Arauto Documentation

Release 0.1.0

Paulo Vasconcellos

May 30, 2020

Contents

ARAUTO

Arauto is an open-source and interactive tool for quick prototyping and experimentation of time series models. You can use it to **build mixed autoregressive moving average models** (AR, MA, ARMA, ARIMA, SARIMA, ARIMAX, SARIMAX).

Arauto offers an intuitive experience, so you can focus on the results of your model. Among other things, it supports exogenous variables and let you customize the whole process, from choosing a specific transformation function to test different parameters. Check it out the main features of Arauto:

- **Support to exogenous regressors** (independent variables);
- Seasonal decompose that let's you know the **Trend, Seasonality and Resid** of your data;
- Stationarity Test using **Augmented Dickey-Fuller** test;
- Customization of data transforming for stationarity: you can use from first difference to seasonal log to transform your data;
- **ACF** (Autocorrelation function) and **PACF** (Parcial correlation function) for terms estimation;
- Customize ARIMA terms or **let Arauto choose the best for you** based on your data;
- **Grid search** feature for parameters tuning;
- Code generation: at the end of the process, Arauto returns the code used to perform each step.

CHAPTER 1

Contents

1.1 Installation

Arauto can be used in three different ways:

1.1.1 Web

If you are just curious about what you can do with Arauto, you can refer to this website.
This version contains some example datasets that you can use to check how Arauto works.

Please note that this version is a Heroku's free tier instance. Due to high traffic, you may experience some poor performance

1.1.2 Docker

Run the following commands to use Arauto with Docker (requires **Docker and Docker-compose**)

```
# Run the docker compose
docker-compose up --build
```

1.1.3 Local installation

Tip: we recommend you to use Anaconda environments

```
# Clone the repository
git clone https://github.com/paulozip/arauto.git
cd arauto
```

(continues on next page)

(continued from previous page)

```
# If you're using Anaconda
conda create --name arauto_env
conda activate arauto_env

# Install dependencies
pip install requirements.txt

# Run Streamlit
streamlit run run.py
```

1.2 How to use Arauto

Arauto was built to be as intuitive as possible. It offers an interface that simulates the line of thinking of analysing and training a model for forecasting. This tutorial will guide you through all the process, starting by the top menus.

1.2.1 Your data menu

This is where you will pick a file for Arauto to analyse and model. You can play with some toy datasets or you can upload your own dataset using the REST API. **You will also select the frequency of your data**. If your data was collected in a daily basis, select **Daily**, if it was collected in a monthly basis, select **Monthly**. This is an important step since **Arauto will use this field to understand the seasonality of the time series**, and apply different techniques to find the best model.

Fields

- **Select a file**: an CSV, TXT, Excel, or delimited file.
- **What is the FREQUENCY of your data?**: the frequency that the dataset was collected. Null values will be replaced by 0.

1.2.2 Choosing columns menu

This is step is composed by 4 fields that will instruct Arauto which columns to use when training the model. In this step you can select **exogenous variables, what are nothing but columns that will help the model to give different ways for your predictions**. For example:

My Daily Sales		
Date	Sales	is_christmas_period
2019-11-29	500	0
2019-11-30	470	0
2019-12-01	200	1
2019-12-02	150	1

Let's say that you are predicting your company's daily sales, and in the Christmas period the sales go down. You can use a column called `is_christmas_period`, where you place `1` for rows (days) on your dataset that was collected in christmas period. Arauto will try to use columns like this to enhance the predictions.

Fields

- **Which one is your DATE column?**: the column used to identify the time point where the data was collected.
- **Which column you want to PREDICT?**: the column with the values that you are trying to predict. It can be data like total sales, temperature measured, stock prices, and so on.
- **Which are your exogenous variables?**: the independent variables (e.g.: `is_christmas_period`) that might be important to give different weights to predictions. **By informing exogenous variables will use automatically models that support regressors, e.g. ARIMAX.**
- **Validation set size**: how many period you want to let for Arauto validate the model? **It's recommended to configure this field with the same amount of periods that you want to forecast**. For example, if your data was collected in a monthly basis, and you want to forecast the next 3 months, let this field with value 3.

1.2.3 Charts menu

In this section you can select some charts to appear on the screen. These charts are helpful to understand some steps of the model training, like the distribution data, seasonal decompose and out-of-sample predictions.

Fields

- **Historical data**: show the absolute distribution of your dataset just like it is. It can be useful to understand your data over time, and identify some interesting points like missing values of unusual scales.
- **Seasonal decompose**: show the components of your time series. It returns informations like the time series trending, seasonality, and resid (what remains when you remove trending and seasonality).
- **Dickey-Fuller statistical test**: to understand if your time series is stationary (one of the properties that make it possible to forecast data), Arauto will execute the Augmented Dickey-Fuller test (a.k.a ADF test). By enabling this option, the transformed data with the best ADF test result (based on the lowest statistical result) will be plotted on Arauto, with its moving average and standard deviation.
- **Train set predictions**: enable this option if you wanna check how your model is predicting the data that it was trained with. Two plots are placed in the figure, one for the observed (real) data (labeled as `y`), and the predicted data (labeled as `ŷ`).
- **Test set predictions**: enable this option if you wanna check the out-of-sample predictions of your model in comparison with unseen data (test set). Two plots are placed in the figure, one for the observed (real) data (labeled as `y`), and the predicted data (labeled as `ŷ`).

1.2.4 Force data transformation menu

For non-stationary time series (that is, for series that doesn't have a constant trend and variance over time), Arauto will find the best transformation technique that will make it stationary, hence, making it possible to model and predict the data. By default, Arauto will iterate over all the transformation techniques in order to find the best one, but you can force it to used one of your favorite transformations.

Using the dropdown menu, **you can even select to not use any transformation technique on your data**. After that, Arauto will execute the Augmented Dickey-Fuller test to check for stationarity. **If the test statistics are not significant, Arauto will show a warning**. You can continue the process of modeling a time series even with a non-relevant statistical test result, but **the model performance might be badly influenced if you select a weak transformation**.

Fields There's just one field on this menu, that is the Transformation technique, which contains the following functions:

- **Choose the best one**: this is the default parameter. **Arauto will iterate over all the transformation techniques and select the best one**, based on the lowest statistical relevant Augmented Dickey-Fuller test result.
- **No transformation**: **No one transformation will be applied on the time series**. The ADF test will be executed on the absolute series, without transformations.
- **First Difference**: a First Difference will be applied to the time series. **This transformation is made by substraction the current observation from the previou observation**. For instance, if you have a series collected in a daily basis, the First Difference of today is equal to: (today_value - yesterday_value). **Most of the time series data uses this techniques, which make it stationary**. Arauto will always execute this technique first.
- **Log Transformation**: each observation of the series will be log transformed. This is useful when we need to penalize higher values on the time series (which is common when we have outliers in our data). The data is transformed using Numpy's Log1p function.
- **Seasonal Difference**: if your data contains seasonality, a seasonal difference will be applied on your series. **It works similar to First Difference**, but instead of substracting the current observation (t) by the previous observation (t-1), Arauto will substract it by the (t-s), where s is the seasonal frequency. For instance, if your data was collected in a monthly basis, and it has a yearly seasonality, the Seasonal Difference will be: (t - t-12).
- **Log First Difference**: this is the first combined transformation that you will see in the dropdown menu. If selected, Arauto will transform your data using Numpy's Log1p function and, **after that, will execute a First Difference transformation (t - t-1)**.
- **Log Difference + Seasonal Difference**: similar to Log First Difference, Arauto will execute a log transformation in your time series, followed by a First Difference transformation (t - t-1) and a Seasonal Difference (t - t-s), where s is the seasonal frequency.
- **Custom Difference**: you can even select a custom difference technique for your data. This option will enable two other parameters: `Difference size` and `Seasonal Difference size`.

  > **IMPORTANT**: it's NOT recommended to use more than 1 Seasonal Difference, or more than 2 Differences combined. In other words, **your total difference should not pass 2 (seasonal + non-seasonal)**.

1.2.5 Model parameters menu

To estimate the right amount of terms to consider for fitting the model, Arauto will look to the ACF and PACF functions to estimate the terms for AR (p), I (d), MA (q), Seasonal AR (P), Seasonal Difference (D), Seasonal MA (Q). By default, Arauto will set a recommended amount of terms for each part, but you can freely select different amounts of terms. Please, refer to How to choose the parameters for the model in order to learn how to select these parameters.

1.2.6 Forecast periods menu

This is the last step of Arauto. Here, you can select how much future periods (days, months, years) you want to forecast.

Fields

- **How many periods to forecast?**: how much period should Arauto forecast?
- **Find the best parameters for me**: if selected, Arauto will execute a grid search process to find the best amount of terms for, p, d, q, and so on. **This is a high computational process, since Arauto will iterate of different amounts of parameters to fit the best model. Be sure your server has enough memory for this process**.

- **Do your Magic!**: once you have all set up, click this button to train your model.

1.2.7 What happens next?

Once your model was trained, **you can check your forecast at the end of you screen**. The out-of-sample forecasts are displayed on the screen using Plotly, you can interate with the chart and export it as a PNG image.

Besides that, Arauto will give the code used to transform the data, generate the charts, train the model, and execute the forecasting. You can copy this code and use it wherever you want.

1.3 How to upload your dataset

A Upload file support will be added to Arauto, but you can use the Arauto REST API to send your dataset. Here's an example of how you can use it using cURL:

```
curl -X POST \
  http://SERVER_ADDRESS:5000/upload_file \
  -H 'content-type: multipart/form-data' \
  -F file=@PATH_TO_YOUR_FILE
```

Example

```
curl -X POST \
  http://0.0.0.0:5000/upload_file \
  -H 'content-type: multipart/form-data' \
  -F file=@/home/my_user/Downloads/dataset.csv
```

1.4 How to choose the parameters for the model

Something it might be dificult to estimate the amount of terms that your model needs, chiefly when it comes to ARIMA. In this part, you be shown to some types of analysis that you can do to estimate the parameters of your model.

Important: by default, Arauto will try to find the best parameters for ARIMA or SARIMA for you. The recommended values will be shown below the ACF and PACF plots, but you also can explore different parameters.

1.4.1 The ACF and PACF function

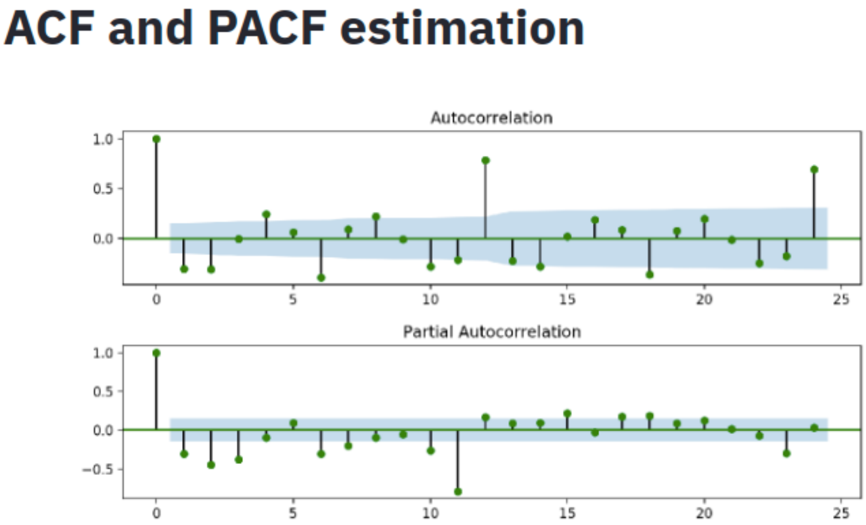
One good and intuitive approach to estimate the terms for seasonal and non-seasonal autoregressive models is to look at autocorrelation function and partial autocorrelation function. We are not going too deep in theorical concepts around these functions, but there are great resources around the web (see References section below). Basically, **the autocorrelation function will show the relationship between a point in time and lagged values**.

For instance, imagine that we have a non-null time series collected in a daily basis. An autocorrelation of lag 1 will measure the relationship between the today's value (Yt) and yesterday's value (Yt-1). The values of the autocorrelation function and partial autocorrelation function can help us to estimate AR (p) terms, and MA (q) terms, respectively.

To estimate the correct amount of terms, **we must use a stationary series, which is basically a time series where there is a constant mean and variance over time**. Arauto provides some resources to make a time series stationary, like log transformations, first differences, and so on (you may want to check the How to Use Arauto section to know more about transformation functions). Also, **Arauto automatically generate plots for autocorrelation function (ACF) and partial autocorrelation function (PACF)**, making it easier to interpret and identify AR and MA terms.

Example

Let's use an example to understand more about ACF and PACF. Here is the plots for the Monthly Wine Sales dataset on Arauto, which is stationary after a log difference transformation.



1.4.2 Estimating AR terms

The lollipop plot that you see above is the ACF and PACF results. **To estimate the amount of AR terms, you need to look at the PACF plot**. First, ignore the value at lag 0. It will always show a perfect correlation, since we are estimating the correlation between today's value with itself. Note that there is a blue area in the plot, representing the confidence interval. To estimate how much AR terms you should use, **start counting how many "lollipop" are above or below the confidence interval before the next one enter the blue area**.

So, **looking at the PACF plot above, we can estimate to use 3 AR terms for our model**, since lag 1, 2 and 3 are out of the confidence interval, and lag 4 is in the blue area.

1.4.3 Estimating I terms

This is an easy part. **All you need to do to estimate the amount of I (d) terms is to know how many Differencing was used to make the series stationary**. For example, **if you used log difference or first difference to transform a time series, the amount of I terms will be 1**, since Arauto takes the difference between the actual value (e.g. today's value) and 1 previous value (e.g. yesterday's value).

1.4.4 Estimating MA terms

Just like the PACF function, **to estimate the amount of MA terms, this time you will look at ACF plot**. The same logic is applied here: how much lollipops are above or below the confidence interval before the next lollipop enters the blue area?

In our example, **we can estimate 2 MA terms, since we have lag 1 and 2 out of the confidence interval**.

1.4.5 Estimating Seasonal AR terms

If your data has seasonality and you want to use a Seasonal ARIMA model, you need to inform the seasonal terms for AR, I, and MA. **The process is quite similar to non-seasonal AR, and you will still using the ACF and PACF function for that**. To estimate the amount of AR terms, you will look one more time to the PACF function. **Now, instead of count how many lollipops are out of the confidence interval, you will count how many seasonal lollipops are out**.

For example, if your data was collected in a monthly basis and you have yearly seasonality, you need to check if the "lollipop" at lag 12 is out of the confidence interval area. **In case of positive result, you need to add 1 term for Seasonal AR. In the plot above, we can see that the value at lag 12 is out of the blue area of the confidence interval**, so we will add 1 terms for seasonal AR (SAR).

1.4.6 Estimating Seasonal I terms

The same logic of estimating non-seasonal differencing is applied here. If you used seasonal differencing to make the time series stationary (e.g. the actual value (Yt) substracted by 12 previous month (Yt-12)), you will add 1 term to seasonal differencing. In our example, **we just used log differencing to make the time series stationary**, we do not used seasonal differencing as well. So, **we will not add 1 terms for seasonal differencing**.

1.4.7 Estimating Seasonal MA terms

For seasonal moving average (SMA), we will be looking at the ACF plot and use the same logic of estimating SAR terms. For our example, **we will be using not 1, but 2 terms for SMA. Why? Because we have significant correlation at lag 12 and lag 24**.

1.4.8 Final considerations

At the end of this process, you will have all the terms needed to build your model. Below the ACF and PACF plot, Arauto will recommend the same amount of terms that we identified in this tutorial for p, d, q, P, D, and Q: (3, 1, 2)x(1, 0, 2). If you want to let Arauto optimize these parameters, you can select the option **"Find the best parameters for me"** and Arauto will apply Grid Search to your model. Keep in mind that this is high computational step, be sure that you have enough resources for this process.

1.5 Need any help?

Hey, I will be happy to help you on whatever you need. Please, refer to my contact infos below to ask for help.

1.5.1 Contact info

- E-mail: phsamus@gmail.com
- Twitter: https://twitter.com/paulo_zip
- Linkedin: http://linkedin.com.br/in/paulovasconcellos

1.5.2 How to contribute

Currently, this project is maintained by just one person. It would be great to have more people collaborating and contributing with this open-source project. If you want to collaborate with Arauto, here's what you can do:

- **Documentation and tutorial**: it's really important to make users get into Arauto as soon as they open the browser. Tutorials and deeper documentation could help us to achieve it.
- **New algorithms**: there are many different algorithms that we could add to Arauto, like ARCH, VAR, and Tree-based algorithms, to name a few. All you need to do is fork the repository, build the feature and open a PR to merge it.
- **Bug fixes**: something might be broken. We need good people to fix it.
- **Tests**: Arauto doesn't contain tests, and this is wrong.