

# GuilhermePires [ML Engineer]

## find me

mail@gpir.es  
colobas.github.io  
github.com/colobas  
linkedin.com/in/guilhermegrijopires

## skills & tools

**very confident:** python, pandas, sklearn, numpy, bash, linux  
**confident:** pytorch, kubernetes, azure, aws, golang, c, dask, docker  
**beginner:** scala, nlp techniques, mesos, rabbitmq, gcp

## technical interests

ml theory, probabilistic modelling, variational inference, optimization

## personal interests

technology, hackathons, climate change, economy, philosophy, the meaning of life, music, running

## education

**MSc degree** | IST - Lisboa  
Electrical and Computer Engineering

Since Sep 2012

**ERASMUS - exchange program** | KIT - Karlsruhe  
Computer Science

Oct - Feb 2015/16

## experience

**Jungle** | ML Engineer

Since Sep 2017

Since Sep-2017 I'm a full time **ML Engineer**. I've worked with diverse **multivariate time-series** datasets, with data sources spanning from **heavy industry**, to **utilities** and **renewables**. Keywords: machine learning, time-series, root cause analysis, forecasting, probabilistic modelling

**Jungle** | Data Engineering Intern

Jul - Sep 2017

I started at Jungle as a **Data Engineering** intern. During my internship I worked on assembling an **infrastructure** to enable Machine Learning at scale. To do so, I looked into and gained experience with technologies such as **Mesos**, **Marathon**, **Kubernetes**, **Dask**, **Docker**. I also built a PoC of a job scheduler, see here

**Snowplow** | Data Engineering Intern

Jan - Mar 2017

I was a remote "wintern" at Snowplow for about 3 months in 2017. I helped them extend some of their tech to **Google Cloud Platform**, having gained introductory experience with **Scala**, and GCP along the way. See the resulting blogpost here

**Quidgest** | SW Engineering Intern

Aug - Oct 2013

My first job in software was a summer internship at Quidgest. I spent my time there review code and refactoring an internal tool, at the R&D department. I worked with C, C#, C++, SQL

## projects

**MSc thesis: Deep Generative Models for Multivariate Time-series** | Python

I'm working with unsupervised learning for multivariate time-series. I'm specially interested in segmenting high dimensional time-series into "regimes of operation", similar to the goal of changepoint detection.

### **PoC of a job scheduler** | Python

During my summer internship at Jungle, I built a PoC of a job scheduler, using RabbitMQ for task queues, and Redis for logging. The goal was to have the concept of Recipes - which would be descriptions of tasks. Basically, to create a Recipe you'd implement a subclass of the AbstractRecipe class. The worker spawners are also abstract, to allow workers of different natures. I only implemented a CPUSpawner which creates workers in different processes.

### **Scraper for politifact.com claim checks** | Python

I was working on an automated claim checking project, and needed a dataset, so I built a scraper for politifact.com

### **Betfair scraper and automated betting bot** | Python

Some friends came up with a soccer betting strategy, based on metrics calculated from live statistics of soccer matches. I scraped Betfair and some other websites, and built a bot that computed the metrics and placed bets according to the strategy.

### **Dynamic Bayesian Network learning** | Java

Implemented a program to learn DBNs from data. See project assignment [here](#)

### **SATPLAN system** | Python

Implemented a SATPLAN system for an AI course. It worked by translating a planning problem into a SAT problem, solving the SAT problem, and translating the solution into the planning semantics. See [here](#) [here](#)

### **PoC of a distributed "e-mail" system** | Go

Implemented a distributed system for nodes to send private messages. It's a combination of a distributed hashtable, to store messages in the network, and an onion routing circuit, so that the node who stores the message is not the node who wants to send it. See [here](#)

### **PoC of a distributed hashtable** | C

Implemented a skeleton for a distributed hashtable.

### **MAXSAT solver - serial, parallel, distributed** | C

Implemented three versions of a MAXSAT solver: serial, parallel (OpenMP), and distributed (MPI). See [here](#)