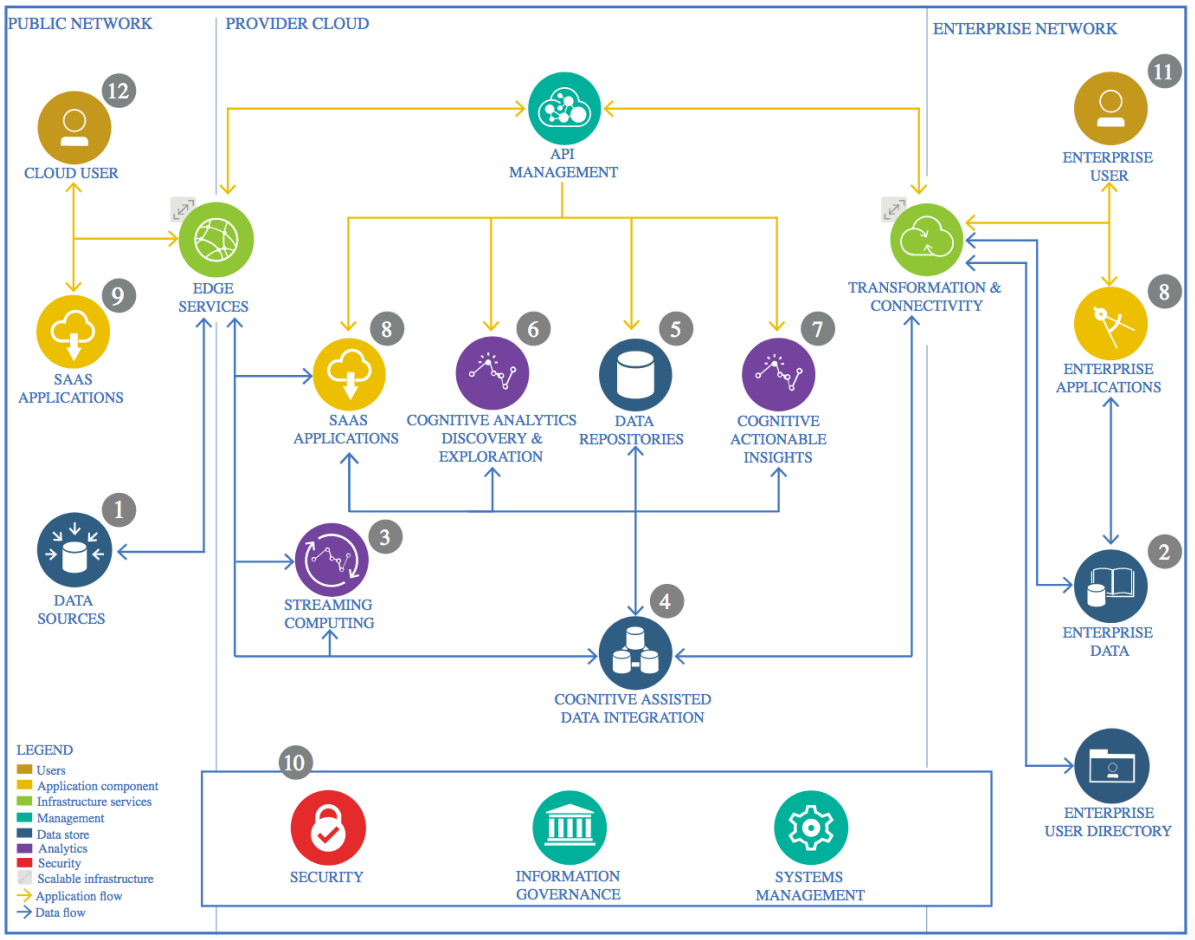
The Lightweight IBM Cloud Garage Method for Data Science

1. Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1. Data Source

1.1.1. Technology Choice
Pandas to download the data into the system.

1.1.2. Justification
CSV file with the results of science research. It is a typical format for the stable research data.

1.2. Enterprise Data

1.2.1. Technology Choice
Please describe what technology you have defined here. Please justify below, why. In case this component is not needed justify below.

1.2.2. Justification
Not needed

1.3. Streaming analytics

1.3.1. Technology Choice
Please describe what technology you have defined here. Please justify below, why. In case this component is not needed justify below.

1.3.2. Justification
Not needed

1.4. Data Integration

1.4.1. Technology Choice
Pandas, sklearn.

1.4.2. Justification
Pandas were selected as it had a simple interface and a big community where you can find the help if you need. It is flexible and easy to use data analysis and manipulation tool. Also, the dataset size is not big, so we don't need parallelization. Sklearn provides easy tools to impute features and do further analytics.

1.5. Data Repository

1.5.1. Technology Choice
Amazon S3

1.5.2. Justification
It's easy to integrate and call from the notebook. It ensures you can save and load data on every step of your pipeline and between the notebooks, So you don't need to repeat the same steps in every notebook. It allows you to divide the process of development into structured steps and save the data on each of them.

1.6. Discovery and Exploration

1.6.1. Technology Choice
Pandas, Matplotlib, Seaborn

1.6.2. Justification
Matplotlib and Seaborn are common, and handy tools for data visualization. They got a lot of built-in functions to plot correlation matrices, histograms, scatter plots, and other useful things.

1.7. Actionable Insights

1.7.1. Technology Choice
PySpark.

1.7.2. Justification
In-Memory cluster computing in Spark, parallelization, speed.

1.8. Applications / Data Products

1.8.1. Technology Choice
Please describe what technology you have defined here. Please justify below, why. In case this component is not needed justify below.

1.8.2. Justification
The realization of our research as data product lies beyond the project, but the possible application, that can be based on it is the website or analytical system, that could provide a doctor help in deciding the chances of a patient, looking for similar cases, deciding type of the therapy. All of the above can be done on the base of the primary analysis, as we proved that even the result after the 5 years could be predicted based on the initial tests. Possible technology is any standard web stack, for example, MySQL + Django application.

1.9. Security, Information Governance and Systems Management

1.9.1. Technology Choice

Please describe what technology you have defined here. Please justify below, why. In case this component is not needed justify below.

1.9.2. Justification

Security should be maintained on the analytics side to avoid un-anonymizing of patients against their will. In our project, we are using already anonymized data, where each patient goes with just an ID, and we don't need to perform any actions on the security side.

2. Development process

The development process is better described and documented in the notebooks, here I recorded architecture choices of the project and answered some crucial questions (below).

2.1 Why have I chosen a specific method for data quality assessment?

I actually did not need chasing any specific method, because I used the research data, as described above in ADD. If we will assume that the "specific method" describes the end data that went into my model, then one important condition for the data in my case - cleaned data. By "cleaned," I mean that I extracted only a particular analysis from patient histories. And standard preprocessing, of course, to make models work.

2.2 Why have I chosen a specific method for feature engineering?

I tested various methods of feature engineering. First of all, I followed the standard guidelines on the question. Scaling and One-Hot encoding are de-facto standards in our days. Also, after every step, I measured the performance indicator of the model and how particularly the engineering process affected it. So, for example, the imputation of missing values was accepted as a part of the model only after I ensured that it is improving the performance.

2.3 Why have I chosen a specific algorithm?

I have chosen Logistic Regression after implementing 7 different classical Machine Learning algorithms and 2 deep learning algorithms and their tuning.

Then I compared the performance indicator of all of the tuned algorithms and chosen the best one.

2.4 Why have I chosen a specific framework

I have chosen an Apache Spark for the In-Memory cluster computing, parallelization, speed. And also because I wanted to practice in it to be able to use new technology in the future.

2.5 Why have I chosen a specific model performance indicator?

I used two performance indicators - accuracy and the AUROC. I have chosen them as standard quality metrics for binary classification. In the end, I decided to concentrate on maximizing the AUROC, as it leaves the opportunity to decide which threshold you want to set in favor of minimizing desired error (type 1 or type 2) but still improves the model. AUC is, I think, is a more comprehensive measure in my case than simple metrics.