



**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](http://www.corporacionfavorita.com/), 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***](http://www.corporacionfavorita.com/) ***Ecuador.***

# Methodology (Project design)

## Data

The data source in Kaggle contains 6 data sets shared by [Corporación Favorita](http://www.corporacionfavorita.com/). 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.

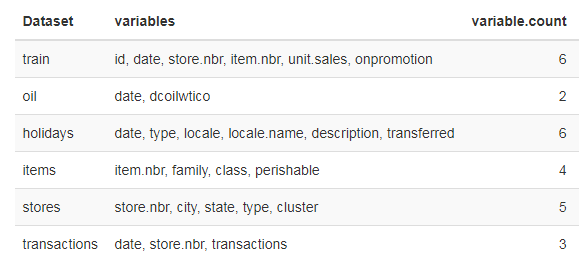
**Stores** (54) includes attributes such as city, state, type and cluster of stores (cluster is a grouping of similar stores)

**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.



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 |

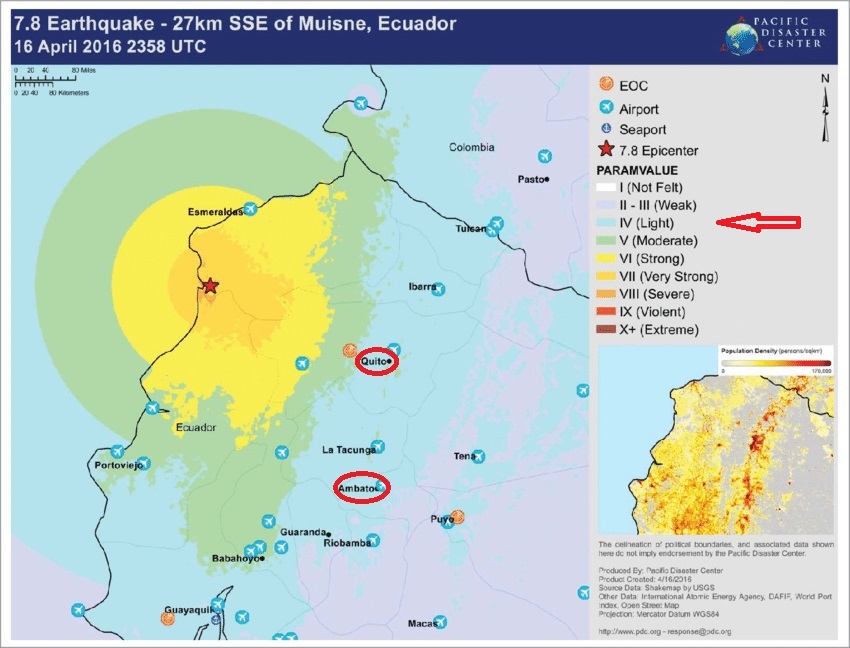
All Seafood product are perishable.



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.

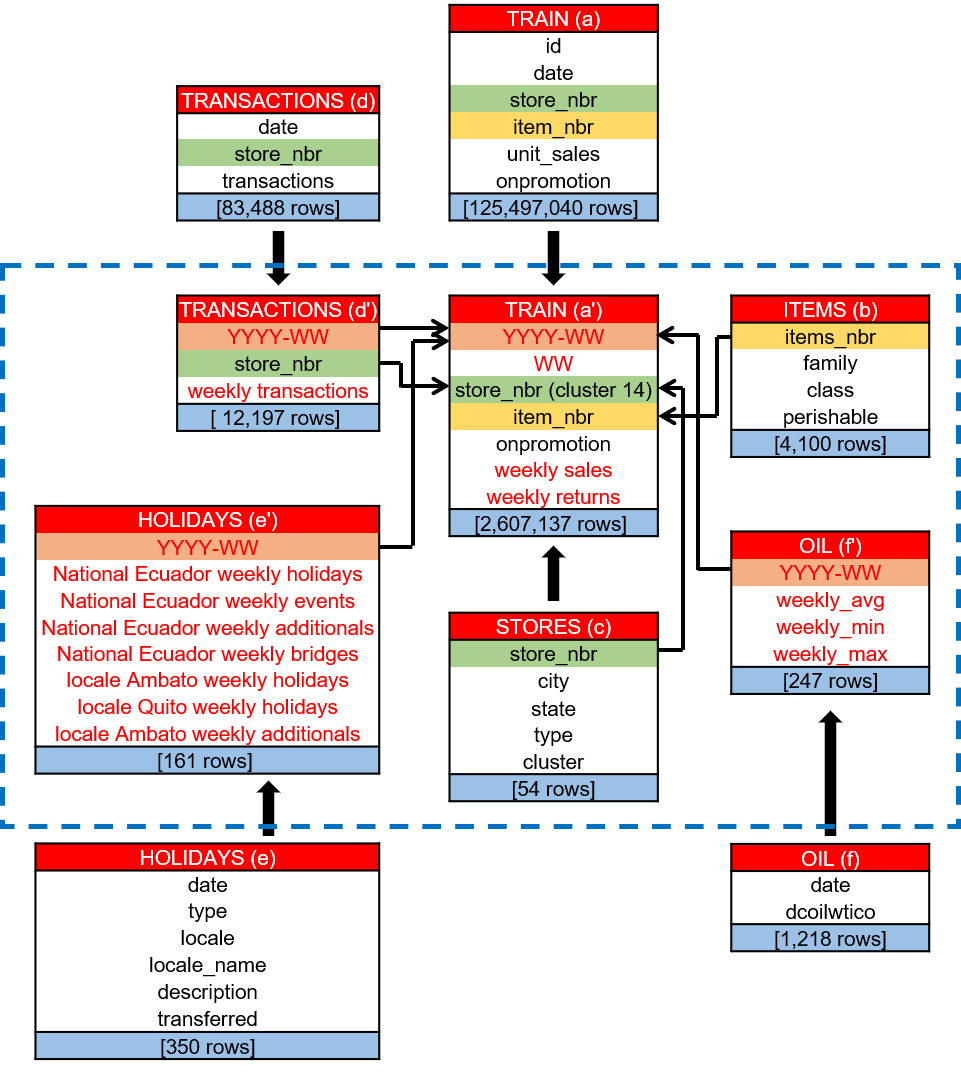


Exploratory Data Analysis

Steps:

1. Explore each of the tables
2. Creating a flat file using SQL

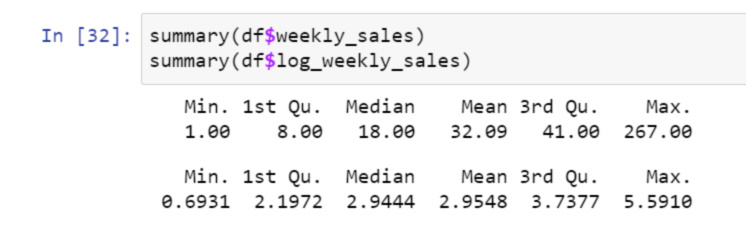
* generate new set of tables based on originals - add new variables & neglect others
* Join tables to a single FF



1. 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).



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

EDA Highlights:

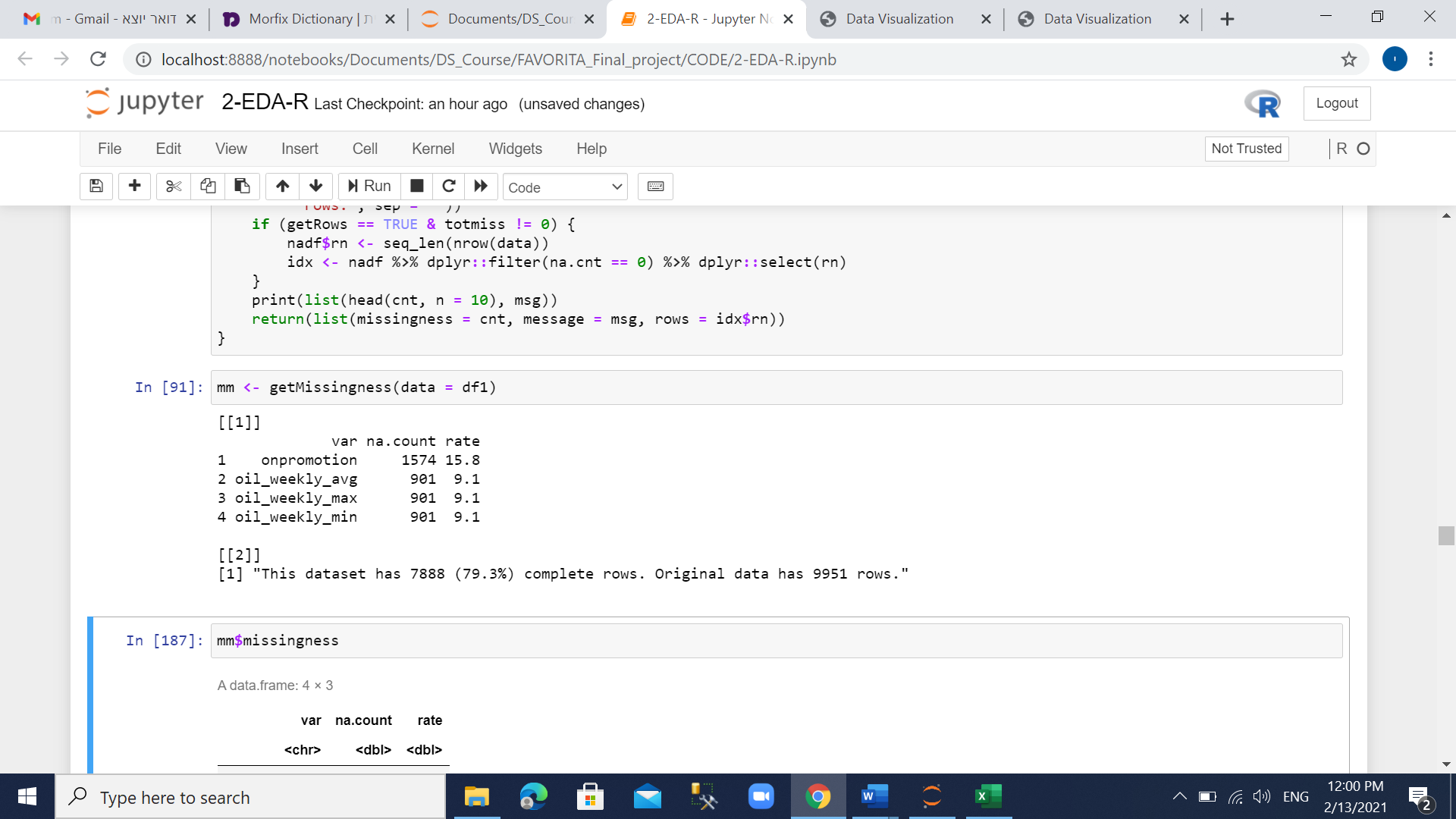
* No outliers
* Due to low exposure of 2 variables in higher levels, we chose to joined levels into one:

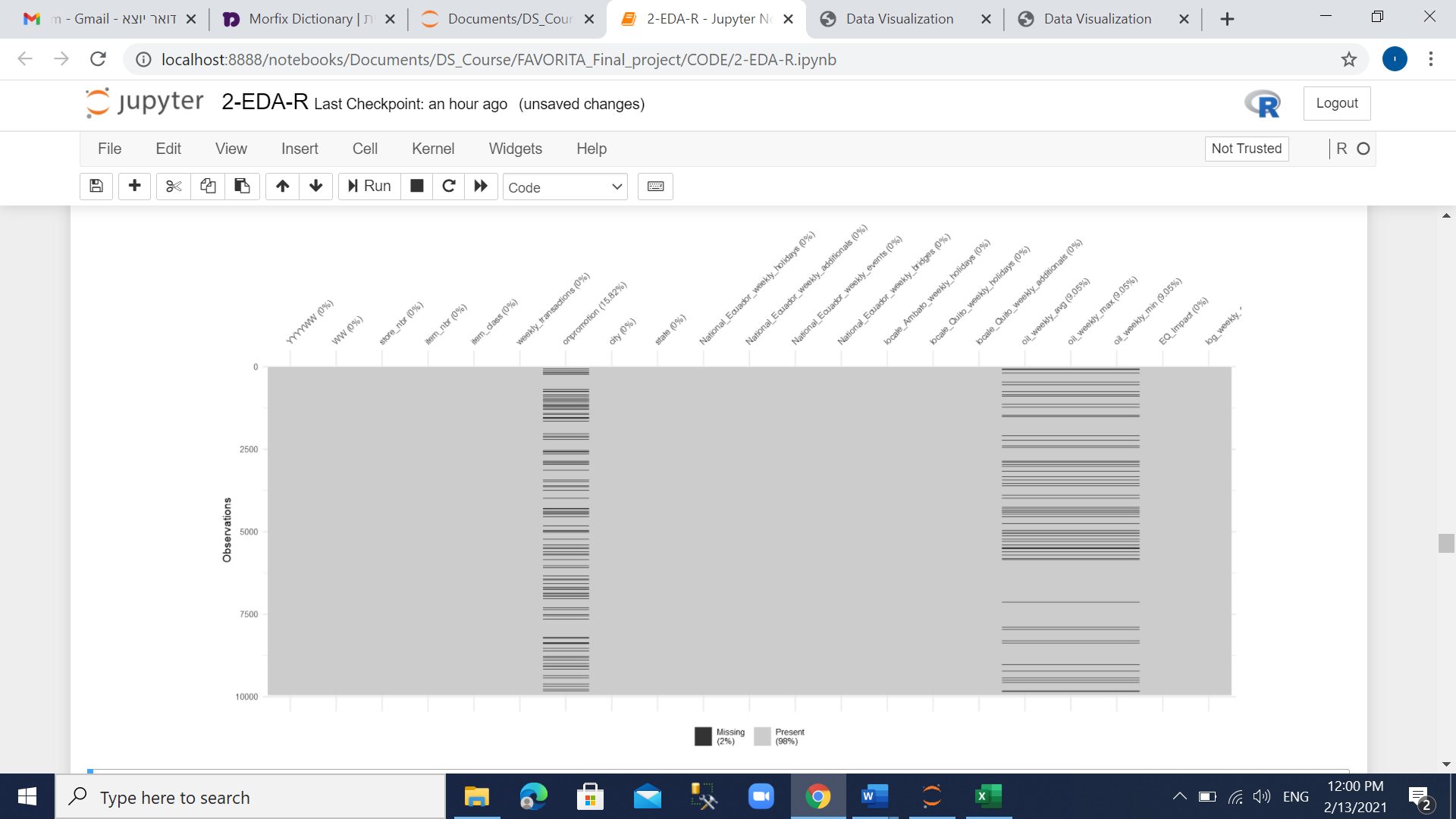
Re categories National Ecuador weekly additionals > 1 as level '2'

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.





* Check the label variable vs. the factor variables.

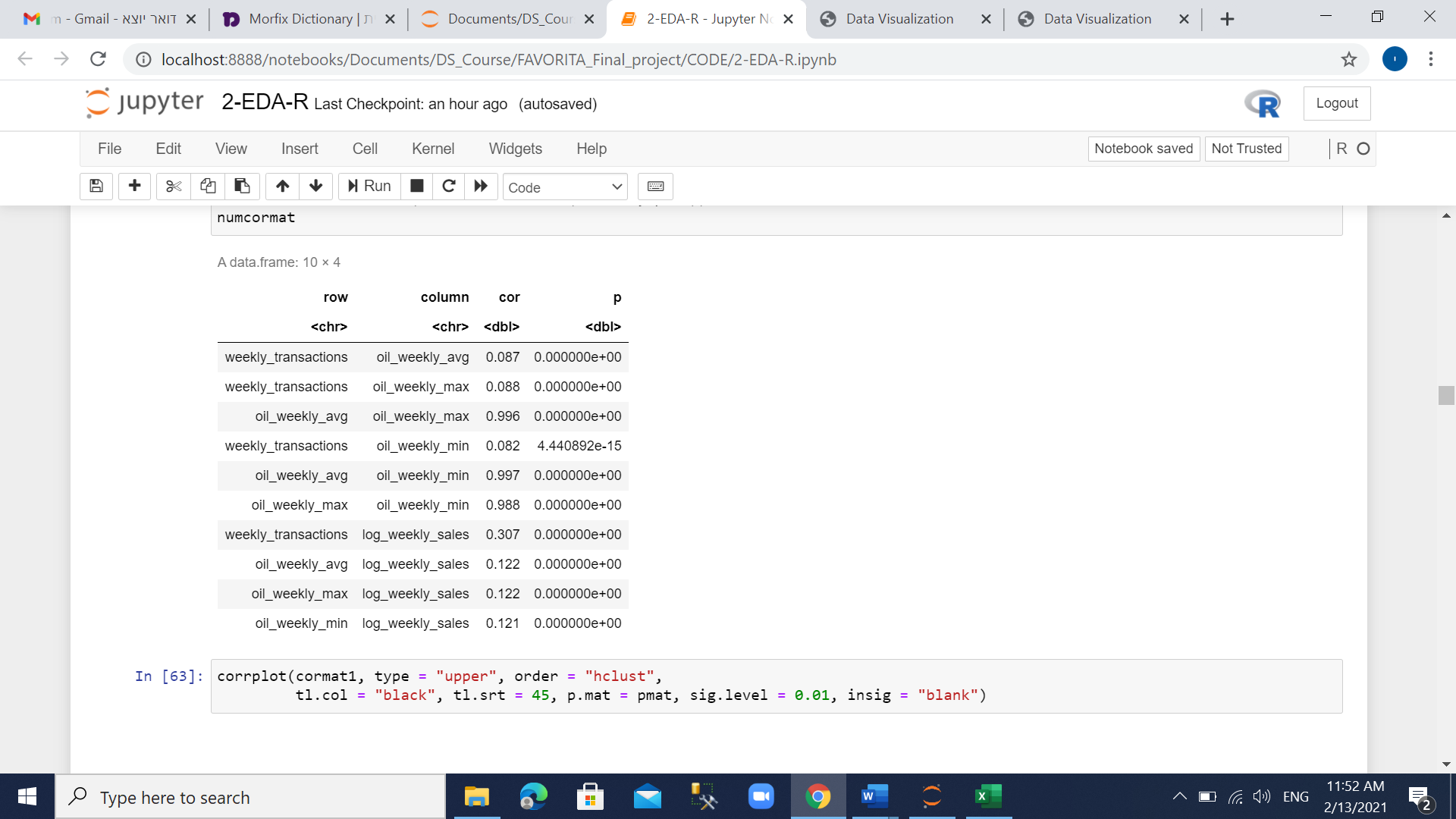
Different distribution between stores, item nbr, item class, onpromotion, cities, states.

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)



1. 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

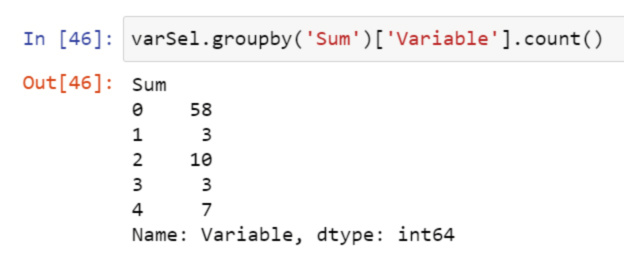
WW

* 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

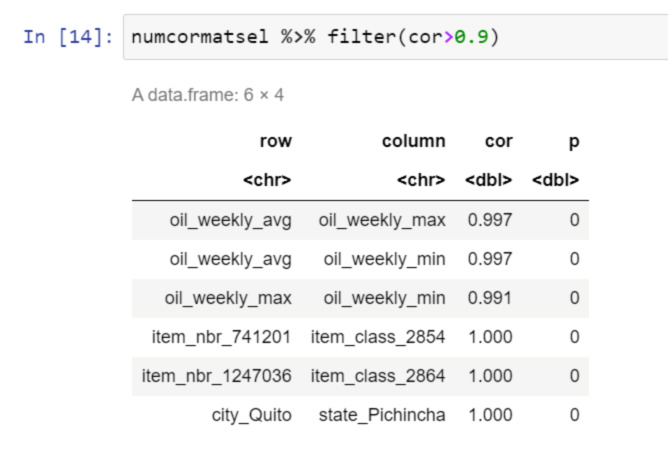
**New data dimension - 82 x 9,951**

1. 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.



Correlation > 0.9 between the selected variables:



* 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:

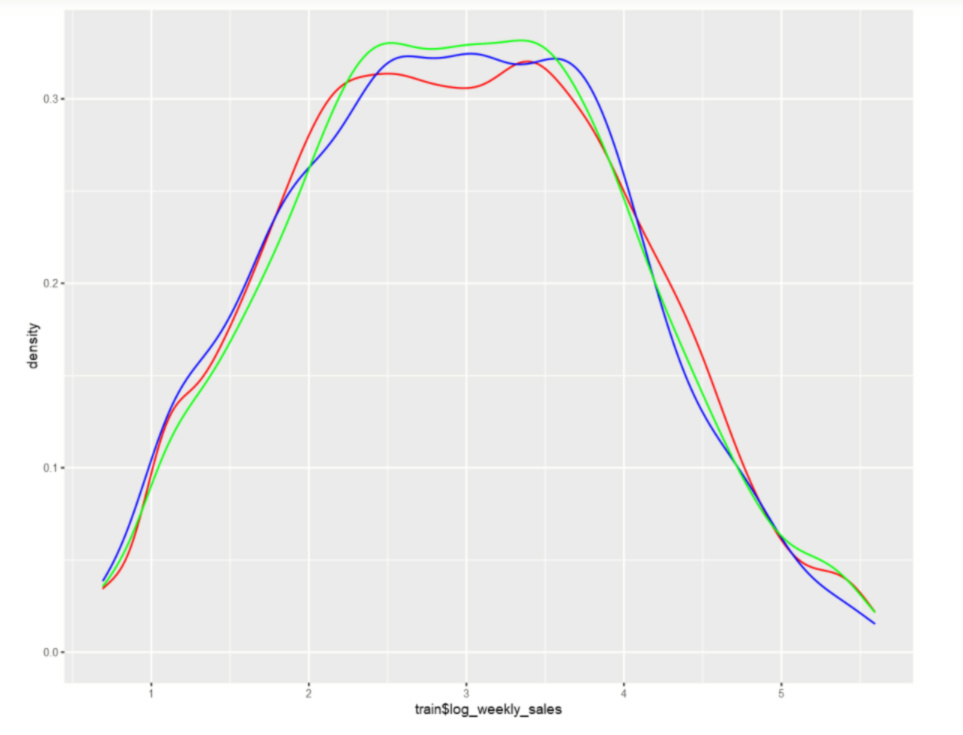
1. Preparing the data for modeling – use 'Table 1' and divide the data into 3 perfectly balanced datasets:

**Test (20%): 18 x 1,991 (green)**

**Dev (16%): 18 x 1,592 (blue)**

**Train (64%): 18 x 6,368 (red)**

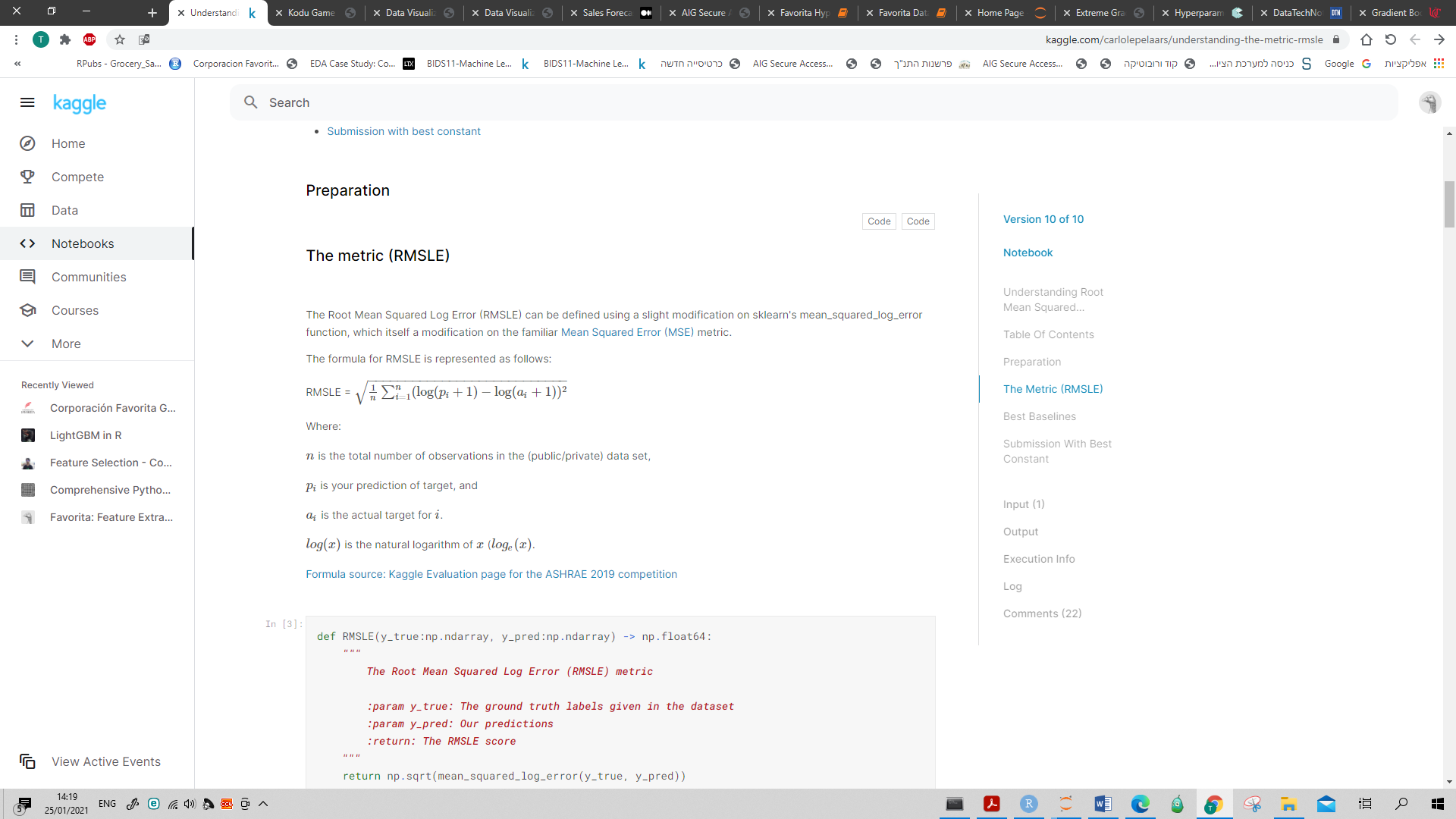
Check the distribution of the outcome on the three subsets.



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

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



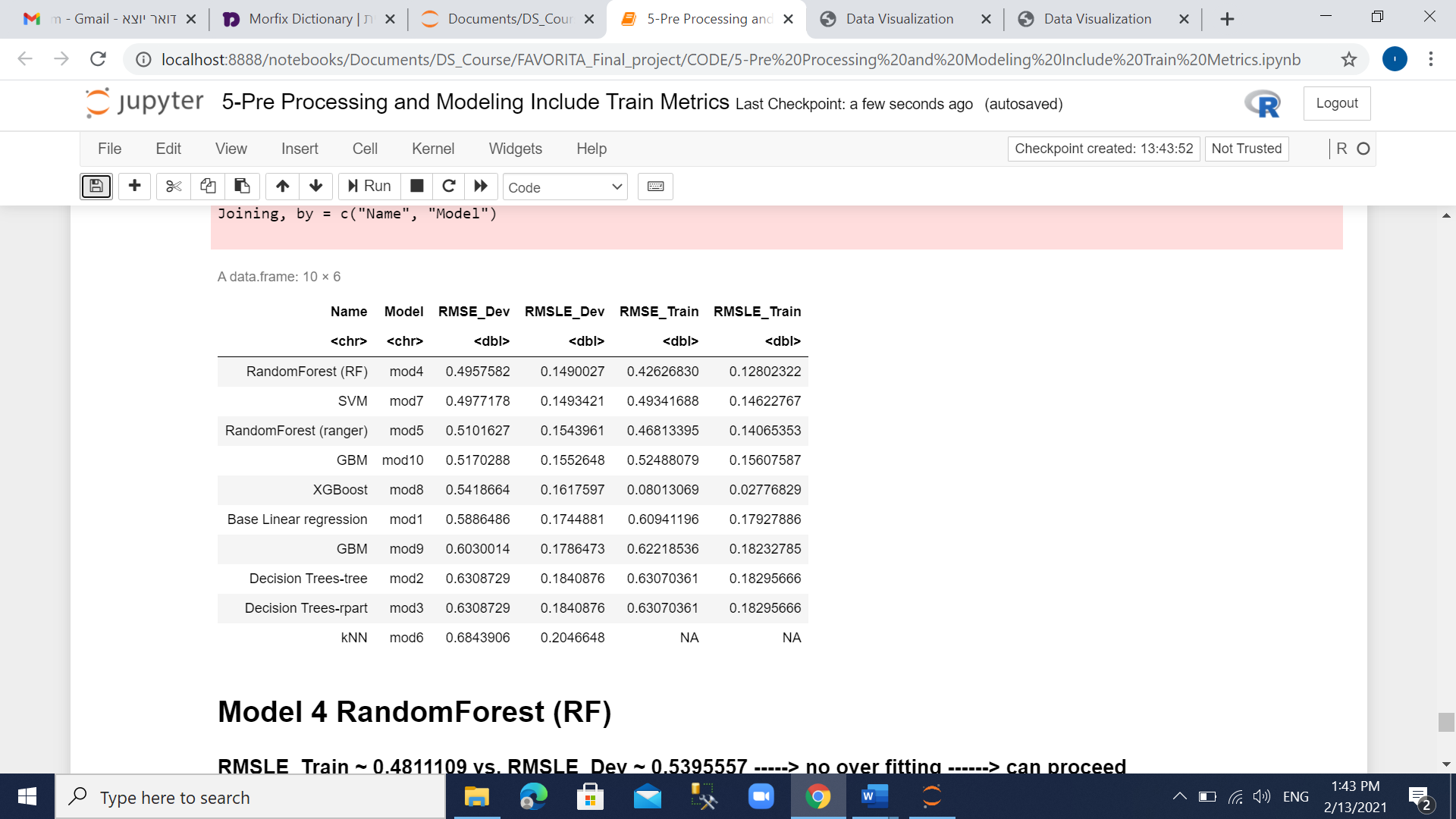


1. Run 10 different models

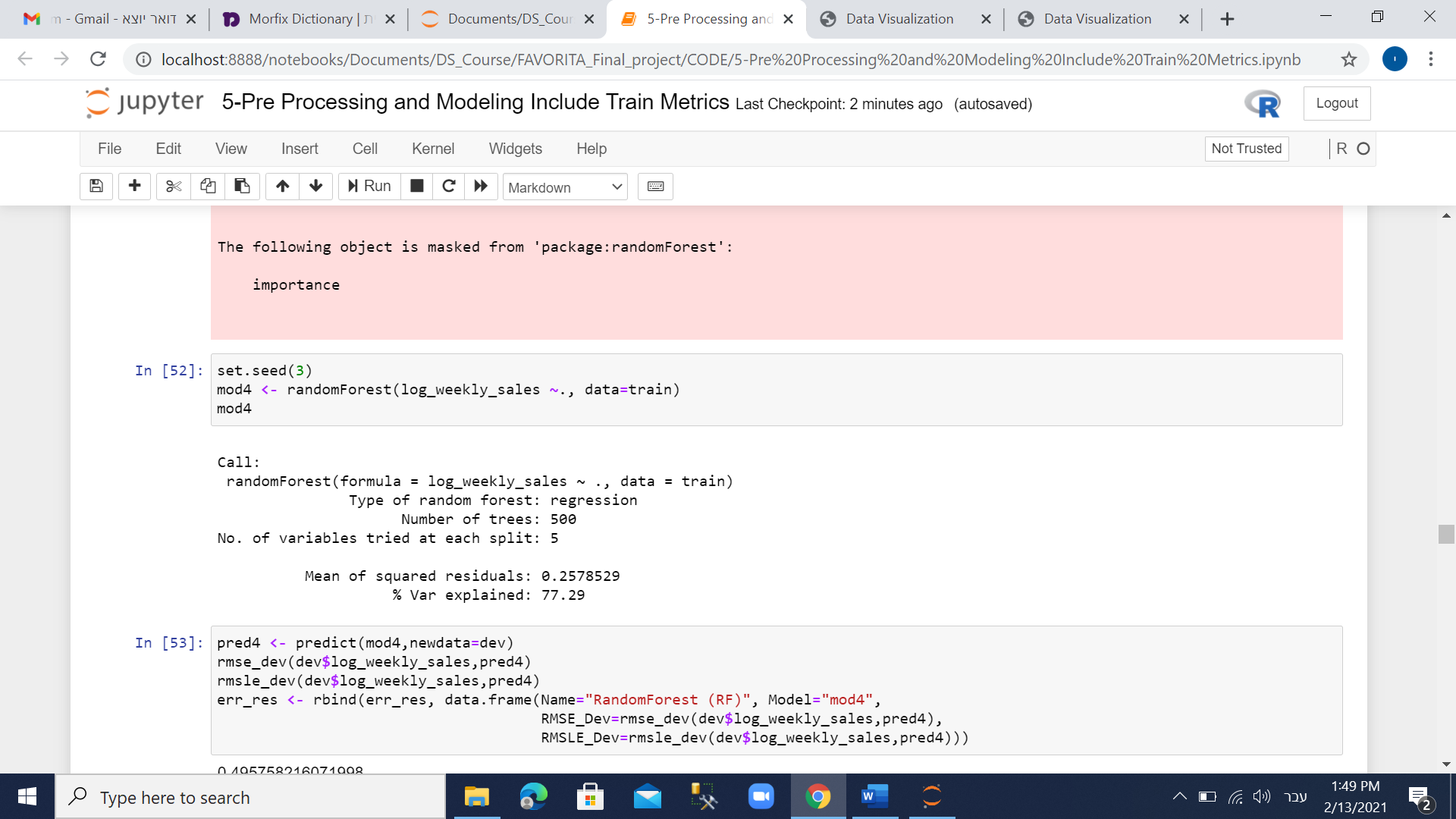
Attached are the metric results (Dev vs. Train results) in order by RMSE\_dev.

It is noticed that the XGBoost model has an overfitting.

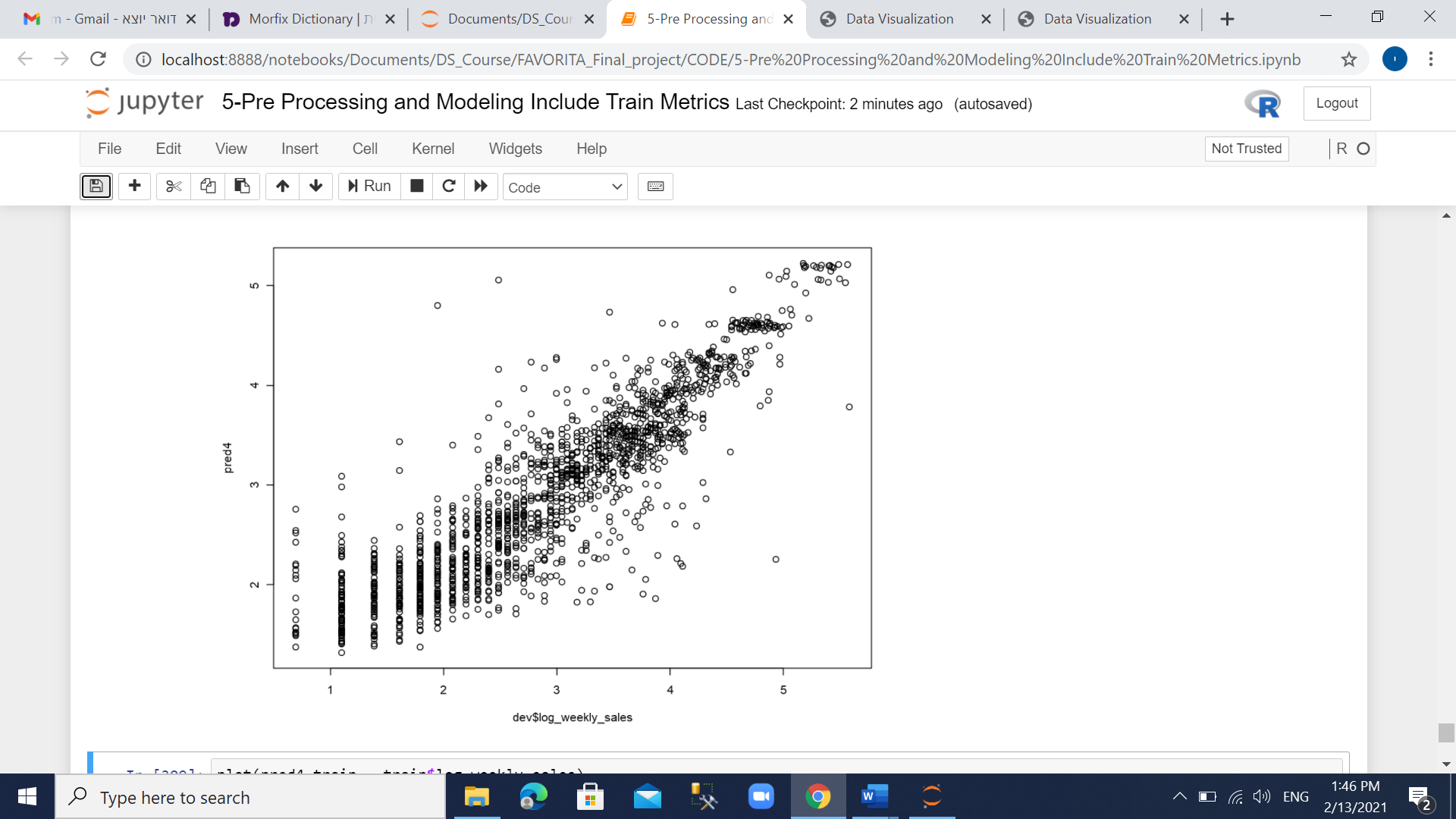
**The selected model is RandomForest (RF-mod4)**



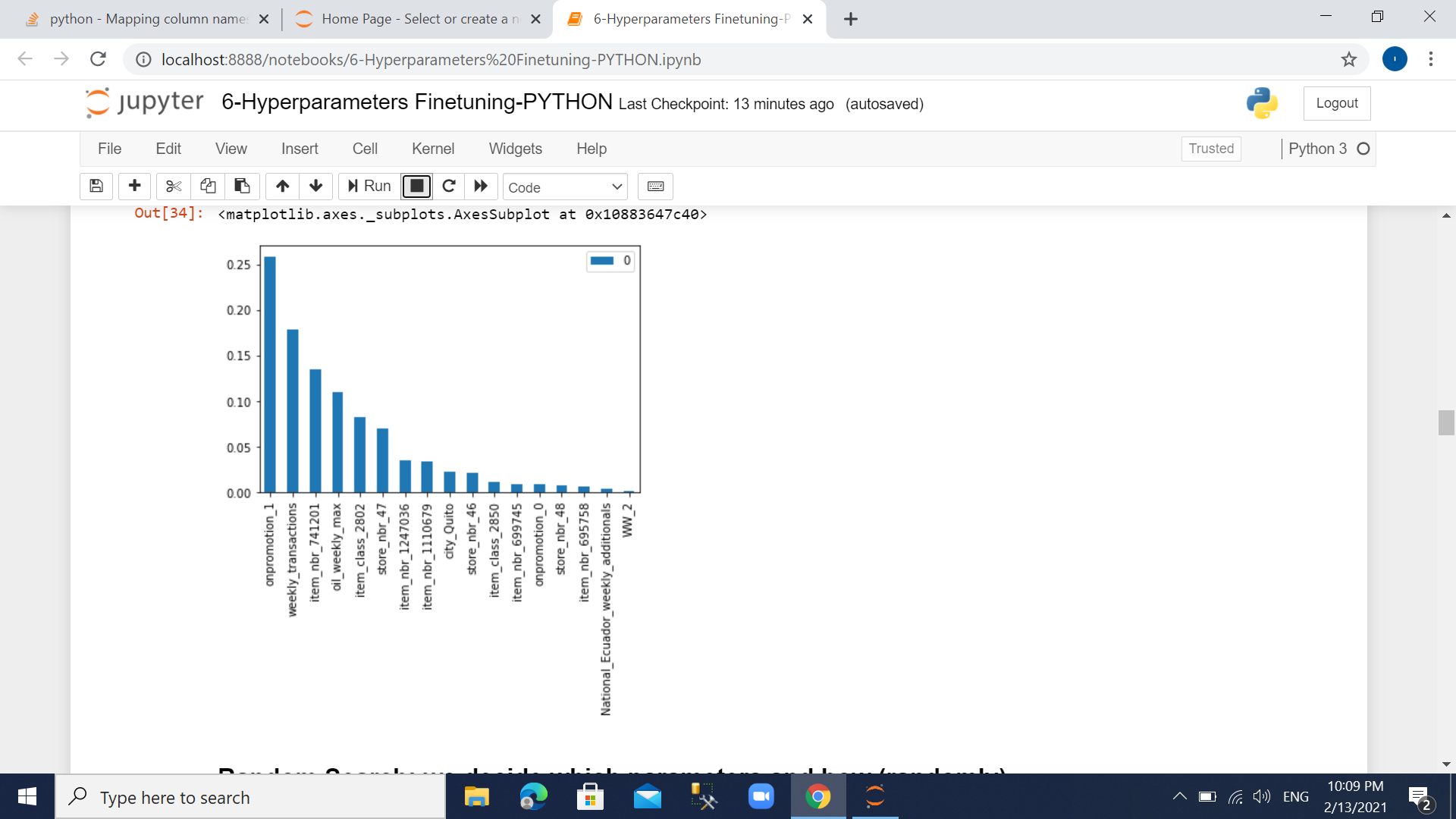
RandomForest (RF-mod4) received % Var explained of 77.29%



The plot graph of the predictive results vs log\_weekly\_sales is



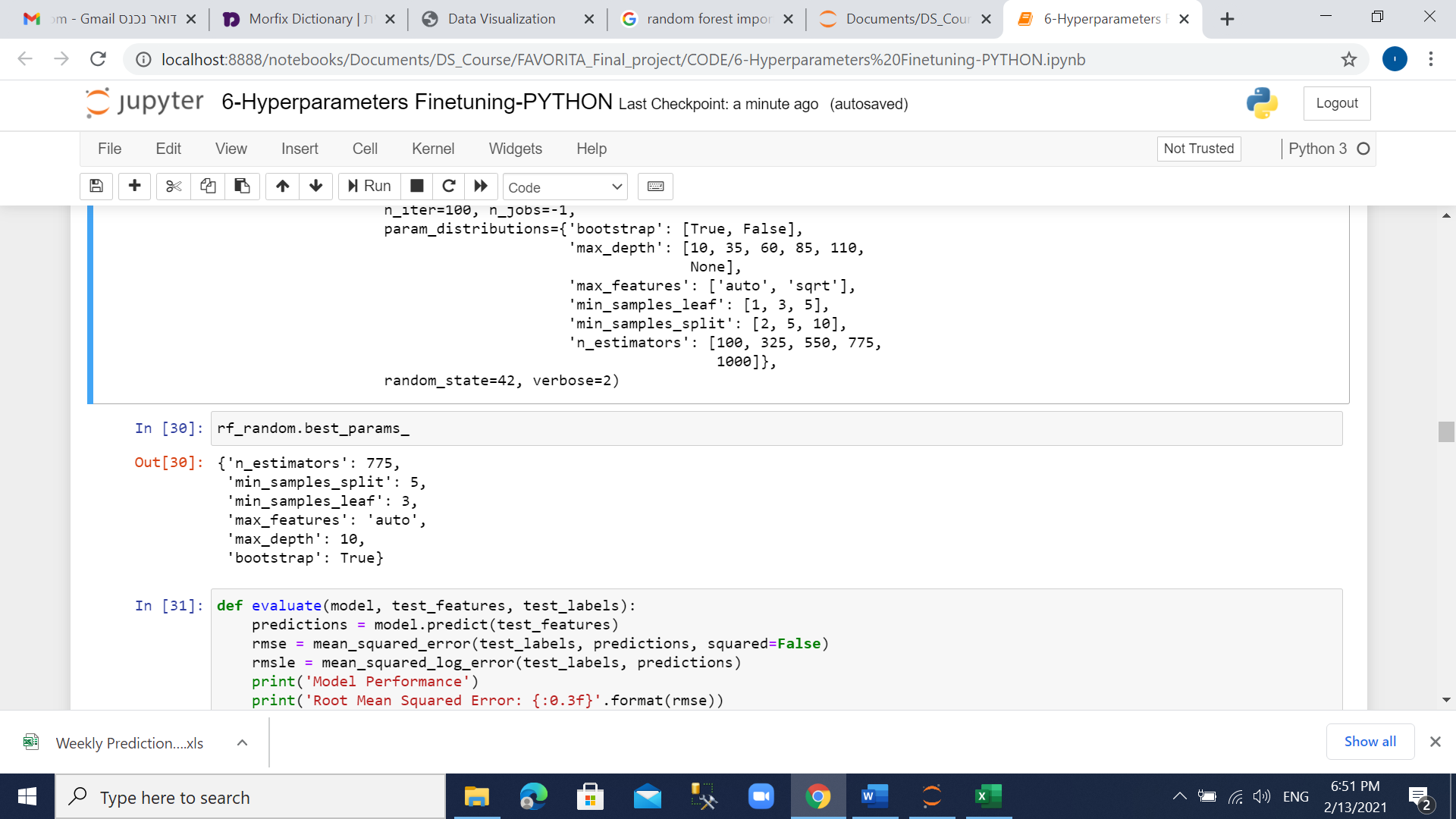
1. Check variable importance



1. 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:

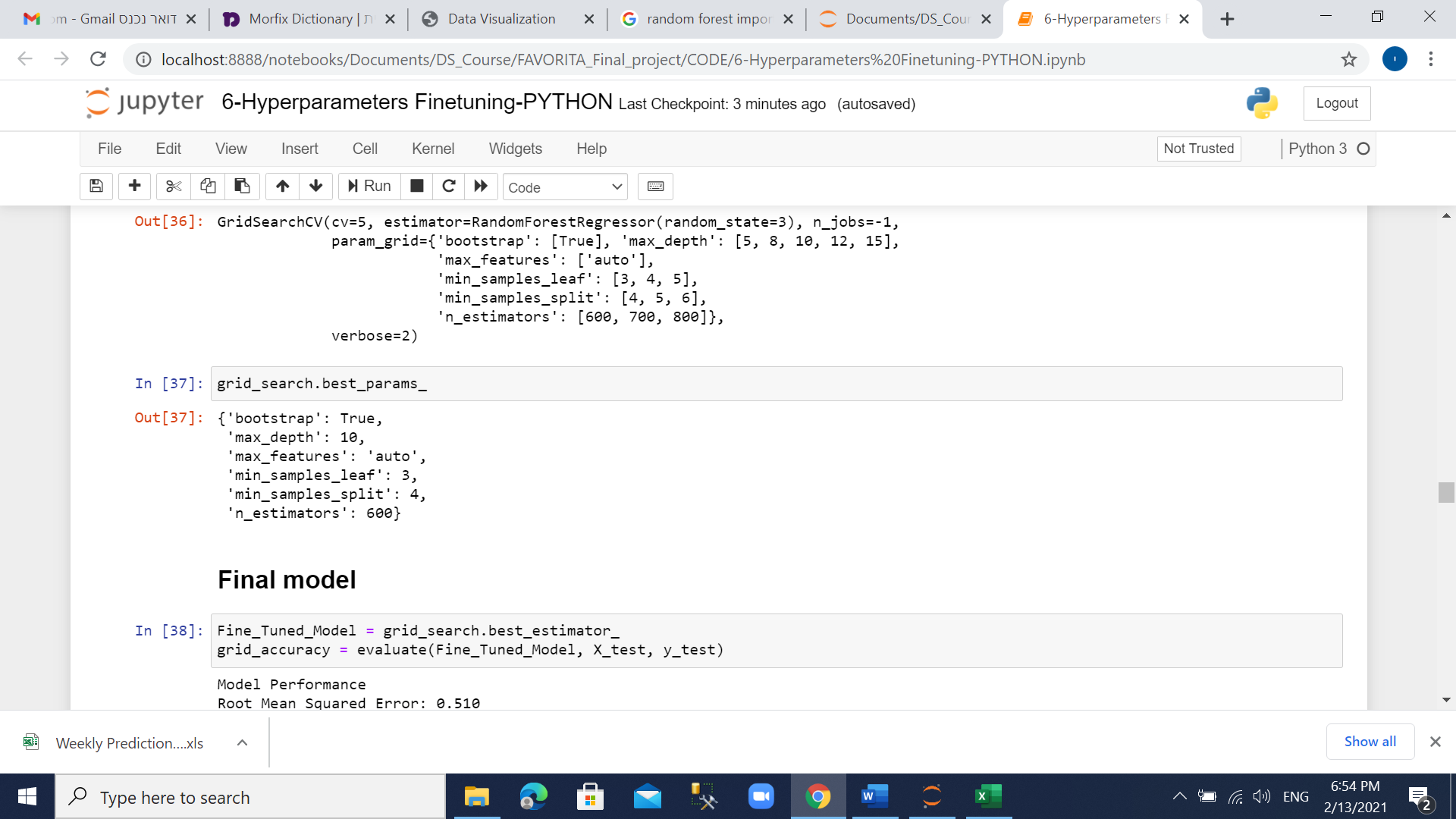


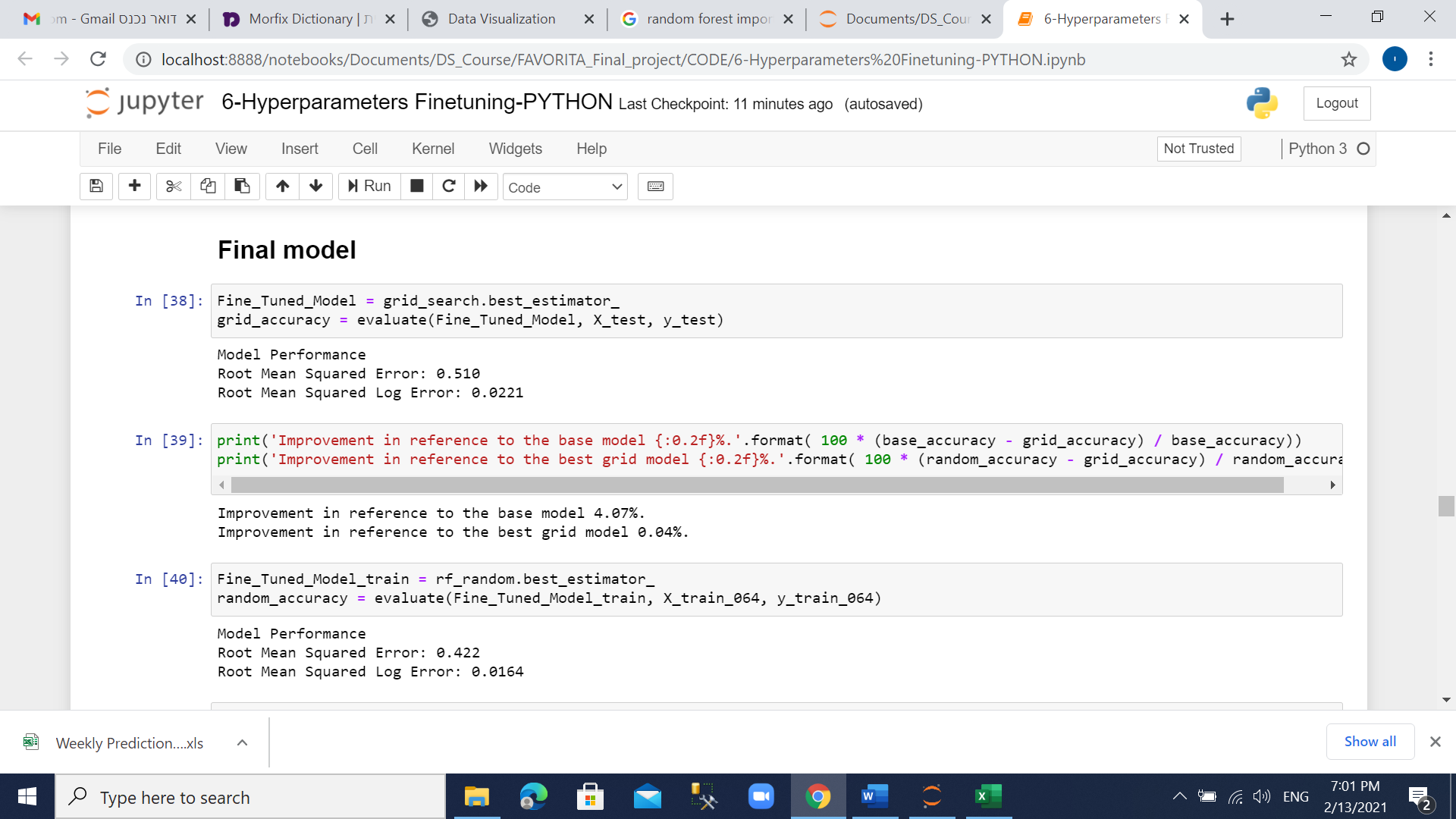
The model was improved by 4.04% (RMSE ~ 0.510) in reference with the base model

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

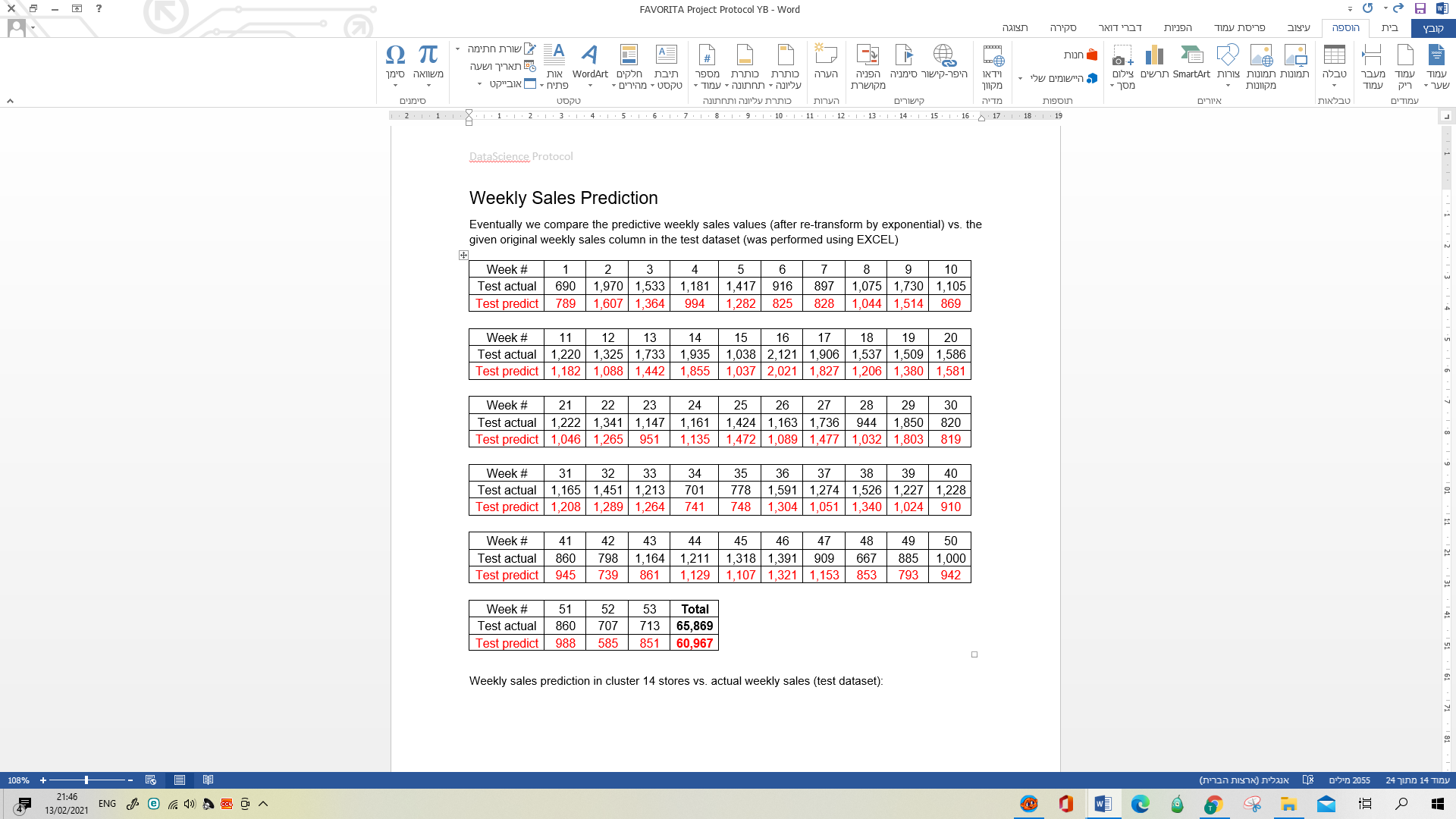




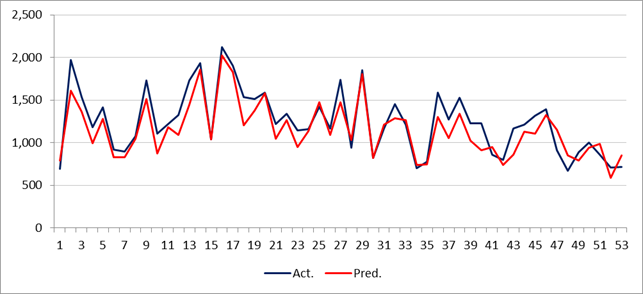
Check fine-tuned model results shows an improvement additional 0.04%

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)



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.

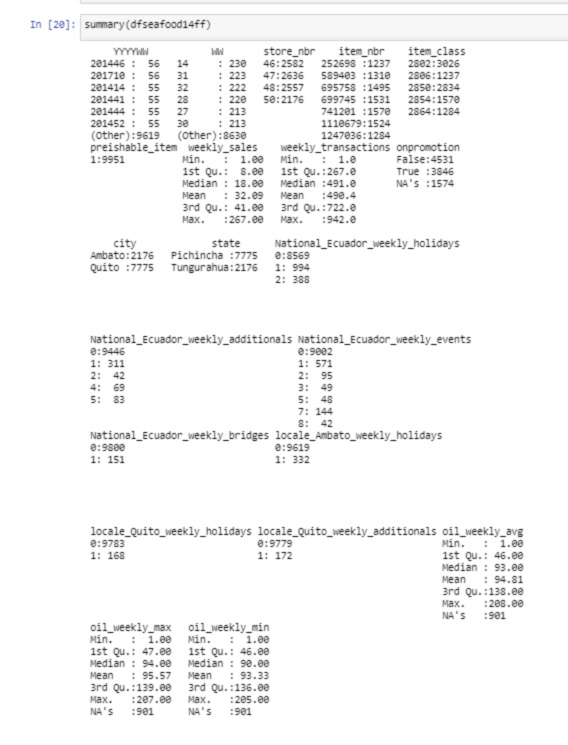
As mentioned above the sales trend over the year was captured well.

**Appendix**

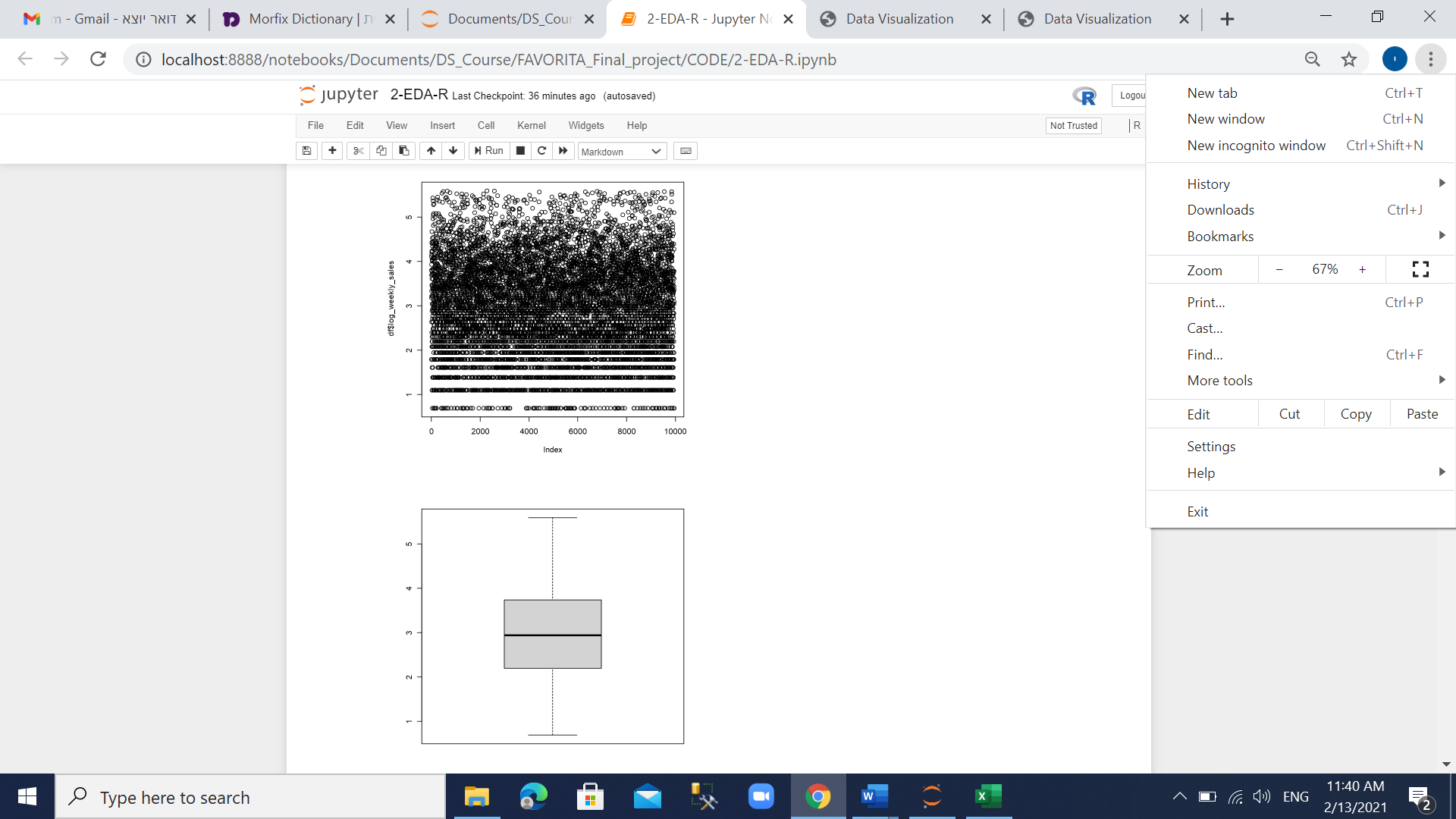
[**Data retrieval protocol**](https://docs.google.com/spreadsheets/d/1pYYjgwZ_8PS1Bcmc2kRNHTL0f_rk__GCJALLs1JHPUQ/edit#gid=0)**:**



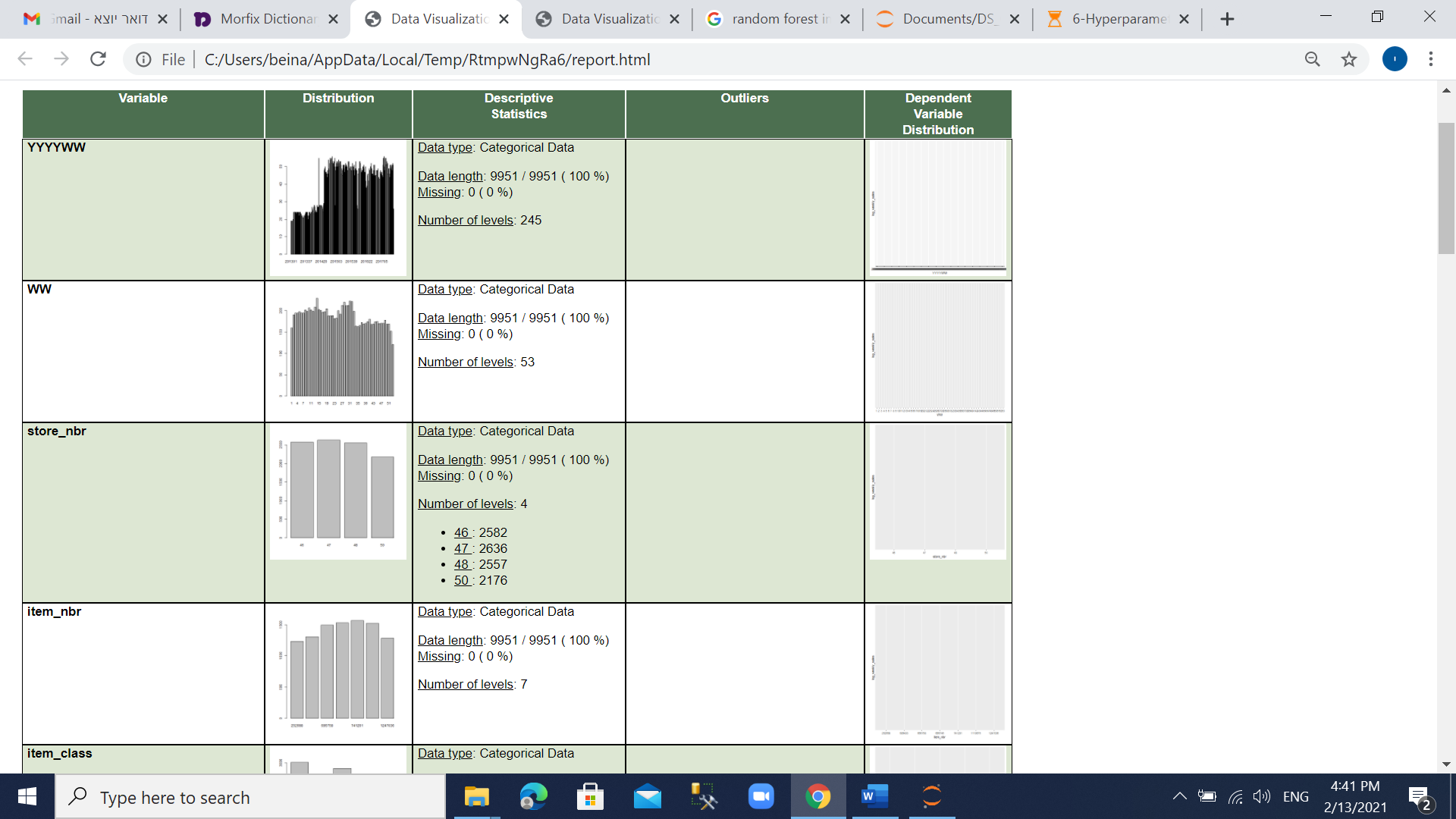
**Summary (data frame):**

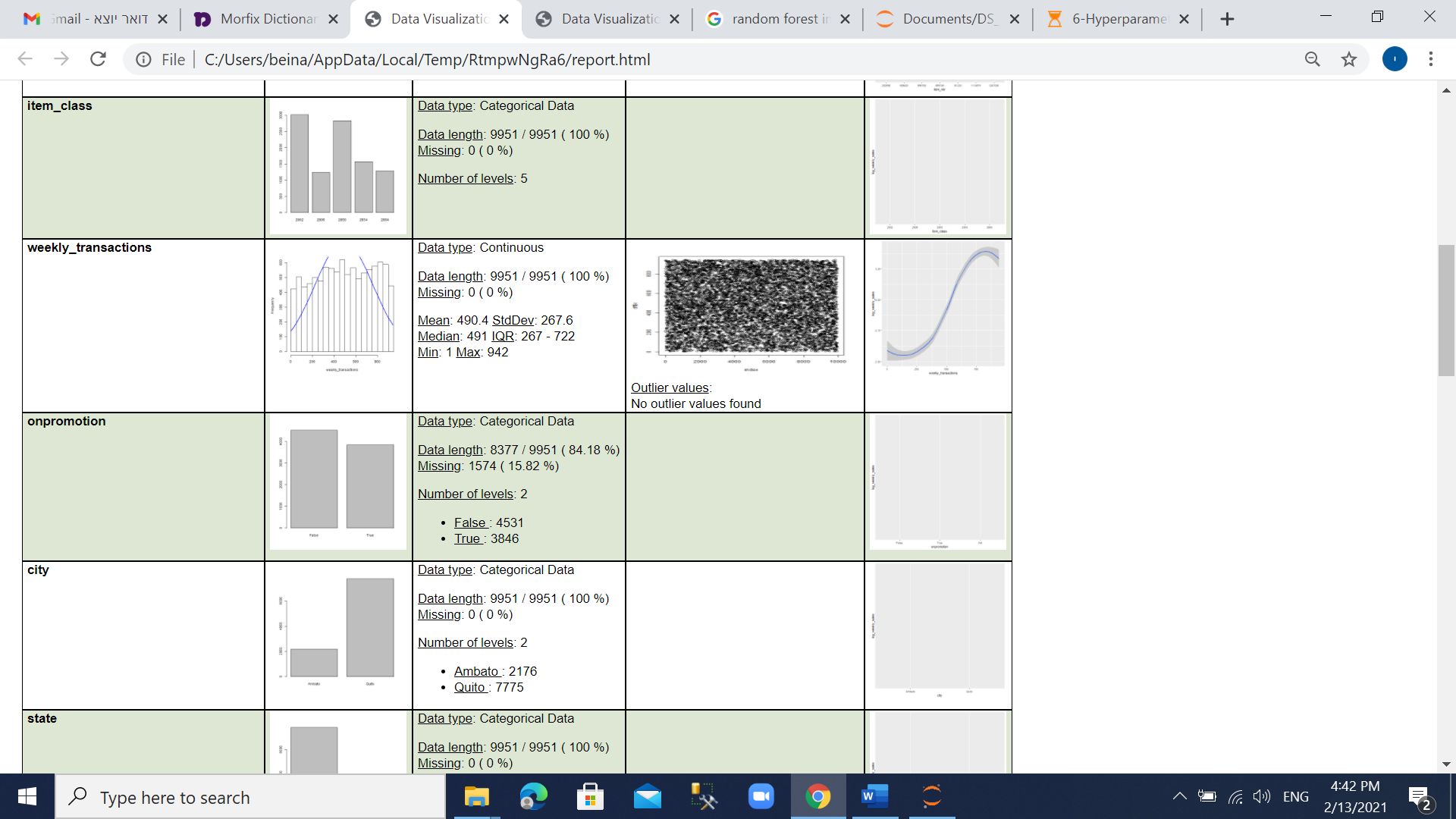


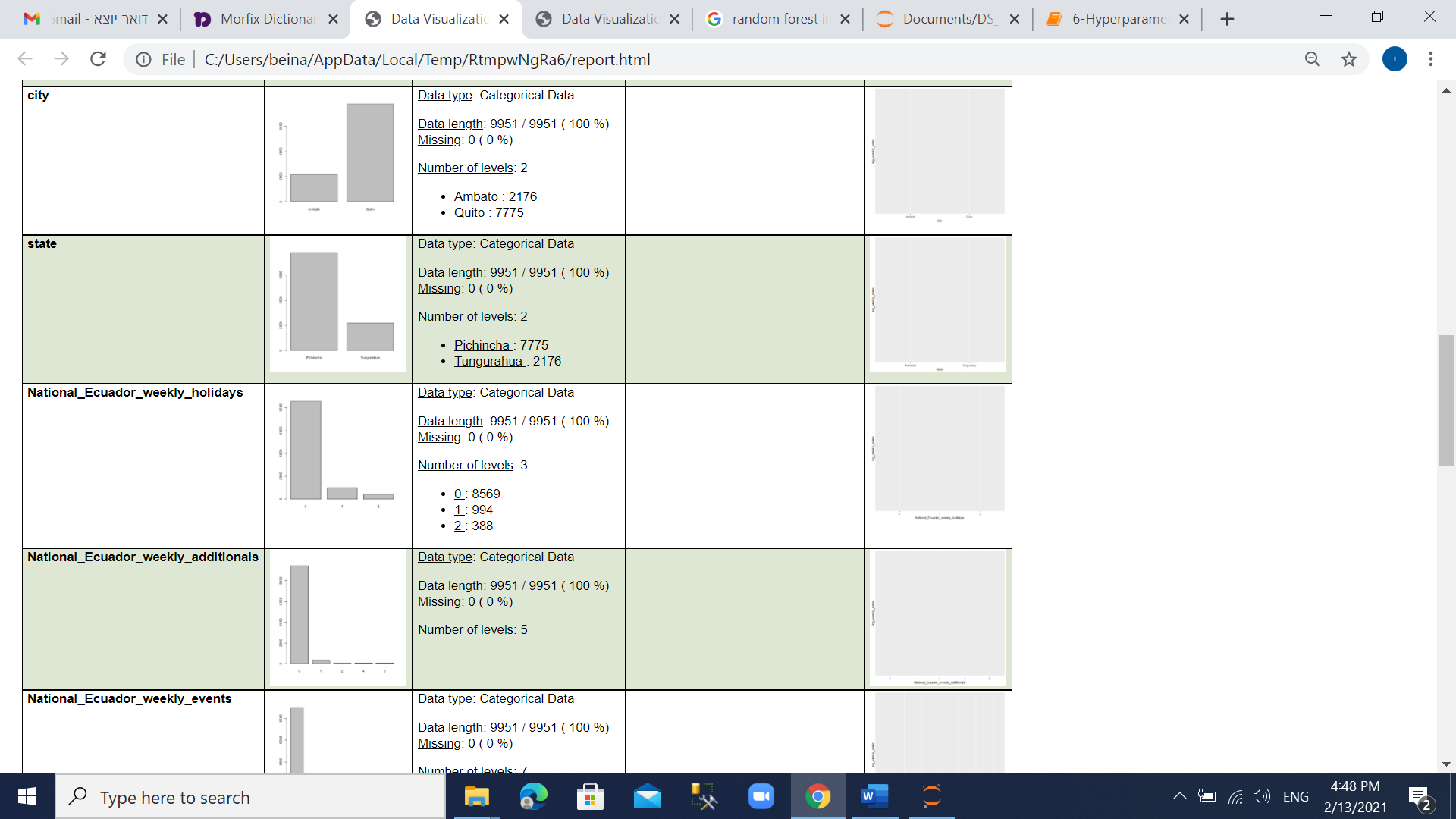
**Transforming the sales count using Log:**

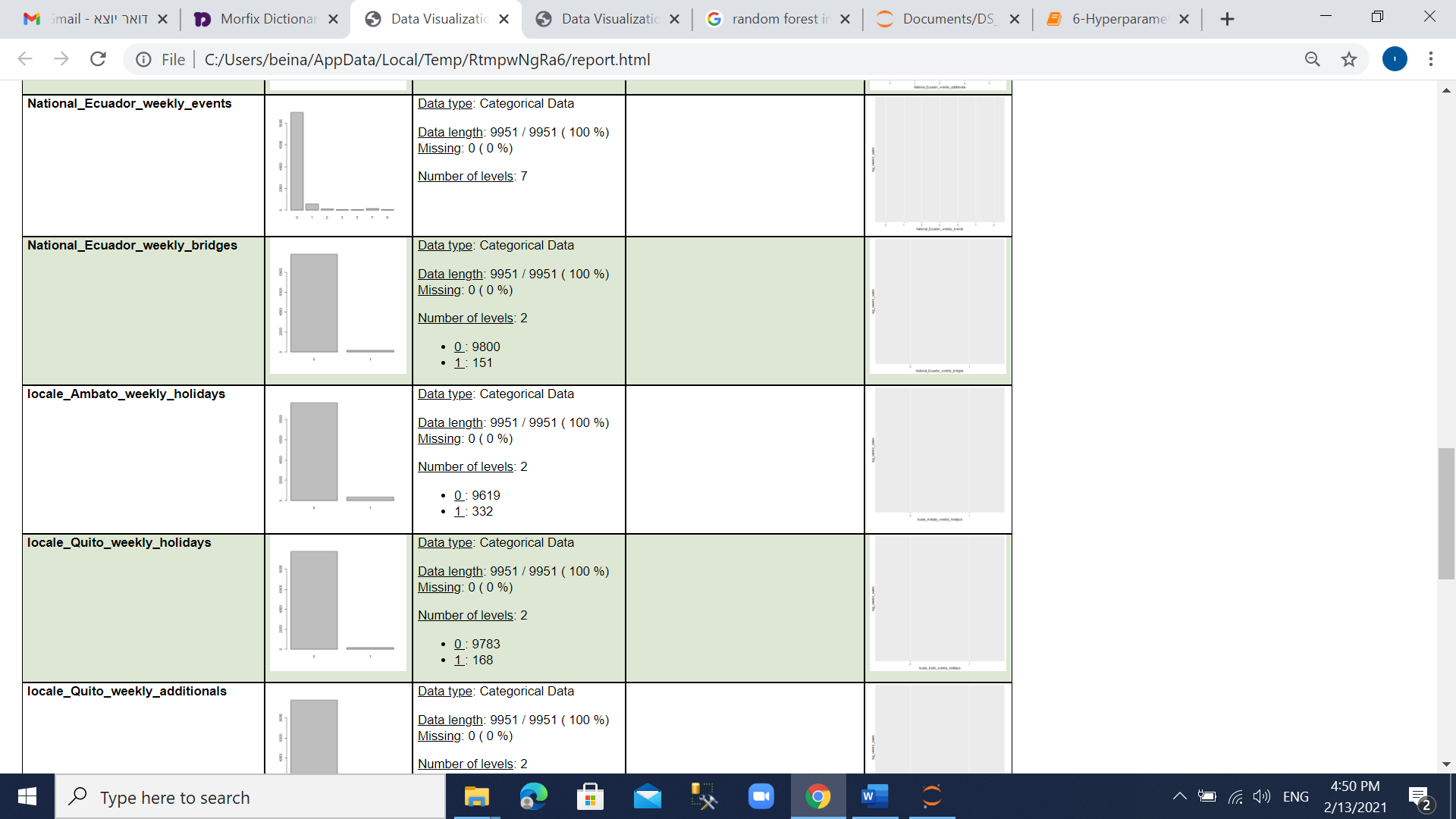


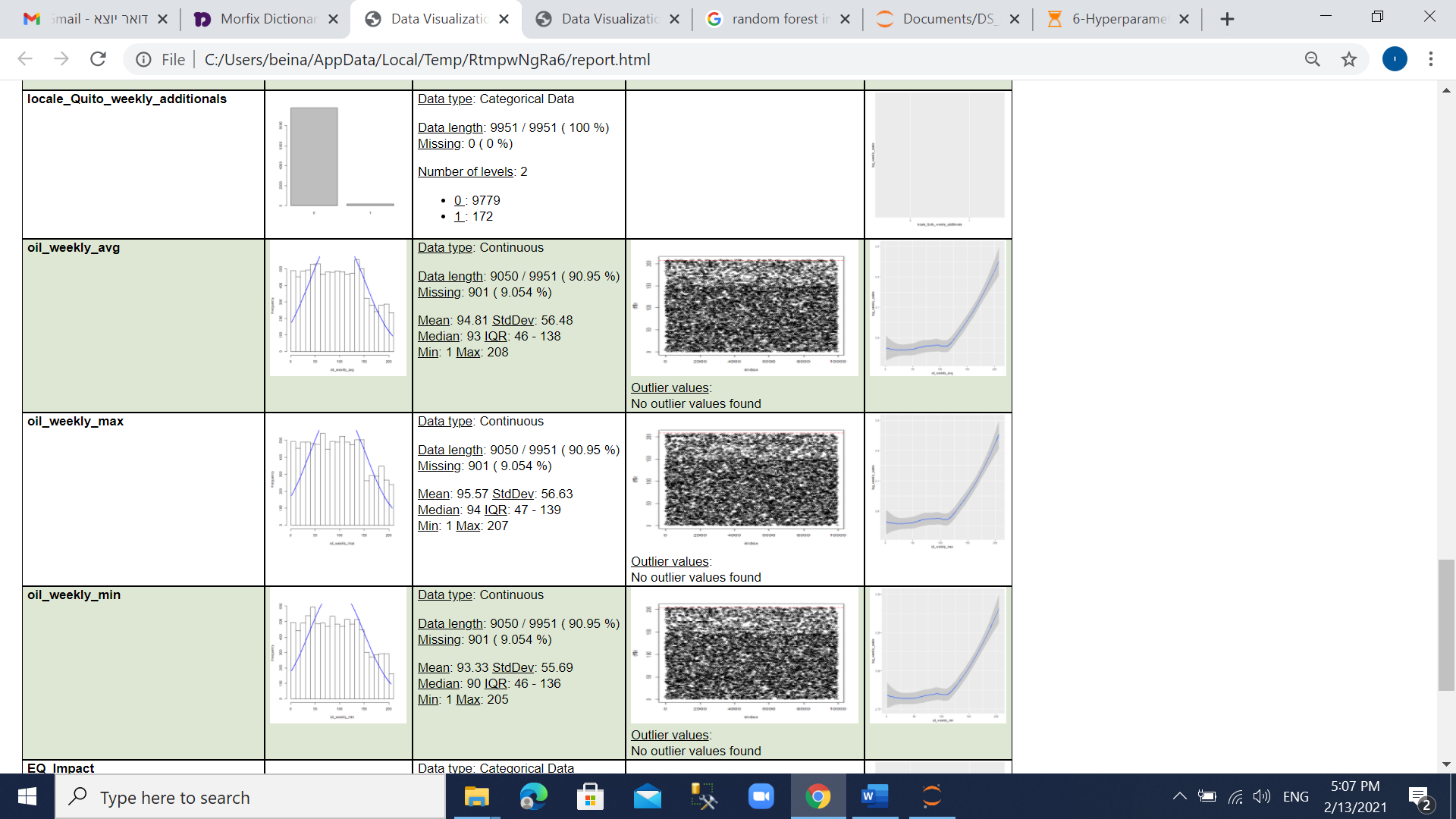
**Data Exploration & Visualization**

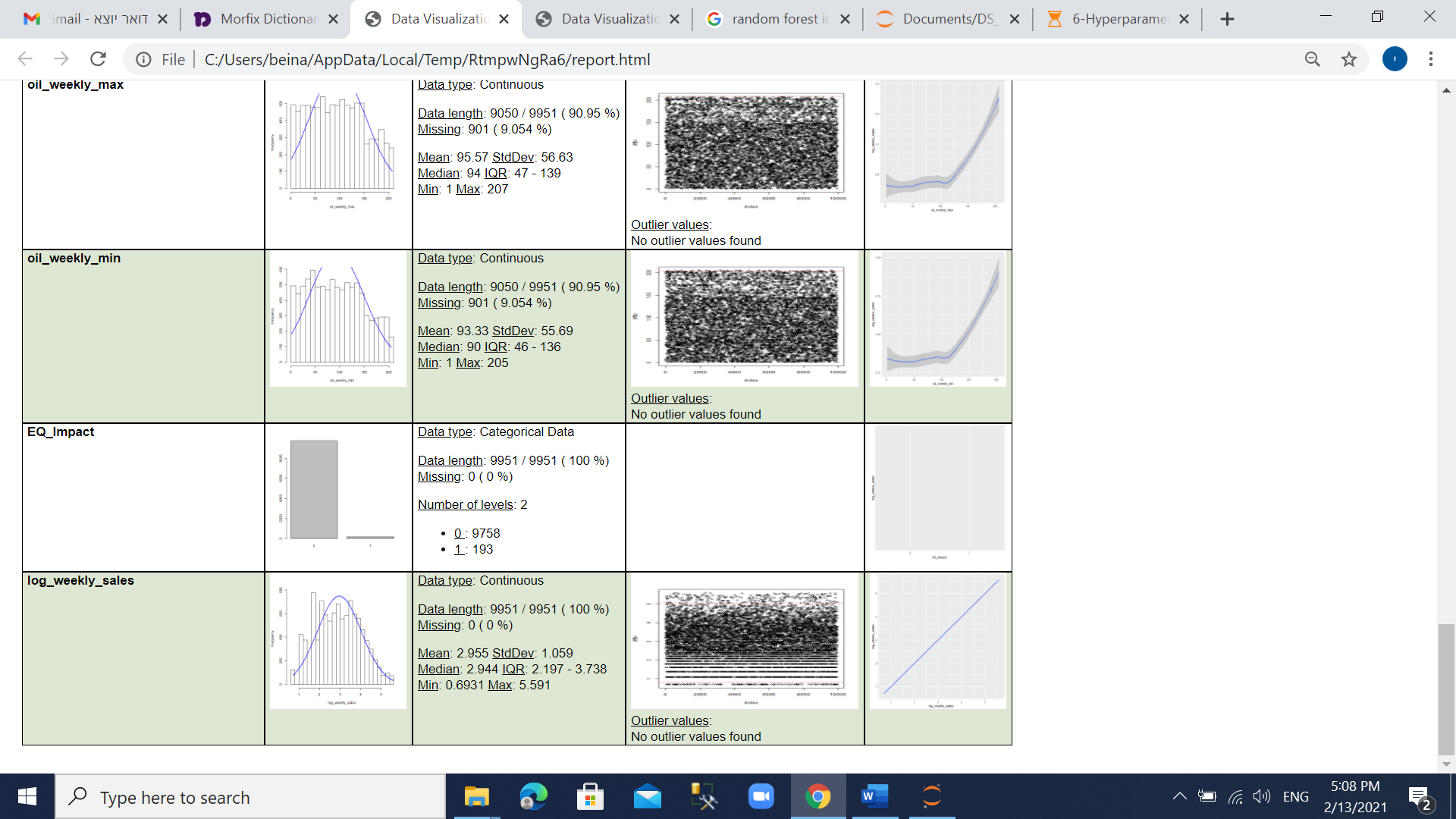




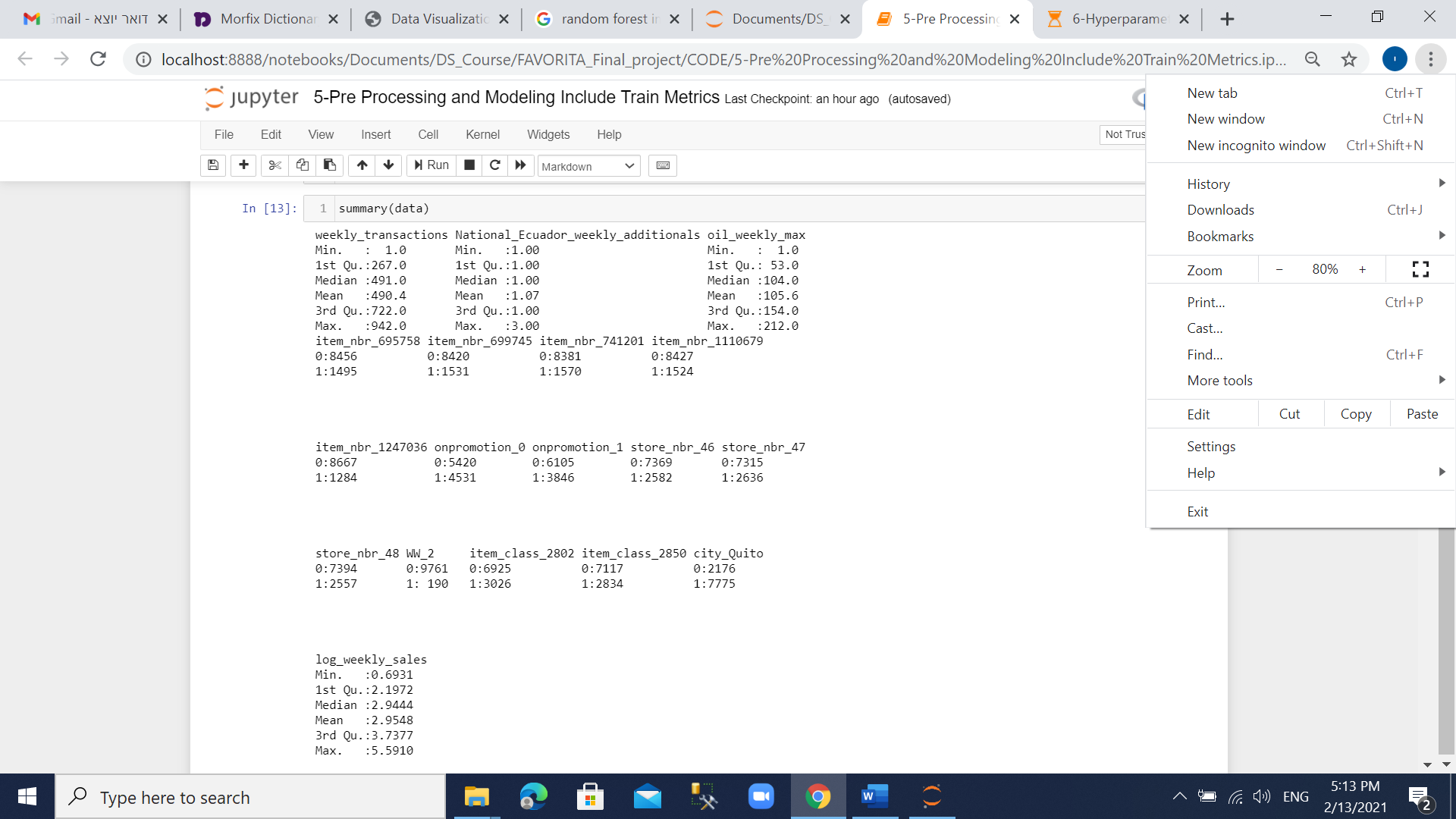








**Data Summary – Selected variables:**



# **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:

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

Excel:

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