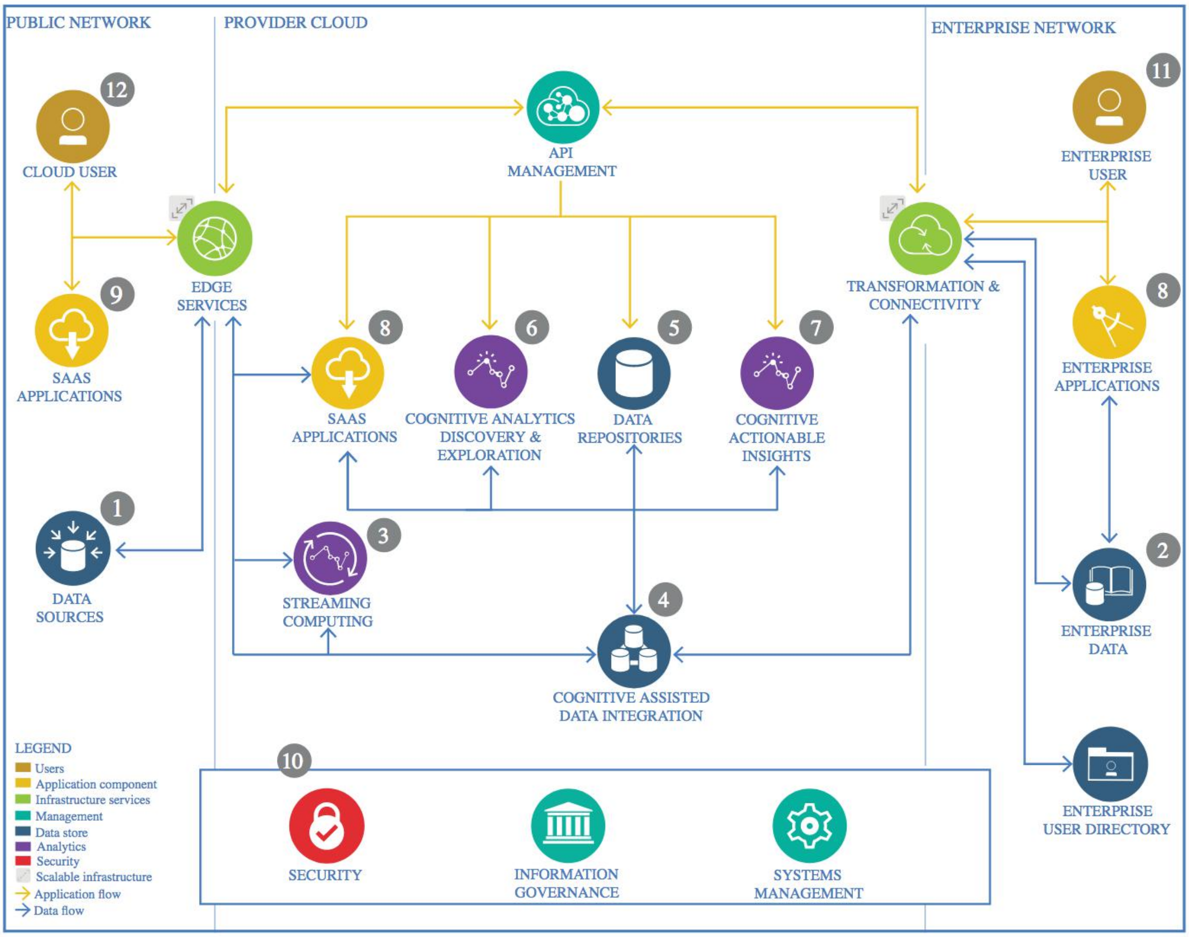
The Lightweight IBM Cloud Garage Method for Data Science

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

The user case is for a private company's manufacturing product quality prediction, the data is extracted from the company's relational database as csv file for the analysis here. To protect IP, all columns have been renamed as generic names.

1.1.2 Justification

Csv file is commonly used for extracting data from relational database and transfer to further analysis.

1.2 Enterprise Data

1.2.1 Technology Choice

Not used here.

1.2.2 Justification

The data is extracted from relational database as csv file, so enterprise data is not needed.

1.3 Streaming analytics

1.3.1 Technology Choice

Not used here.

1.3.2 Justification

The user case is to predict the final product quality metric using inline test results. While it's doable to add the streaming analytic function later to continuously monitor the inline test results, from business procedure perspective, we should demonstrate the model first to gain more support from management and users and then implement the streaming analytic function.

1.4 Data Integration

1.4.1 Technology Choice

The data is first inspected. Then all columns are set to float data type because they are all continuous values as inline test results. Then all feature columns are standardized using the sklearn standardscaler function. We also tried PCA function for feature dimension reduction, while we see the F1-score is worse using PCA components (this is expected since we reduce the feature size), such understanding is useful when we later decide to further increase feature size and need to make a trade-off between model training effort and prediction accuracy.

1.4.2 Justification

Common steps for data integration are used here, including correcting data types, standardization and PCA analysis. These methods are chosen based on the dataset's quality. We observed some feature columns are int type instead of float type, so we changed all columns to float type. We saw values vary significantly (on orders' level) among feature columns, so we applied standardization. We also observed correlation between feature columns, so we tried PCA.

1.5 Data Repository

1.5.1 Technology Choice

The csv file is uploaded to IBM cloud object storage, as integral solution of Watson Studio.

1.5.2 Justification

IBM cloud object storage is used here as integral solution of Watson Studio.

1.6 Discovery and Exploration

1.6.1 Technology Choice

We examined the correlation matrix of the dataset and also visualize some of the correlations that have relatively higher correlation coefficient. The visualization methods are general statistical metrics of each column (using describe function to get mean, std dev, etc. for each column), heat map of the correlation matrix, scatter plot of two features that have high correlation coefficient and box plot of a feature column between final test result 0 and 1. From the correlation matrix, we can see the final test result (the column we would like to predict) shows correlation to some feature columns but not very strong, which indicates we many need a lot of features to ensure a good prediction. Also we see some features are correlating to each other, which indicates a PCA analysis may be useful to reduce dimension if we need to make a trade-off between model accuracy and training time.

1.6.2 Justification

Dataset inspection methods used here are correlation matrix and PCA. Visualization methods used here are heat map, scatter plot and box plot. These are the general data assessment methods so we chose to use them. Since there are correlations observed among features and final label, scatter plot / box plot are used to visualize such correlations.

1.7 Actionable Insights

1.7.1 Technology Choice

3 machine learning models are tried here: 2 non-deep-learning models of Gradient-Boosted Tree (GBT) and Supported Vector Machine (SVM); 1 deep-learning model of Multiple Layer Perceptron (MLP). Since the user case is a binary classification problem, according to literature, GBT should have the best performance. SVM and MLP are commonly used model for classification, one is a traditional machine learning model while the other one is deep-learning model, so they are used to compare with GBT. For each model, 3 data preprocessing scenarios are tried: no preprocessing, standardization, PCA + standardization. We chose to use sklearn as framework since our dataset size is not too large and sklearn is capable of handling it and we don’t have to go to Apache Spark for big data.

1.7.2 Justification

3 machine learning models are tried, including the most promising non-deep-learning model GBT according to literature.

1.8 Applications / Data Products

1.8.1 Technology Choice

The model’s prediction performance F1-score is generated (0.956) to show to the stakeholders. If approved, we can follow up with creating an application for user to input the test result and get predicted final product pass/fail.

1.8.2 Justification

Use model performance F1-score to demonstrate model’s capability because the user case is a binary classification problem and F1-score is capable of comparing among models.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

Not used here.

1.9.2 Justification

After the model is approved by the stakeholders, we can discuss with the company’s IT system to implement it with security. This can be a follow-up, but not included in the current project.