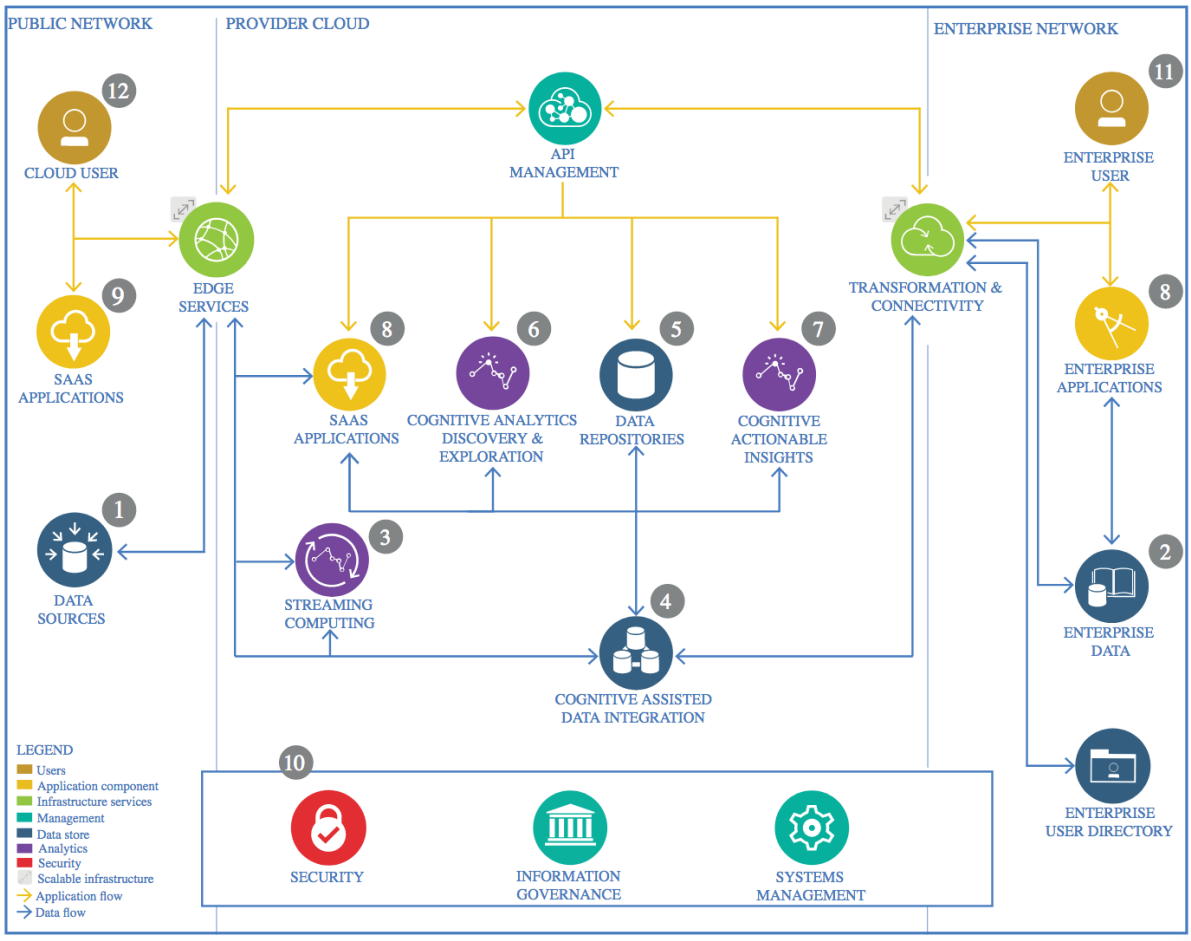
The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

Skin Lesion Prediction Project

Dermatological computer-aided classification of Pigmented Skin Lesions

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

Data sources are available thorough Kaggle or Harvard Dataverse in CSV format. Data includes a large set of dermatoscopic RGB images of skin lesions and a patient metadata file containing information including patient age, sex, and lesion location.

1.1.1 Technology Choice

Excel files are used to export the dataset and metadata file.

1.1.2 Justification

Reading large files is much easier for the end user. Can be added to as more data is collected in the future.

1.2 Enterprise Data

Not applicable. Not working with data for a specific commercial enterprise.

1.2.1 Technology Choice

Component not needed.

1.2.2 Justification

Data is obtained once from external data source for modeling.

1.3 Streaming analytics

Not applicable. We are not working with streaming data.

1.3.1 Technology Choice

Component not needed.

1.3.2 Justification

Project does not involve real time data streaming.

1.4 Data Integration

In the data integration stage, data is cleansed, transformed, and if possible, downstream features are added.

1.4.1 Technology Choice

Feature extraction/creation, data cleaning and formatting are performed using Python, and pandas data frames.

1.4.2 Justification

Feature extraction/creation, data cleaning and formatting are performed to create a reliable data set suitable for achieving high performance with the algorithms used.

1.5 Data Repository

Persistent storage for the data.

1.5.1 Technology Choice

Cloud object storage.

1.5.2 Justification

1. Object storage is the cheapest option for storage.
2. Any data type is supported.
3. Scales to the petabyte range.
4. Can access specific storage locations through folder and file names and file offsets.

1.6 Discovery and Exploration

Visualizations and statistical analysis of data set features. Examine numeric and categorical distributions, and correlations.

1.6.1 Technology Choice

Pandas, Seaborn, NumPy and Matplotlib are used for analysis and visualizations.

1.6.2 Justification

Pandas, Seaborn, NumPy and Matplotlib are widely used, open-source libraries with extensive documentation and community support.

1.7 Actionable Insights

We have a multiclass classification task. The metrics will be Accuracy. We also review Precision, Recall/Sensitivity, and Specificity This is also a medical classification; Sensitivity and Specificity are used for evaluation in this field.

- # Algorithms

Non-Deep Learning:

- Dummy Classifier
- Logistic Regression (Altered for multiclass classification)
- Decision Tree
- Random Forest

- XGBoost Classifier

Neural Networks with loss function: (sparse_categorical_crossentropy)

- Single fully connected layer with same number of neurons as the input variables, one hidden layer with 512 neurons and softmax output layer.
- Second model is the same as the first with an additional hidden layer of 64 neurons and a dropout layer at (0.2)

Best performing models:

- Neural Network is the best performing model.
- XGBoost algorithm is the second-best performer.

1.7.1 Technology Choice

Python with libraries (pandas, scikit-learn, numpy, seaborn, tensorflow and keras) are used with IBM Watson Studio and Jupyter notebooks.

1. Pre-run Jupyter notebooks with explanations of each step in the IBM lightweight Cloud Garage Method for data science, named according to convention.
2. Illustrated Power Point presentations for stakeholders and data science piers.
3. Video presentation of the entire project.

1.7.2 Justification

Project components are open source and have extensive documentation and community support. Scikit-learn, TensorFlow and Keras support a wide range of state-of-the-art models.

Accuracy is the proportion of correct predictions over the total number of predictions.

Accuracy = (TP + TN) / All Predictions

Precision (Positive Predictive Value - PPV), out of all predicted positive cases, how many were actually positive.

Precision = TP / (TP + FP)
Sensitivity (Recall), out of all actual positives how many were predicted as positive.

Sensitivity (Recall) = TP / (FN + TP)

Specificity (Selectivity or True Negative Rate – TNR), out of all actual negatives (not a certain lesion), how many were predicted as negative.

Specificity = TN / (TN + FP)

Out of the various combinations of Precision, Recall/Sensitivity, and Specificity applied to algorithms we are looking for those where we have **High Precision, High Recall/Sensitivity**, and **High Specificity**.

1.8 Applications / Data Products

1.8.1 Technology Choice

Jupyter notebook and trained model with explanation of the entire process.
Trained model and weights ready for integrating into a mobile device application.
Final deployment will depend upon the intended use scenario and resources available.

1.8.2 Justification

A Jupyter notebook with a trained model and explanation is simple to use for testing in a local environment.

The model can be used to classify skin lesions as an augmentation of in clinic diagnoses.

Exported model can be pushed to mobile devises for use outside of a clinical setting, even where internet service may be sparce or non-existent, thereby potentially expanding access to low-cost medical care to almost anywhere.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

Project Jupyter notebooks and saved model stored on the IBM Cloud Platform and GitHub repository. Data files in cloud object storage.

1.9.2 Justification

Security for these low or no cost services is managed by the vendors.