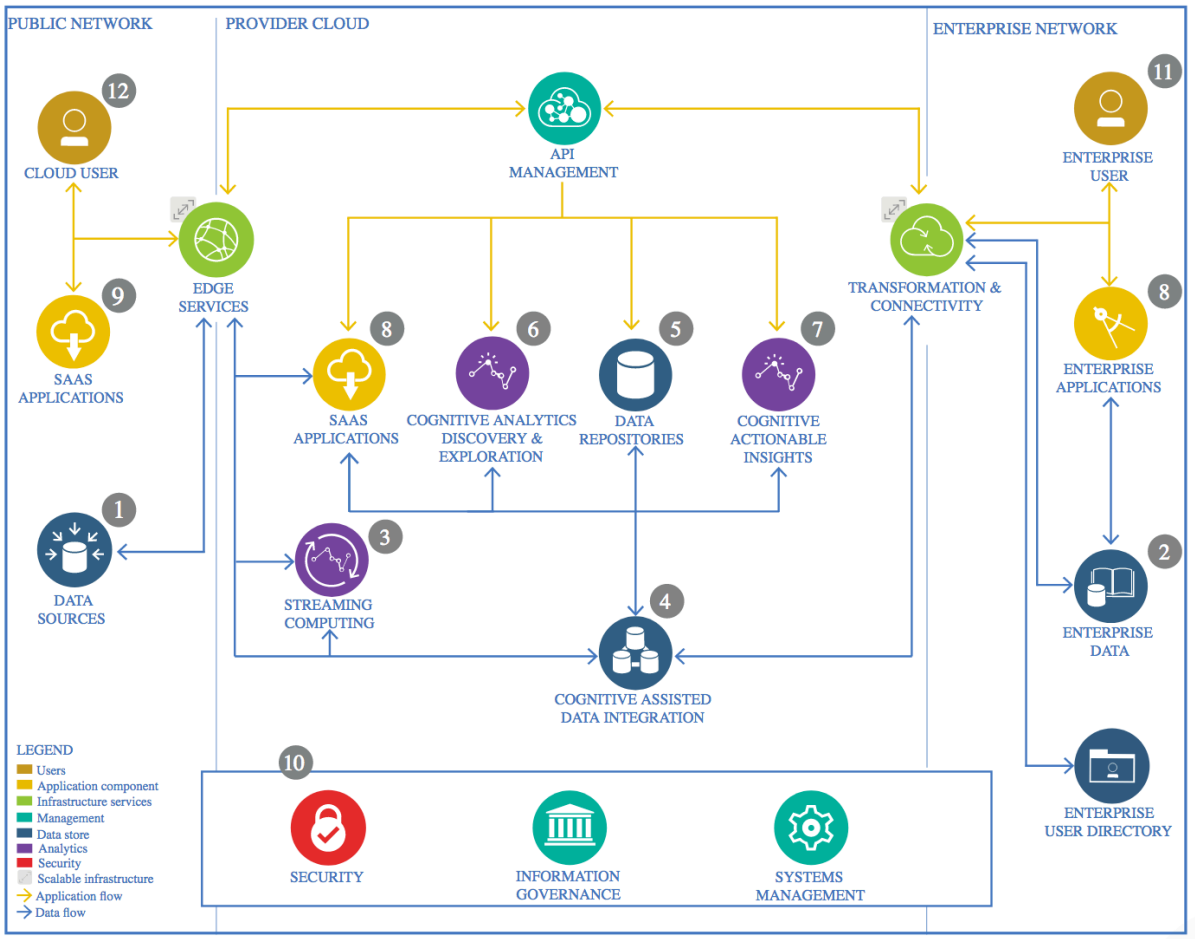
Sentiment Analysis for Financial News

The use case of this project is to create a model that is capable of detecting the sentiment of business related news' headlines. Such a model can be utilized to get an overview of the market sentiment, or the sentiment of a specific company to help decision making in stock trading.

The output of the model can also be used as an input feature for other purposes such as tracking or predicting the sentiment of news related to some specific company to understand better how the public sees the company.

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

Kaggle provides a dataset called "**Sentiment Analysis for Financial News**" (available at: https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news/metadata).
The dataset is an csv -file, and contains 4837 samples (size 656kB).

1.1.2 Justification

The above mentioned dataset has been chosen for this project since it contains business news titles, is labelled, and publicly available for both, commercial and research purposes. In addition to this, the dataset language is English, and the titles are from high quality sources – not for example based on Twitter tweets or forum conversations.

1.2 Enterprise Data

1.2.1 Technology Choice

Enterprise data has been decided to leave out of this project.

1.2.2 Justification

Justification for not utilizing enterprise data is that there is no need to continuously transfer subsets of enterprise data to the cloud nor current plans on extending the enterprise data model.

1.3 Streaming analytics

1.3.1 Technology Choice

Streaming analytics has been decided to leave out of this project.

1.3.2 Justification

For now, the use case has no need for real time analytics capabilities. Solution such as Spark could, however, be implemented in the future if needed – especially if the data amount increases exponentially.

1.4 Data Integration

1.4.1 Technology Choice

Data integration consist of loading the data, cleaning it, and extracting a new feature. The cleaning consists of lowercasing text as well as lemmatizing it. Also punctuation, stopwords and numbers are removed. Due to the high difference between the total sample amounts among different categories, normalization is also performed. After cleaning the data, the text is transformed into a new feature using tfidf-vectorizer that is suitable for training the model.

For deep learning model, a vocabulary is defined using the training data, which is then vectorized using GloVe word2vec model.

1.4.2 Justification

Cleaning the data is needed for higher quality text analysis and prediction. Especially the removal of stopwords, punctuation and numbers make the vocabulary much more compact, which saves storage space and makes the computation faster. It also removes parameters that could be considered as noise. Normalization is performed to avoid some categories dominating over others.

Tfidf-vectorizer is the standard option used for text vectorization purposes. It is fast and takes into account the "value" of the word instead of just counting the words. Similarly, GloVe is the de facto word2vec model. Such a model helps creating high quality vectors from the sample texts.

1.5 Data Repository

1.5.1 Technology Choice

IBM Cloud Storage has been selected as data repository.

1.5.2 Justification

IBM Cloud Storage is a natural place for storing data for long periods of time. Keeping the data for example in local memory or at local computer would be very impractical.

1.6 Discovery and Exploration

1.6.1 Technology Choice

The used dataset consist of only one feature, news title, as well as its label for supervised learning. In the initial data exploration, focus is on testing the data quality, analyzing variations between samples and categories, as well as understanding the title contents.

1.6.2 Justification

Analysing the data quality is important in order to understand how the model will react on the training. It can also be helpful in understanding the achieved results. For example, mislabeled samples can be considered as noise which will decrease the accuracy. Missing values, on the other hand, can cause failures in the performance, and therefore need to fixed or removed. For these reasons, the data quality analysis will consist of counting the number of rows with missing values, and a manual analysis of random data samples to get an idea of the quality of the labeling.

Data analysis itself helps to understand, what kind of features might be valuable for the modelling, e.g. text length. It also helps to understand if there are differences between different categories when it comes to total sample amounts per category, or title lengths per category.

1.7 Actionable Insights

1.7.1 Technology Choice

Please describe what technology you have defined here. Please justify below, why. In case this component is not needed justify below.

1.7.2 Justification

Please justify your technology choices here.

1.8 Applications / Data Products

1.8.1 Technology Choice

This project is built on IBM Cloud Dataplatform. For modelling purposes, sklearn and pyTorch have been chosen. From sklearn, SGDClassifier, and from pyTorch LSTM were chosen as model types.

The training will consist of 20 % of total data amount. Evaluation is performed using accuracy, F1-score, and confusion matrices. The coding will be implemented as Notebooks.

The model will be deployed in IBM Watson Machine Learning.

1.8.2 Justification

IBM Cloud is a natural choice for this project due to its free usage, Data Storage and Machine Learning possibilities, provided security, as well as easy deployment.

Sklearn is considered a professional, up to date package for machine learning modeling. The creator of this document also has a deep understanding on the package, and it is one of the most popular among data scientists. SGDClassifier was chosen as a model due to its generally good performance for such classification tasks.

Tensorflow and Keras were the original choice for deep learning modeling, but were changed to pyTorch due to having problems updating Tensorflow to version 2. It might be that IBM Cloud does not currently support version 2. pyTorch is considered also professional, and easier than Tensorflow to use, so it can be seen as a good alternative for deep learning purposes. Since the task focuses on text classification, LSTM model was chosen.

The training will consist of 20 % of total data amount. This could be implemented even better using batching but was decided to leave out from this project for simplicity reasons. Evaluation is performed using accuracy, F1-score, and confusion matrices. Accuracy gives a good general idea about the performance of the model for management and public that might not be aware of the less known performance indicators. F1-score, on the other hand, can be considered one of the most extensive parameters for describing the overall performance of the model, while confusion matrix helps understanding which categories are problematic and get misclassified. This is very valuable information also for production – it is more dangerous if positive and negative tend to get mixed while detecting positive or negative as neutral is much more acceptable error.

The IBM Watson Machine Learning was chosen for deployment due to its simplicity, and excellent client capabilities.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

The data will be stored in IBM Data Object Storage, the model will be deployed in IBM Cloud, and the API calls will be secured without access from outsiders. Only the account admin has access to the data.

1.9.2 Justification

IBM Data Object Storage, and IBM Cloud are secure, battle tested platforms, which decreases the security risks significantly. Secured API calls is the de facto way of transporting information nowadays.

Only the admin has a need to see the data stored in IBM Data Storage.