The Lightweight IBM Cloud Garage Method for Data Science Hamza jamshaid

Architectural Decisions Document

1 Architectural Components Overview

1.1 Data Source

1.1.1 Definition

Understanding the data is an important step when designing a machine learning algorithm. In the notebook, I will use a data set published by Smartlab - Università degli Studi di Genova, Genoa (I-16145), Italy. The data was recorded using smartphone sensors for monitoring movement (Gyro-scope, Accelerometer). The data is labeled, hence very useful and easier to evaluate. The data is pre-partitioned 70% training 30% test data. the data was recorded from 30 volunteers with ages varying between 19-48. The data is available at UCI machine learning repo for downloading.

1.1.2 Technology Choice

- IBM Watson Studio jupyter notebooks, Python, scikit-learn, pandas, matplotlib.
- Numpy, Tensorflow, Keras

1.1 Data Integration

1.1.1 Technology Choice

- IBM Watson Studio jupyter notebooks, scikit-learn, pandas, matplotlib.
- Numpy, Tensorflow, Keras

1.1.1.1.1 Python

Python is often the primary choice data processing and machine learning. The advantages Python has over other technologies is its flexibility and easy to understand syntax as it is a programming hence the primary choice for people with computer science background.

- What skills are required?
  At least advanced SQL skills are required and some familiarity with either C++ or any basic programming language.
- What data types must be supported?

Python is very dynamic in this sense as it works well with all sorts of data and can easily structure data according to requirements. It supports datasets, arrays and other data structures as well. However for machine learning purposes numpy.array is best suited and supported.

- What source systems must be supported?
  Python can access a variety of SQL and NoSQL data based as well as file source out of the box. A common data source architecture allows adding capabilities. 3rdparty project add functionality as well

1.1.1.2 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

1.1.1.3 IBM Watson Studio

IBM Watson Studio provides tools for data scientists, application developers and subject matter experts to collaboratively and easily work with data to build and train models at scale. It gives you the flexibility to build models where your data resides and deploy anywhere in a hybrid environment so you can operationalize data science faster.

1.1.1.4 Pandas

It's important to notice that data integration is mostly done using ETL tools or plain SQL or a combination of both. ETL tools are very mature technology and an abundance of technologies exist. Pandas is very efficient at converting object files into data frames which can be transformed and engineered for future tasks

1.2 Data Repository

1.2.1 Architectural Decision Guidelines

There exists an extremely huge set of technologies for persisting data. Most of them are relational databases. The second largest group are NoSQL databases and file system (including Cloud Object Store) form the last one. The most important questions to be asked are:

- How does is the impact of storage cost?
- Which data types must be supported?
- How good must point queries (on fixed or dynamic dimensions) be supported?
- How good must range queries (on fixed or dynamic dimensions) be supported?
- How good must full table scans be supported?
- What skills are required?
- What's the requirement for fault tolerance and backup?
- What's the amount of storage needed?
- How does the growth pattern look like?
- What are the retention policies?

1.2.2 Technology choice

- Cloud file storage (Object Storage, raw files)

1.2.3 Justification

1.2.3.1 Object Storage

In this project, I will use raw txt files to store and read the dataset, the total size of the training and testing data set is (<100mb). Hence no special storage requirements are needed.

- Very low storage cost.
- Linearly scalable.
- Schema-less, hence seamless migration

1.3 Discovery and Exploration

1.3.1 Definition

This component allows for visualization and creation of metrics of data.
In various process models, data visualization and exploration is one of the first steps. Similar tasks are also applied in traditional data warehousing and business intelligence. So for choosing a technology, the following questions should be kept in mind:

- What type of visualizations are needed?
- Are interactive visualizations needed?
- Are coding skills available / required?
- What metrics can be calculated on the data?

1.3.2 Technology Choice

- Pandas, Numpy, Matplotlib, Seaborn, sci-kit learn

1.3.3 Justification

1.3.3.1 Jupyter, python, scikit-learn, pandas, matplotlib, seaborn

The components mentioned above are all open source and supported in the IBM Cloud. Some of them have overlapping features, some of them have complementary features. This will be made clear by answering the architectural questions

- What type of visualizations are needed?Matplotlib supports the widest range of possible visualizations including bar charts, run charts, histograms, box-plots and scatter plots.
- Are interactive visualizations needed?Whereas matplotlib creates static plots, pixiedust supports interactive ones
- Are coding skills available / required?Whereas matplotlib needs coding skills,. For computing metrics, some code is necessary in Python
- What metrics can be calculated on the data?Using scikit-learn and pandas, all state-of-the-art metrics are supported

- Do metrics and visualization need to be shared with business stakeholders?Watson Studio supports sharing of jupyter notebooks, also using a fine grained user and access management system

1.4 Actionable Insights

1.4.1 Technology Choice

- Python, Pandas and Sci-kit learn.
- Keras, Tensorflow

1.4.2 Justification

1.4.2.1 Python, pandas and scikit-learn

Python is a much cleaner programming language than R and easier to learn therefore. Pandas is the python equivalent to R dataframes supporting relational access to data. Finally, scikit-learn nicely groups all necessary machine learning algorithms together. It's supported in the IBM Cloud via IBM Watson Studio as well.

- What are the available skills regarding programming languages? Python skills are very widely available since python is a clean and easy to learn programming language.
- What are the cost of skills regarding programming languages? Because of python's properties mentioned above, cost of python programming skills are very low
- What are the available skills regarding frameworks? Pandas and scikit-learn are very clean and easy to learn frameworks, therefore skills are widely available.
- What are the cost of skills regarding frameworks? Because of the properties mentioned above, cost of skills are very low.
- Is model interchange required? All scikit-learn models can be (de)serialized. PMML is supported via 3rdparty libraries.

1.4.2.2 Tensorflow and Keras

Tensorflow is a great library to learn and dive into deepleanring. For our project we selected keras sequential model to implement a deep learning algorithm for our application. Keras is easier to understand and build a deep learning model.

1.5 Model Evaluation

1.5.1 Choice

- Confusion Matrix
- Precision, Recall and F1-Score

1.5.2 Justification

1.5.2.1 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

We use a confusion matrix in order to identify the miss classified labels which is very important since we are dealing with multiple classes and our dataset has more than 500 features.
Getting a count of all the wrongly classified data helps in better evaluation of our model.

1.5.2.2 Precision, Recall and F1-Score

1.5.2.2.1 Precision and Recall

It can be more flexible to predict probabilities of an observation belonging to each class in a classification problem rather than predicting classes directly. The reason for this is to provide the capability to choose and even calibrate the threshold for how to interpret the predicted probabilities.

1.5.2.2.2 F1-Score

F1 Score is needed when you want to seek a balance between Precision and Recall. Accuracy can be largely contributed by a large number of True Negatives which in most business circumstances, we do not focus on much whereas False Negative and False Positive usually has business costs (tangible & intangible) thus F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

1.6 Model Deployment

1.6.1 Choice

- Jupyter Notebook

1.6.2 Justification

1.6.2.1 Jupyter Notebook

Our model is a research product and further working is still under way, hence it is being developed and currently deployed as a Python notebook.