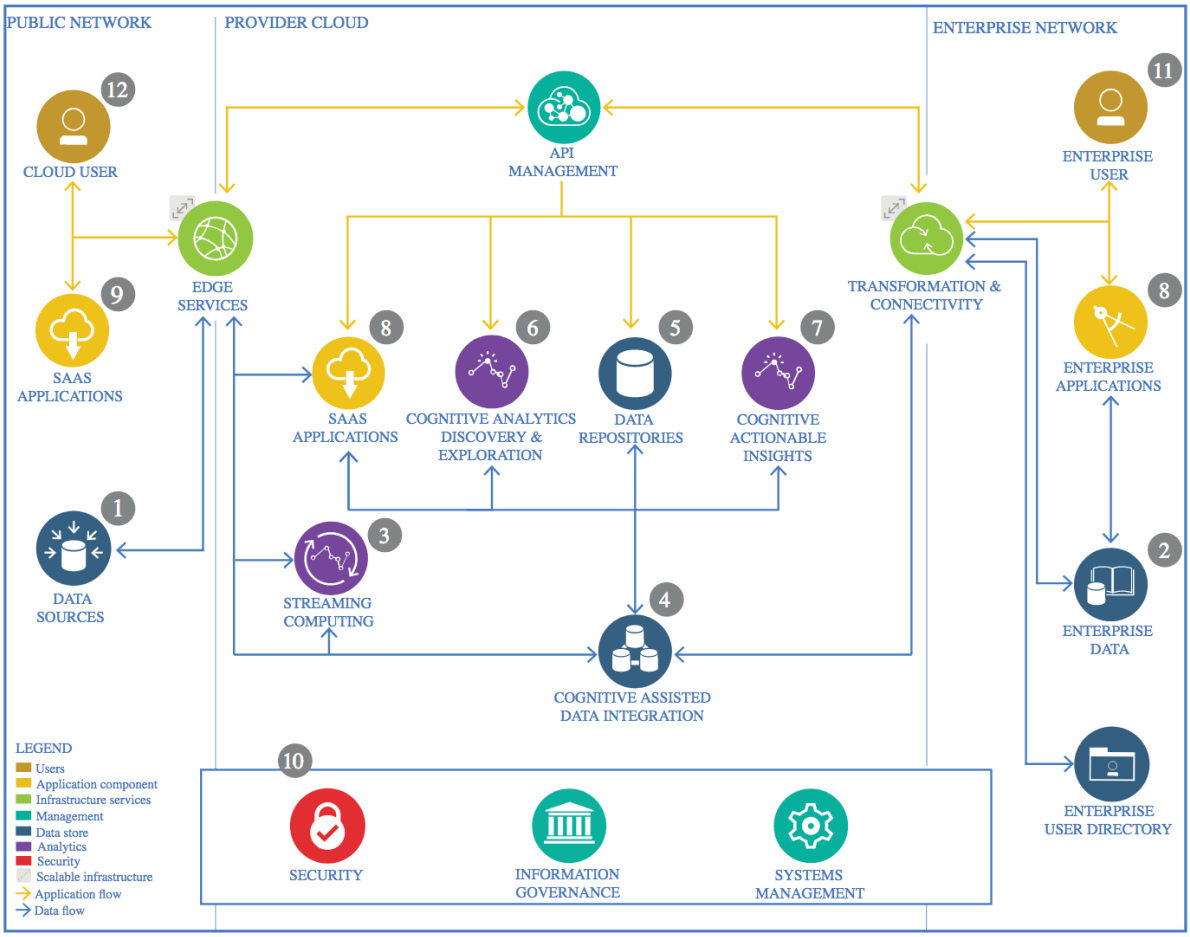
Sentiment Prediction Model

Architectural Decisions Document

By **Damilola Esan**

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

CSV (Comma-separated values) file format is chosen as the data source for this project. CSV is a widely used format for structured data, and it is easily accessible and manageable.

1.1.2 Justification

The Sentiment140 dataset is available in CSV format on Kaggle, and it contains the required sentiment labels and tweet texts for sentiment analysis. CSV allows easy data extraction and processing using libraries like pandas and is compatible with various data processing tools.

1.2 Data Quality Assessment

1.2.1 Technology Choice

Manual inspection to visually identify noise and automated preprocessing techniques using regular expressions (re), pandas, and the Natural Language Toolkit (NLTK) to ensure clean text data.

1.2.2 Justification

A manual inspection enables the visual identification of noise, while the combined use of regular expressions, pandas, and NLTK ensures automated preprocessing for consistent and clean text data. Regular expressions assist in pattern matching and removal of specific noise elements, pandas offer efficient data manipulation capabilities, and NLTK provides tools for comprehensive text preprocessing.

1.3 Streaming analytics

1.3.1 Technology Choice

Not applicable

1.3.2 Justification

Streaming analytics is not applicable to this project as we are working with a static dataset, and real-time analysis is not a requirement.

1.4 Data Integration

1.4.1 Technology Choice

For this project, data integration was not a major concern as the Sentiment140 dataset was already provided in a single CSV file.

1.4.2 Justification

Since the dataset is available in a single CSV file, there is no need for additional data integration processes.

1.5 Data Repository

1.5.1 Technology Choice

IBM Cloud Object Storage was selected as the ideal data repository for the Sentiment Prediction Model project due to its unmatched scalability. This choice ensures the efficient storage and accessibility of the Sentiment140 dataset, comprising 1.6 million labeled tweets.

1.5.2 Justification

IBM Object Storage's scalability effortlessly accommodates the dataset's size, allowing it to grow seamlessly with project needs. Its advanced security measures, including encryption and access controls, guarantee the protection of data.

1.6 Discovery and Exploration

1.6.1 Technology Choice

Seaborn was used for data visualization and the unique capabilities of the WordCloud library were utilized to visualize textual data in an insightful manner.

1.6.2 Justification

Seaborn is a pillar in the data visualization realm, extensively utilized for its versatility and adaptability. Also, the incorporation of WordCloud enriches the approach, providing a visually captivating representation of text, and highlighting the most common words.

1.7 Feature Engineering

1.7.1 Technology Choice

For Logistic Regression and Random Forest models, the process involves utilizing Scikit-Learn's TF-IDF vectorization for text representation. In the case of the LSTM model, Keras was used for tokenization, padding of sequences, and GloVe pre-trained embedding for capturing word relationships.

1.7.2 Justification

Scikit-Learn's TF-IDF vectorization is a well-established method for converting text data into numerical features for traditional models. Keras provides a user-friendly syntax for preprocessing text data and building neural network architectures, while GloVe pre-trained embedding enhances the LSTM model's capacity to understand semantic relationships among words.

1.8 Algorithm

1.8.1 Technology Choice

Logistic Regression, Random Forest, and an LSTM-based deep learning model were utilized. These models are implemented using Python libraries such as scikit-learn and TensorFlow’s Keras.

1.8.2 Justification

Scikit-Learn offers a wide array of machine-learning algorithms, making it an ideal choice for implementing Logistic Regression and Random Forest models. These algorithms are well-established and widely used for binary classification tasks. For the LSTM model,

TensorFlow's Keras provides a robust framework for building and training neural networks, especially suited for sequence data analysis like text.

1.9 Framework Choice

1.9.1 Technology Choice

The Jupyter Notebook environment provided by IBM Watson Studio is selected as the primary framework for developing and documenting the entire project.

1.9.2 Justification

The Jupyter Notebook environment in IBM Watson Studio provides an integrated ecosystem for seamless data analysis and model development. It allows for easy integration of code, text explanations, and visualizations in a single document. This enhances the project's reproducibility and facilitates effective collaboration among team members. The interactive nature of Jupyter Notebooks enables real-time experimentation, making it well-suited for exploring data, testing algorithms, and fine-tuning models.

1.10 Model Performance Indicator

1.10.1 Technology Choice

Metrics such as accuracy, precision, recall, and F1-score are calculated using Scikit-Learn's library functions.

1.10.2 Justification

The chosen metrics provide a comprehensive evaluation of model performance in binary classification scenarios. Accuracy indicates the overall correctness of predictions, precision highlights the proportion of true positive predictions among positive predictions, recall measures the proportion of actual positives correctly predicted, and the F1-score balances precision and recall. Utilizing well-known metrics facilitates a clear and standardized assessment of model effectiveness.

1.11 Applications / Data Products

1.11.1 Technology Choice

The primary output of this project is sentiment predictions for given text inputs. We have chosen to save and load machine learning and deep learning models locally using Joblib and Keras for future deployment. Additionally, we use Python to build a function to predict sentiment using the deployed models.

1.11.2 Justification

Saving models locally and deploying them as a function allows for easy integration into various applications or data products. This approach provides flexibility in utilizing the

sentiment analysis models for different use cases, such as sentiment analysis in customer feedback forms or social media monitoring.

1.12 Security, Information Governance, and Systems Management

1.12.1 Technology Choice

Security, information governance, and systems management are crucial aspects of this project. We rely on IBM Cloud Object Storage's built-in security features and access controls to protect the Sentiment140 dataset. Additionally, we implement access restrictions and encryption for local model storage.

1.12.2 Justification

Leveraging IBM Cloud Object Storage for data storage ensures robust security measures are in place, including encryption and access control. For local model storage, access restrictions and encryption further enhance security. These measures align with our commitment to data security and governance.