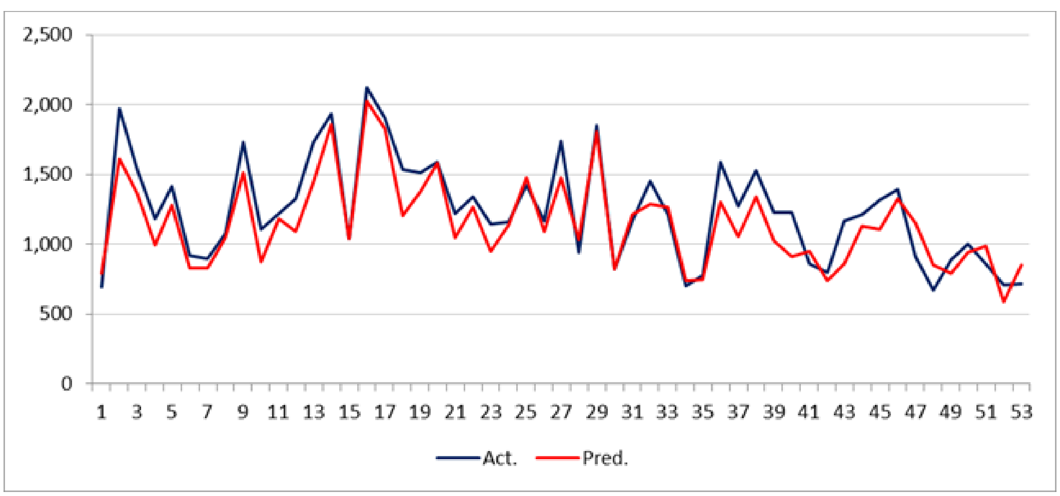
Week #	21	22	23	24	25	26	27	28	29	30
Test actual	1,222	1,341	1,147	1,161	1,424	1,163	1,736	944	1,850	820
Test predict	1,046	1,265	951	1,135	1,472	1,089	1,477	1,032	1,803	819

Week #	31	32	33	34	35	36	37	38	39	40
Test actual	1,165	1,451	1,213	701	778	1,591	1,274	1,526	1,227	1,228
Test predict	1,208	1,289	1,264	741	748	1,304	1,051	1,340	1,024	910

Week #	41	42	43	44	45	46	47	48	49	50
Test actual	860	798	1,164	1,211	1,318	1,391	909	667	885	1,000
Test predict	945	739	861	1,129	1,107	1,321	1,153	853	793	942

Week #	51	52	53	**Total**
Test actual	860	707	713	**65,869**
Test predict	988	585	851	**60,967**

Weekly sales prediction in cluster 14 stores vs. actual weekly sales (test dataset):



The trend over the year is predicted pretty well. The average variance is around 7%

Results and Conclusions

The origin Favorita data contains ~ 125M rows.

In the current project, we decided to focus on the seafood family of products, in four stores out of 54 (from same type/cluster) and predict its weekly sales over one year.

The initial Seafood cluster 14 dataset contains ~ 10K rows with 21 independent variables.
The label (weekly unit sales) distribution over the years shown stable.
No outliers were found, and the missing data was handled by imputation or adding category level to replace NA values.

In model selection phase we divided the data into 3 balanced datasets:
Train (64% of the data), Dev (16%) and Test (20%);
Executed 10 different models
The best model picked by its lowest RMSE score was RandomForest.
Except for the XGBoost model, no overfitting was found.

We set a grid for random search with RandomForest hyper parameters, and execute it over the Train and Test partition to improve the base model.
We performed several fine tuning cycles according to random grid base parameters results, over Train and Test partitions.

Eventually, we set the best fine-tuned model as the final model which represent the best metric, lowest RMSE ~ 0.510 which represent an improvement of 4.07% vs. the base model.

Finally, we performed a prediction of weekly sales based on the Test dataset.
As mentioned above the sales trend over the year was captured well.

Appendix

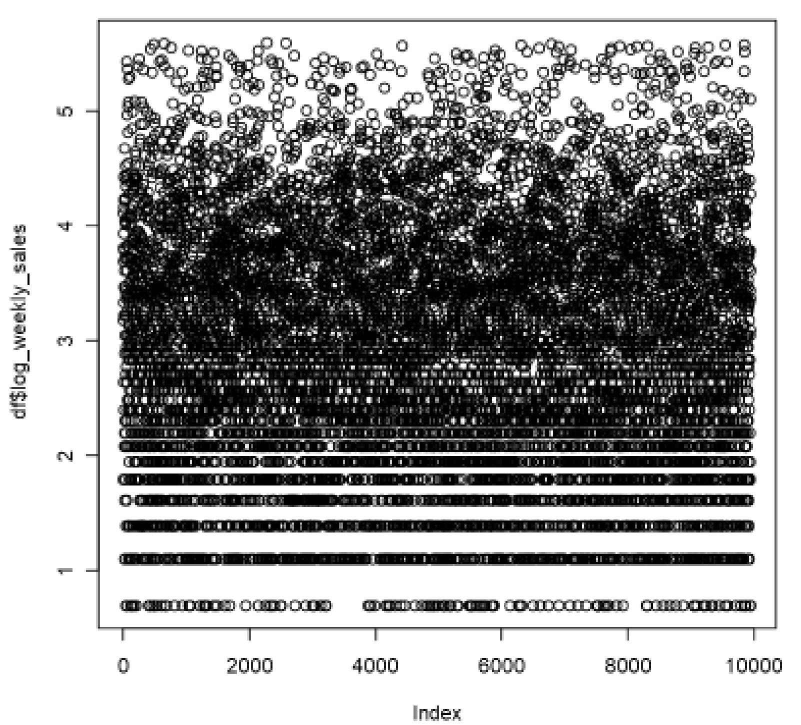
Data retrieval protocol:

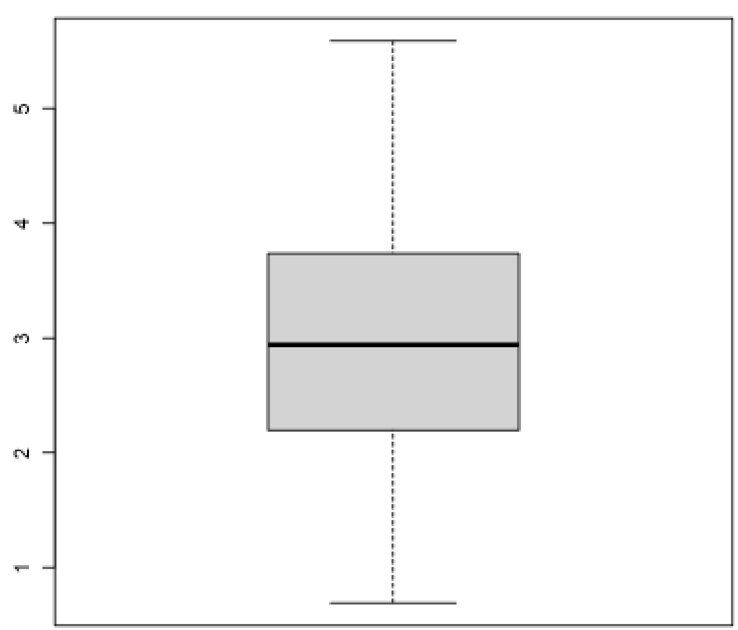
FAVORITA Data
Retrieval Protocol.xls:

Summary (data frame):

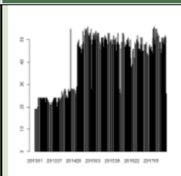
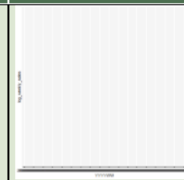
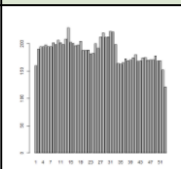
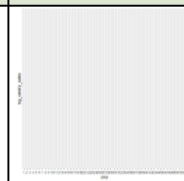
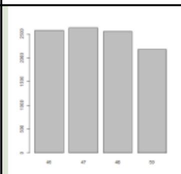
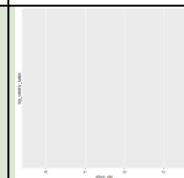
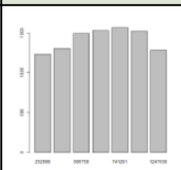
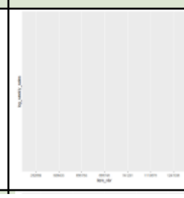
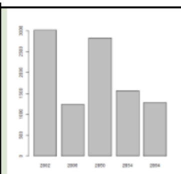
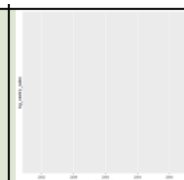
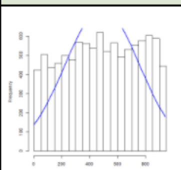
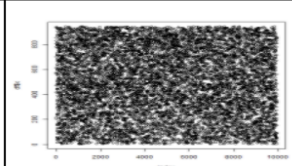
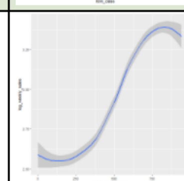
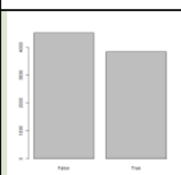
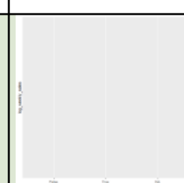
```
In [20]: summary(dfseafood14ff)

     YYYYWW           WW         store_nbr    item_nbr      item_class
 201446 :   56   14      : 230   46:2582   252698 :1237   2802:3026
 201710 :   56   31      : 223   47:2636   589403 :1310   2806:1237
 201414 :   55   32      : 222   48:2557   695758 :1495   2850:2834
 201441 :   55   28      : 220   50:2176   699745 :1531   2854:1570
 201444 :   55   27      : 213             741201 :1570   2864:1284
 201452 :   55   30      : 213             1110679:1524
 (Other):9619   (Other):8630             1247036:1284
 preishable_item  weekly_sales     weekly_transactions onpromotion
 1:9951          Min.   :  1.00   Min.   :  1.0       False:4531
                 1st Qu.:  8.00   1st Qu.:267.0       True :3846
                 Median : 18.00   Median :491.0       NA's :1574
                 Mean   : 32.09   Mean   :490.4
                 3rd Qu.: 41.00   3rd Qu.:722.0
                 Max.   :267.00   Max.   :942.0

      city              state      National_Ecuador_weekly_holidays
 Ambato:2176   Pichincha :7775   0:8569
 Quito :7775   Tungurahua:2176   1: 994
                                 2: 388



 National_Ecuador_weekly_additionals National_Ecuador_weekly_events
 0:9446                              0:9002
 1: 311                              1: 571
 2:  42                              2:  95
 4:  69                              3:  49
 5:  83                              5:  48
                                     7: 144
                                     8:  42
 National_Ecuador_weekly_bridges locale_Ambato_weekly_holidays
 0:9800                          0:9619
 1: 151                          1: 332



 locale_Quito_weekly_holidays locale_Quito_weekly_additionals oil_weekly_avg
 0:9783                       0:9779                          Min.   :  1.00
 1: 168                       1: 172                          1st Qu.: 46.00
                                                              Median : 93.00
                                                              Mean   : 94.81
                                                              3rd Qu.:138.00
                                                              Max.   :208.00
                                                              NA's   :901
 oil_weekly_max   oil_weekly_min
 Min.   :  1.00   Min.   :  1.00
 1st Qu.: 47.00   1st Qu.: 46.00
 Median : 94.00   Median : 90.00
 Mean   : 95.57   Mean   : 93.33
 3rd Qu.:139.00   3rd Qu.:136.00
 Max.   :207.00   Max.   :205.00
 NA's   :901      NA's   :901
```

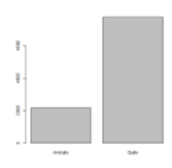
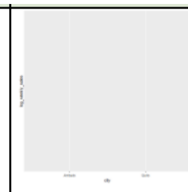
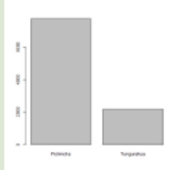
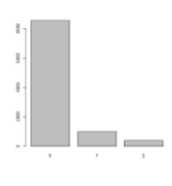
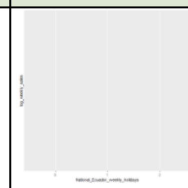
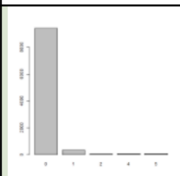
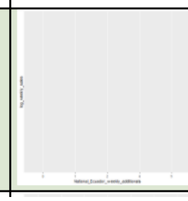
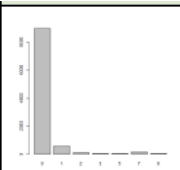
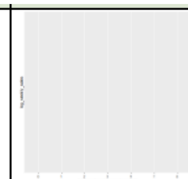
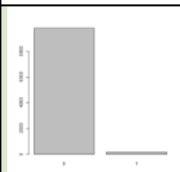
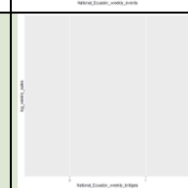
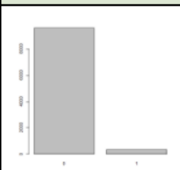
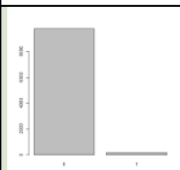
Transforming the sales count using Log:

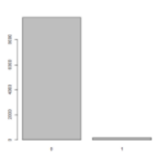
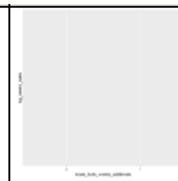
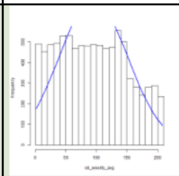
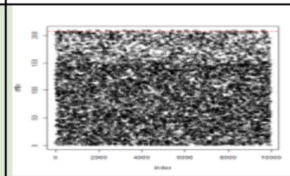
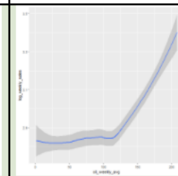
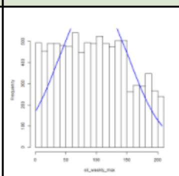
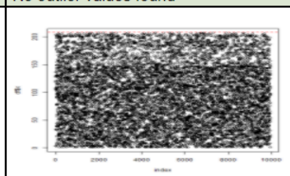
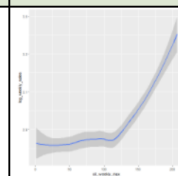
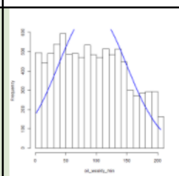
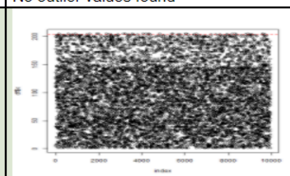
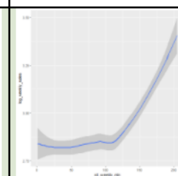
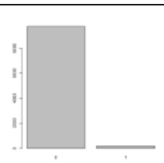
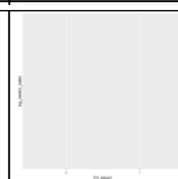
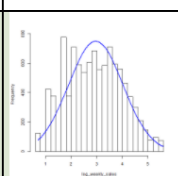
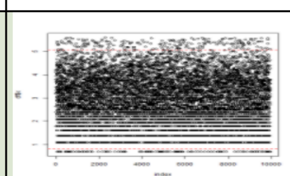
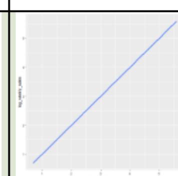




Data Exploration & Visualization

Variable	Distribution	Descriptive Statistics	Outliers	Dependent Variable Distribution
YYYYWW		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 245		
WW		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 53		
store_nbr		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 4 • 46 : 2582 • 47 : 2636 • 48 : 2557 • 50 : 2176		
item_nbr		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 7		
item_class		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 5		
weekly_transactions		Data type: Continuous Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Mean: 490.4 StdDev: 267.6 Median: 491 IQR: 267 - 722 Min: 1 Max: 942	 Outlier values: No outlier values found	
onpromotion		Data type: Categorical Data Data length: 8377 / 9951 (84.18 %) Missing: 1574 (15.82 %) Number of levels: 2 • False : 4531 • True : 3846		

city		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • Ambato : 2176 • Quito : 7775		
state		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • Pichincha : 7775 • Tungurahua : 2176		
National_Ecuador_weekly_holidays		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 3 • 0 : 8569 • 1 : 994 • 2 : 388		
National_Ecuador_weekly_additionals		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 5		
National_Ecuador_weekly_events		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 7		
National_Ecuador_weekly_bridges		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • 0 : 9800 • 1 : 151		
locale_Ambato_weekly_holidays		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • 0 : 9619 • 1 : 332		
locale_Quito_weekly_holidays		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • 0 : 9783 • 1 : 168		

locale_Quito_weekly_additionals		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • 0 : 9779 • 1 : 172		
oil_weekly_avg		Data type: Continuous Data length: 9050 / 9951 (90.95 %) Missing: 901 (9.054 %) Mean: 94.81 StdDev: 56.48 Median: 93 IQR: 46 - 138 Min: 1 Max: 208	 Outlier values: No outlier values found	
oil_weekly_max		Data type: Continuous Data length: 9050 / 9951 (90.95 %) Missing: 901 (9.054 %) Mean: 95.57 StdDev: 56.63 Median: 94 IQR: 47 - 139 Min: 1 Max: 207	 Outlier values: No outlier values found	
oil_weekly_min		Data type: Continuous Data length: 9050 / 9951 (90.95 %) Missing: 901 (9.054 %) Mean: 93.33 StdDev: 55.69 Median: 90 IQR: 46 - 136 Min: 1 Max: 205	 Outlier values: No outlier values found	
EQ_Impact		Data type: Categorical Data Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Number of levels: 2 • 0 : 9758 • 1 : 193		
log_weekly_sales		Data type: Continuous Data length: 9951 / 9951 (100 %) Missing: 0 (0 %) Mean: 2.955 StdDev: 1.059 Median: 2.944 IQR: 2.197 - 3.738 Min: 0.6931 Max: 5.591	 Outlier values: No outlier values found	

Data Summary – Selected variables:

```
weekly_transactions National_Ecuador_weekly_additionals oil_weekly_max
Min.   :  1.0       Min.   :1.00                        Min.   :  1.0
1st Qu.:267.0       1st Qu.:1.00                        1st Qu.: 53.0
Median :491.0       Median :1.00                        Median :104.0
Mean   :490.4       Mean   :1.07                        Mean   :105.6
3rd Qu.:722.0       3rd Qu.:1.00                        3rd Qu.:154.0
Max.   :942.0       Max.   :3.00                        Max.   :212.0
item_nbr_695758 item_nbr_699745 item_nbr_741201 item_nbr_1110679
0:8456          0:8420          0:8381          0:8427
1:1495          1:1531          1:1570          1:1524
```

```
item_nbr_1247036 onpromotion_0 onpromotion_1 store_nbr_46 store_nbr_47
0:8667           0:5420        0:6105        0:7369       0:7315
1:1284           1:4531        1:3846        1:2582       1:2636
```

```
store_nbr_48 WW_2     item_class_2802 item_class_2850 city_Quito
0:7394       0:9761   0:6925          0:7117          0:2176
1:2557       1: 190   1:3026          1:2834          1:7775
```

```
log_weekly_sales
Min.   :0.6931
1st Qu.:2.1972
Median :2.9444
Mean   :2.9548
3rd Qu.:3.7377
Max.   :5.5910
```

RandomForest parameters for tree:

n_estimators = number of trees in the foreset

max_features = max number of features considered for splitting a node

max_depth = max number of levels in each decision tree

min_samples_split = min number of data points placed in a node before the node is split

min_samples_leaf = min number of data points allowed in a leaf node

bootstrap = method for sampling data points (with or without replacement)

Working Files

SQL server:

1. Generate flat file → 1-SQL.sql

Jupyter Notebooks:

2. EDA → 2-EDA-R.ipynb
3. Feature selection (multivariate) → 3-Fearture Selection-PYTHON.ipynb
4. Feature selection (univariate analysis) → 4-Univariate analysis-R.ipynb
5. Model selection → 5-Pre Processing and Modeling Include Train Metrics-R.ipynb
6. Hyperparameters and finetuning → 6-Hyperparameters Finetuning-PYTHON.ipynb

Excel:

7. Weekly Prediction Results for cluster 14 seafood items.xls