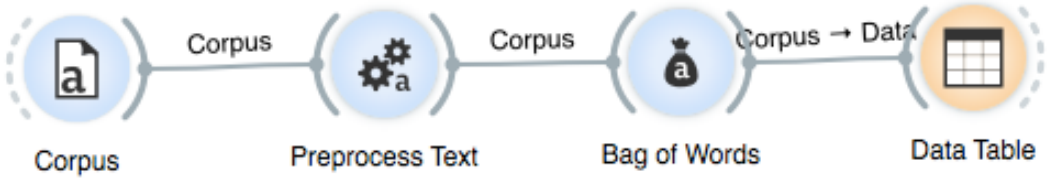
Text e Machine Learning

- Dobbiamo trasformare i testi in rappresentazioni numeriche, ad es. per contare quante volte ogni parola appare nel testo
- Questo approccio si chiama: **Bag of words**.
- Il widget Bag of Words fornisce alcune funzioni di peso (TF, IDF, Binary etc.) e rappresenta ogni testo in un vettore di parole pesate (vd. la matrice)
- BoW è necessario per la modellazione predittiva, l'apprendimento automatica non supervisionato e molti altri metodi

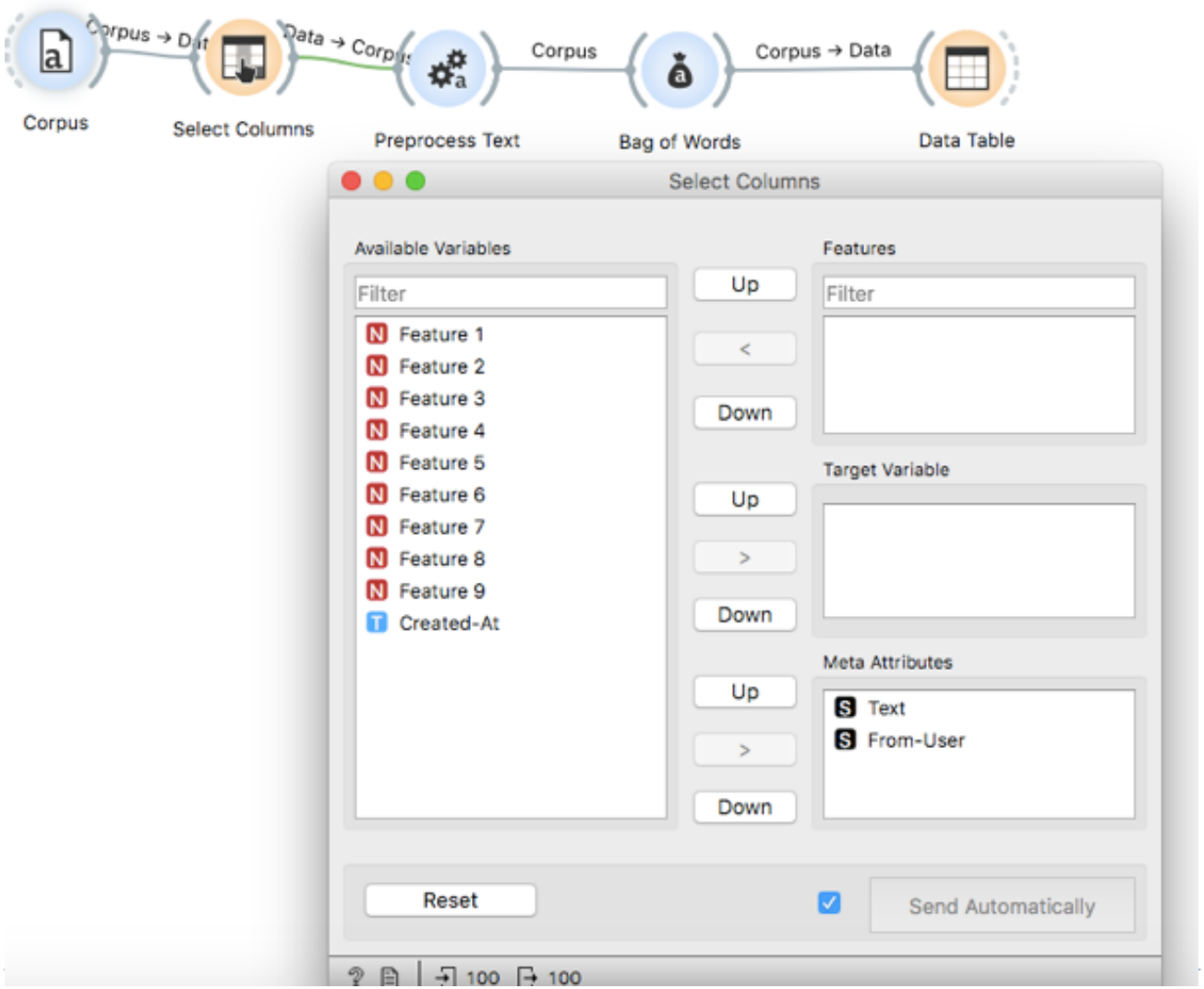


Problema lettura Bag of words

- Per caricare un file excel di tweet e usare correttamente Bag of words si deve:

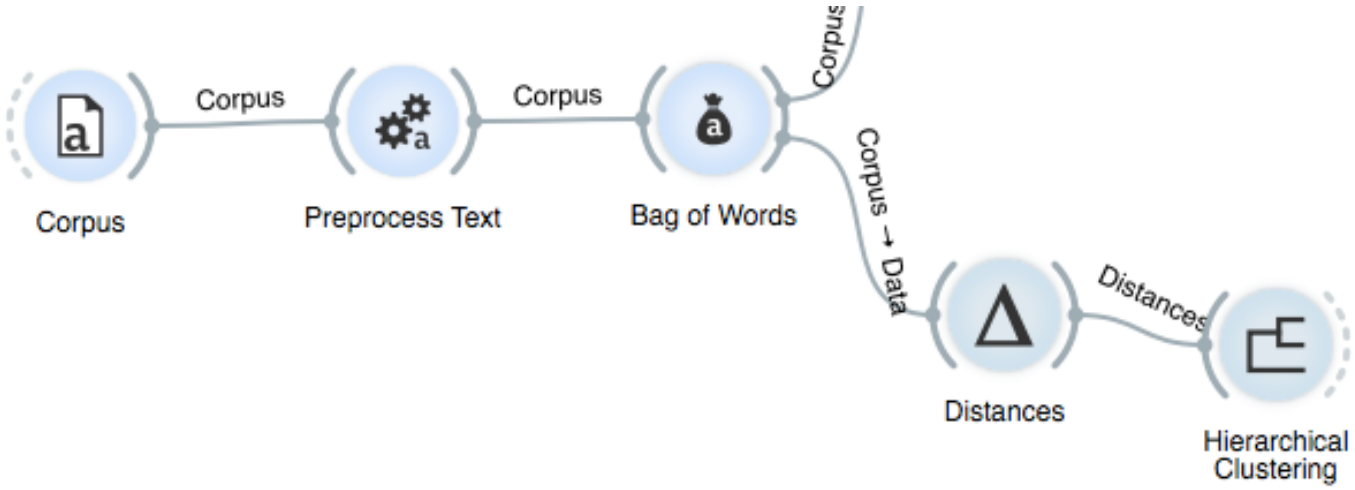
1. Modificare il file excel inserendo due righe con i tipi di attributo e i ruoli per ogni colonna (come spiegato a lezione e nelle prime slide su Orange)
2. Inserire tra Corpus e Preprocess Text un operatore Select Columns
3. Nei parametri di Select Columns spostare tutte le voci presenti nel campo Features a sin nel campo Available Variable (selezionandole e cliccando sulla freccia <)

Problema lettura Bag of words



Text Clustering (Hierarchical)

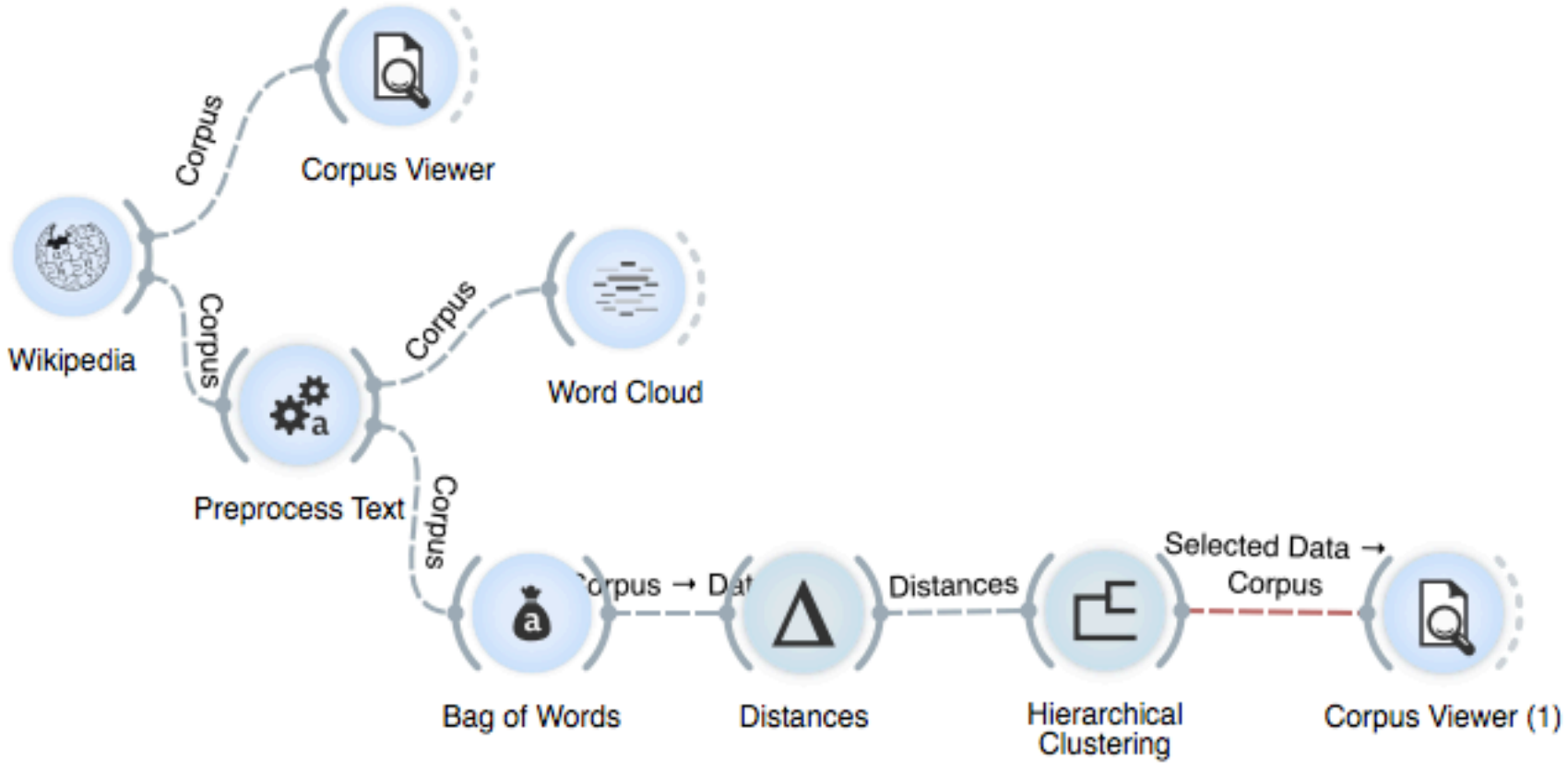
- Con Bag of words possiamo trovare gruppi di documenti simili tra loro
- Collega il widget Distances e usa la distanza del Coseno
- Quindi collega un widget per creare cluster (ad esempio Hierarchical o MDS)



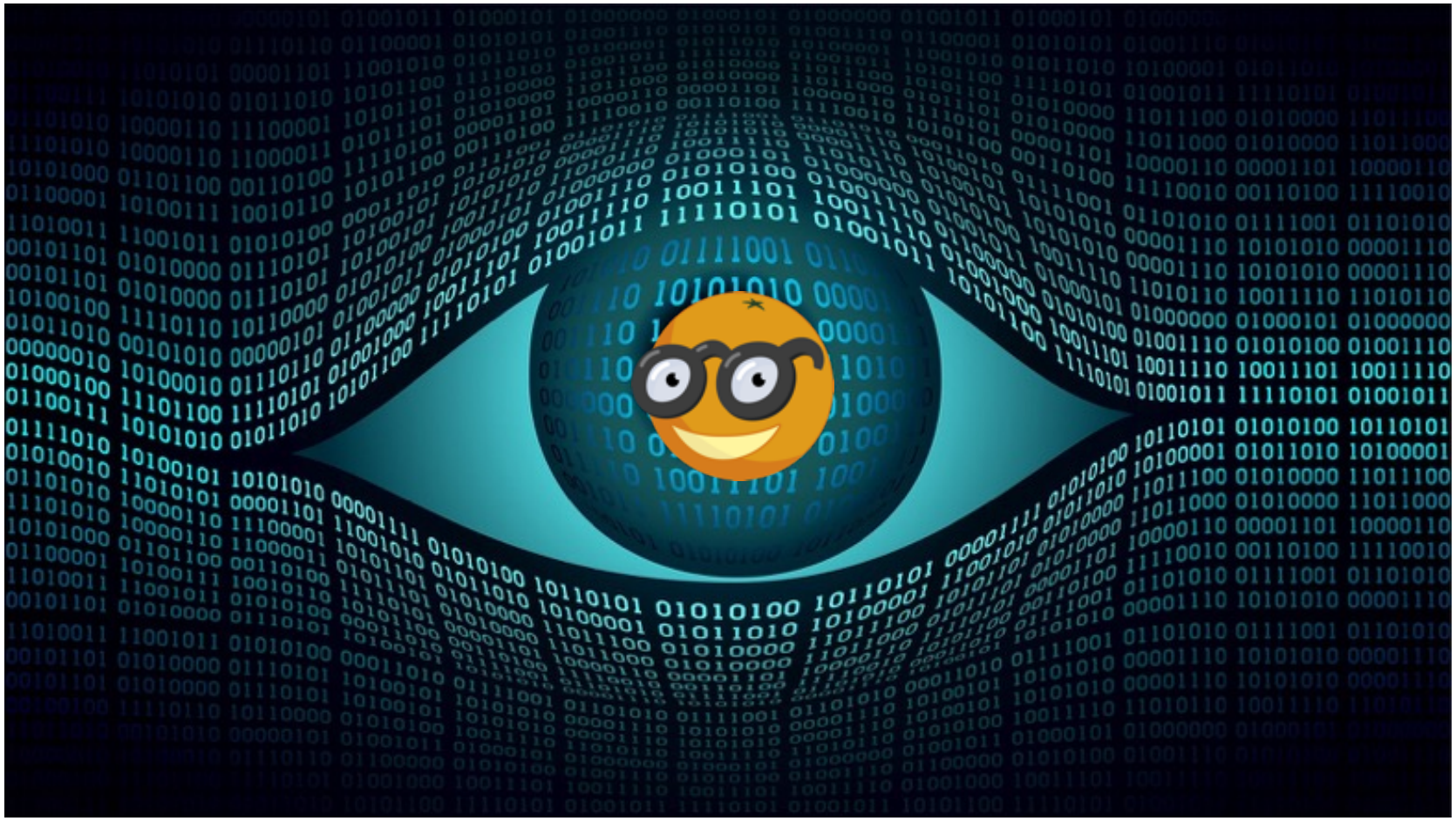
Wikipedia articles Clustering

- Il widget Wikipedia consente di recuperare testi dall'API di Wikipedia e può gestire più query
- Interroghiamo Wikipedia per articoli su, ad es. Euripide e Sofocle (in italiano), e visualizzarli nel Corpus Viewer
- Quindi applichiamo Preprocess Text, Bag of words, Distances e Hierarchical Clustering per trovare articoli simili (visualizzati in un Corpus Viewer)

Wikipedia articles Clustering



Orange Image Analytics



Images Analytics

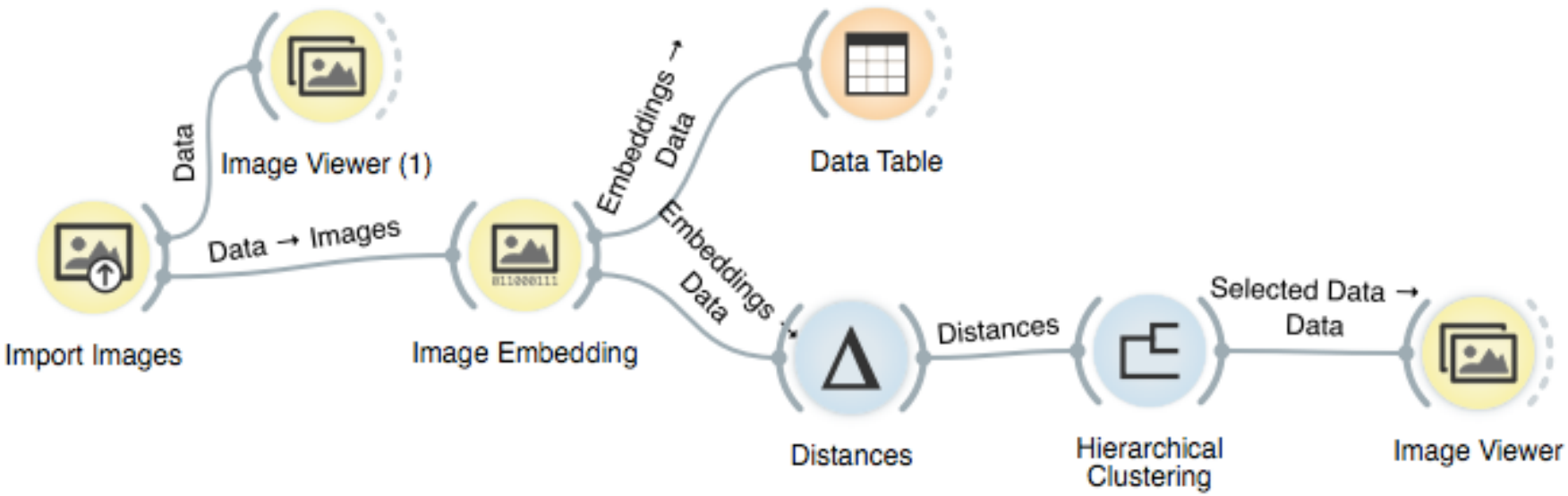
- Abbiamo bisogno del add-on Image Analytics
- Scarica e decomprimi domestic-animals.zip e Paintings.zip
- Usa il widget Import Images per caricare queste immagini e il Image Viewer per visualizzarle
- Per acquisire la rappresentazione numerica di queste immagini, invieremo le immagini al widget Image Embedding (Inception v3)
- Misuriamo la distanza tra queste immagini e vediamo quali sono le più simili: usa Distances e Hierarchical clustering
- Collega Image Viewer per vedere i risultati

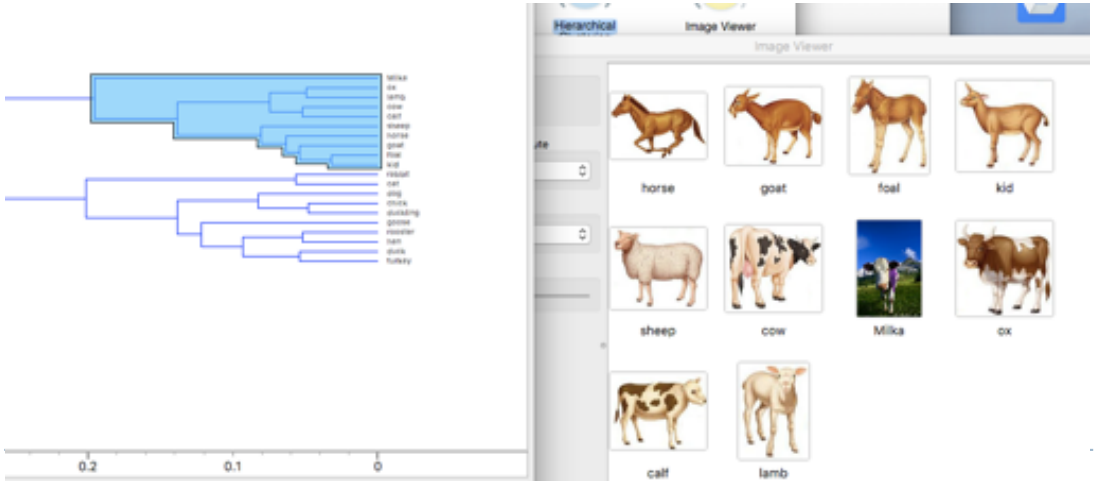
Image Embedding

- Il parametro più importante per il widget **Image Embedding** è l'algoritmo:

  - **SqueezeNet**: Small and fast model for image recognition trained on ImageNet
  - **Inception v3**: Google's deep neural network for image recognition, model trained on ImageNet
  - **VGG-16**: 16-layer image recognition model trained on ImageNet
  - **VGG-19**: 19-layer image recognition model trained on ImageNet
  - **Painters**: A model trained to predict painters from artwork images
  - **DeepLoc**: A model trained to analyze yeast cell images
  - **Openface**: Face recognition model trained on FaceScrub and CASIA-WebFace datasets

Image clustering





Trovare immagini simili

- Usiamo il dataset Paintings.zip
- In Image Embedding scegliere il modello Painters
- Collegare il widget Neighbors, numero di vicini = 2, metrica Cosine
- In Data Table selezionare un'immagine, e.g. Monet, per trovare in Image Viewer se ci sono immagini simili a quella