**1.ML\_Predictive Analytics**

This graph runs the PA automated analytics application on streaming string input data, using a model trained on the Census data set as example. It contains a dataGenerator that mocks census input data, an Automated Analytics Execution operator, a JavaScript operator to throttle the stream for demonstration purposes and a terminal operator to display the prediction results.

**Configure & Run the Graph**

Follow the steps below to run the example graph:

1. Click on the Automated Analytics component.
2. In the configuration panel, set the “modelPath” to the path to your kxen model data generated by PA Automated Analytics. NOTE: This path is relative to the blobs/ directory in the Vora VFlow repository.
3. Then set “modelName” to a model that you want to select.
4. You can optionally set a schema description file if the data that is streamed does not match the data representation of the trained model.
5. In the tool bar, select “save graph” (disk button) and then “run graph” (play button).
6. The Status panel indicates if the graph is running.
7. The graph runs and terminates itself on success.

**2.Classify Images and Videos with Inception**

Classify Images with Inceptioncom.sap.ml.tensorflow.classifyImages

# Classify/label images

This graph applies the inception neuronal network (downloaded from <http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz>) on any image that is fed into the input stream.

The graph can be separated into three main components:

* Stream generation: The file consumer reads an image directory and feeds the image file names into the classify operator.
* Inference: The classification operator applies the inception model on each image and streams the results into the output pipeline.
* Output pipeline: The results of the classification are fed into a terminal, which displays them.

## Prerequisites

* (Only if run in local mode) Operators in this graph use the following libraries:
  + Python 2.7+
  + TensorFlow 1.7.0
  + SciPy 1.0+ & Pillow 5.0+

**3.SAP Leonardo ML**

**Prerequisites**

* InferenceClient operator configuration:
  + **oauthClientId** (type string): Client ID used for the OAuth2 authentication.
  + **oauthClientSecret** (type string): Client Secret used for the OAuth2 authentication.
  + **oauthTokenUrl** (type string): Url for the address where the OAuth2 authentication will be performed.
  + **deploymentAPI** (type list): Url where the status of the server will be checked and the certificate and model host/port will be acquired.
  + **numberResults** (type integer): Maximum number of results to be returned.
  + **modelName** (type string): Model name which will process the input.
  + **signatureName** (type string): Server signature name defined when building the model.
  + **inputTag** (type string):
  + **inputShape** (type list): List of integers with the input dimensions
* A file which will be used as the input to the query locally available. Example: an .jpg image.

**4.Text Analysis**

This is an operator that allows to run various natural language processing tasks on unstructured textual data by using the Vora text analysis server.

Request parameters

* endpoint: Host name and port number of the vora-textanalysis service. For example: ‘vora-textanalysis:2204’.
* taconfig: Text Analysis default configuration. One of: LINGANALYSIS\_BASIC, LINGANALYSIS\_STEMS, LINGANALYSIS\_FULL, EXTRACTION\_CORE, EXTRACTION\_CORE\_ENTERPRISE, EXTRACTION\_CORE\_PUBLIC\_SECTOR, EXTRACTION\_CORE\_VOICEOFCUSTOMER. If empty, LINGANALYSIS\_BASIC is used.
* languages (optional): A list of languages used for language detection specified in ISO 639-1 codes. For example: ‘EN,DE,ES’. If no language is specified, then automatic detection is attempted.
* mime\_type (optional): The type of input documents. Allowed values are ‘text/plain’, ‘text/html’, ‘text/xml’, and ‘text’. The value ‘text’ indicates that the input is one of plain text, HTML or XML. If not set, or if value is ‘text’, document identification and conversion are performed. Note that the analysis of documents in binary formats is also supported, but automatic format detection is used in this case.
* text\_encoding (optional): If the document contains text, this parameter indicates the encoding. For example: ‘UTF-8’. If not set and the MIME type indicates text, encoding detection and conversion are performed.