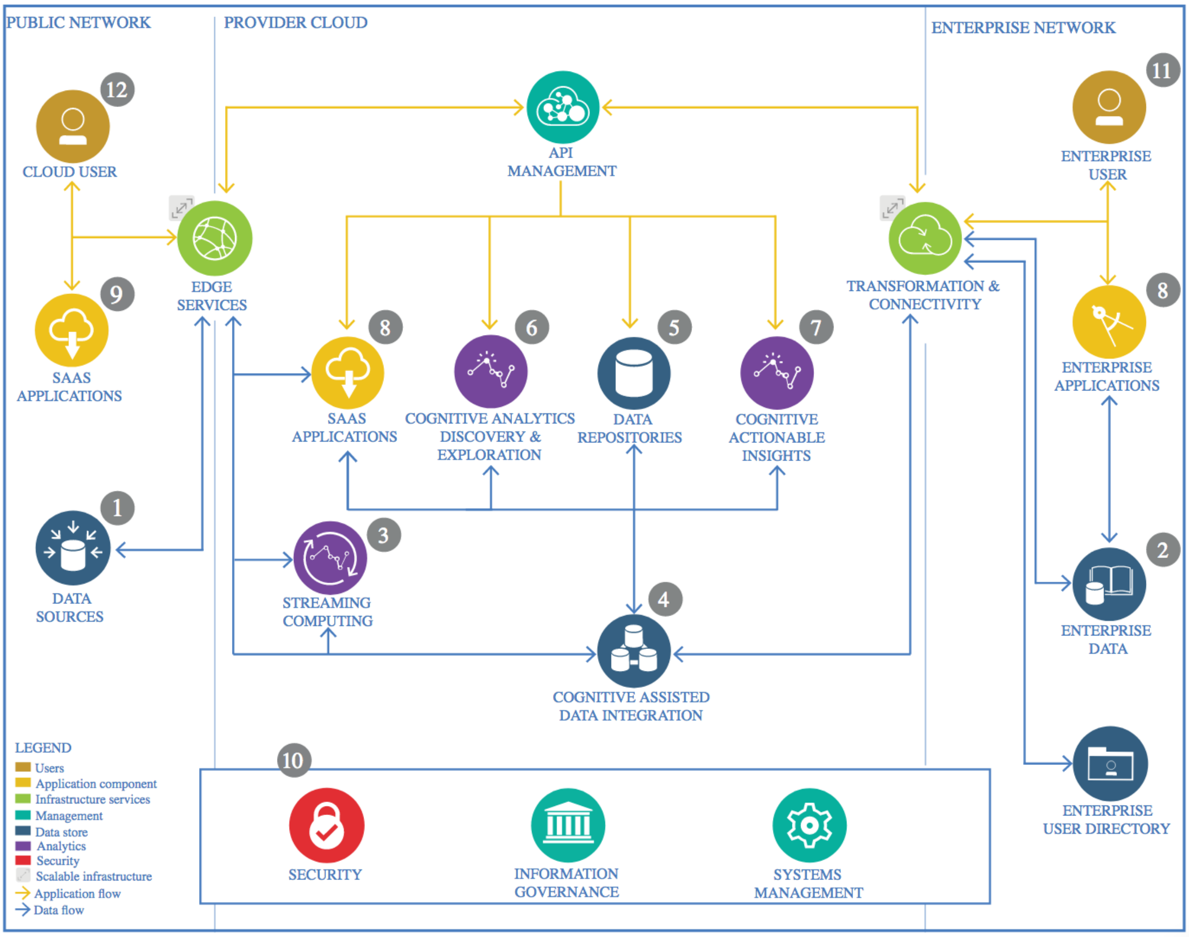
Prediction of Vehicle Resale Price

Architectural Decisions Document

IBM Advanced Data Science Capstone Project
By Nikolaj Andresen

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

CSV file downloaded from Kaggle:
https://www.kaggle.com/austinreese/craigslist-carstrucks-data
Additional information is collected through web scraping from the following sources:

- https://people.sc.fsu.edu/~jburkardt/datasets
  - John Burkardt's "repository" contains readily available information on the latitude and longitude of capitals in the United States, as well as neighboring states.
- YouGov
  - British market research site, which has popularity and fame ratings on roughly 60 vehicle brands.
- NHTSA
  - Free VIN decoder to obtain additional information, where applicable, on each vehicle.
- Wikipedia
  - List of current electric cars

1.1.2 Justification

The choice of Kaggle is because it was an easy way to obtain relevant main data for the project scope. Scraping data from different websites to obtain extra information seemed the logical choice, instead of simply copy and pasting, in order to make the data load more dynamic.

1.2 Enterprise Data

1.2.1 Technology Choice

Not applicable.

1.2.2 Justification

There is no need to transfer data to the cloud, for example, as this dataset is static and would only need to be updated quarterly, monthly or maybe once a week, depending on the amount of data the real business would be able to accumulate.

1.3 Streaming analytics

1.3.1 Technology Choice

Not applicable

1.3.2 Justification

There is no streaming data used in this project.

1.4 Data Integration

1.4.1 Technology Choice

Local Python environment consisting of the following packages (by group):

- Data manipulation:
    - NumPy (matrix/vector operations)
    - Pandas for data manipulation and ETL
    - GeoPandas (for geographical data)
    - Pandasql (SQL like syntax for pandas)
- Visualization:
    - Matplotlib
    - Seaborn
- Data extraction/scraping:
    - Requests (for REST API calls)
    - Selenium (for web scraping)
    - BeautifulSoup (for parsing html)
- Machine learning:
    - Scikit-learn (preprocessing tools such as scaling and one-hot-encoding, and pipelines)
    - LightGBM (for decision tree-based machine learning)
    - Tensorflow/Keras (for deep learning)

Exploration and modelling done in Jupyter notebooks. Web app written in regular Python.

1.4.2 Justification

My local environment gives me easy access to a lot of memory and computing power from an Nvidia GPU. It also gives the freedom to easily install, modify, and execute packages and code.

1.5 Data Repository

1.5.1 Technology Choice

Data stored locally on desktop PC.

1.5.2 Justification

As the development was done locally, it made sense to keep it all readily available on a local machine.

1.6 Discovery and Exploration

1.6.1 Technology Choice

Pandas, matplotlib and seaborn in Visual Studio Code with Jupyter notebooks.

1.6.2 Justification

Jupyter notebook allows for inline visualizations and code in combination with Word like text. This allows for a fantastic environment for a data scientist to show of his/her work, not only to a technical crowd, but business stakeholders as well in form of a report. Most cloud services such as Watson Studio, Google Colab, and so on, provide Jupyter notebook as tools for development.

1.7 Actionable Insights

1.7.1 Technology Choice

Visual Studio Code with Jupyter notebooks in combination with packages listed in 1.4.1

1.7.2 Justification

As an extension of 1.6.2, Jupyter makes sense for machine learning training, as it provides easier inline execution of single code cells, as opposed to mainstream IDEs that in some cases only allows for execution of the entire script at once. This makes it easier to use when debugging code. For production, plain Python would be preferrable.

1.8 Applications / Data Products

1.8.1 Technology Choice

Web application built in Python with Flask (with extensions), HTML and some JavaScript.

1.8.2 Justification

A web application seemed like the best choice to show off the results of the machine learning model, as a live tool is sometimes more effective than graphs. It allows the user to play around with different values for the variables.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

Not applicable

1.9.2 Justification

This is not needed as the data is freely available and contains no personally identifiable information.