The Lightweight IBM Cloud Garage Method for Data Science

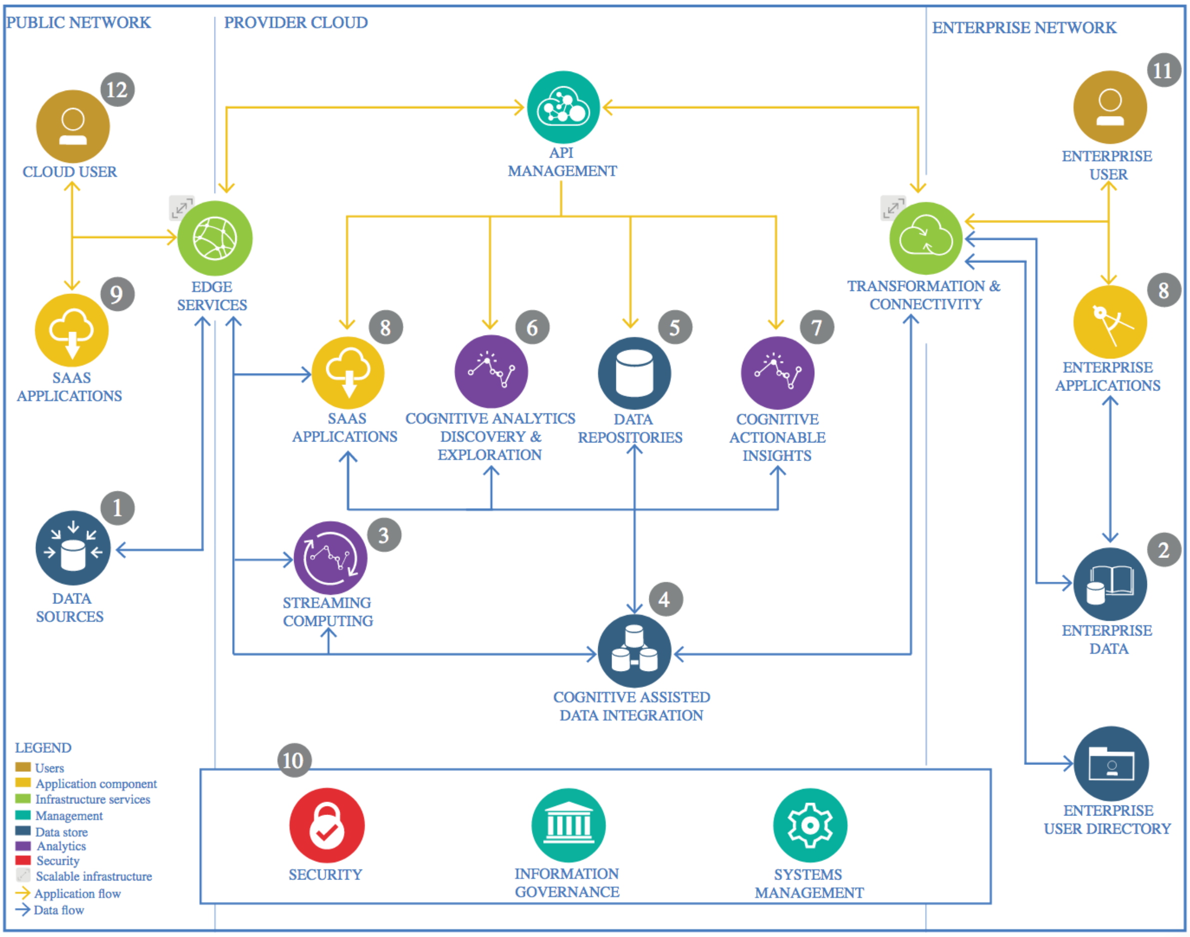
Architectural Decisions Document

ECG Heartbeat Categorization

This project focuses on accurately classifying five different rhythms:

- Normal
- Atrial Premature Contraction
- Premature Ventricular Contraction
- Fusion of Ventricular and Normal Contractions
- Paced, Unclassifiable heartbeats

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

The dataset is composed of two collections of heartbeat signals derived from two datasets in heartbeat classification: MIT-BIH Arrhythmia Dataset and The PTB Diagnostic ECG Database. The dataset is also available through Kaggle. Dataset Link: https://www.kaggle.com/shayanfazeli/heartbeat?select=mitbih_train.csv

1.1.1 Technology Choice

Excel/CSV file store the dataset

1.1.2 Justification

Excel/CSV files are easier for the end user and can be expanded as more data becomes available.

1.2 Enterprise Data

Not applicable. Data is not from a specific commercial enterprise.

1.2.1 Technology Choice
Component not needed.

1.2.2 Justification
NA

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 streaming data.

1.4 Data Integration
In the data integration stage, the data is cleaned, transformed and if necessary 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
Data preparation is necessary for creating a dataset suitable for achieving high performance with machine learning algorithms.

1.5 Data Repository
Persistent storage is used.

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
Statistical analysis and visualizations of ECG samples are used to examine ECG signals, numeric and categorical distributions and correlations of the dataset features.

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

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

Algorithms

Non-Deep Learning:

- Decision Tree
- Random Forest
- XGBoost Classifier

Neural Network Architectures:

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

Sequence Learning Architecture:

- 1D Convolutional layer with 10 filters, MaxPooling1D layer, three LSTMs with 100 units each, BatchNormalization and a Dense softmax output layer.

Model Performance:

- **Random Forest** Classifier on balanced dataset has an accuracy of 0.996.
- **Neural Network** with second model configuration as above has an accuracy of 0.991 on balanced dataset with PCA applied.
- **Sequence Model** performs with 0.981 accuracy on the standard dataset.

1.7.1 Technology Choice

Python with libraries (pandas, scikit-learn, numpy, seaborn, tensorflow and keras) are used with IBM Watson Studio and pre-run Jupyter notebooks with explanations of each step.

1.7.2 Justification

The 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.

TensorFlow is one of the most widely used deep learning frameworks. At its core, it is a linear algebra library supporting automatic differentiation. Keras provides an abstraction layer on top of TensorFlow.

Model Selection

Decision tree - The process behind the model is easy to understand and clear when viewing a plot of the model. Only the important attributes in making a decision are included in its rule. Attributes that do not contribute to the accuracy of the model are ignored.

Random forests - operate by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. Random forests correct for individual decision tree overfitting.

XGBoost - Extreme Gradient Boosting is a supervised leaning gradient boosted trees algorithm. The gradient boosting technique attempts to accurately predict targets by combining the estimates of simpler, weaker models.

Neural Network - Neural Networks can recognize patterns and produce output that is not limited to the input provided to them. Neural networks learn from examples and can apply them in similar situations.

Sequence Learning Model - Traditional models assume that all inputs (and outputs) are independent of each other. A sequence learning model considers previous inputs to be important for predicting the next output. ECG signals are a time series.

Performance Metrics

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 an 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 ECG signals as an augmentation of in clinic diagnoses.

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.