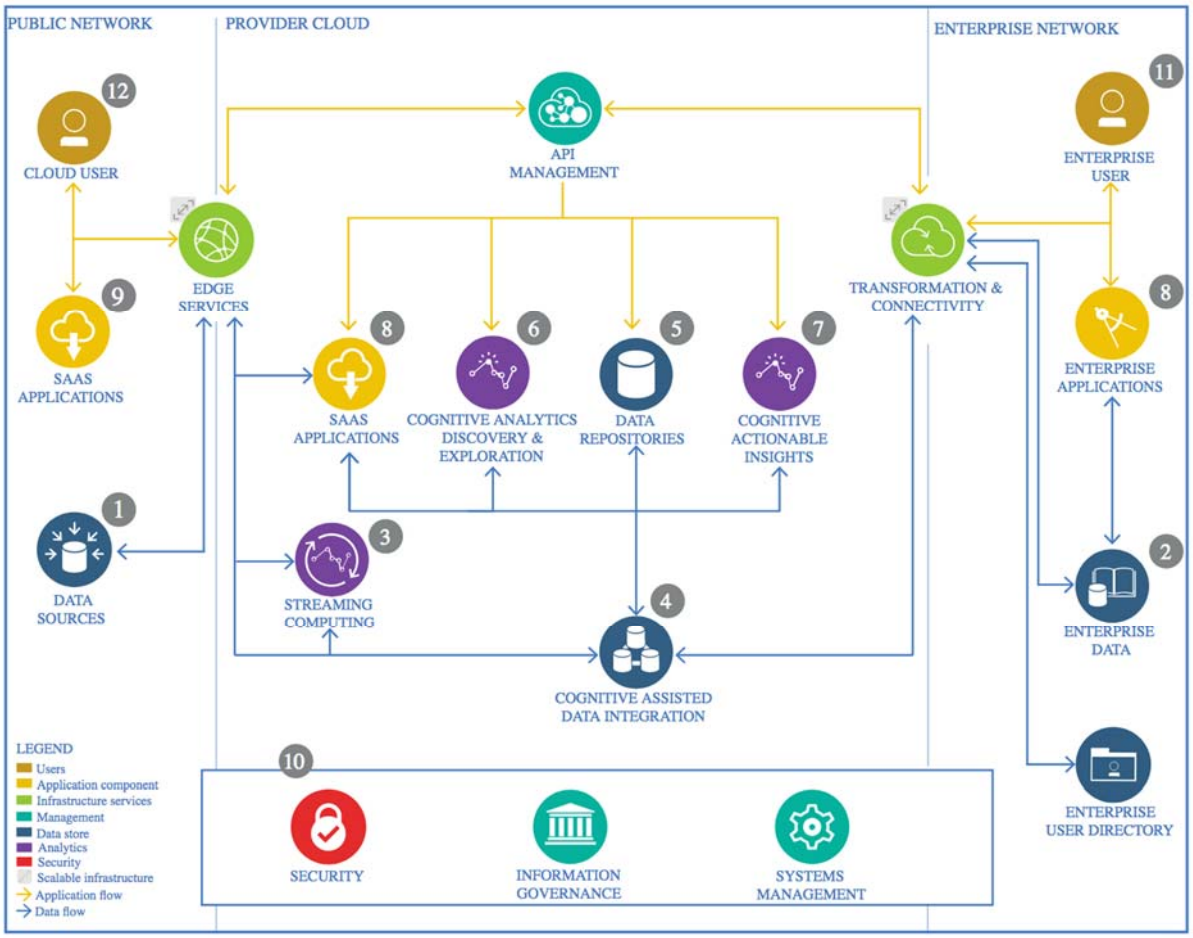
COVID-19 Medical care need

Author: Victor Pontello

The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

This data comes from a renowned Hospital in Brazil from multiple patients with different diagnosis, this Hospital now very committed to fight the COVID-19 pandemic and make use of a IT-Center to manage the healthcare data which are generated in the hospital and try to make use of it to improve the healthcare services offered from them. The Technology choice for Data Source was the simple download from open source dataset from Kaggle. The dataset is open source and was offered in

1.1.2 Justification

There are plenty of interesting data, which are available to use. And the choice was made because of the relevance of the theme and the seriousness of the organization which publicize the data. In these crisis times I am willing to help in the fight against the COVID-19 and to make it with Machine Learning an Data Science Techniques is as a very interesting opportunity to help the world and develop my skills at the same time.

1.2 Enterprise Data

1.2.1 Technology Choice

Seeing me as the Enterprise, that is going to offer the Data Science Services, the technology was to rely on the local computational infrastructure, with one computer with 8 cores and 16 threads, 16 Gb RAM and 4Gb GPU.

1.2.2 Justification

The local computational power is higher than the normal free computational power available in the cloud and is already disponible without extra costs.

1.3 Streaming analytics

1.3.1 Technology Choice

For the deployment of the model it was not needed to use any Streaming analytics technologies like Apache Spark, IBM Streams or NodeRED.

1.3.2 Justification

Because the data is steady, do not change with the time.

1.4 Data Integration

1.4.1 Technology Choice

For this project there was no need to use some extra Data Integration tool like IBM Data Stage on Cloud or Apache Spark.

1.4.2 Justification

It’s relatively small dataset and could be downloaded without problems directly and run locally.

1.5 Data Repository

1.5.1 Technology Choice

For the data repository there was no need found to use some persisting data technology like Relational Databases, NoSQL and Object Storage.

1.5.2 Justification

Mostly because the amount of storage needed is not worth it to create some extra data repository. The impact of cost of storing the data is inexistent and there is no data growth, because the data is steady.

1.6 Discovery and Exploration

1.6.1 Technology Choice

For the discovery and exploration, it is needed to visualize and deal with the data unbalance, lots of missing data und lots of features, which are available on the dataset, but not in real world. To discover the importance of each feature the correlation was calculated, and to impute the missing values the median was used. The choice about dropping features which are not available in the real world was made with support from domain experts. The whole project was developed in Python on the Jupyter Notebook. For the data exploration and the libraries Numpy and Pandas were the mostly used, for the visualization tasks Matplotlib and Seaborn.

1.6.2 Justification

Python was the chosen programming language because it is open source and is the language with most development benefits based on my skills and the disponible resources. Jupyter notebook is a platform, which allies the documentation visualization and the programing on the same place, making the project development dynamic and fast. The libraries Pandas, Numpy, Matplotlib and Seaborn are the standard libraries to perform Data Exploration and, because the dataset was relatively small, there was no need to use some extra tools, like pyspark to these tasks.

1.7 Actionable Insights

1.7.1 Technology Choice

The model is developed locally in Python in the Jupyter Notebook. The additional libraries used were tensorflow 2.0 with keras, kerastuner, sci-kit learn, XGBoost, LightGBM, Imblearn, pandas, numpy and for the data visualization matplotlib library and seaborn. For the development of this project no need of parallel- or GPU-based training or scoring was required, as well as no model interchange. The use of the aforementioned resources offered a good basis to develop the project and also some nice out-of-the-box solutions to the challenges encountered, but some programming tasks were needed mostly to integrate the resources from different libraries. The main metrics which was used was the f1-score. for unbalanced datasets, and that was the case. In that way it was possible to compare the results.

After the models were created and performed really well on both test and datasets, more performance metrics were used, these here mainly the ROC-AUC curve and PR-Curve.

1.7.2 Justification

The option for Tensorflow 2.0 with Keras, XGBoost Classifier and LightGBM Classifier was based on the characteristics of the dataset and the lack of knowledge about the performance of these algorithms in this task. The objective was to try out the performance of different algorithms and then figure out which one performed better. After trying out all of them, the need to deal with the unbalanced data was realized, because the best achieved performance in that time was about 74% f1-score. The key point for the performance rise was the use of the over-sampling algorithm SMOTE which was used to balance the dataset. With the balanced dataset the algorithms learned much better and the performance improved to 97% f1-score. The best trained models were one NN with hyperparameter tuning based on a random search and the LightGBM and XGBoost classifiers in that order, but with almost the same performance (97,03%, 96,99% and 96,99% f1-score respectively). The f1-score was the main choice for the metrics because it combines in an effective way both precision and recall and using that it is possible to compare the performances with and without use of over-sample technique and to find out the real improvement of using that. On the end, after the models with over-sampling technique performed really well and almost the same, further metrics like the ROC-Curve and PR-Curve werde used to compare the final models performances.

1.8 Applications / Data Products

1.8.1 Technology Choice

The from the project generated data product are both the models and the the Jupyter Notebook, where all the project steps are described. The models can be used as plug and play classifiers on the Technology Center of the Hospital to help it's healthcare management and the notebook can be used as inspiration for the Technology datacenter from the Hospital to improve their models and to get insights to deal with this issue. In that form, no need of choosing technologies like Node-RED or D3 was detected.

1.8.2 Justification

The justification about the form of the data product was made based on the explained demand of the Hospital and the it's healthcare professionals. It is a deal to improve the work capacity in this crisis time and to let the professionals from the Data Science collaborate with the fight against the COVID-19.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

Due to the open source spirit of the project, the only issue was to be sure that the data input is the right data and that the versions deployed are the right ones as well. The data and the project are open source and available to anyone which wants to check it,

collaborate with it or use it as source of inspiration for the development of their own projects.

1.9.2 Justification

This decision was made based on the solidarity principle and spirit of the project. That is not a commercial project and the data product is not do be sold. The main goal is to help the healthcare workers to do their job in an efficient way and so help the ill people. That is why, the main issue about the project is to be sure that the right model version be disponible and that the right data is used to create the model, and these are project management and data governance tasks.