The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

The project “**Analyze and Forecast Electricity Market Prices in Germany**” uses the lightweight IBM Cloud Garage Method process model. The lightweight IBM Cloud Garage Method for data science includes a process model to map individual technology components to the reference architecture. This method does not include any requirement engineering or design thinking tasks. Because it can be hard to initially define the architecture of a project, this method supports architectural changes during the process model.

# Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

# Data Source

Germany’s and its neighbouring countries market data can be found on the SMARD website. SMARD is an information platform that belongs to “Bundesnetzagentur” which is a federal bureau of the German state. This platform enables anyone to look up data on electricity generation, consumption, price, imports, exports and more. It aims to increase transparency and is very comprehensive.

## Technology Choice

Data can be either downloaded as csv and xlsx files from the website directly. Moreover, a RESTful API is provided. The details for this API can be found at smard.api.bund.de. At the time of composing of this document the API version is 0.0.1. OA S3.

# Enterprise Data

## Technology Choice

Data can be downloaded as csv or xlsx (or pdf). Alternatively, it can be downloaded by a REST API, which is more practical for automated downloading. The latter solution would be preferred. Unfortunately, the API is not well documented and is according to the documentation broken.

## Justification

Python’s request API is easy to use and capable of accepting Json files. Moreover, Python scripts can easily be adjusted for a later automated download of new data and updating the model. In the meanwhile Python is also an excellent choice for working with csv files.

# Streaming analytics

## Technology Choice

SMARD provides data on Electricity Market with approximately 30 to 60 minutes delay. Some data about market prices and stock markets take a day or two days till they are listed.

Granularity of the data is 15 minutes for energy consumption and energy generation. Granularity for Market Prices is 60 minutes. That means, the prediction model can later be fed in almost real-time, and we can have a streaming analytics and forecasting.

In this project, streaming analytics is not used, for there is no streaming API provided by SMARD.

## Justification

No data streaming available. Data is retrieved by download and – if working again – later by Restful API.

# Data Integration

## Technology Choice

Use of Pandas for ETL, EDA and Feature Generation. Keras 2.10 and Tensorflow 2.10 will be used RNN architecture, as it runs out of the box on MS Windows with CUDA Support. Consequently, a NVIDIA GeForce GPU is used for local execution of the RNN.

## Justification

Python is already used, so no additional language is needed for use here. Tensorflow is fully supported through the Keras library.

# Data Repository

## Technology Choice

GitHub.

## Justification

It is easy to use and should the requirement arise. Git can be used to commit and update changes to the repository.

# Discovery and Exploration

## Technology Choice

Python with Pandas, Matplotlib and Plotly

## Justification

Python is already used. Plus Plotly is for free.

# Actionable Insights

## Technology Choice

Karas 2.10. and Tensorflow 2.10 with Python 3.10.x and Pandas, SciKit-learn and Numpy.

## Justification

EDA and Feature Generation. Keras 2.10 and Tensorflow 2.10 will be used RNN architecture, as it runs out of the box on MS Windows with CUDA Support. Consequently, a NVIDIA GeForce GPU is used for local execution of the RNN.

# Applications / Data Products

## Technology Choice

No technology has been defined here, as only a model is generated and trained here by Keras / Tensorflow, which could be used later on.

## Justification

The product is the consequential result of the used frameworks.

# Security, Information Governance and Systems Management

## Technology Choice

Not applicable

## Justification

Not applicable.