ProApes

proapes11@gmail.com

User Manual

Version	1.0.0-1.10
Approval date	2020-05-15
Project Manager	Federico Carboni
Editors	Federico Carboni Alessandro Discalzi
Verifiers	Fiammetta Cannavò
State	Approved
Distribution list	*ProApes* *Prof. Tullio Vardanega* *Prof. Riccardo Cardin*
Usage	External

Summary

This document contains the definition of the functionalities of *Predire in Grafana*, and a usage guide.

History

Version	Data	Changes	Author	Role
v1.0.0-1.10	2020-05-15	*Document approved for RA*	Federico Carboni	*Project Manager*
v0.1.0-1.10	2020-05-13	*Reiew and corrections of coherence and cohesion(*Verifier*: Fiammetta Cannavò)*	Alessandro Discalzi	*Programmer*
v0.0.3-1.10	2020-05-12	*Written down §4, 5, §6 and appendice §A (*Verifier*: Fiammetta Cannavò)*	Alessandro Discalzi	*Programmer*
v0.0.2-1.10	2020-05-11	*Written down §1, §2, §3 (*Verifier*: Fiammetta Cannavò)*	Federico Carboni	*Programmer*
v0.0.1-1.9	2020-05-09	*LaTeX document created*	Federico Carboni	*Project Manager*

Index

1 Introduction **4**

1.1 Purpose of this document 4

1.2 Product Perspective 4

1.3 Glossary 4

2 System requirements **5**

2.1 Prerequisite 5

2.2 Hardware recommendation 5

2.3 Software requirements 5

2.4 Bug report 5

3 `Training Module`: installation guide **6**

4 `Prediction Module`: installation guide **8**

5 `Training Module`: how to use **9**

5.1 Choosing the algorithm 9

5.2 File format and uploading 9

5.3 Options editing 9

5.4 Start the training 9

5.5 Download predictor 9

5.6 Reset 9

5.7 File format 9

5.7.1 *CSV* file 9

5.7.1.1 *Linear Regression* 10

5.7.1.2 *Support Vector Machine* 10

5.7.2 File *JSON* 10

5.8 Training options 11

5.8.1 *Linear, Exponential and Logarithmic Regression* 11

5.8.2 *Support Vector Machine* 12

6 `Prediction Module`: how to use **13**

6.1 Enabling the application and creating a sample dashboard 13

6.2 Importing panels in a preexisting dashboard 16

6.3 Use of the *Prediction Panel* 17

A Glossary **20**

List of figures

1 Example of the usage of *command* 6
2 Application startup 7
3 Sample of a *CSV* file used for *RL* training 10
4 Sample of a *CSV* file used for *SVM* training 10
5 Sample of a *JSON* file obtained from a previous training 10
6 *Regression* options 11
7 *SVM* options 12
8 `Plugins` tab 13
9 Result of the research in `Plugins` tab 13
10 Enabling the application 14
11 Creating a sample dashboard 14
12 Sample dashboard 15
13 Internal training tab 15
14 `Add panel` button 16
15 Selecting `Prediction Panel` 16
16 Example of a query 17
17 Panel `Edit` button and import of a new predictor. 17
18 Manual alert configuration 18
19 Guided alert configuration 18
20 Example of a query 19

1 Introduction

1.1 Purpose of this document

The purpose of this document is to explain all the features of the *Predire in Grafana plug-in*. The end-user will have all the information that could be needed for the correct software usage.

1.2 Product Perspective

The tender specifications$_{\mathbf{G}}$ **C4 -** *Predire in Grafana* arise from the need to constantly monitor resources, applications and information, after applying a DevOps$_{\mathbf{G}}$ approach to the software lifecycle. *ProApes* group propose is to develop a *Grafana*$_{\mathbf{G}}$ *plug-in*$_{\mathbf{G}}$ for *Zucchetti S.p.A.*
The *plug-in* will apply *Linear Regression*$_{\mathbf{G}}$, *Support Vector Machine*$_{\mathbf{G}}$, *Exponential Regression*$_{\mathbf{G}}$ and *Logarithmic Regression*$_{\mathbf{G}}$ prediction algorithms to the data flow; its goal is to predict whenever there could be a problem in production to fix that before it happens.

1.3 Glossary

In this document, there could be terms with ambiguous or contradictory meanings. To avoid any incomprehension, at the end of the document, in appendix §A, you can find a glossary with these terms and their explanations. In the following sections, to promote clarity without being redundant, glossary words will be marked with a subscripted "G" at their first occurrence of each section.

2 System requirements

This section lists the minimum hardware and software requirements necessary for a proper use of *Predire in Grafana* plug-in$_{\mathbf{G}}$.

2.1 Prerequisite

- *Grafana v6.7.x*;
- *NodeJS v13.7.x* or above;
- *Yarn v1.22.0* or above;

2.2 Hardware recommendation

Predire in Grafana is composed of two modules, a *training module* and an *internal module*; we recommend to follow ***Grafana*** hardware recommendation:

- 2GB RAM;
- dual-core CPU.

For further information, refer to:

`https://grafana.com/docs/grafana/latest/installation/requirements/`

2.3 Software requirements

- **OS:** *Windows, MacOS, Debian, Ubuntu and RPM-based Linux*;
- **Browser:**
  - *Google Chrome* v58 or above;
  - *Microsoft Edge* v14 or above;
  - *Mozilla Firefox* v54 or above;
  - *Apple Safari* v10 or above.

2.4 Bug report

If you find bugs or any other kind of problem using the software, please report it at:

proapes11@gmail.com

3 Training Module: installation guide

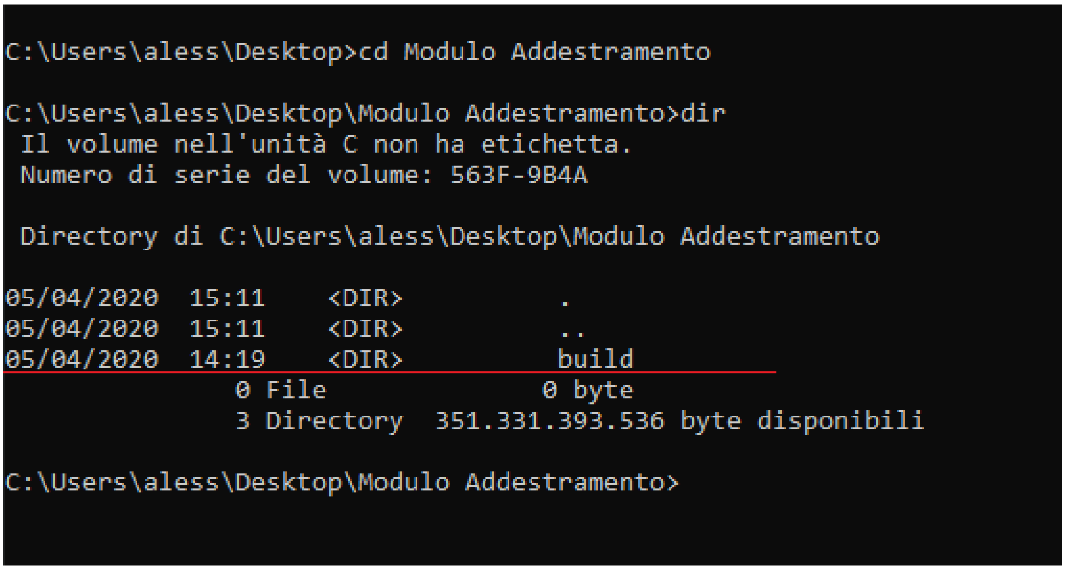
To install the training module you need to:

- download `dist` folder from the *GitHub Repository*$_{\mathbf{G}}$

https://github.com/Kero2375/proapes-predire-in-grafana/tree/master/Training%20Module

- open the command prompt (*Windows*) or the terminal (*Linux/MacOS*);
- place yourself in the folder you have just downloaded using the command

```
cd /path
```



```
C:\Users\aless\Desktop>cd Modulo Addestramento

C:\Users\aless\Desktop\Modulo Addestramento>dir
 Il volume nell'unità C non ha etichetta.
 Numero di serie del volume: 563F-9B4A

 Directory di C:\Users\aless\Desktop\Modulo Addestramento

05/04/2020  15:11    <DIR>          .
05/04/2020  15:11    <DIR>          ..
05/04/2020  14:19    <DIR>          build
               0 File              0 byte
               3 Directory  351.331.393.536 byte disponibili

C:\Users\aless\Desktop\Modulo Addestramento>
```

Figura 1: Example of the usage of *command*

If it's your first application startup:

- type in the terminal:

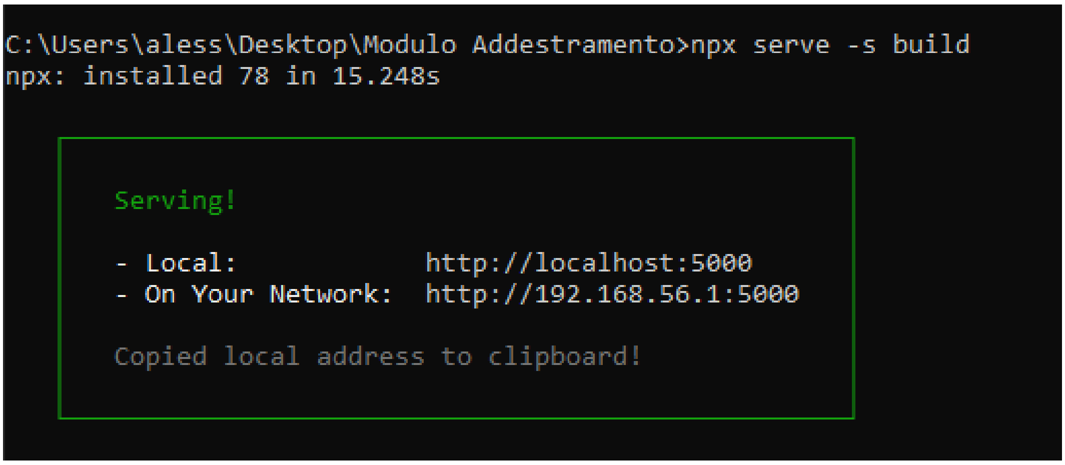
```
yarn global add serve
```

- then type:

```
serve -s build
```

- finally open the link displayed in the terminal (e.g: `http://192.168.56.1:3000`).

N.B.: `build` is the name of the unzipped folder. If you changed the name of that folder, change also `build` with the new name in the terminal.



```
C:\Users\aless\Desktop\Modulo Addestramento>npx serve -s build
npx: installed 78 in 15.248s

   ┌──────────────────────────────────────────────────┐
   │                                                  │
   │   Serving!                                       │
   │                                                  │
   │   - Local:            http://localhost:5000      │
   │   - On Your Network:  http://192.168.56.1:5000   │
   │                                                  │
   │   Copied local address to clipboard!             │
   │                                                  │
   └──────────────────────────────────────────────────┘
```

Figura 2: Application startup

4 Prediction Module: installation guide

To install the internal module you need to:

- download `dist` folder from the *Repository*

  https://github.com/Kero2375/proapes-predire-in-grafana/tree/master/Prediction%20Module

- import the plug-in on *Grafana* following this guide:

  https://grafana.com/docs/grafana/latest/plugins/installation/;

- if you are using *Linux* o *MacOS*, restart *Grafana* using the command:

  ```
  sudo systemctl restart grafana-server
  ```

- if you are using *Windows*, terminate the application and then reopen the executable;

5 `Training Module`: how to use

This section will provide useful information for the correct use of the Training Module.

5.1 Choosing the algorithm

The first step you have to do when using the Training Module is to select the training algorithm. Once an algorithm has been selected you can only change it thanks to the `Reset` button.
At the actual state of the product you can choose between four algorithms:

- *Linear Regression*$_{\mathbf{G}}$ (*RL*);
- *Exponential Regression*$_{\mathbf{G}}$ (*RegExp*);
- *Logarithmic Regression*$_{\mathbf{G}}$ (*RegLog*);
- *Support Vector Machine*$_{\mathbf{G}}$ (*SVM*);

5.2 File format and uploading

Once the desired algorithm has been selected, you can proceed to upload a *CSV*$_{\mathbf{G}}$ file containing the training data. The *CSV* file can be uploaded by clicking the `Choose file` button, near the label `Import CSV data`. After the file has been uploaded, its data will be visible in the graph. The correct file format is described in §5.7. Apart from the *CSV* file, also a *JSON*$_{\mathbf{G}}$ file, containing the training configuration option, can be uploaded; the file format is described in §5.7.2.

5.3 Options editing

The Training Module allows you to change the training configuration option used by the training algorithm itself; if you are not comfortable changing these parameters you can just use the default ones.
Each algorithm has different types of editable options, these are described in §5.8.

5.4 Start the training

You can start the training process by clicking on the `Train` button. When the training process is done, the hyperplane representing the obtained predictor$_{\mathbf{G}}$ will be visible. Thanks to the graph you can now compare the predictor to the raw data contained in the *CSV* file.

5.5 Download predictor

By clicking on `Download JSON` button, the *JSON* file containing the predictor that has been calculated in the previous step will be downloaded.

5.6 Reset

Thanks to the `Reset` button, it's possible to go back at the algorithm-choosing point. We suggest to use it with caution because this operation will delete from the page every data that has been uploaded or calculated.

5.7 File format

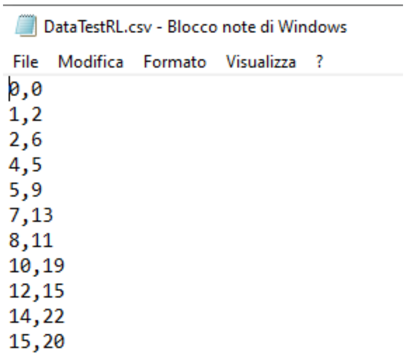
5.7.1 *CSV* file

The *CSV* file, as shown below, have to be formatted according to the chosen algorithm.

5.7.1.1 *Linear Regression*

Two values separated by a comma for each line.

e.g. `"Value 1,Value 2"`


DataTestRL.csv - Blocco note di Windows
File Modifica Formato Visualizza ?

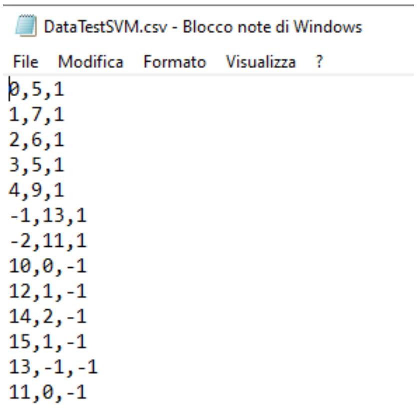
```
0,0
1,2
2,6
4,5
5,9
7,13
8,11
10,19
12,15
14,22
15,20
```

Figura 3: Sample of a *CSV* file used for *RL* training

5.7.1.2 *Support Vector Machine*

Two values and a class separated by a comma for each line.

e.g. `"Value 1,Value 2,Value 3"`

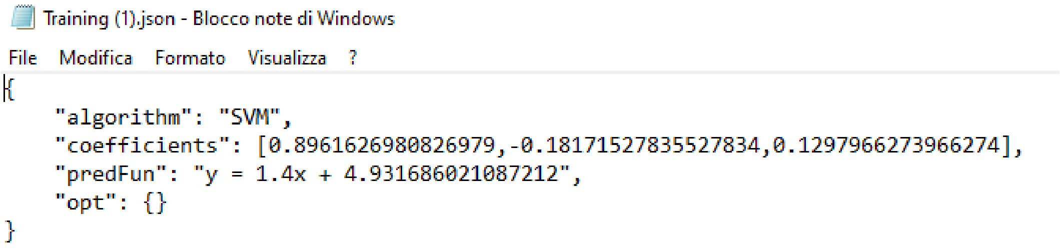

DataTestSVM.csv - Blocco note di Windows
File Modifica Formato Visualizza ?

```
0,5,1
1,7,1
2,6,1
3,5,1
4,9,1
-1,13,1
-2,11,1
10,0,-1
12,1,-1
14,2,-1
15,1,-1
13,-1,-1
11,0,-1
```

Figura 4: Sample of a *CSV* file used for *SVM* training

5.7.2 File *JSON*

The *JSON* file is the file you have downloaded from a previous training, which means that you should have done the regular training at least once. In the *JSON* file you can find also a value representing the training correctness (written as R^2 for regression algorithms, or as *F-measure* for classifications).


Training (1).json - Blocco note di Windows
File Modifica Formato Visualizza ?

```
{
    "algorithm": "SVM",
    "coefficients": [0.8961626980826979,-0.18171527835527834,0.1297966273966274],
    "predFun": "y = 1.4x + 4.931686021087212",
    "opt": {}
}
```

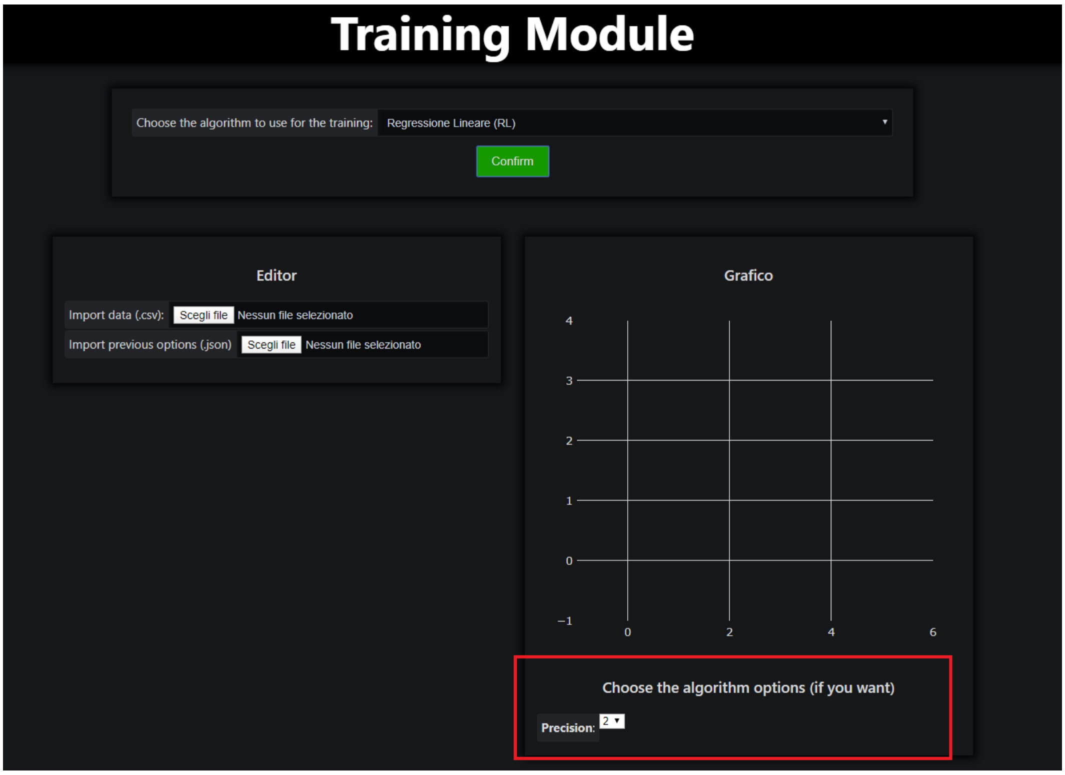
Figura 5: Sample of a *JSON* file obtained from a previous training

5.8 Training options

This section will describe the editable configuration options for each selectable algorithm.

5.8.1 *Linear, Exponential and Logarithmic Regression*

- **Precision**: a numerical value representing the precision of the known term and the angular coefficient in the equation.



Training Module

Choose the algorithm to use for the training: Regressione Lineare (RL)

Confirm
Editor
Import data (.csv): Scegli file Nessun file selezionato
Import previous options (.json) Scegli file Nessun file selezionato
Grafico
4
3
2
1
0
−1
0
2
4
6

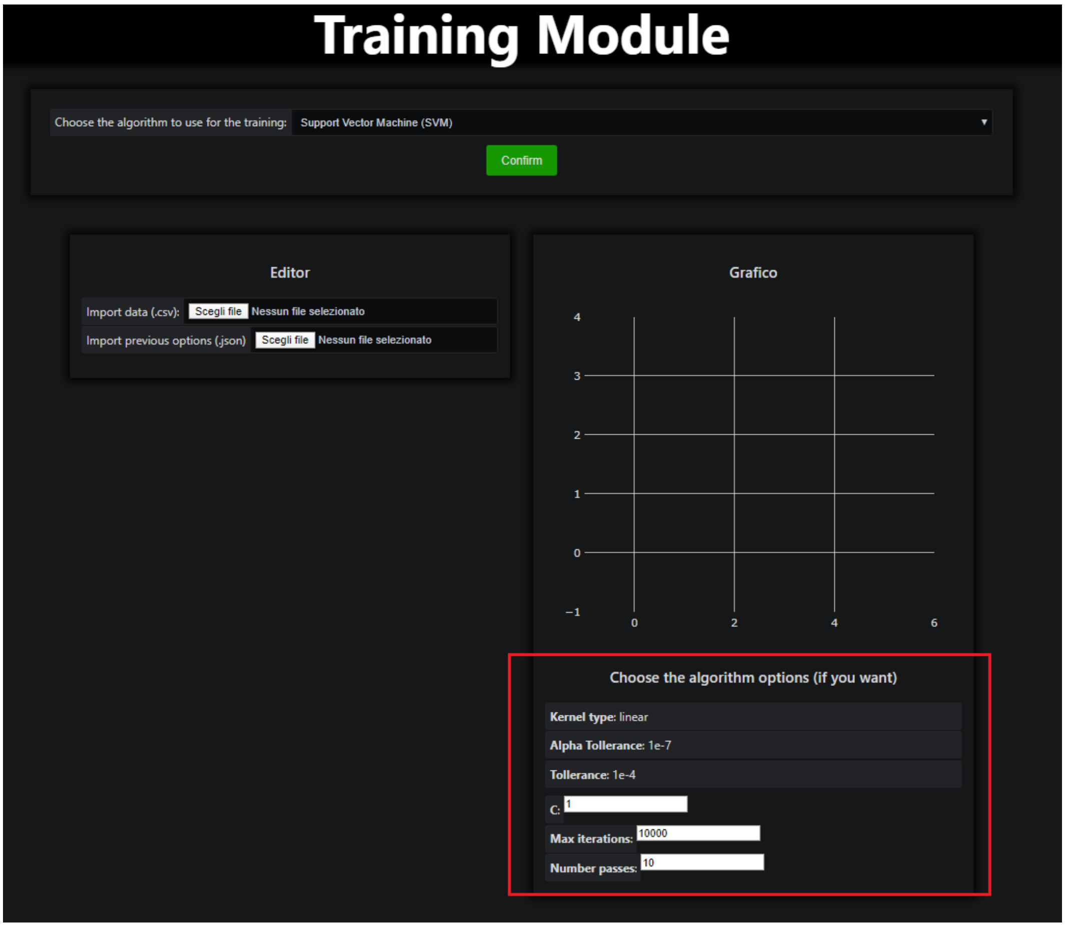
Choose the algorithm options (if you want)

Precision: 2

Figura 6: *Regression* options

5.8.2 *Support Vector Machine*

- **Max iteration**: the upper limit of the number of iteration the algorithm can do; it may be increased for large amounts of data;
- **C**: with a higher value you trust your data more, with a lower one ther will be more regularization. C values should be in range of around *1e-2* and *1e-5*;
- **Number of passes**: can be increased for an higher precision of the result but the computing will be slower.



Choose the algorithm to use for the training: Support Vector Machine (SVM)

Import data (.csv): Nessun file selezionato

Import previous options (.json) Nessun file selezionato

Choose the algorithm options (if you want)

Kernel type: linear

Alpha Tollerance: 1e-7

Tollerance: 1e-4

C: 1

Max iterations: 10000

Number passes: 10

Figura 7: *SVM* options

6 Prediction Module: how to use

This section will provide useful information for the correct use of the Prediction Module.

6.1 Enabling the application and creating a sample dashboard

The provided *plug-in* gives the user the opportunity to use a *Grafana* application as a starting point to the *plug-in* itself. The application, thanks to the latest changes, implements an internal `Training Module`. The internal `Training Module` works the same way the external one does, so you can follow the usage instruction at §5.
To enable the application you have to:

- open the `Settings` tab;
- click onto `Plugins` label;

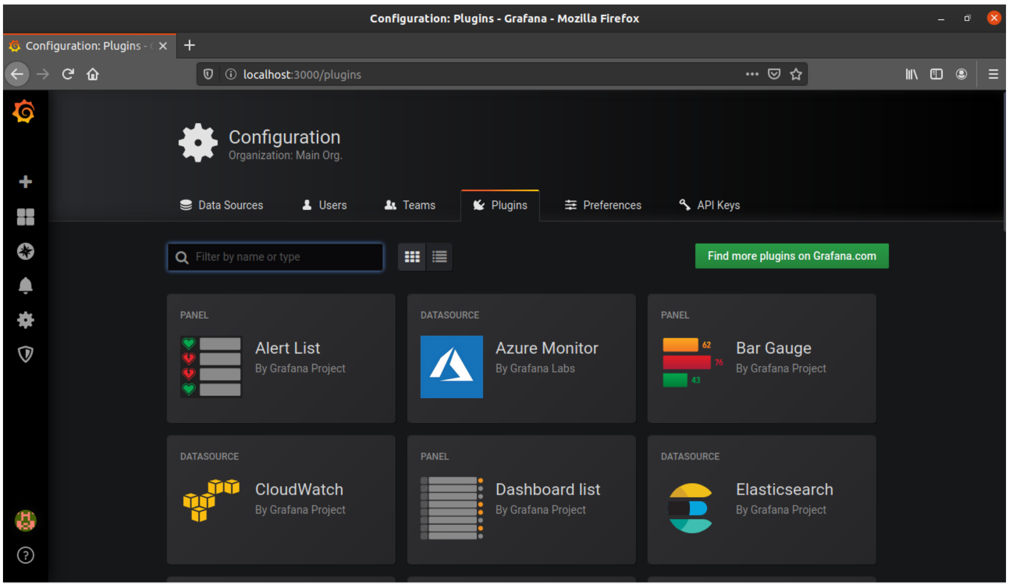


Figura 8: `Plugins` tab

- now search for `ProApes` in the search bar;

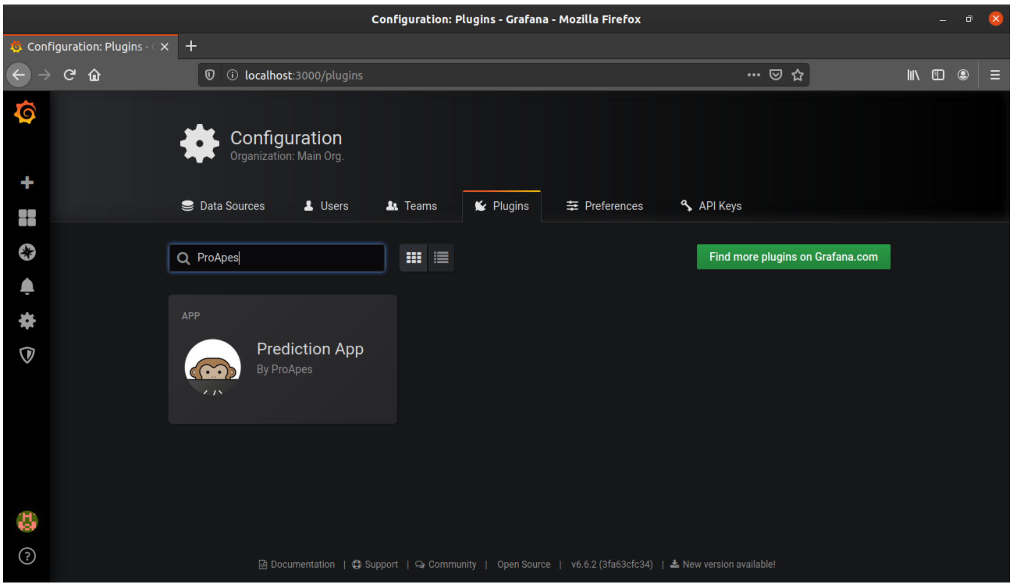

Configuration: Plugins - Grafana - Mozilla Firefox
Configuration: Plugins
localhost:3000/plugins
Configuration
Organization: Main Org.
Data Sources
Users
Teams
Plugins
Preferences
API Keys
ProApes
Find more plugins on Grafana.com
APP
Prediction App
By ProApes
Documentation | Support | Community | Open Source | v6.6.2 (3fa63cfc34) | New version available!

Figura 9: Result of the research in `Plugins` tab

- click on the search result (figure above);
- click on the `Enable` button in the `Config` tab;

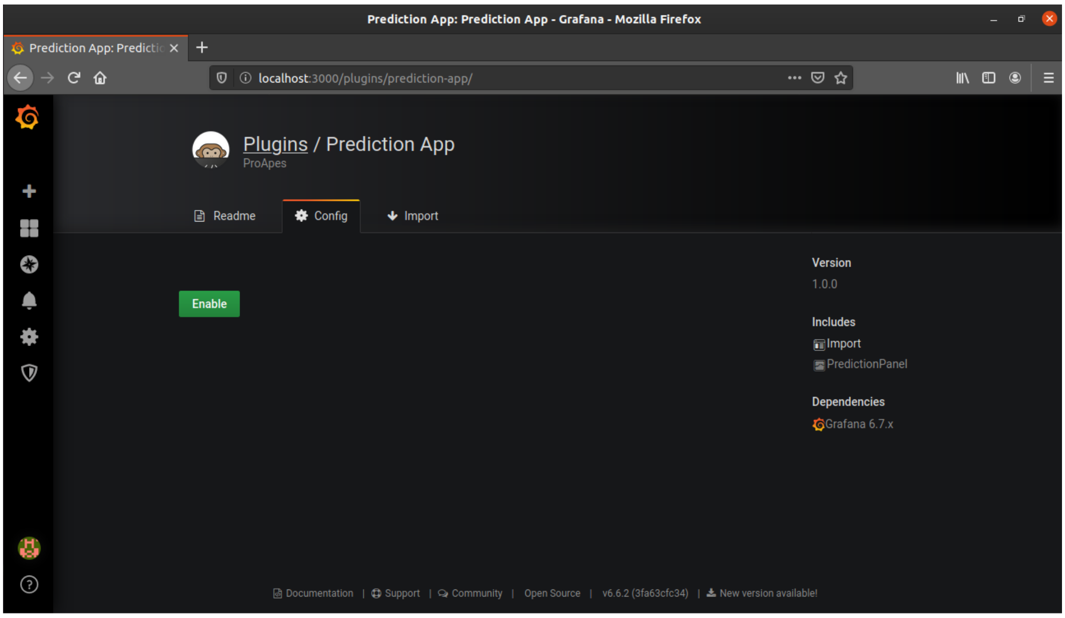

Prediction App: Prediction App - Grafana - Mozilla Firefox
Prediction App: Predictio
localhost:3000/plugins/prediction-app/
Plugins / Prediction App
ProApes
Readme
Config
Import
Enable
Version
1.0.0
Includes
Import
PredictionPanel
Dependencies
Grafana 6.7.x
Documentation | Support | Community | Open Source | v6.6.2 (3fa63cfc34) | New version available!

Figura 10: Enabling the application

To import the sample Dashboard$_G$:

- open the `Import` tab;
- click on `Sample Dashboard`.

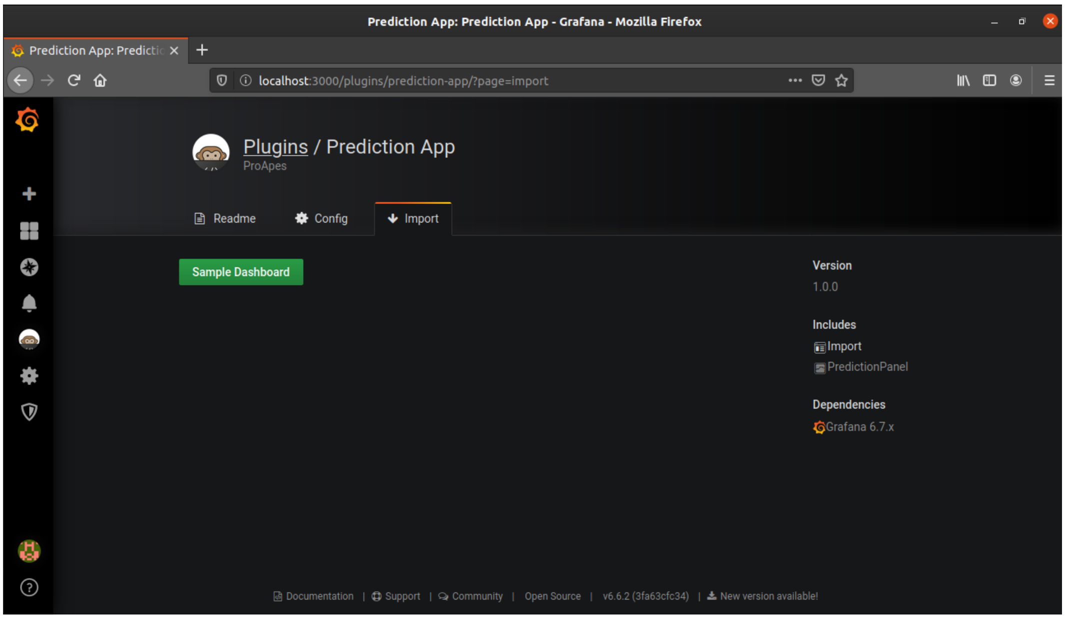
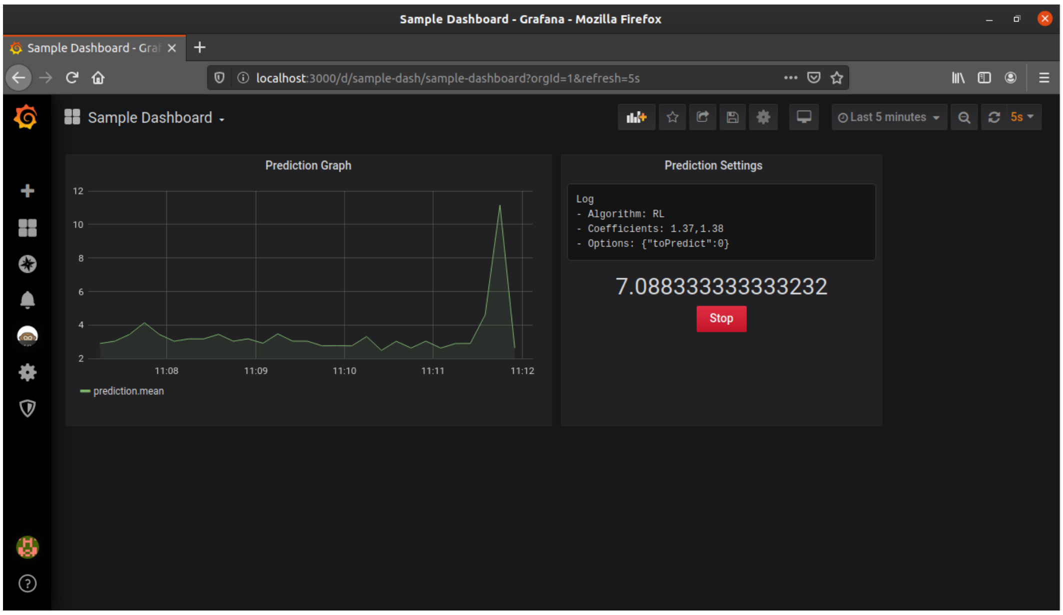

Prediction App: Prediction App - Grafana - Mozilla Firefox
Prediction App: Predictio
localhost:3000/plugins/prediction-app/?page=import
Plugins / Prediction App
ProApes
Readme
Config
Import
Sample Dashboard
Version
1.0.0
Includes
Import
PredictionPanel
Dependencies
Grafana 6.7.x
Documentation | Support | Community | Open Source | v6.6.2 (3fa63cfc34) | New version available!

Figura 11: Creating a sample dashboard

A sample dashboard has now been created. This Dashboard can be used as an entry point to create more complex dashboards, or even to test the panels you want to import in preexisting dashboards.

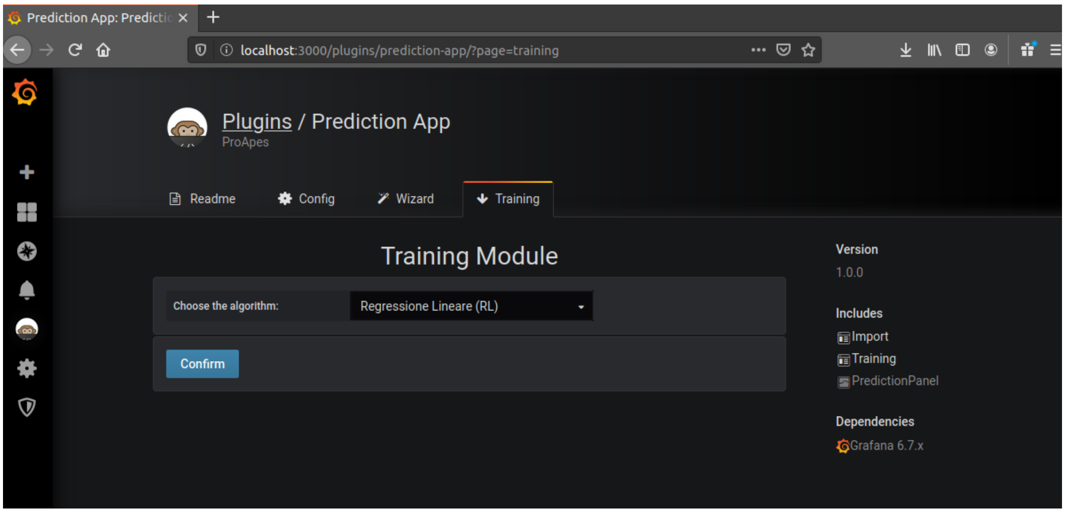


```
Log
- Algorithm: RL
- Coefficients: 1.37,1.38
- Options: {"toPredict":0}
```

7.0883333333333232

Figura 12: Sample dashboard

As written before, in the `Training` tab is possible to execute the training (as explained in §5) directly from the `Prediction Module`.



Choose the algorithm: Regressione Lineare (RL)

Figura 13: Internal training tab

6.2 Importing panels in a preexisting dashboard

If you want to import *Predire in Grafana*'s panel in a preexisting dashboard, you have to:

- click on `Add panel`

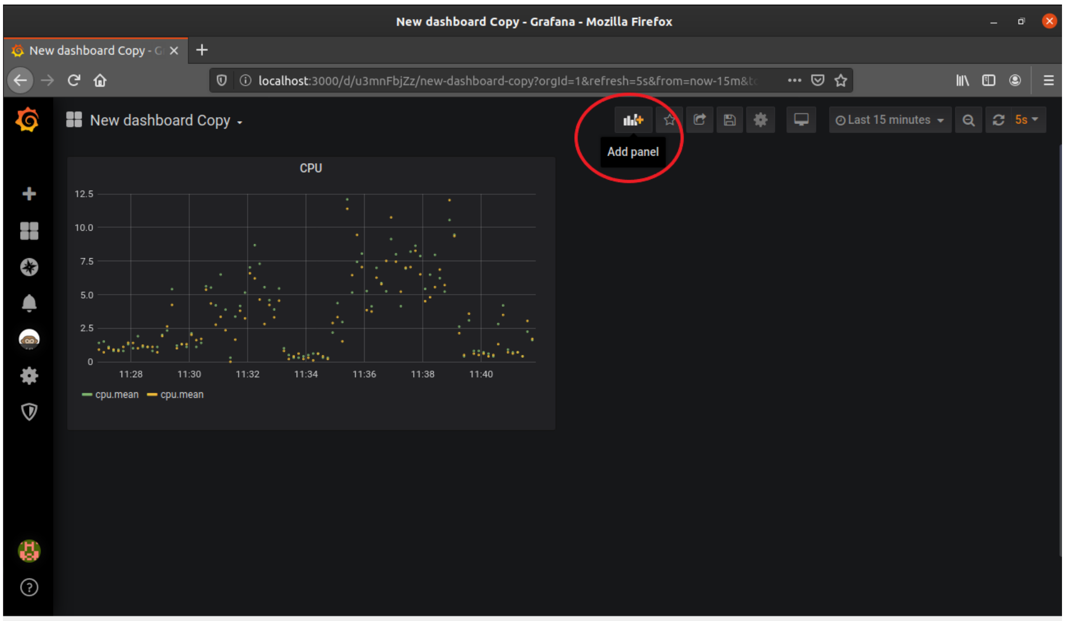

New dashboard Copy - Grafana - Mozilla Firefox
New dashboard Copy
localhost:3000/d/u3mnFbjZz/new-dashboard-copy?orgId=1&refresh=5s&from=now-15m&t
Add panel
Last 15 minutes
5s
CPU
12.5
10.0
7.5
5.0
2.5
0
11:28
11:30
11:32
11:34
11:36
11:38
11:40
cpu.mean
cpu.mean

Figura 14: `Add panel` button

- click on `Choose visualization`;
- select `Prediction Panel`;

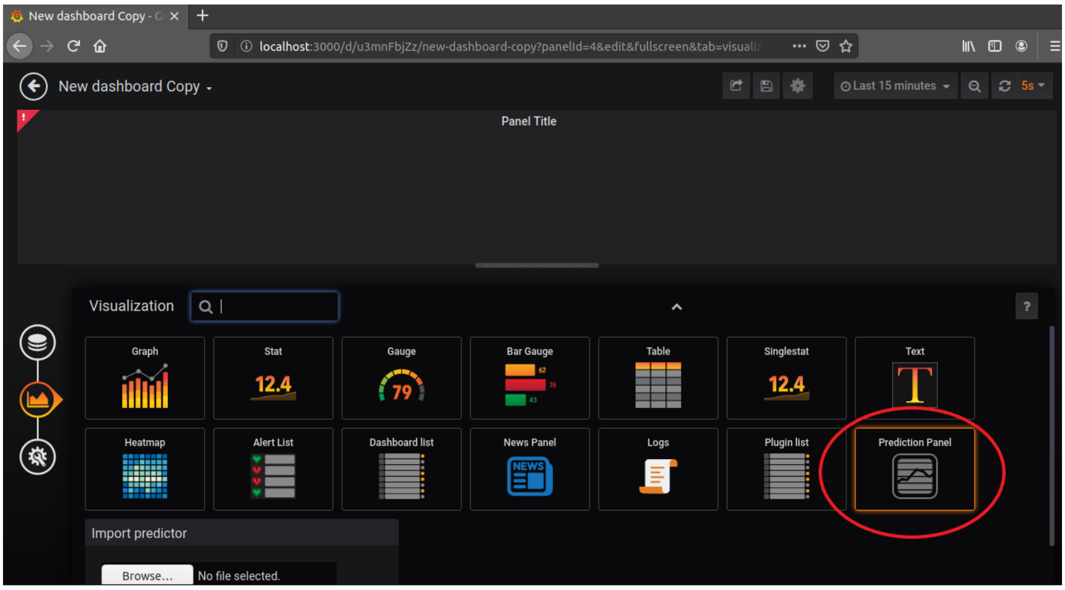
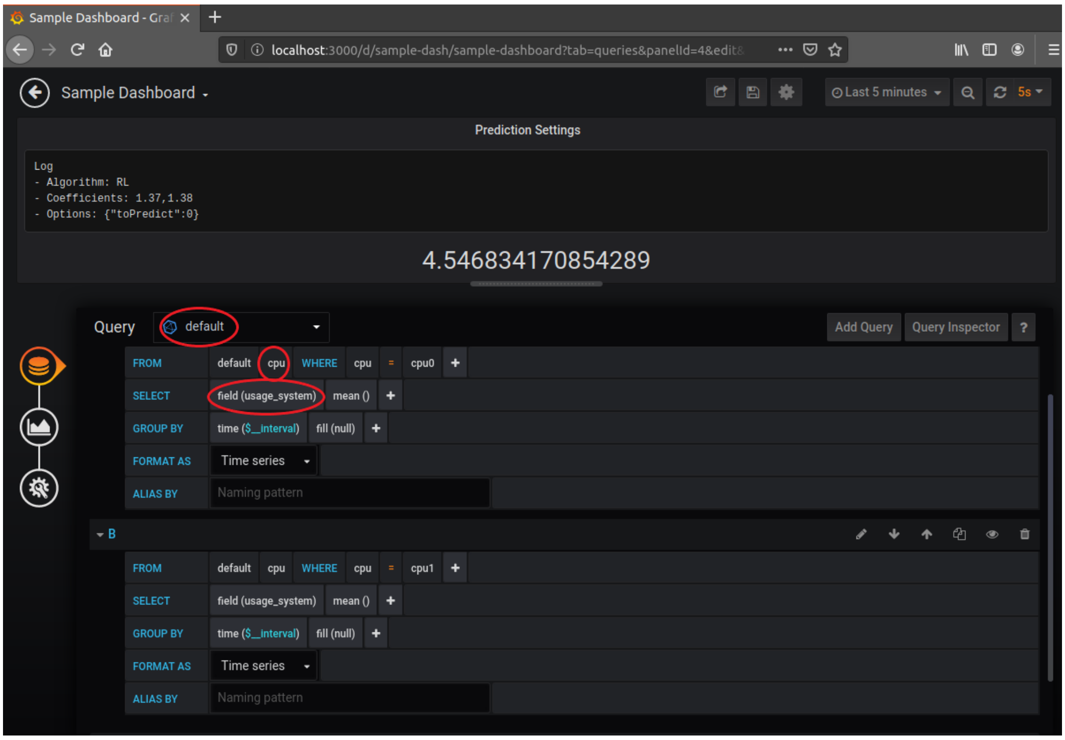


Figura 15: Selecting `Prediction Panel`

- import the predictor, as described in §6.3: the panel can start predicting values.

6.3 Use of the *Prediction Panel*

To use the `Prediction Panel` it's mandatory to define at least two queries from the editor's query section. These queries will get the data you want to predict. In the query editor, you have to select the *data source* (among those previously set in *Grafana*), the measure type and the desired value.



```
Log
- Algorithm: RL
- Coefficients: 1.37,1.38
- Options: {"toPredict":0}
```

4.546834170854289

Query: default

FROM	default cpu WHERE cpu = cpu0
SELECT	field (usage_system) mean ()
GROUP BY	time ($__interval) fill (null)
FORMAT AS	Time series
ALIAS BY	Naming pattern

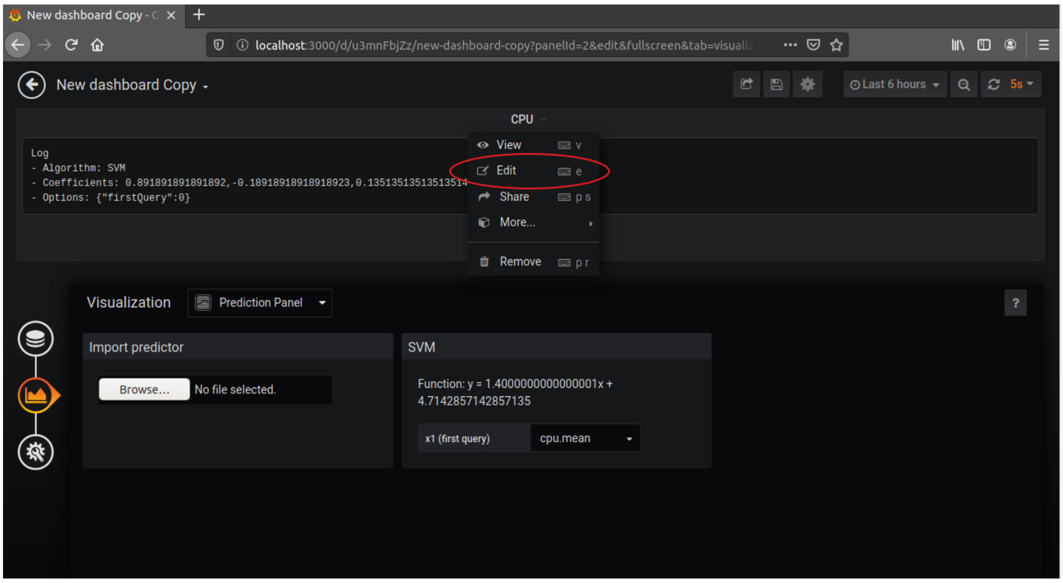
B

FROM	default cpu WHERE cpu = cpu1
SELECT	field (usage_system) mean ()
GROUP BY	time ($__interval) fill (null)
FORMAT AS	Time series
ALIAS BY	Naming pattern

Figura 16: Example of a query

Thanks to the `Prediction Panel` you can:

- import a new predictor using the `Import predictor` function, which you can find on the `Visualization` section. You can get to the `Visualization` section through the panel edit menu;



```
Log
- Algorithm: SVM
- Coefficients: 0.891891891891892,-0.18918918918918923,0.1351351351351351
- Options: {"firstQuery":0}
```

Function: y = 1.4000000000000001x + 4.7142857142857135

x1 (first query): cpu.mean

Figura 17: Panel `Edit` button and import of a new predictor.

- pause the predictione thanks to the `Stop` button in the panel view;
- view settings of the imported predictor;
- view the prediction value of whatever you are monitoring.
- create and configure specific alerts for predictions. Alerts can be created thanks to the grafic interface (by pressing on the threshold value above which you want the alert to appear, a horizontal red line representing the threshold will appear) or through the dedicated section of *Grafana*; in the second scenario the guided configuration can be used.

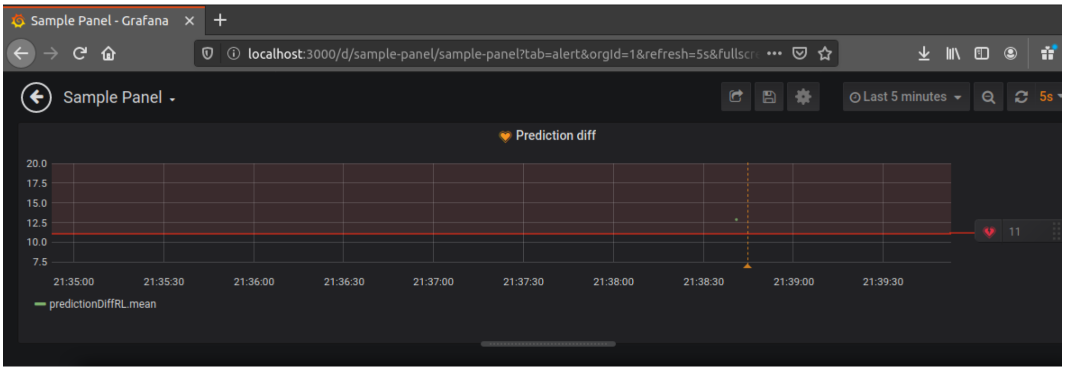
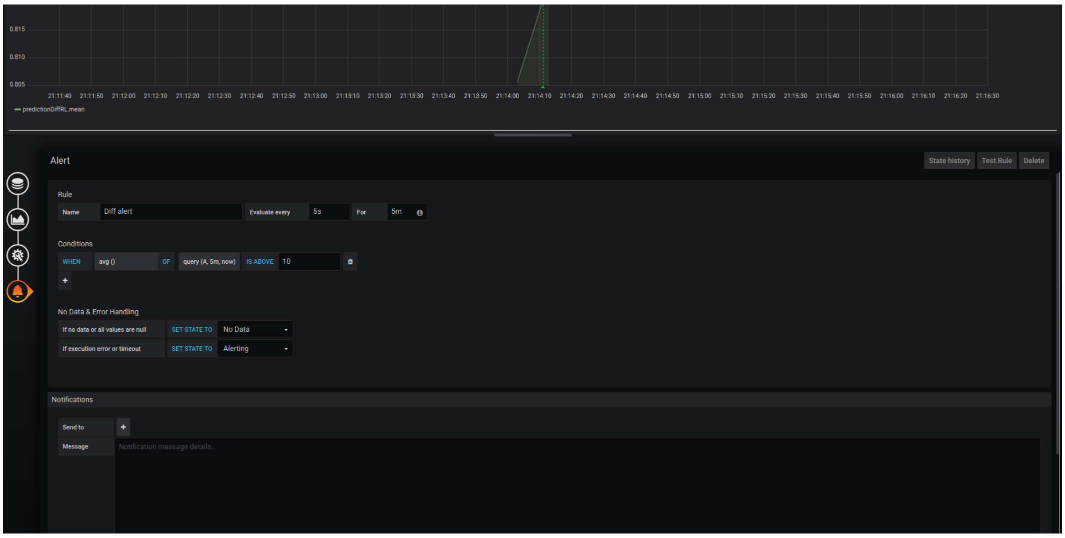


Figura 18: Manual alert configuration



Alert

Rule

Name: Diff alert
Evaluate every: 5s
For: 5m

Conditions

WHEN avg () OF query (A, 5m, now) IS ABOVE 10

No Data & Error Handling

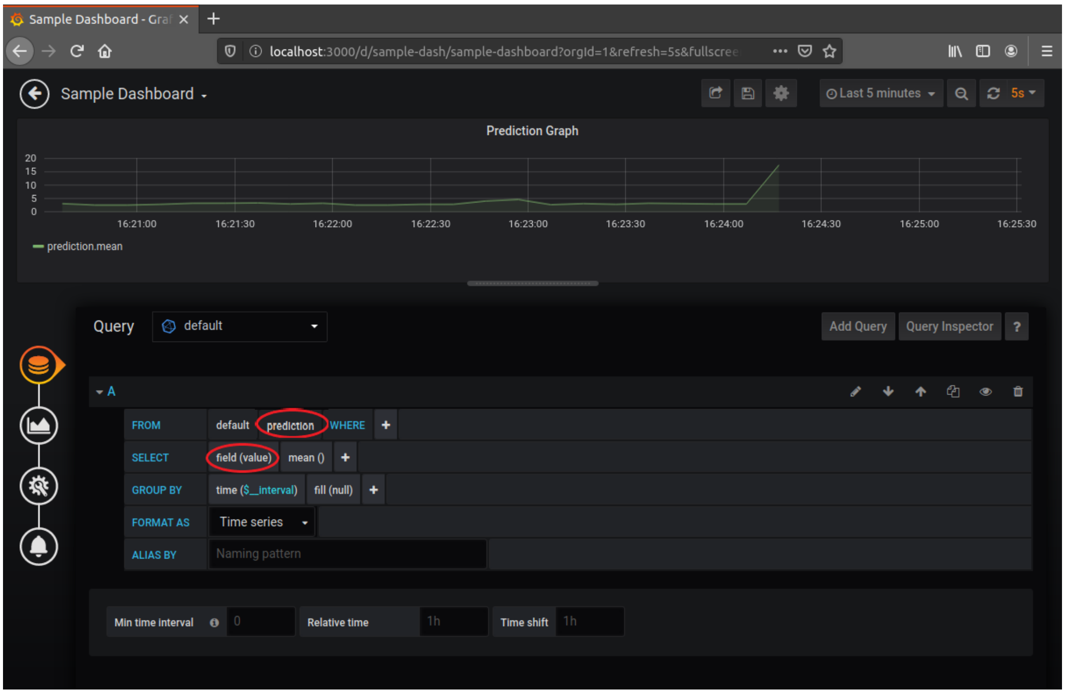
If no data or all values are null SET STATE TO No Data
If execution error or timeout SET STATE TO Alerting

Notifications

Send to
Message: Notification message details...

Figura 19: Guided alert configuration

To compare how much the predicton is different from the real value, or simply to view the predicted values, you can use the visualization format you desire instead of the `Prediction Panel` defined in §6.3. For example you could use a `Graph`, `Stat` and so on.
If you want to use other visualization formats you just need to set the query on the predicted value by selecting the `value` field from the `Prediction` tab.



FROM default prediction WHERE +
SELECT field (value) mean () +
GROUP BY time ($__interval) fill (null) +
FORMAT AS Time series
ALIAS BY Naming pattern

Min time interval 0 Relative time 1h Time shift 1h

Figura 20: Example of a query

N.B. After setting a query you need to refresh the page to make it work.

A Glossary

C

CSV: *comma-separated values* is a text-based file format used to import and to export tables of data.

D

Dashboard: group of graphical objects used to make visible a bunch of information of different nature at complexity in real-time.

DevOps: set of practices that combines *software development (Dev)* and *information-technology operations (Ops)* which aims to shorten the system development life cycle and provides continuous delivery for high software quality.

E

Exponential Regression: a *Linear Regression*$_{\mathbf{G}}$ variation used when the given data can't be represented as a straight line; with this algorithm data are transformed by mathematical operations so that they can be represented as an exponential function.

G

Grafana: system used by *Zucchetti S.p.A.* to monitor applications; the monitor functionality becomes even more important in a *DevOps*$_{\mathbf{G}}$ scenario and it has three jobs:

- controlling the system health status and verifying that performances agree with expectation;
- identify weaknesses that must be fixed by the developers;
- supply with elements used to define the scale of priorities for improvements and new implementations.

The monitoring scope is quite broad; if any extreme situation is detected, alerts (e-mails) are sent to accountable.

J

JavaScript: object and event oriented scripting language, used both in client-side and server-side web programming.

JSON: (JavaScript Object Notation) is a data exchange file format: it's easy to read for people, while for machines it's easy to generate and analyze its syntax. It's based on a *JavaScript*$_{\mathbf{G}}$ subset, Standard ECMA-262 third edition - December 1999.

L

Logarithmic Regression: a *Linear Regression*$_{\mathbf{G}}$ variation used when the given data can't be represented as a straight line. With this algorithm data are transformed by mathematical operations in a way they can be represented as a logarithmic function.

Linear Regression: abbreviated as *RL*, it's a numerical prevision technique that uses the Ordinary Least Squares method. *Linear regrassion* approximates the solution of overdetermined systems by minimizing the sum of the squares of the residuals made in the results of every single equation.

N

NodeJS: an *open-source*$_{\boldsymbol{G}}$, multiplatform, event oriented *JavaScript*$_{\boldsymbol{G}}$ runtime, used to execute *JavaScript*$_{\boldsymbol{G}}$ code.

P

Plug-in: non-autonomus program that interacts with another program to extend its functionalities.

Predictor: statistics (function of data) defined to make previsions on one or more variables.

S

Support Vector Machines: abbreviated as *SVM*, a classification algorithm used to resolve the "curse of dimensions" which arises when dealing with large amounts of data. In this context, it's noticeable a thinning of data when considering more dimensions. *SVM* algorithm goal is to find the hyperplane that divides in the best way data in two classes; this way, even while considering a lot of dimensions the classification will be good.

T

Tender specifications: is a technical document, generally attached to a contract, in which technical specifications of the work that will be carried out as a product are defined.

Y

Yarn: *NodeJS* package manager, compatible with *npm* package registry.