Step by step EVO TEST

1 - I started with Cloud Google for looked the meteorological data and created an BigQuey account, after that a project without bills. Also, I aggregated the data from Global Historical Climatology Network Daily Weather Data with SQL quey to check the data.

```
SELECT *
FROM `bigquery-public-data.samples.gsod`
LIMIT 5;
```

```
SELECT DISTINCT year
FROM `bigquery-public-data.samples.gsod`
ORDER BY year DESC

LIMIT 10;
```

Locate the location using latitude and longitude

```
SELECT id, ROUND(latitude, 1) AS latitude, ROUND(longitude, 1) AS longitude
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_stations`
WHERE (latitude = 51.2 AND longitude = -1.6)
        OR (latitude = 51.1 AND longitude = -3.0)
        OR (latitude = 52.2 AND longitude = -1.4)
        OR (latitude = 51.4 AND longitude = 0.7)
        OR (latitude = 51.0 AND longitude = 0.9)
        . . .
        OR (latitude = 55.3 AND longitude = -4.0);
```

To generate the list below, I proceeded with R code:

```
store_master <- read.csv("url", header = TRUE, sep= ';')
sales_data <- read.csv(url, header=TRUE, sep= ';')

output <- paste("OR (latitude =", store_master$latitude, "AND longitude =",
store_master$longitude, ")")

cat(output, sep = "\n")
```

My last query didn't has any match. For this reason I looked for patterns using R finding that all coordinates are between lat: 50:57 and log: -5-5. My next Query was the next one:

```
SELECT id, latitude, longitude

FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_inventory`
WHERE latitude >= 50 AND latitude <= 56 AND  longitude <= 6 AND longitude >=-6;
```

After that I detected that the code UK is a a common pattern, giving the option to generate a more specific query:

```
SELECT DISTINCT id, ROUND(latitude,2) AS latitude, ROUND(longitude,2) AS longitude

FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_inventory`
WHERE id LIKE 'UK%';
```

This table was saved as csv for future uses
To check meteorological data, I analyzed factors that could modify the human buy behaviors Here, I decided to used sun hours, precipitation, snow and temperature.

```
SELECT date, id,element
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2017`
WHERE id LIKE ‘UK%’

UNION ALL
SELECT date,id,element
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2018`
WHERE id LIKE ‘UK%’

UNION ALL
SELECT date,id,element
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2019`
WHERE id LIKE ‘UK%’
ORDER BY date;
```

Transposition in order to improve our data lecture, data normalization and reduce the files quantity

```
SELECT date, id,
    MAX(CASE WHEN element = 'TAVG' THEN value END) AS TAVG,
    MAX(CASE WHEN element = 'SNOW' THEN value END) AS SNOW,
    MAX(CASE WHEN element = 'TSUN' THEN value END) AS TSUN,
    MAX(CASE WHEN element = 'PRCP' THEN value END) AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2017`
```

```
WHERE element IN ('PRCP', 'SNOW', 'TAVG', 'TSUN') AND id LIKE 'UK%'
GROUP BY date, id;
UNION ALL

SELECT date, id,
    MAX(CASE WHEN element = 'TAVG' THEN value END) AS TAVG,
    MAX(CASE WHEN element = 'SNOW' THEN value END) AS SNOW,
    MAX(CASE WHEN element = 'TSUN' THEN value END) AS TSUN,
    MAX(CASE WHEN element = 'PRCP' THEN value END) AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2018`
WHERE element IN ('PRCP', 'SNOW', 'TAVG', 'TSUN') AND id LIKE 'UK%'
GROUP BY date, id;
UNION ALL

SELECT date, id,
    MAX(CASE WHEN element = 'TAVG' THEN value END) AS TAVG,
    MAX(CASE WHEN element = 'SNOW' THEN value END) AS SNOW,
    MAX(CASE WHEN element = 'TSUN' THEN value END) AS TSUN,
    MAX(CASE WHEN element = 'PRCP' THEN value END) AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2019`
WHERE element IN ('PRCP', 'SNOW', 'TAVG', 'TSUN') AND id LIKE 'UK%'
GROUP BY date, id;
```

Now, due to a significant amount of missing data regarding snow and sunlight, these variables were decided to be eliminated from the model, leaving only rainfall and temperature. Subsequently, normalization was applied, categorizing rainfall greater than 4 mm per day as FALSE (based on the average rainfall in the given coordinates, a daily rainfall of more than 4 mm was considered a rainy day) or TRUE.

For temperature normalization, initially, I attempted to use TAVG, but encountered a high percentage of missing values (around 23% of the data). Therefore, I tried using TMAX and TMIN, with TMAX having the highest occurrence rate. We decided to proceed with this variable. Furthermore, based on the estimated average temperature in the UK for the given coordinates, values below 6º were classified as COLD (0), values between 6-15º as MILD (1), and values above 16º as HOT (2). Additionally, the temperature values were expressed in tenths of degrees, so division by 10 was necessary.

```
SELECT date,
    CASE
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) < 6 THEN 0
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) > 15 THEN 2
        ELSE 1
    END AS TMAX,
```

```
    CASE
       WHEN MAX(CASE WHEN element = 'PRCP' THEN value END)<4 THEN 0
       ELSE 1
    END AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2017`
WHERE id LIKE 'UK%'

UNION ALL
SELECT date,
    CASE
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) < 6 THEN 0
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) > 15 THEN 2
        ELSE 1
    END AS TMAX,
    CASE
       WHEN MAX(CASE WHEN element = 'PRCP' THEN value END)<4 THEN 0
       ELSE 1
    END AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2018`
WHERE id LIKE 'UK%'

UNION ALL
SELECT date,
    CASE
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) < 6 THEN 0
        WHEN MAX(CASE WHEN element = 'TMAX' THEN value/10 END) > 15 THEN 2
        ELSE 1
    END AS TMAX,
    CASE
       WHEN MAX(CASE WHEN element = 'PRCP' THEN value END)<4 THEN 0
       ELSE 1
    END AS PRCP
FROM `datos-meteorologicos-389713.ghcn_daily.ghcnd_2019`
WHERE id LIKE 'UK%'
```

DETAILED WORK STEPS - INSTRUCTIONS
1) Load data from CSVs

2) Before data analysis
Look at the numbers, and think, "top down":
- Would you use all the data?

- Would you use all the available variables or not? (eg store, category)

- Would you add other variables in the analysis?

Take a quick decision to produce your final result, so to simulate real conditions; reasoning on the approach that you would use to respond to the points above, even without doing all the calculations. Please document your choices and prepare to talk about them later. Choose 10 items to forecast and motivate your choice properly.

To address the previously stated problem, a representative random sampling could be employed to avoid using the entire dataset, resulting in computational savings (especially crucial when working with cloud-based data that incurs usage costs). However, while this approach reduces the data, it may introduce errors due to inherent randomness, which would necessitate developing multiple models, still incurring computational expenses.

Taking the above into consideration, I will proceed with using the data from the last two years (2018-2019) for the following reasons::

1. Reduce the amount of data to process
2. Relate last year trends.
3. Avoid atypical events that can trigger our sales.

However, to develop the learning models, around 70% of the data will be used for training, avoiding overfitting, and 30% for functionality checking.

Regarding the variables when analyzing the movement of units and not having complete information on the environment where the products operate, I consider it pertinent to add the weather and holidays. I also consider it appropriate to remove the column

1. "item_category" don’t has any value.

Finally, and in order to select 10 items to forecast, the following critical points are proposed:

1. Those who have the largest volume of product movement.
2. Take into account the item unit_price (price that we pay for each of the elements) detecting which ones will generate greater losses in case the flow of units is delayed.
3. Also, we could perform a sensitivity analysis regarding movement on holiday dates and how the weather affects movement within the warehouses.

4. According to the trends of the last six months, detect which are the products with the greatest growth in storage
5. Detect seasonal products that may suffer a large drop or rise in their storage levels.

Given these parameters, the outputs provided by R:

Historical qty

	item	total_qty	percentage
	<chr>	<int>	<dbl>
1	185	2577551	5.24
2	186	1941479	3.94
3	416	1587703	3.23
4	385	1360864	2.76
5	519	1112612	2.26
6	179	1106302	2.25
7	408	972094	1.97
8	346	931859	1.89
9	352	868853	1.77
10	158	864918	1.76

Historical sales

	item	total_sales
1	186	56338051.
2	416	38555461.
3	385	36632655.
4	179	35433506.
5	48	31620673.
6	410	23499356.
7	127	22861446.
8	163	21853251.
9	185	21311796.
10	436	20790258.

Highest qty in the last 6 months

item	total_qty
416	433854
186	393811
528	351127
385	271319
589	247194
465	199954
179	188733
596	186638
526	183831
415	182596

TOP 10 of sales on the last 6 months

month	item1	item2	item3	item4	item5	item6	item7	item8	item9	item10
2018-10	416	526	408	385	186	519	528	346	309	352
2018-11	416	186	408	385	526	519	528	561	309	163
2018-12	186	385	408	416	526	519	528	561	309	163
2019-01	528	416	186	519	408	385	179	589	465	415
2019-02	416	186	528	385	408	589	519	465	309	596
2019-03	416	186	385	589	528	526	465	309	596	415

TOP 10 of Items for season

year	season	item_1	item_2	item_3	item_4	item_5	item_6	item_7	item_8	item_9	item_10
2017	invierno	185	186	385	416	170	446	295	179	296	158
2017	otoño	185	186	416	385	179	170	413	295	286	346
2017	primave	185	158	186	385	170	414	179	295	416	43
2017	verano	185	186	170	179	385	416	414	286	295	159
2018	invierno	186	519	408	416	385	179	346	502	352	436
2018	otoño	416	186	526	408	519	385	528	346	309	589
2018	primave	519	408	186	416	385	179	158	286	436	346

2018	verano	519	526	408	528	186	527	416	179	309	159
2019	invierno	416	528	186	385	519	408	589	465	179	596
2019	primave	416	186	385	589	528	526	465	309	596	415

From the analysis of trends by season, the only clear pattern visible is item 186. We can see that the amounts transacted for this item have a certain cyclical behavior around three months.

With the results of the data, I will choose to continue my analysis with the following items:

- 179, for being the 4th most representative item in our balances and constant appearance in summer seasons.
- 186, 2nd item with the largest number of units and who represents our highest monetary value, being permanently in our TOP 3 units
- 309, 66% of appearances in our TOP TEN of the last months and shows a growth in the market share position.
- 385, 4th item with the largest number of units and who represents our third highest monetary value, usually being the TOP 5 of units with the greatest movement
- 408, an item that represents a great variation in our historical TOP 10 and currently has lost presence in the market but in recent months it was among those that have represented the most units
- 416, our last 6 months it was the No. 1 of units with movement four times.
- 465 shows signs of market share growth over the past three months advancing one position per month.
- 526, TOP 10 in the last 6 months.
- 528, 3rd in movements in the last 6 months with peak in January and rapid reduction in quantity.
- 589, has been growing by leaps and bounds in market share in terms of quantity
- 596, great growth in the last two months and seasons.

3) Model data

Apply an appropriate forecasting model, for example a multi-variate regression, time series or another model of R (there are many available modules), as long as the results are a good prediction for the variable Qty. The model can account for special factors e.g. holiday effect.

The results obtained from the evaluated models were as follows:

Metric	MVregression	ARIMA	Random Forest	XBoost
MAE	156.4186	148.634	155.3947	154.8423
RMSE	281.4475	284.3188	281.4764	281.0004
R2	0.008456686	-0.01187776	0.008252956	0.01160437

Based on the results, the XGBoost model shows slightly better performance in terms of MAE, RMSE, and R2 compared to the other models. However, it is important to note that the performance values are quite similar between the models and are quite low overall, indicating that neither model provides a highly accurate prediction.

Given this, it is advisable to further evaluate the models, consider other relevant factors, and make adjustments or explore other modeling methods to further improve the predictions.

It could be a data sampling problem, so I decided to also use the data from 2017. Without improving results.

Therefore, it was attempted by using the variable store and item and the modification to 0.8 of training. The results resulted in a significant improvement being the following:

Metric	MVregression	ARIMA	Random Forest	XBoost
MAE	135.757	128.6488	131.6297	19.28
RMSE	249.2807	225.5795	230.5863	32.21
R2	0.008456686	-2.880928e-06	0.2894963	0.9735041

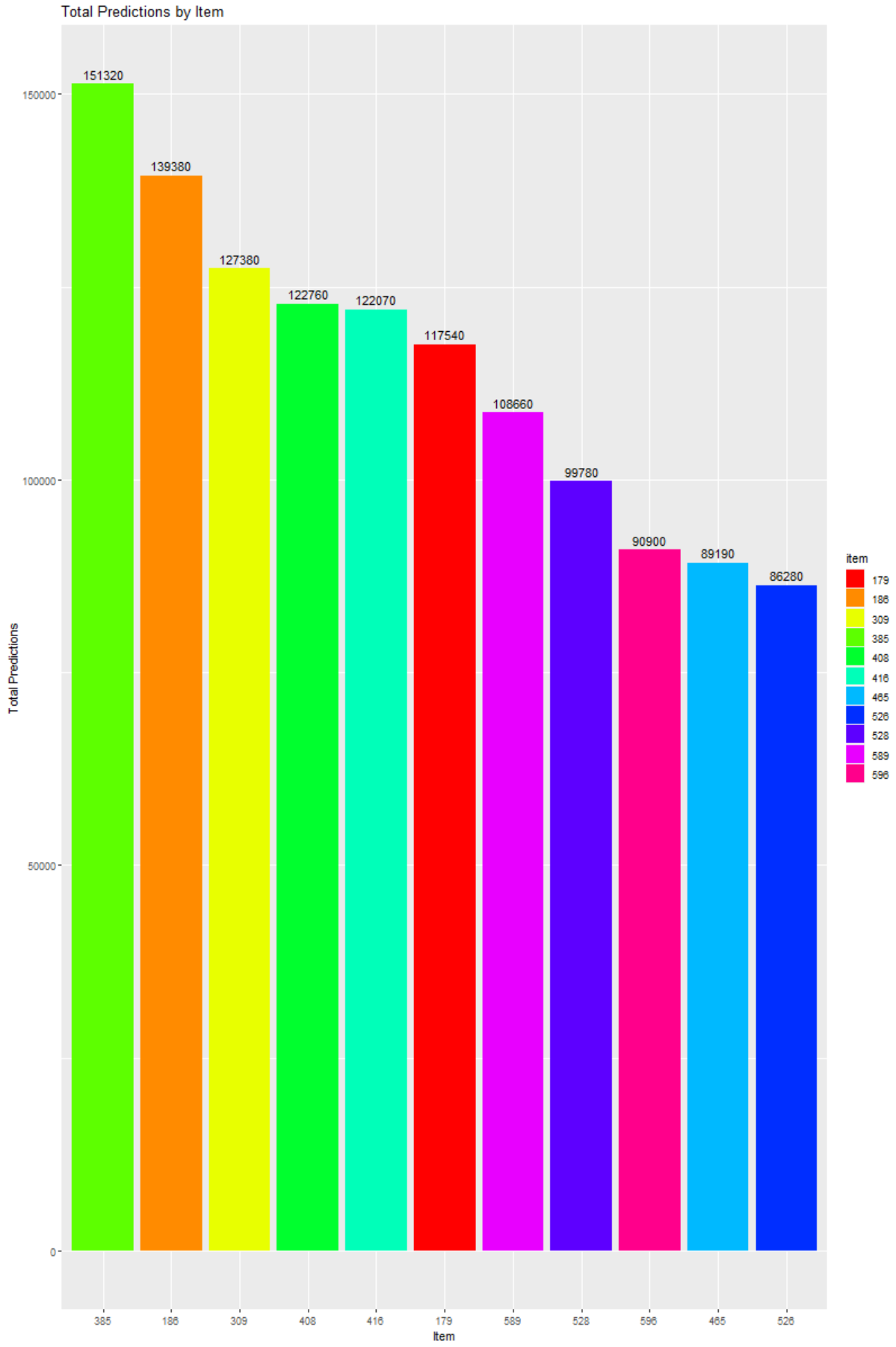
I tried the training with the 0.7 again, just improving the performance, so the model at this point I think is beginning to be promising.

Finally, an attempt was made to add the variable season to the analysis without results of great interest.

Therefore, I will continue with the estimation of future values using the XGBoost model of 0.7 training.

4) Estimated future values

Once the model is built, make an estimate for 30 subsequent days for the 10 items you chose.
Take values for future input variables (if any) that are - in your opinion - reasonable. Obviously you miss a lot of context, so you have to make some independent assumptions (as it happens in reality); perhaps this test has a 'more extreme' lack of context, but substantially you will face a situation not much unlike in reality. Let yourself be guided by common sense and ease of implementation.


Total Predictions by Item
151320
139380
127380
122760
122070
117540
108660
99780
90900
89190
86280
150000
100000
50000
0
Total Predictions
385
186
309
408
416
179
589
528
596
465
526
Item
item
179
186
309
385
408
416
465
526
528
589
596

5) Error rate and self-feedback

What error measure would you choose to use to assess your own results? How well do you think your code performs? What other future adjustments or work could further improve this?

I think that in this case with an r2 of 0.97 only 147 values will be wrong, so the code works with a good prediction. In case you want to improve, you could obtain a particular analysis on a calendar of events that trigger sales, such as presidential elections, sporting events, and others.