Multivariate
time series
forcasting challenge

1 – Problem description

t	F^1	F^2	F^3	…	F^P
1	x_1^1	x_1^2	x_1^3	…	x_1^p
2	x_2^1	x_2^2	x_2^3	…	x_2^p
3	x_3^1	x_3^2	x_3^3	…	x_3^p
4	x_4^1	x_4^2	x_4^3	…	x_4^p
…	…	…	…	…	…
n	x_n^1	x_n^2	x_n^3	…	x_n^p

- *P* features (F^1 to F^P) / *N* timesteps (t_1, t_2, …., t_n)
- $x_i^j \in [a_1, a_2, \ldots, a_z]$ $(i \in [1..n],\ j \in [1..p])$
- *Z numeric values, shared by all features:* a_k $(k \in [1..z])$
- $x_i^1 < x_i^2 < x_i^3 < x_i^4 < \ldots < x_i^p$ $(i \in [1..n])$

2 – Example

#	H1	H2	H3	H4	H5	H6	H7	M1	M2	M3	M4	M5	M6	L1	L2	L3	L4	L5	L6	L7
00001	1	2	3	5	9	11	13	14	24	27	32	45	47	58	60	64	65	67	69	70
00002	2	4	5	7	11	12	22	23	27	28	31	41	44	46	48	53	54	62	66	70
00003	3	5	9	16	21	24	28	33	36	37	39	42	43	44	46	53	57	60	65	70
00004	5	7	10	17	26	28	32	33	35	36	39	42	44	46	49	52	57	58	60	68
00005	3	7	8	9	11	12	18	22	26	29	30	33	40	46	48	49	55	58	60	63

- Above is a sample of the first 5 rows with header of the dataset
- Complete dataset in the CVS file
- The dataset contains:
  - More than 15,000 observations ➔ ***n***
  - 20 features (7 High, 6 Middle, 7 Low) ➔ ***p***
  - 70 unique values ➔ ***z***

For information only, in real life, there are 43 features and 255 values (00 to FF)

3 – Objective #1

t	F^1	F^2	F^3	…	F^P
1	x_1^1	x_1^2	x_1^3	…	x_1^p
2	x_2^1	x_2^2	x_2^3	…	x_2^p
3	x_3^1	x_3^2	x_3^3	…	x_3^p
4	x_4^1	x_4^2	x_4^3	…	x_4^p
…	…	…	…	…	…
n	x_n^1	x_n^2	x_n^3	…	x_n^p
n+1	$x_n^{1}{}_{+1}$	$x_n^{2}{}_{+1}$	$x_n^{3}{}_{+1}$	…	$x_n^{p}{}_{+1}$

➔ **Forcast the *p* values of the next timestep t_{n+1}**

Alternative:

As it can be difficult (or not possible) to forcast exactly the unique solution of *p* values at timestep $\mathbf{t_{n+1}}$, an alternative way can be, for each feature F^i, to define a small set of possible values, as for example:

F^1: [2, 9, 17] F^2: [5, 9, 16, 21] F^3: [10, 29, 35] etc…

And then, combine them each other, to generate a set of possible combinations of *p* values: the exact solution should be inside this set.

4 – Objective #2

➔ **Forecast some values from the previous timesteps t_n, t_{n-1}, t_{n-2}, t_{n-3}, ... in order to:**

- **remove those for which we are sure they will not be present at timestep t_{n+1}**
- **select those for which we are sure they will be present at timestep t_{n+1}**

As we have to find the solution of *p* values at timestep t_{n+1}, identifying these values can be very helpfull for the algorithm or model to be more efficient to achieve the goal #1, because it will reduce the set of possible values.

Example:

If $[a_1, a_2, ..., a_z]$ = [2, 9, 12, 23, 24, 30, 33, 48, 50, 65] then

If we are sure that at timestep t_{n+1} :

- values [12, 30] will not be present
- value [2] will be present

Then the set of possible values for the remaining values to be predicted will be reduced to:

[9, 23, 24, 33, 48, 50, 65]

5 – Expected results at timestep t_{n+1}

- Goal #1 ➔ predict **solutions** of *p* values:

unique and **exact** solution	**sets** of possible values ➔ **multiple** solutions
undefined, none, $x_n^{1}{}_{+1}$, $x_n^{2}{}_{+1}$, $x_n^{3}{}_{+1}$, $x_n^{4}{}_{+1}$, … , $x_n^{p}{}_{+1}$	undefined, none, F^1:[a_1, a_5, a_8], F^2:[a_2, a_4, a_{10}, a_{11}], …, F^P:[a_7, a_{10}, a_{12}, a_{19}]

A **B**

- Goal #2 ➔ predict **values** which will:

be present	**not be** present
undefined, none, (*) p_1, p_2, p_3, …	undefined, none, (*) q_1, q_2, …

C **D**

(*) In both case (C&D), if the value is not « undefined » nor « none », the number of values selected must be at least 1

Note:
- *undefined = the system can not predict any value/solution*
- *none = the system predict explicitly that no value/solution will be/not be present at timestep t_{n+1}*

6 – Evaluation criteria for a solution

In order to evaluate each solution predicted, a system with points has been defined to measure the performance of a solution.

Goal #1 / Case A & B

Unique (A) or multiple (B) forcasted solution of *p* values

For each solution (A or B):

Number of values in timestep t_{n+1}	Number of points
20	1 000 000
19	250 000
18	62 500
17	12 500
16	2 500
15	500
14	100
13	20
12	5
11	2
10	1

Goal #2 / Case C

Values which will be present at timestep t_{n+1}

Number of values selected	Number of values in timestep t_{n+1}	Number of points
10	10	200 000
	9	2 000
	8	100
	7	10
	6	5
	5	2
9	9	40 000
	8	100
	7	20
	6	8
	5	2
	4	1
8	8	8 000
	7	100
	6	20
	5	5
7	7	3 000
	6	70
	5	5
	4	2
6	6	900
	5	30
	4	2
5	5	80
	4	10
	3	2
4	4	50
	3	5
3	3	10
	2	2
2	2	6
1	1	2

Note: if the number of values selected is greater than 10, generate all combinations of 10 values. Then, for each of them, refer to the table above to know the number of points corresponding to each combination. The final number of points in this case is the sum of the number of points of all combinations

Goal #2 / Case D

Values which will not be present at timestep t_{n+1}

Number of values removed	Number of values not in timestep t_{n+1}	Number of points
10	10	2
9	9	1
8	8	2
7	7	1
6	6	1
5	5	1
4	4	1
3	3	1
2	2	1
1	1	1

Note: if the number of values removed is greater than 10, generate all combinations of 8 values. Then, for each of them, refer to the table above to know the number of points corresponding to each combination. The final number of points in this case is the sum of the number of points of all combinations

7 – Evaluation criteria for a model (1/2)

The CVS file contains a complete dataset which can be split in several parts to train, test and validate the model(s).

For the final validation, about 100 new datasets will be used to evaluate efficiency, accuracy and performance of the model(s) and then, the quality of the prediction.

For each new dataset, the 4 cases can be forecasted:

- The unique solution of *p* values ➔ goal #1 / case A
- All sets of possible values for each feature, and then, all combinations of *p* values ➔ goal #1 / case B
- Values wich will be present at timestep t_{n+1} ➔ goal #2 / case C
- Values wich will not be present at timestep t_{n+1} ➔ goal #2 / case D

Note: in the response of the final prediction, at least 1 case must be given. It means that it is not necessary to evaluate all cases for the final prediction: A, B&C, A&C&D or A&B&C&D are valid responses

For each prediction, the number of points will be calculated for each case (A, B, C and D) according to the rules defined in the previous slide.

7 – Evaluation criteria for a model (2/2)

For each case, the following rules (in red) must be respected after processing the new datasets to validate the model(s).

	Goal #2 ➔ values which will be, at t_{n+1}		**Goal #1 ➔ Predicted solution at t_{n+1}**	
	present	not present	unique	multiple
Case	**C**	**D**	**A**	**B**
Initial points at start	10	10	10	70
Cost of a solution *(a)*	1 point	1 point	1 point	1 point / combination
Number of points *(b)*	0 to 200 000 points	0 to 2 points	0 to 1 000 000 points	0 to ? points
Sum of points *(after processing the new datasets)*	Must be positive ➔ *((number of points – cost) for each solution for all new datasets) > 0*			
Performance required	80 %	80 %	15 %	50 %

Important:

- *For each case (A, B, C & D):*
  - *« undefined » cost and give 0 point, and is not considered as successfull prediction,*
  - *« none » cost 1 point and give 0 point, and is not considered as successfull prediction*
  - ***As each new prediction costs 1 point (or more if case B), if for a new prediction there is not enough point to predict case A, B, C and/or D, then the evaluation stops: no more prediction can be done for the considered case!***
- *Performance is calculated as this: (Number of successfull predictions / Total number of predictions) x 100 ➔ « Successfull prediction » means prediction where the number of points is positive (see slide 6 – Evaluation criteria for a solution)*

8 – Prize

The total prize pool for this job is **5,000 US$**, and is structured as shown in the table below:

CASE	A	B	C	D
Prize US$	2,000	1,000	1,000	1,000

The prize is due <u>if and only if</u> the expected performance is <u>fully</u> reached.

In case of the performance obtained for a given case (A, B, C or D) is less and close (up to 20% less max.) to the expected performance, the amount of the prize <u>for the considered case</u> will be determined <u>according to our appreciation</u>.

In case of the performance obtained for a given case (A, B, C or D) is greater than the expected performance, a bonus will be applied according to the following rules, <u>for the considered case</u>:

- 0% < P ≤ 10% ➔ bonus = prize x 0.20
- 10% < P ≤ 20% ➔ bonus = prize x 0.85
- 20% < P ➔ bonus = prize x 1.50

$$P = 100\ x\ \frac{(Perf.\ obtained\ - Perf.\ expected)}{Perf.\ expected}$$

If results are not reached, a fixed revenue (**250 US$**) will be paid for all that work.

9 – Others

Keep in mind that, if the proposed solution fully meets the requested objective, this solution will be installed on the client side: this means additionals tasks as support to the installation, training if new tools/software are used, etc… This is already include in the prize.

The operating mode (progress, milestones, intermediate deliveries, validation process, etc.) will be defined between the parties before the mission starts.