



Akai Kaeru

AK Analyst User Guide

Version 1.0.5 (January 27, 2023)

Table of Contents

1. [Introduction](#)
2. [Landing Page](#)
3. [AK Analyst Workspace Overview](#)
4. [AK Analyst Actions](#)
5. [Load Data Action](#)
6. [Load Data Lake Action](#)
7. [AK Transformer Action](#)
8. [Aggregate Action](#)
9. [Join Action](#)
10. [Split Data Action](#)
11. [AK Pattern Mining Action](#)
12. [AK What-If Analysis Action](#)
13. [SK Learn Action](#)
14. [Predict Action](#)
15. [Rolling Regression Action](#) (Experimental)
16. [AK Pattern Browser Action](#)
17. [AK Visualizer Action](#)
18. [Launching the Workspace](#)
15. [Useful Tips](#)

Introduction

The AK Analyst is a No-code / Low-Code platform for data science geared toward analytics that involves complex high-dimensional data. The platform features an easy-to-use GUI that facilitates the creation of a wide variety of dedicated analytics pipelines. The pipelines are constructed by simply dragging, dropping, and connecting action widgets that map into a rich set of specially designed data analysis tools.

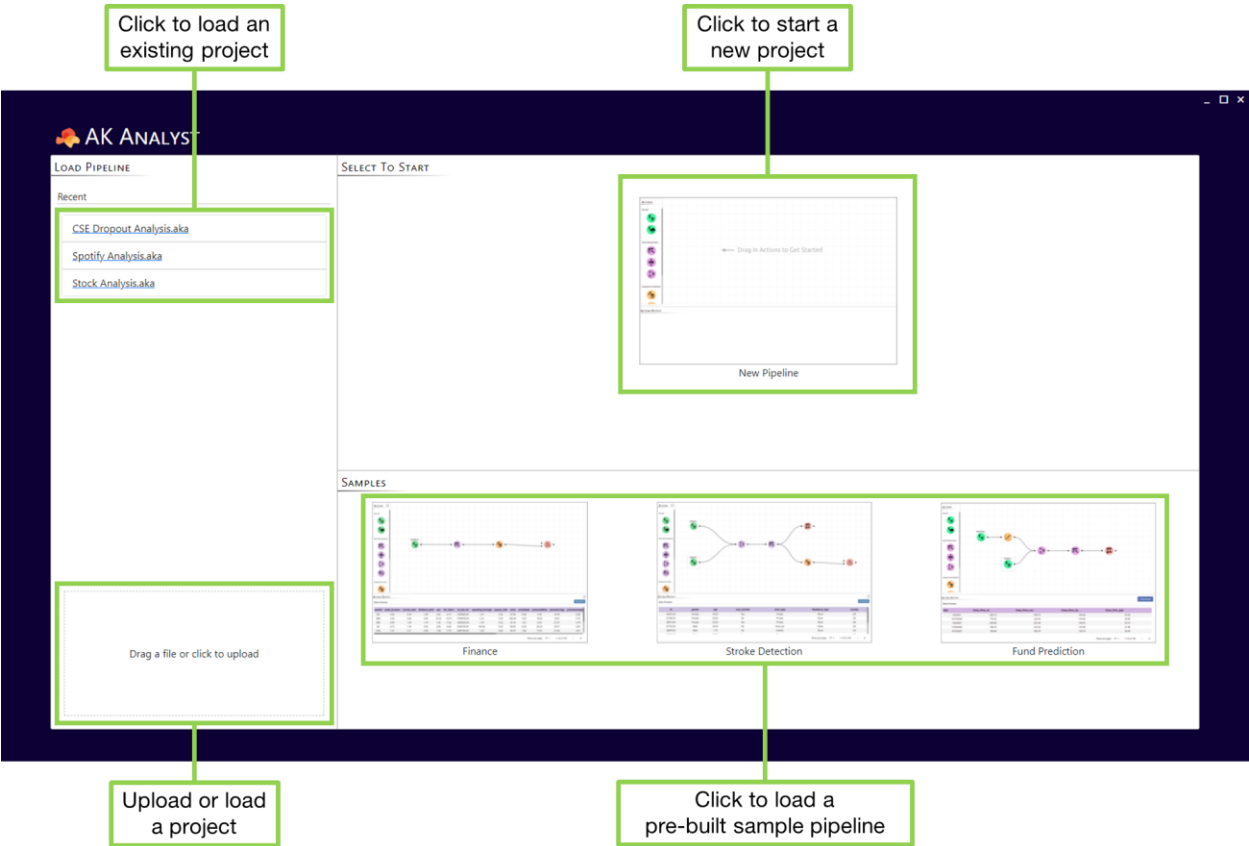
The platform allows you to load, clean, and transform data with visual feedback. It provides a wide set of innovative analysis components that can extract relations in high-dimensional data, identify features that drive a target of interest, and perform predictive analytics with concise explanations of model behavior.

In this guide, we will introduce the AK Analyst platform, explain each action supported by the platform, and how these actions can be combined to create solution-focused data analysis pipelines.

For further information please contact us at info@akaikaeru.com

Landing Page

Upon launching the AK analyst microservice you will be greeted with the landing page. From here, you will be able to (1) start a new project, (2) load an existing project, (3) upload a project, or (4) launch a pre-built sample pipeline. Clicking any of these options will take you to the AK Analyst workspace.



The AK Analyst landing page

AK Analyst Workspace Overview

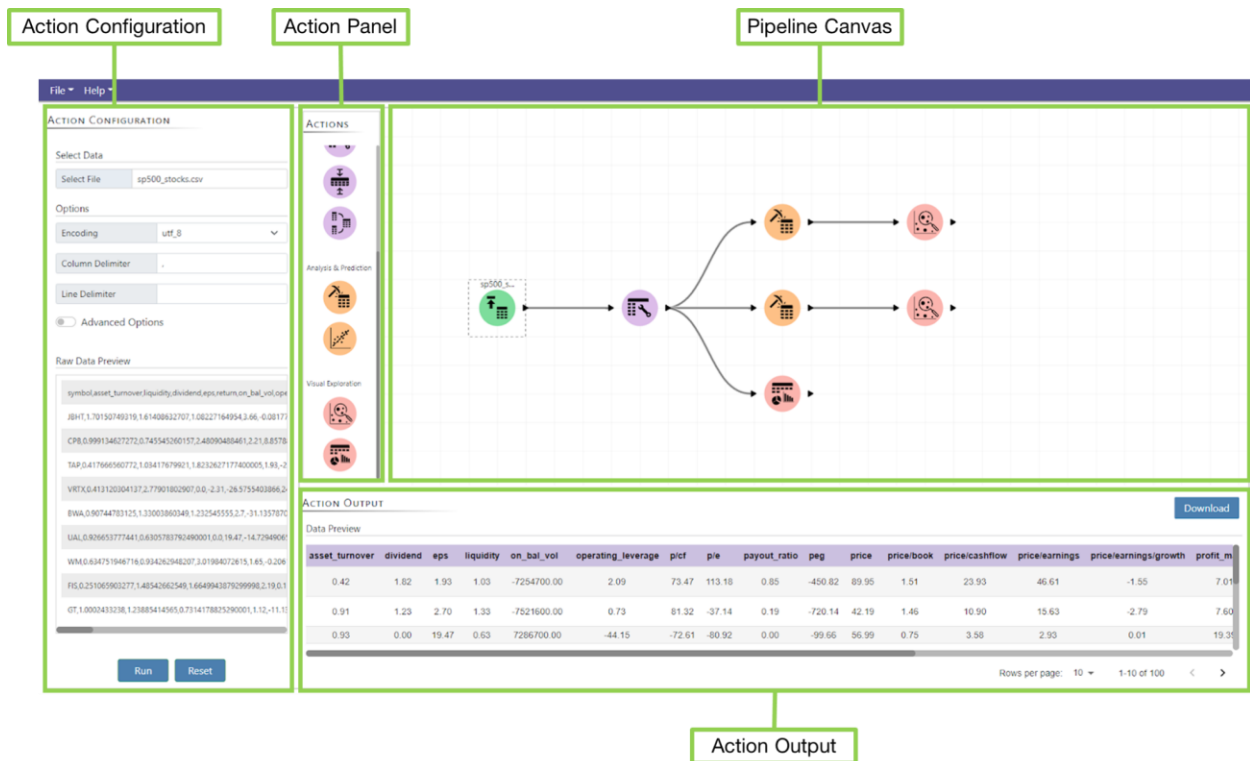
The AK Analyst workspace is an easy-to-use drag-and-drop data analytics pipeline designer. An example of this interface with a complete pipeline already set up is shown below. It consists of four main panels and components:

Action Panel: The action panel contains actions that are components that allow you to perform tasks on your data. They must be dragged into the pipeline canvas and connected to each other to form an analytics pipeline. Connections can be made by holding the mouse down over an action's output port (i.e. right ►) and releasing it after it snaps to another action's input port (i.e. left ►).

Pipeline Canvas: The pipeline canvas is where pipelines reside and can be edited.

Action Configuration: This panel is used to configure each action in the pipeline. Clicking on an action will load its configuration in the panel where it can be set up.

Action Output: This panel displays the output of the other panels. If the output is a data frame you can download that data frame from this panel.



AK Analyst Actions

Actions in the AK Analyst are the core components that allow you to perform tasks on your data. Actions are represented by colored circles with an icon at the center. They also have input and/or output ports to connect them to other actions. The color of an action indicates its category. An action's color changes to gray when it is not ready which indicates that it may not have an input or is not configured correctly.



There are four categories of actions:

- **Input/Output** – Actions for moving data in and out of the platform
- **Data Manipulation** – Actions for performing data manipulations such as column transformations, row aggregations, various cleaning operations, dataset merging, and splitting datasets.
- **Analysis and Prediction** – Actions for performing an analysis of the data and for building predictive models. This includes our proprietary AK Pattern Mining algorithm and What-If Analysis tool.

■ **Visual exploration** – Actions that allow you to visually explore the data at every stage of your pipeline. This includes our AK Browser specifically designed for studying and organizing patterns identified by the AK Miner.

Load Data



Load Cloud Data



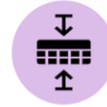
Transform Data



Merge Tables



Aggregate Rows



Split Data



AK Pattern Miner



Model Builder



Predictor



What-If Analysis



AK Pattern Browser



AK Visualizer



Load Data Action



The load data action is one of the first actions that you will use. It allows you to import data into the AK Analyst. It currently supports the loading of structured data files such as delimited text files (e.g. CSV, TSV).

The configuration panel (shown on the right) for this action allows you to specify various file loading parameters:

Encoding type: Encoding to use when reading/writing (e.g. utf-8).

Column delimiter: Delimiter to use to indicate the end of a column.

Line delimiter: Delimiter to use to indicate the end of a row.

Header row: The row at which the header is located.

Skip rows after header: The number of rows to skip after the header.

Escape character: One-character string used to escape other characters.

Comment character: One-character string used to indicate the remainder of the line should not be parsed.

Thousands separator: Character to recognize as the thousands separator.

Decimal character: Character to recognize as decimal point.

NA Values: Strings in the file that should be treated as NaN/NA

Skip Empty Rows: A flag that if set tells the loader if an empty row should be skipped when reading in the file. If it is not set, empty rows are read in with NaN/NA values.

ACTION CONFIGURATION

Select Data

Select File

Options

Encoding

Column Delimiter

Line Delimiter

Advanced Options

Header Row

Skip Rows after Header

Escape Character

Comment Character

Thousands Separator

Decimal Character

NA Values

Skip Empty Rows

Load Data Lake Action



The load data lake action enables importing data from a data lake hosted in the cloud. The configuration panel is very similar to the load data action – with the only difference being the addition of “Datalake Credentials” which is needed to access data stored in the cloud.

The “Datalake Credentials” include:

IP Address: The IP address of the data lake to access.

Username: The data lake username.

Secret Key: The secret key associated with the data lake.

Bucket: The bucket to access.

File Path: File path within the data lake bucket to import into the pipeline.

The remaining “Options” and “Advanced Options” are identical to the load data action.

ACTION CONFIGURATION

Datalake Credentials

IP Address	127.0.0.1:9000
Username	johndoe
Secret Key	*****
Bucket	datalake
File Path	johndoe/sp500_data.csv

Options

Encoding	utf_8	▼
Column Delimiter	,	
Line Delimiter		

Advanced Options

AK Transformer Action



The data transformer action provides a visual interface for transforming and cleaning the data. Upon clicking on the data transformer icon, the Action Configuration panel will update to display a button called “Launch Data Transformer” (see right). There is also an option to use sampling which, when enabled, will use a random sample of the data within the data transformer.

ACTION CONFIGURATION

Sample Options

Use Sample # Samples 1000

Launch Data Transformer

Transformation Summary

No transforms applied.

Upon clicking the “Launch Data Transformer” button, the main display will navigate to the data transformer visual interface. This interface initially contains two panels – the “Attributes” panel (left) and the “Data Preview” panel (right).

ATTRIBUTES

- symbol Nominal
- asset_turnover Numeri...
- liquidity Numeri...
- dividend Numeri...
- eps Numeri...
- return Numeri...
- on_bal_vol Numeri...
- operating_leverage Numeri...
- payout_ratio Numeri...
- price Numeri...
- price/book Numeri...
- price/cashflow Numeri...
- price/earnings Numeri...
- price/earnings/gr... Numeri...

Add Derived Attribute Transform Multiple

DATA PREVIEW

Select an attribute from the left to view details.

symbol	asset_turnover	liquidity	dividend	eps	return	on_bal_vol	operating_leverage	payout_ratio	price	price/book	price/cashflow	price/earnings	price/earnings/growth
JBHT	1.70	1.61	1.08	3.66	-0.08	-4200100.00	37.00	0.23	77.61	2.89	10.29	21.21	1.34
CPB	1.00	0.75	2.48	2.21	8.86	4629500.00	3.62	0.56	50.30	2.98	13.31	22.76	-1.55
TAP	0.42	1.03	1.82	1.93	-2.14	-7254700.00	2.09	0.85	89.95	1.51	23.93	46.61	-1.55
VRTX	0.41	2.78	0.00	-2.31	-26.58	2449000.00	-0.42	0.00	130.36	15.76	-85.91	-56.43	2.13
BWA	0.91	1.33	1.23	2.70	-31.14	-7521600.00	0.73	0.19	42.19	1.46	10.90	15.63	-2.79
UAL	0.93	0.63	0.00	19.47	-14.73	7266700.00	-44.15	0.00	56.99	0.75	3.58	2.93	0.01
WM	0.63	0.93	3.02	1.65	-0.21	1559200.00	1.49	0.93	51.00	1.29	9.26	30.91	-0.76
FIS	0.25	1.49	1.66	2.19	0.12	4639900.00	-4.77	0.47	62.46	0.74	15.63	28.52	-4.19
GT	1.00	1.24	0.73	1.12	-11.14	-19155200.00	-1.94	0.22	34.18	0.80	5.46	30.52	-0.35
GS	0.05	0.00	1.36	12.14	-11.42	-18675800.00	0.00	0.21	187.75	0.00	12.16	15.47	-0.54
NWL	0.81	1.25	1.73	1.29	-11.50	-18509800.00	-0.17	0.59	43.85	2.23	20.87	33.99	-7.65
GE	0.24	0.00	3.25	-0.61	-8.06	171119500.00	0.00	-1.51	28.28	0.00	14.09	-46.37	0.33
GD	0.98	1.17	0.00	9.08	-2.21	-13387400.00	3.72	0.00	142.00	2.33	18.25	15.64	0.70
VAR	0.86	1.83	0.00	4.09	-4.46	1463300.00	-2.40	0.00	71.57	3.22	15.19	17.50	2.58
GM	0.75	1.09	4.20	5.91	-11.47	-65664000.00	-84.31	0.23	32.85	0.42	4.35	5.56	0.02
ALK	0.86	0.92	1.02	6.56	-12.69	-3734400.00	8.15	0.12	78.70	2.14	6.37	12.00	0.25
MAS	1.26	1.33	1.25	1.02	-6.85	-16986500.00	-0.99	0.36	29.70	3.18	14.44	29.12	-0.51
MAR	2.38	0.43	0.00	3.15	-8.38	-4635400.00	3.29	0.00	70.20	6.57	13.10	22.28	0.93
MAT	0.87	1.94	6.69	1.08	-1.51	37988500.00	3.24	1.41	22.71	1.58	10.58	21.03	-0.82
SNI	0.45	1.77	1.67	4.66	12.48	-2722300.00	0.85	0.20	55.14	1.25	8.78	11.83	0.55
XRAY	0.61	2.51	0.47	1.76	-1.35	946100.00	1.86	0.16	61.32	2.19	17.30	34.84	-1.63
SIG	0.91	3.29	0.57	4.75	-2.12	-1010000.00	0.03	0.15	127.20	2.04	35.93	26.78	6.43
XYL	0.78	2.44	0.02	1.87	-2.41	-1271700.00	0.45	0.00	36.64	1.73	14.28	19.59	8.96
TSN	1.80	1.52	0.86	2.95	1.84	23947200.00	5.12	0.14	49.39	0.53	4.01	16.74	0.68
AFL	0.18	0.00	2.51	5.85	-5.66	-13305200.00	0.52	0.27	62.93	0.00	4.00	10.76	-1.08

Rows per page: 25 1-25 of 100

100.00% of the data remains after filtering.
Attribute Count: 27 (27) | Data Item Count: 374 (374)

Initial view of the data transformer visual interface

Clicking on an attribute will update the interface to show more information about the clicked attribute. Specifically, the interface will replace the “Data Preview” panel with 3 more panels – “Attribute Details”, “Notifications”, and “Transforms”.



Visual interface after clicking on the “liquidity” attribute

The “Attribute Details” panel contains relevant information about the selected attribute including summary statistics and a histogram showing the distribution of the selected attribute. From this panel various transformations can be applied via the control panel at the bottom. The data type of the selected attribute can also be changed.

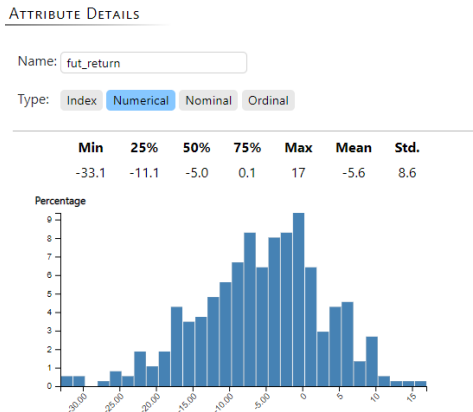
The “Notifications” panel contains information and warnings about the selected attribute. Information is marked with **i** and indicates potentially useful information (e.g. correlated attributes). The **⚠** icon indicates a potentially serious issue with the selected attribute. This can include missing values (e.g. NaNs), collinear attributes, or high cardinality for nominal attributes. The “Resolve” button to the right of the warning provides a hint on how to deal with these issues. Note that these are merely suggestions and it is not required that all warnings be addressed.

Attribute Details

The AK Analyst supports 5 attribute types - Date/Time, Numerical, Nominal, Ordinal, and Index. Based on the attribute type the “Attribute Details” panel appearance changes.

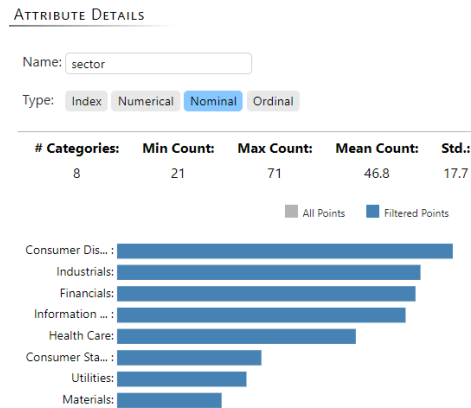
Date/Time & Numerical Attributes

For numerical or quantitative attributes, the AK analyst reports summary statistics values and a histogram showing the distribution of the data.



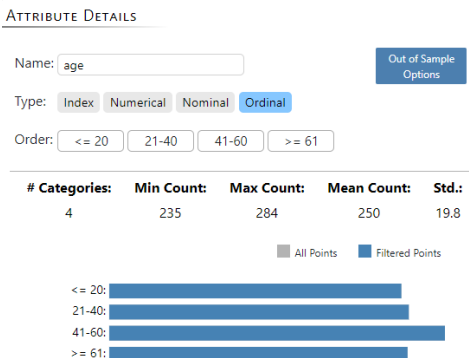
Nominal Attributes

For nominal or categorical attributes, the AK analyst reports summary statistics and a bar chart showing the count of each category. The bars are ordered in descending order of the count.



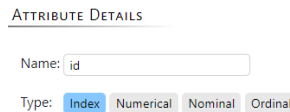
Ordinal Attributes

Ordinal attributes are identical to the nominal attributes but here the user is able to specify the order of the attributes. The bars are ordered according to the user-specified ordering.



Index Attributes

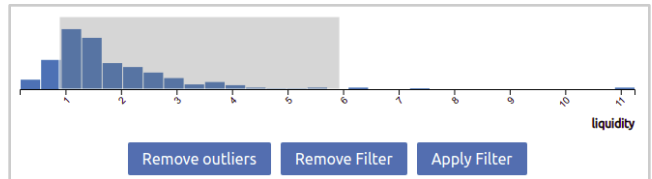
Index attributes are attributes that are used to index the rows of the data set and should have unique values for each row. Thus, we do not show any summary for these attributes.



Numerical Attribute Transforms

The following transformations are available for “Numerical” attributes.

Filter: A filter transformation can be applied by clicking and dragging the edges of the gray box. This will update the histogram to show only those values within the gray box. Clicking on “Remove outliers” will update the gray box to automatically exclude outliers. In both cases Clicking “Apply Filter” applies the filter transform and updates the data.



Clamp: The clamp transform replaces any value below/above “Clamp Min”/“Clamp Max” with the value at “Clamp Min”/“Clamp Max”.

Clamp Min: Clamp Max:

Normalize: The normalize transform normalizes the data between “Lower Bound” and “Upper Bound.”

Lower Bound: Upper Bound:

Log: The log transform applies a log transformation with the specified base.

Base:

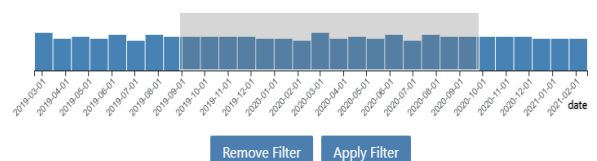
Custom: The custom transform provides a way to enter python code to perform custom transformations.

Enter custom python operation:
liquidity =
x[x<10] = x[x<10] + y[y<10]
Here x and y should be attribute names

Date/Time Attribute Transforms

The following transformations are available for “Date/Time” attributes.

Filter: A filter transformation can be applied to a date/time attribute just like we do for the numerical attribute described above. However, the option to remove outliers is not available for the date/time attribute.



Nominal Attribute Transforms

The following transformations are available for “Nominal” attributes:

Filter: The filter transform allows for inclusive/exclusive filtering by category name. In the figure on the right, for example, clicking on the “Apply” button filters the dataset to only include those data items whose *sector* values are “Utilities” or “Materials.”

Utilities:	[Orange bar]
Materials:	[Orange bar]
Consumer Dis...:	[Blue bar]
Industrials:	[Blue bar]
Financials:	[Blue bar]
Information ...:	[Blue bar]
Health Care:	[Blue bar]
Consumer Sta...:	[Blue bar]

Filter Type: Include | Categories to filter by: Utilities Materials | Apply

Replace: The replace transform provides a way of merging categories into another larger class of categories (e.g. replacing a set of infrequent categories with an “Other” category).

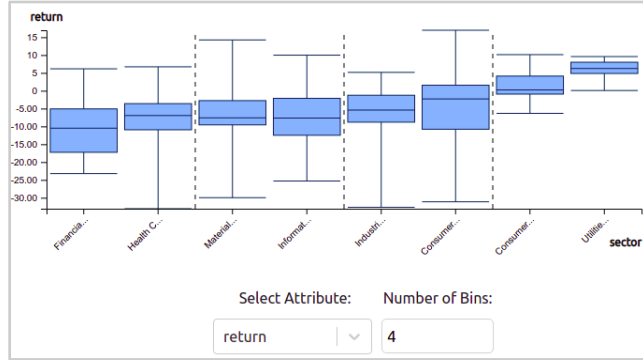
New Category Name: Other = Categories to be merged: Select...

One-Hot Encode: One-hot encoding creates a new set of attributes (i.e. 1 for each category level). If “Bind to” is set to “None”, then each newly created attribute will be binary (i.e. 1 if the data item belongs to that category and 0 otherwise). If “Bind to” is set to another attribute, then the newly created attribute will take on the bound attribute’s value whenever the data item belongs to that category and NaN otherwise.

Applying this transform will create N new columns where N is the number of categories in the attribute(s) shown above. Each new column will have a true value if the data item belongs to that category otherwise its value will be false. If you choose to bind to , another attribute each new column will have the bound attribute's value if the data item belongs to that category

Bind to: None |

Rank: The rank transform provides a way of ranking the categories based on another numerical attribute and bin the categories based on this ranking. In the figure on the right, for example, the `sector` categories are ranked/ordered based on `return`. Consecutive categories are grouped (indicated by dashed lines) into 4 bins (e.g. “Financials” and “Health Care” are grouped in the same bin). This transformation creates a new attribute called `sector_rank4_return` with categories “rank_0”, “rank_1”, etc. based on this binning.



Cell Split: The cell split transform provides a way to split a nominal cell into multiple columns. This is especially useful for splitting arrays stored as strings for example “[‘a’, ‘b’, ‘c’, ‘a’]”. It provides two split types - ordered and unordered.

The ordered method will create N columns where N is the maximum number of items after the split across all rows. The items will be placed in the columns in which they occur. For example, the array above will be split into 4 columns with col1 having the value a, col 2 will have b, col 2 will have c, and col 2 will have a. The unordered method behaves more like one hot encoding. Here columns will be created for each unique item after the split and every row that has an item will have the count of the items in the column for that item. For example, the array above will cause 3 columns to be created with the column labeled a having a value of 2 while columns b and c will have a value of 1. In order to format the split, the user has to specify 3 parameters - the delimiter string, chars to be stripped, and quotation marks to be stripped. In our example, the delimiter is ‘,’ and we would want the brackets ‘[]’ to be stripped while the single quotes around each character would be the quote parameter.

Custom: The custom transform provides a way to enter python code to perform custom transformations not provided elsewhere by the interface.

```
sector = x**2
x[x<10] = x[x<10] + y[x<10]
Here x and y should be attribute names
```

Ordinal Attribute Transforms

Attributes that have “Ordinal” data types contain the same set of transformations as “Nominal” attributes. The main difference is the addition of a drag-and-drop mechanism for setting the order of the categories. The order is specified from left to right (e.g. in the right image “Female” is rank 1 and “Male” is rank 2). For subsequent actions in the pipeline, ordinal categories will be replaced with their integer rank internally (e.g. in the pattern mining action, instead of Female / Male there will be 1 and 2).

ATTRIBUTE DETAILS

Name:

Type: Index Numerical Nominal Ordinal

Order:

[Out of Sample Options](#)

If the data transformer was launched with sampling turned on, then a button will appear on the top right to handle categories that are out of sample. Clicking on this button will trigger a pop-up that allows you to enter additional categories not included in the current sample (see below). In subsequent actions, any out-of-sample categories not specified will be appended to the current ranking in alphabetical order.

OUT OF SAMPLE OPTIONS

You are working with a sample / subset of your dataset. Sampling improves run-time performance. You may see differences in the summary statistics for the full dataset. Some categories in nominal / ordinal columns may not have been sampled. To handle issues related sampling please use the options below:

Sampled Categories:

×

[Add Unsampling Category](#)

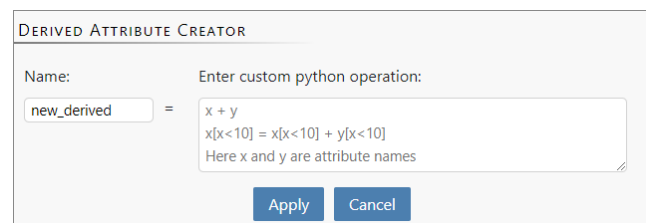
Any additional categories not included in this sample will be appended to the list above in alphabetical order.

[Set Ordering](#) [Close](#)

Out of sample options pop-up

Derived Attribute Transform

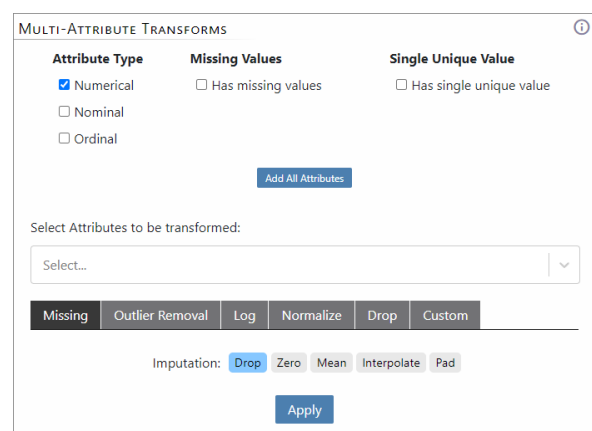
Derived attributes can also be created by clicking on the “Add Derived Attribute” button. This brings up the “Derived Attribute Creator” pop-up. Similar to the “Custom” transform, python code can be entered to create a new attribute based on some combination of existing attributes.



The screenshot shows a dialog box titled "DERIVED ATTRIBUTE CREATOR". It has a "Name:" field containing "new_derived" followed by an equals sign. To the right is a large text area for a "custom python operation" containing the code: `x + y`, `x[x<10] = x[x<10] + y[x<10]`, and a note "Here x and y are attribute names". At the bottom right are "Apply" and "Cancel" buttons.

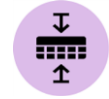
Multi-Attribute Transform

Clicking on the “Transform Multiple” button brings up the “Multi-Attribute Transform” pop-up. This is a convenience feature that allows for applying a single transformation to multiple attributes. It also includes buttons for selecting all columns with missing values or single unique values – allowing the user to handle them all simultaneously.



The screenshot shows a dialog box titled "MULTI-ATTRIBUTE TRANSFORMS". It has three sections: "Attribute Type" with checkboxes for "Numerical" (checked), "Nominal", and "Ordinal"; "Missing Values" with a checkbox for "Has missing values"; and "Single Unique Value" with a checkbox for "Has single unique value". Below these is an "Add All Attributes" button. A "Select Attributes to be transformed:" dropdown menu is currently empty. At the bottom, there are buttons for "Missing", "Outlier Removal", "Log", "Normalize", "Drop", and "Custom". Below these is an "Imputation:" section with buttons for "Drop", "Zero", "Mean", "Interpolate", and "Pad". An "Apply" button is at the bottom right.

Aggregate Action



The aggregate action allows you to combine multiple rows in a table into a single row. To use this action, drag it into the canvas and connect it to any data source.

To be able to aggregate rows you must select a key column that will contain repeating keys that indicate which rows should be combined. You also must select aggregation functions for the remainder of the columns that tell the software how to combine the rows. An example of aggregating salaries by year (key column) is shown below.

AGGREGATOR

Select Column to Aggregate Over

work_year

Select Aggregation Functions for Columns

One-Hot Encoding | job_title x

Bind: None

Method: Mean

Bind: salary_in_usd

Method: Mean

+

Mean | salary_in_usd x

Add Aggregation Function

job_title_Principal Data ...	job_title_Research Scie...	job_title_Research Scie...	job_title_Staff Data Scie...	job_title_Staff Data Scie...	salary_in_usd_mean	work_year
148261.0	0.027777777777777776	246000.0	0.0	NaN	95813.0	2020
239152.4	0.04608294930875576	83003.6	0.004608294930875576	105000.0	99853.79262672811	2021
162674.0	0.012578616352201259	105569.0	0.0	NaN	124522.00628930818	2022

Apply Done

For numerical variables, you can choose between computing the mean, max, min, standard deviation, variance, and the sum total of all the rows. You can also choose to just select the first or last value or use the count (i.e. number of rows) belonging to a key with or without NaN values in the numeric column. In the example above we aggregate the salary_in_usd for each year

For nominal variables, you can choose between selecting the most frequent value or performing a one-hot encoding and aggregating the one-hot encoded columns. With one-hot encoding, you can choose to apply numerical aggregations (e.g. mean, max,

etc.) to them as well as bind other attributes to them and perform aggregations (see data transformer section for more information on binding). In the example above we apply the one-hot coding to the job_title attribute creating columns for each job title. We also add two aggregation types to these columns. The first has no data bound to the new column and we aggregate the one-hot encoded columns by computing the mean. This creates columns with the percentage of jobs for each job_title each year. The second has the salary_in_usd bound to the one-hot encoded columns which are again aggregated with the mean function. This creates columns with the mean salary for each job title for each year.

Join Action



The join action allows you to join two data tables along a column. To join the two tables, they should share one or more key columns. The key columns indicate which rows match in both tables.

To use this action, drag it into the canvas and connect two data sources to it. This is illustrated in the right panel of the figure below. The left panel shows how to set up the configuration for the join action. At least one pair of keys is required. In the example, *id* in both sources serves as the key. More key columns can be specified, if necessary, by clicking the button with the '+' sign. Finally, a join type must be selected to complete the configuration. Four join types are supported – left, right, inner, and outer.

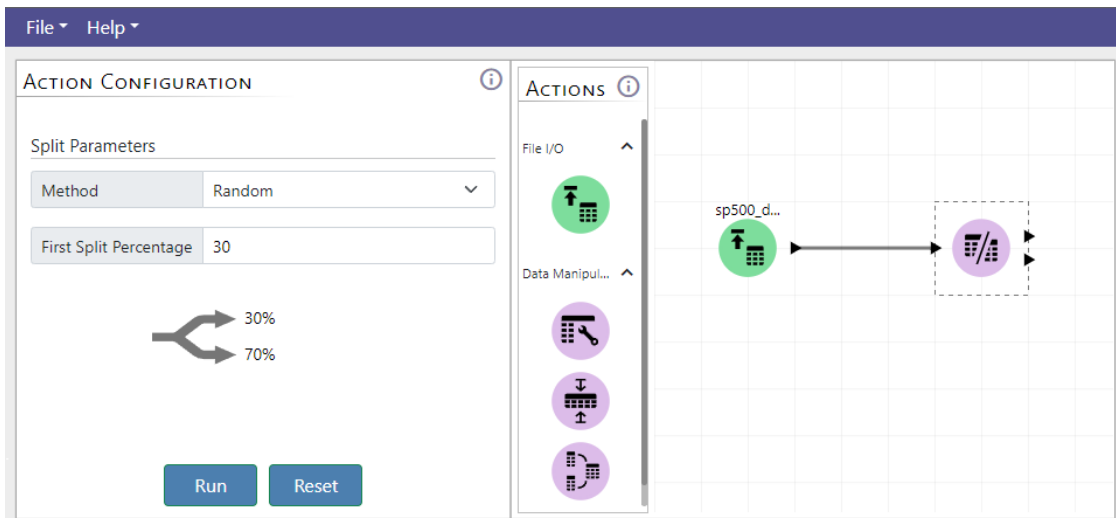
The screenshot displays a software interface with a dark blue header containing 'File' and 'Help' menus. The main workspace is divided into three panels:

- ACTION CONFIGURATION:** This panel is used to set up the join action. It features a 'Select Join Column' section with two input fields. The first field is labeled 'patient_health.csv' and has 'id' selected in a dropdown menu. The second field is labeled 'patient_demographics.csv' and also has 'id' selected. A '+' button is located to the right of the second dropdown. Below this, the 'Join Type' section has four radio buttons: 'Left' (selected), 'Right', 'Inner', and 'Outer'. At the bottom of this panel are 'Run' and 'Reset' buttons.
- ACTIONS:** This panel shows a list of available actions. Under the 'File I/O' category, there are two icons: one with an upward arrow and one with a downward arrow. Under the 'Data Manipulation' category, there are four icons: a join icon (purple), a filter icon (purple), a sort icon (purple), and a table icon (purple).
- Canvas:** The rightmost panel shows a grid-based workspace. Two data source icons, labeled 'patient...', are connected by arrows to a central 'Join Action' icon (purple). The 'Join Action' icon is highlighted with a dashed border, indicating it is the active element.

Split Data Action



The split data action allows you to split a data table into two data tables. This is useful for creating train and test sets for machine learning. To use this action, drag it into the canvas and set up the parameters for the split in the action configuration panel to the left. This is illustrated in the figure below.



This action allows the user to split the data with two methods - Random and In-Order. With the random method, rows are randomly selected for each split. The user provides the percentage of the data they want in the first split and the remaining data will be in the second split. For the in order method, the first N data items are placed in the first split, and the remaining items are placed in the second split. The user can specify N as a percentage of the data or the raw count of data items as shown in the figure to the right.

Split Parameters

Method	In Order
Split By	Absolute Count
Absolute Count	30



AK Pattern Miner



The AK Pattern Mining action uses our proprietary automated, AI-driven process which identifies factors that affect a selected target attribute. It extracts statistically well-defined patterns or groups of data items. A pattern is defined by a set of data items that fall within identified ‘interesting’ ranges of important factors or attributes. These data points perform unusually high or unusually low in terms of a user-specified target variable, and at the same time are defined by a set of common feature value ranges.

The screenshot displays the AK Pattern Miner interface. On the left is the 'ACTION CONFIGURATION' panel with the following settings:

- Sample Options:** Use Sample, # Samples: 1000
- Target Details:** Target: return, Mine Type: numeric
- AK Miner Parameters:** Max Pattern: 100, Threshold: 0.6, Holdout: 1, Min. pattern size: 0.01 (percentage of dataset)

At the bottom of the configuration panel are 'Run' and 'Reset' buttons. The main workspace shows a workflow diagram where a data source 'sp500_s...' is connected to the AK Pattern Miner action, which then branches into three paths leading to other actions. A vertical toolbar on the left lists various actions under categories like 'File I/O', 'Data Manipulation', and 'Analysis & Prediction'. The 'ACTION OUTPUT' section at the bottom right shows the following mining results:

Input Data Properties	
Item Count	374
Feature Count	25

Mined Patterns - Details	
Pattern Count	36
Largest Pattern	284
Smallest Pattern	27
Maximum Feature Count	3
Minimum Feature Count	1

To use this action, drag it into the canvas and connect a data source to it. This action can connect to any action that outputs a data table. An example is shown in the figure above. Next, configure the action as shown in the panel. The configuration parameters are:

Sample Option: You can opt to mine on a random sample of your data or the entire data by configuring the sample option. Sampling is necessary when the data set is too large (i.e. too large to fit in RAM at any one time).

Target: This is the attribute whose performance you want to study based on the ranges of other attributes in the dataset.

Mine Type: *Numeric* or *Binary*. Select *Numeric* if your target is a continuous type numerical variable. Select *Binary* if your target is a two-class variable (e.g. Yes and No, Present and Absent). We do not support multi-class variables (e.g. Red, Blue, Green), however, these can be converted to two-class variables and analyzed by converting all values except one to a different value (e.g. Red and Other).

Max Pattern: The approximate maximum number of patterns to mine for.

Threshold: The minimum effect size for a pattern to be considered 'interesting'. The effect size is the common language effect size for the *Numeric* mine type (default is 0.6) and the odds ratio for the *Binary* mine type (default is 2).

Holdout: Number of holdout sets to test patterns against. Increasing this value reduces false positives but can increase false negatives.

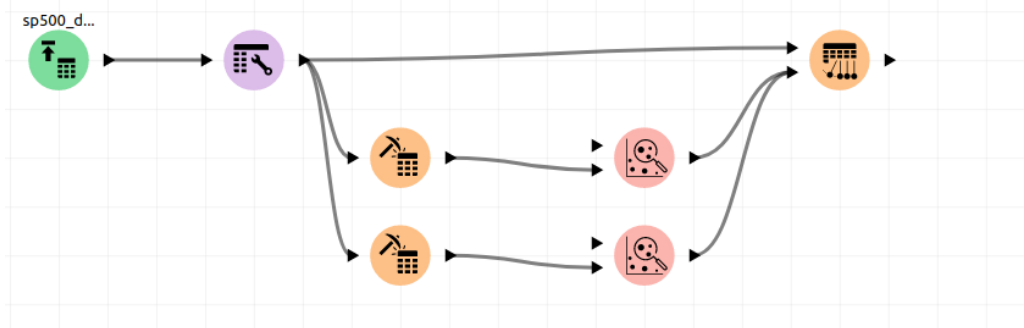
Min Pattern Size: The minimum number of data points a pattern must contain. This is specified as a percentage of the total number of data points.

AK What-If Analysis

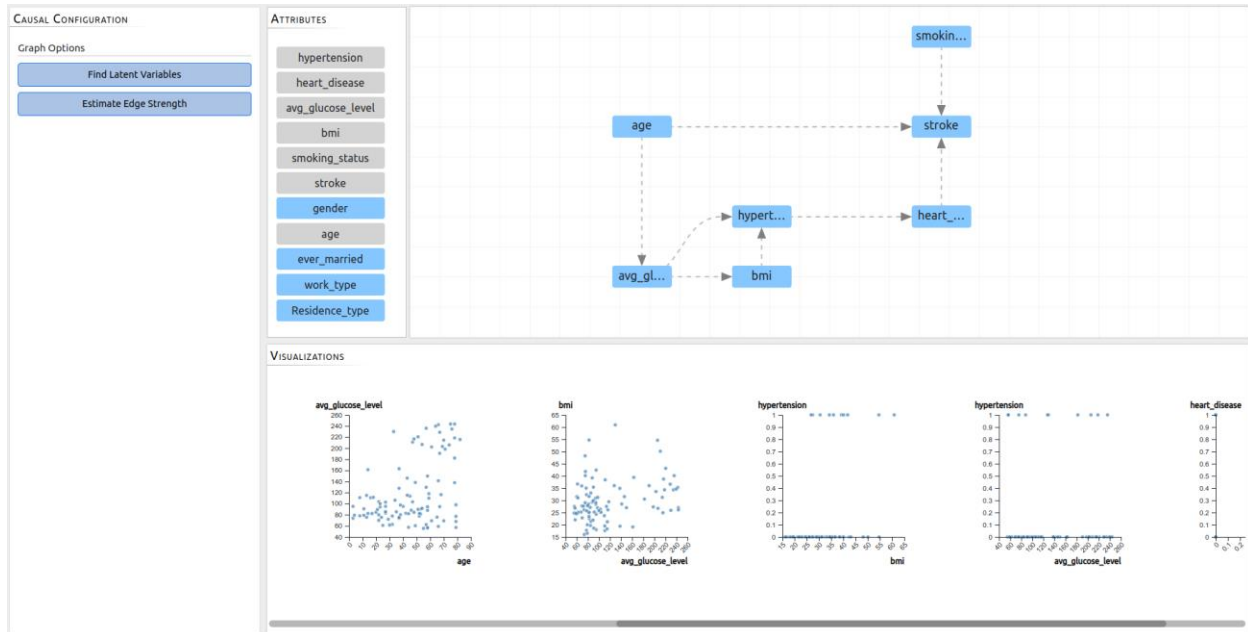


The What-If analysis action allows you to combine insights derived from the other actions in the AK Analyst to model the underlying data generating process. This gives a more complete understanding of the data and allows for answering “What-if” type questions. This can be used to better understand possible interventions (e.g. what happens to sales if I increase ad spending by 10%?).

The figure below shows how the What-If action interacts with the other actions in the AK Analyst. The What-If action takes as input any action that outputs a dataframe (e.g. Load Data Action, AK Transformer Action, etc.) and any number of Pattern Browser actions. The attributes of any patterns added to the output within the Pattern Browser are then automatically added to the graph within the What-If interface (i.e. edges are drawn from the pattern attributes to the target variable the pattern is associated with).



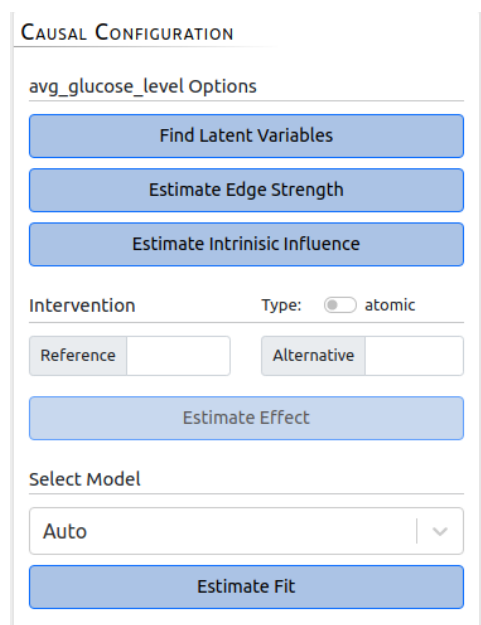
The figure below shows an example of the What-if interface for the stroke analysis sample pipeline. The Causal Configuration panel shows different analysis options. The Attributes panel shows the different attributes in the dataset. These can be dragged into the center panel to create a graph. Finally, the bottom panel shows scatter plots for each edge in the graph (e.g. a scatter plot showing avg_glucose_level v.s. age appears in this panel because of the edge age → avg_glucose_level in the graph).



Causal Configuration: This panel is where different analyses can be performed and What-if questions can be posed. The options shown in the panel change if a node in the graph is selected. Two options are shown in the figure above. “Find Latent Variables” will identify potential latent variables that are in the dataset but not included in the graph. “Estimate Edge Strength” will update the edges in the graph by mapping explained variance to edge thickness.

The figure on the right shows the additional options available when clicking on a node in the graph. “Estimate Intrinsic Influence” will identify the ancestors of the current node which have the biggest influence. This will update the graph to map node size to the intrinsic influence (i.e. higher influence is mapped to larger nodes).

The intervention section within the Causal Configuration panel can be used to estimate the effects of interventions on descendants of the current node. There are two types of interventions that can be applied – atomic and shift. Atomic interventions allow for comparisons between a reference value and an alternative value. These values are synonymous with control and treatment values respectively. The shift intervention can only be applied on



continuous attributes and estimates the effect of a positive or negative shift in the current attribute value. For example, applying a shift of +10 on the *age* attribute can answer the question “what is the effect on heart disease if everyone in the dataset was 10 years older”?

Each of the analyses described above are dependent on a machine learning model associated with each endogenous attribute. The model associated with the current node can be changed via the “Select Model” section of the Causal Configuration panel. The default is “Auto” which defaults to a linear regression model if the attribute is continuous and a logistic regression model if it is nominal. Clicking on “Estimate Fit” will display the R^2 value next to the “Select Model” label.



Model Builder

The Model Builder action provides you with access to the SK Learn and Statsmodel modeling tools. It allows you to build machine learning models supported by the [SK Learn](#) library or [statsmodels](#) library without writing any code.

The screenshot displays the Model Builder interface. On the left is the 'ACTION CONFIGURATION' panel, and on the right is the 'ACTIONS' canvas. Below the canvas is the 'ACTION OUTPUT' section.

ACTION CONFIGURATION

- Sample Options: Use Sample, # Samples: 1000
- Package: Scikit Learn
- Select Model: RandomForestClassifier
- Target Details: Target: Exited
- Predictors: CreditScore, Age, Tenure, Balance
- Cross Validate
- Configuration: n_estimators: 100, criterion: gini, max_depth: [empty], min_samples_split: 2, min_samples_leaf: 1
- Buttons: Fit, Reset

ACTIONS

The canvas shows a workflow: 'Bank Ch...' (data source) → 'RandomForestClassifier' → 'Model Fit Results'.

ACTION OUTPUT

Model Fit Results

Data Properties		RandomForestClassifier Accuracy		Confusion Matrix	
Item Count	3500	Accuracy	1.000	2802	0
Feature Count	8	AUC	1.000	1	697

To use this action, drag it into the canvas and connect a data source to it. This action can connect to any action that outputs a data table. An example is shown in the figure above. Next, configure the action as shown in the panel. The configuration parameters are:

Package: The package or library to use to create the model. The options are SK Learn and statsmodels.

Model: The model to be used. This is a searchable dropdown list whose options depend on the selected package.

Target: This is the attribute that you want to model or predict.

Predictors: These are the list of attributes that will be used to predict the target.

In addition to these parameters, we have the option to add cross-validation and model-specific configuration parameters. To learn more about these parameters please refer to the documentation for [SK Learn](#) and [statsmodels](#).

Predict



The Predict action allows you to use a model created with the model builder action to make predictions on new data or a test set. To use this action, drag it into the canvas and connect a model builder action and a data source. An example is shown in the figure below. Here we split the data into a train and test set with the split data action. The predictor action outputs a data table that contains the predicted value for each data point. The configuration panel allows the user to add the predictor values, target values, and probability for the prediction value to this table.

The screenshot shows a software interface for configuring and running a Predict action. The interface is divided into several sections:

- File Help**: A menu bar at the top.
- ACTION CONFIGURATION**: A panel on the left with "Output Options" and three checked checkboxes: "Include Predictors", "Include Target", and "Include Probability".
- ACTIONS**: A central canvas with a workflow diagram. It starts with a "Bank Ch..." data source, followed by a "Split Data" action, then a "Predictor" action, and finally a "Predict" action. Arrows indicate the flow of data between these components.
- ACTION OUTPUT**: A panel at the bottom showing the results of the action. It has two tabs: "Prediction Results" (selected) and "Error Results".

The "Prediction Results" tab displays three tables:

Data Properties	
Item Count	6500
Feature Count	7

RandomForestClassifier Accuracy	
Accuracy	0.795
AUC	0.720

Confusion Matrix	
4752	369
961	378

At the bottom left of the interface, there are "Run" and "Reset" buttons.

Rolling Regression (Experimental)



The rolling regression action provides a means for modeling time-varying relationships. It is marked experimental, as it is not yet fully supported. We still include it, however, as it can be useful for smaller-sized datasets and will work as expected for most cases.

The configuration panel (see right) for this action include:

Sample Options: For performing rolling regression on a sample of the dataset.

Target: The target variable to model.

Predictors: The list of covariate attributes to model the target variable.

Window: Rolling window size to use to model the target variable.

Confidence Interval: The confidence interval (in %) to include in the output.

Feature Selection: When toggled, a backward elimination method is used to include the significant covariates for each window.

ACTION CONFIGURATION

Sample Options

Use Sample # Samples 1000

Regression Parameters

Target qyld_returns ▼

Predictors return_dji x return_gspc x return_ixic x return_nya x return_rut x return_stoxx x return_vix x x ▼

Window 60

Confidence Interval 95

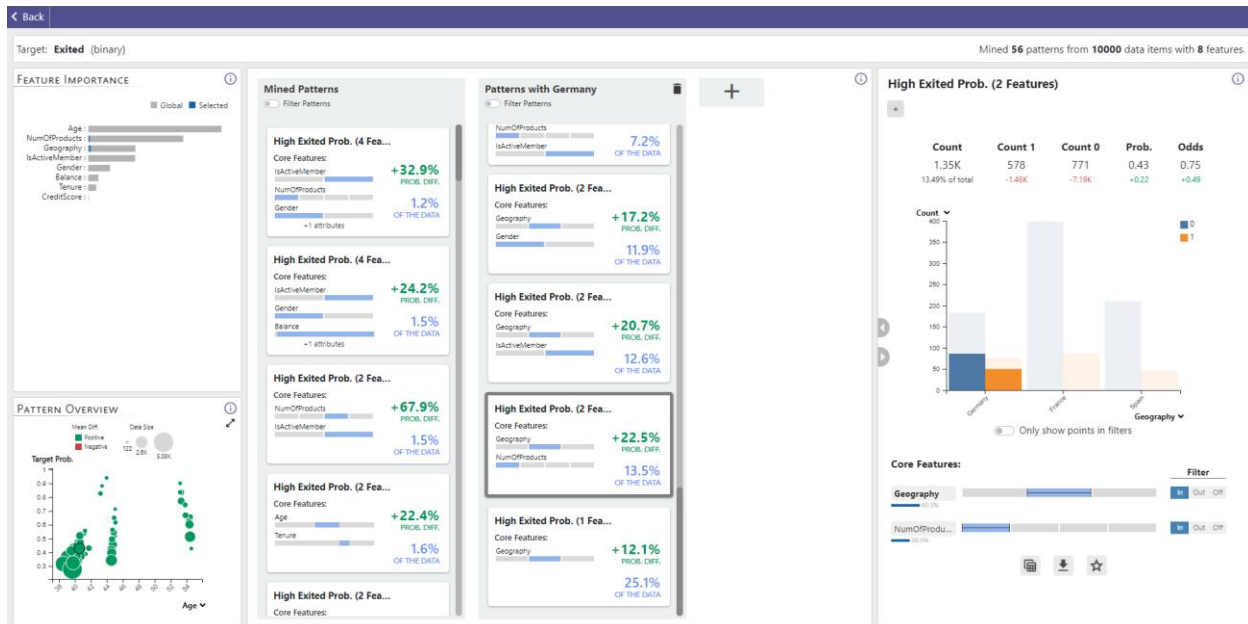
Feature Selection

The rolling regression action takes as input a dataframe and outputs a dataframe that appends the predictors' beta coefficients and lower / upper confidence intervals to the input dataframe.

AK Pattern Browser



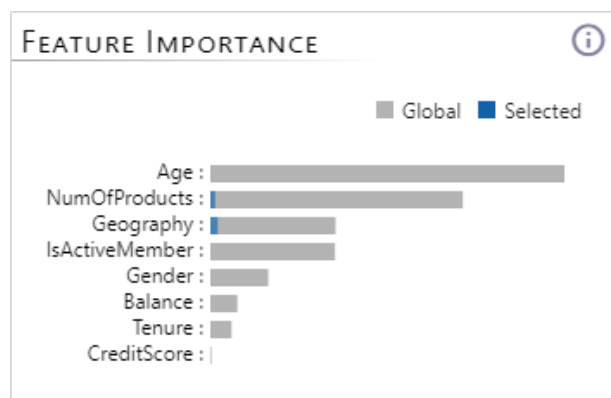
The AK Pattern Browser action brings up a visual interface for exploring a set of patterns mined via the AK Pattern Miner. It includes a high-level overview of the factors or drivers of the specified target attribute as well as detailed information about specific patterns/groups.



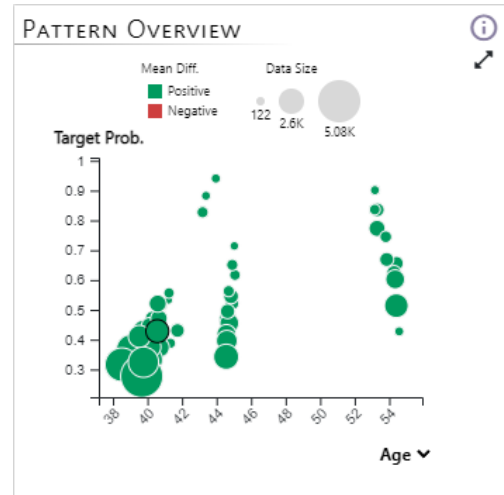
The AK Pattern Browser visual interface

The AK Pattern Browser interface includes the following panels:

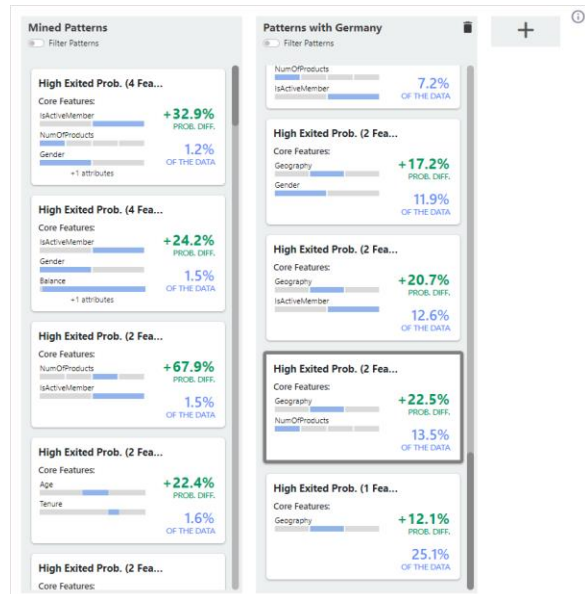
Feature Importance: The feature importance panel shows a bar for each attribute indicating the relative predictive power of each attribute or factor. The gray bars show the global importance while the blue bars indicate the local feature importance (i.e. the features important to a specific pattern/group). In the figure on the right, for example, while *age* is the most important feature globally, it is not important for the selected group.



Pattern Bubble Chart: Each pattern in this chart is represented by a colored circle. The color is based on the target variable (i.e. whether the target variable is unusually **high/low**). The size of the circle is based on the number of points that fall within the pattern. The position is based on the median values for the x and y axes. The x-axis can be set by clicking on the corresponding attribute in the feature importance panel.

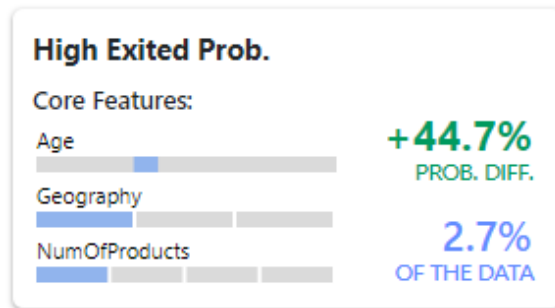


Pattern Lists: The patterns lists are located in the center of the interface and are the main component. It is a set of one or more lists within which patterns are organized. When initialized there will be a single 'Mined Patterns' list with all the patterns identified by the AK Miner. Users can create additional lists based on their requirements and preferences. In the example on the right we see two lists that contain multiple patterns represented by card-like designs (explained below). The first list is the 'Mined Patterns' list and the second is a list created by the user to store and organize all patterns with a driving factor or attribute set to Germany.

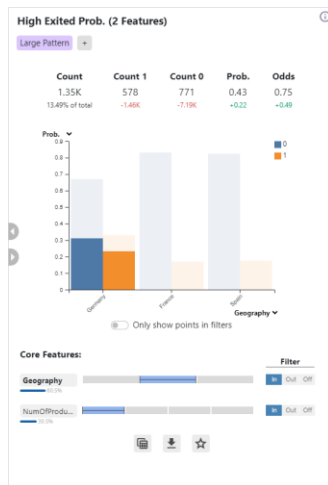


Pattern Card: The pattern card is a visual summary of a pattern that provides the user with the most important information in order to place or organize it in the pattern lists. An example is shown below. The visual summary shows the top 3 core features or factors that drive the target for the data points that belong to that pattern. The features are represented by gray bars which show the entire range of the attributes while the blue segments of the gray bar indicate which parts of the range define the pattern. The card also reports the probability difference (increase/decrease) in the target attribute for the points within the pattern versus those in the entire data set. It also shows the percentage of the dataset covered by the pattern. In the image below, the core

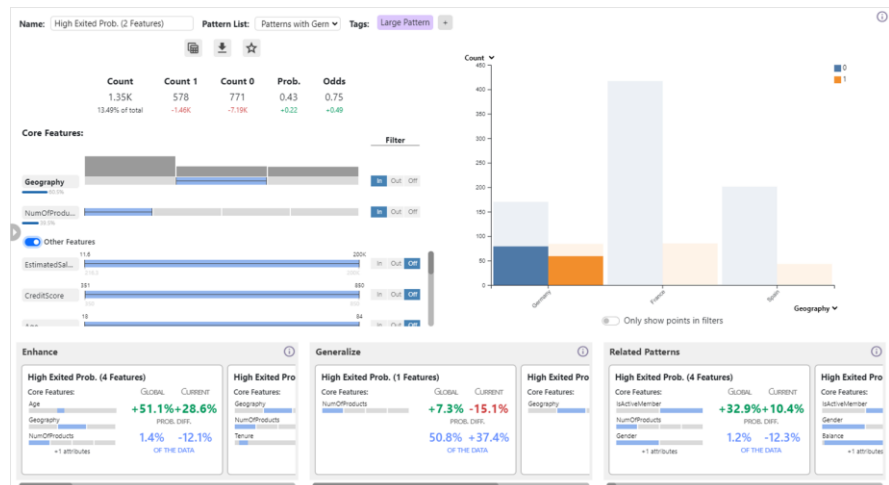
features reported are Age, Geography, and NumOfProducts. Age is a continuous variable represented by a bar with no gaps where a small range in the middle of the entire age range (highlighted in blue) defines the pattern. Geography and NumOfProduct are categorical attributes with categories represented by segmented bars. Each has one bar or category selected (highlighted in blue) which further defines the pattern. The representation also informs us that the pattern increases the probability of the target by 44.7% (green text) and 2.7% of the data or customers fall into this pattern (blue text).



Pattern Detail: Clicking on a pattern card in one of the lists or a circle in the pattern bubble chart causes a pattern to be selected and a panel with its details to pop up. We have two levels of detail - a Peek View and a Detail View. Both are shown below.



Peek View



Detail View

The peek view is used to look into the core details of the pattern while keeping the other components of the interface described above on screen. This is shown in the first

image of this section - the pattern browser interface. The detail view has all the components of the peek view with a few additions. It occupies the entire screen causing all the components except the list containing the selected pattern to be removed from the screen. Each component of the peek and detail view is described below.

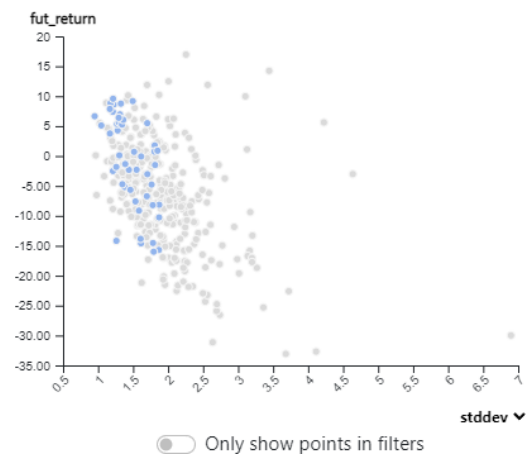
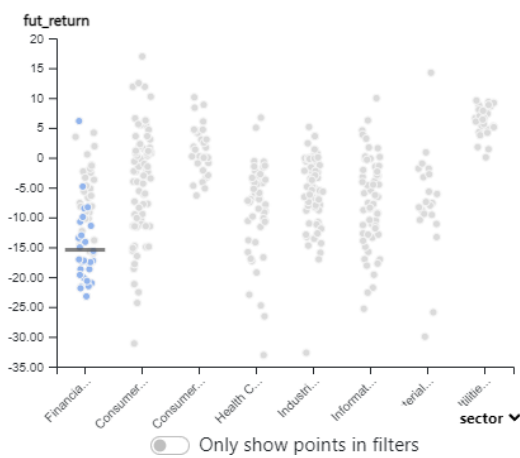
Summary Statistics: The summary statistics panel shows the statistics for the selected pattern along with information on how it compares to the summary statistics for the entire dataset (i.e. whether it is **higher/lower** than the entire dataset). For patterns with numeric targets, we report the Mean, Standard Deviation, Minimum, Median, and Maximum value of the target variable for the data points in the pattern along with how it compares to the entire dataset (values below each stat). For patterns with binary targets, we report the number of data items with the value of 1 (Count 1) and the number of data items with the value of 0 (Count 0) for the target variable for the data points in the pattern. We also report the Probability and Odds of the target being 1 for the data points in the pattern. For both target types, we report the count or number of data points in the pattern and what percentage of the dataset they represent. An example of the summary statistics for numeric and binary targets are shown in the images below.

Count	Mean	Std.	Min.	Med.	Max.
30	3.23	5.56	-14.18	5.3	9.57
8.02% of total	+8.86	-3.02	+18.89	+10.31	-7.38

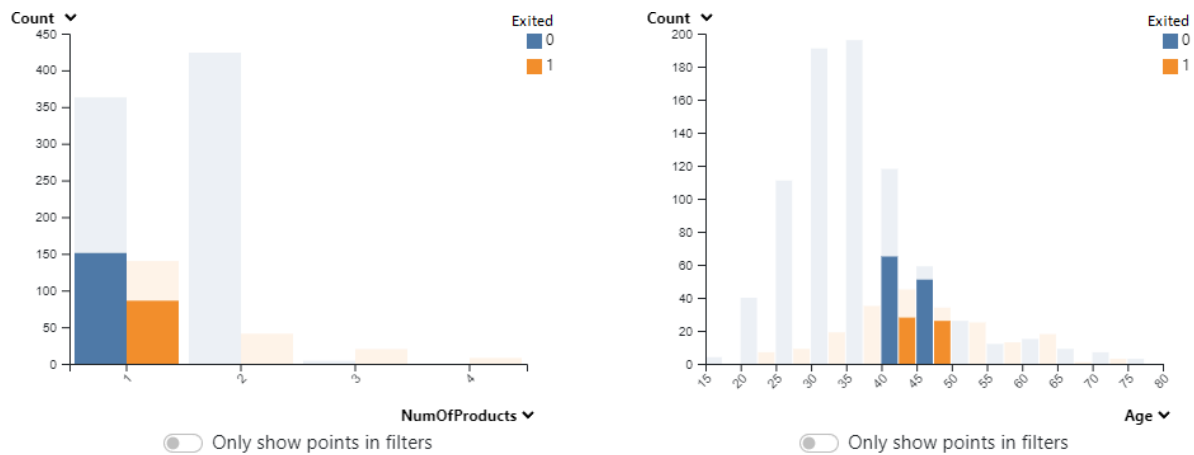
Count	Count 1	Count 0	Prob.	Odds
2.52K	924	1597	0.37	0.58
25.21% of total	-1.11K	-6.37K	+0.16	+0.32

Data Point Visualization: For a better understanding of how the data points in the pattern are distributed and how they compare to the global distribution we present a visualization of the data points. Depending on the target type (numeric or binary) and the selected feature type (numerical or nominal) we represent the data with appropriate visualizations.

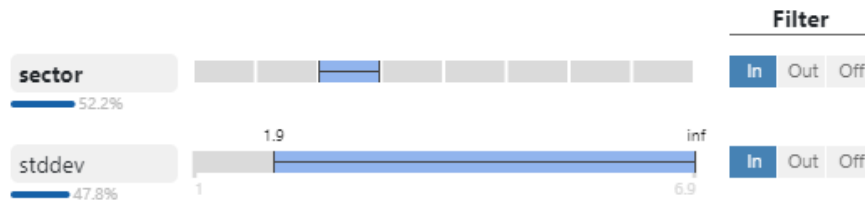
For numerical targets, we use scatter plots as shown in the images below. We highlight selected points in blue; by default, the selected points are points that belong to the pattern. The user can change this selection as described in the following section. In the images below the target *fut_return* is shown on the y-axis of the two scatter plots. The first scatter plot has a nominal variable *sector* on its x-axis thus we jitter the points along the x-direction to avoid overlap and provide the user with a better view of the distribution; we also add a horizontal line to show the mean value of the target for the highlighted points. The second scatterplot has a numerical variable *stddev* on its x-axis, in this case, we do not jitter any points.



For nominal targets, we use a bar chart or histogram as shown in the images below. We use the color blue for data points that have a target value of 0 and orange color for those that have a target value of 1. For points that are outside the current selection, we reduce the opacity of the bar colors thus making the data points within the selection stand out. By default, the selection is points that belong to the pattern. The user can change this selection as described in the following section. In the images below, the target *exited* is shown with blue (*Exited=0*) and orange (*Exited=1*) bars. For nominal values along the x-axis we show two bars (orange and blue) for each category, and for numerical values we create a histogram and show two bars (orange and blue) for each bin.



Feature Visualization: As described previously, each pattern is defined by a subset of different attributes and their specific ranges that drive the value of the target attribute. To provide the user with an informative and easy-to-read view of these attributes and their ranges, we create a visualization based on the bullet chart. An example is shown below, where we see two features - a nominal feature *sector* and a numeric feature *stddev*. Below each feature name, there is a small blue bar with a percentage value; these bars report the extent to which each feature impacts the target in the current pattern. To the right of each feature name there is a large bar. For the nominal feature (*sector*) the bar is divided into 8 segments for each of its eight nominal values sorted in alphabetical order. Hovering the mouse over each bar will show its name. The third segment is highlighted in blue and it has a whisker on it. The whisker indicates that the pattern is defined by that nominal value i.e. data points must have *sector* set to that nominal value to be in this pattern. For the numeric feature (*stddev*) we have a single bar with part of the bar highlighted in blue and a whisker on the bar. Here again, the whisker indicates the range of the *stddev* attribute that defines the pattern i.e. data points must have a *stddev* value within the range to belong to this pattern. To the right of each bar, there are three buttons - in, out, and off - that are used to filter each attribute. Based on the selected filter we highlight parts of the bars in blue. By default, the filters are set to 'In' which highlights the ranges that define the pattern i.e. the ranges shown with the whisker. Setting the filter to 'Out' will highlight the portions of the bar outside the whisker. i.e. it will invert the gray and blue color in the image below. Setting the filter to off will highlight the entire bar thus highlighting the entire range. These filters affect the selection of data points highlighted in the data point visualizations discussed above.



The example in the image above is a visual representation of core features in the 'Peek View'. In the 'Detail View' we add a histogram showing the distribution of a selected attribute and we also show the other features that do not contribute to the pattern. An example is shown below. Note that the whiskers in the bars for 'Other Features' also show the range of the values of data points in the pattern, however, these ranges and features do not define the pattern.



Tags: Users can create custom tags to mark patterns. An example of a tag created for a pattern is shown in the image below. The tag appears as a rectangle with the tag text and color below the pattern title. In the example we can see the tag with the text 'Lowest Return' and a red color appears below the pattern title 'Low fut_return' in the card and peek view. Clicking on the add tag button will pop up a menu by which the user can create, edit, and assign tags to a pattern. In the example below the tag editor panel is shown on the right; here there are two tags that were already created - Highest Return (green) and Lowest Return (red). The red tag is checked as it was assigned to a pattern. Below the tags are a color picker and a text box that the user can use to create/edit tags.

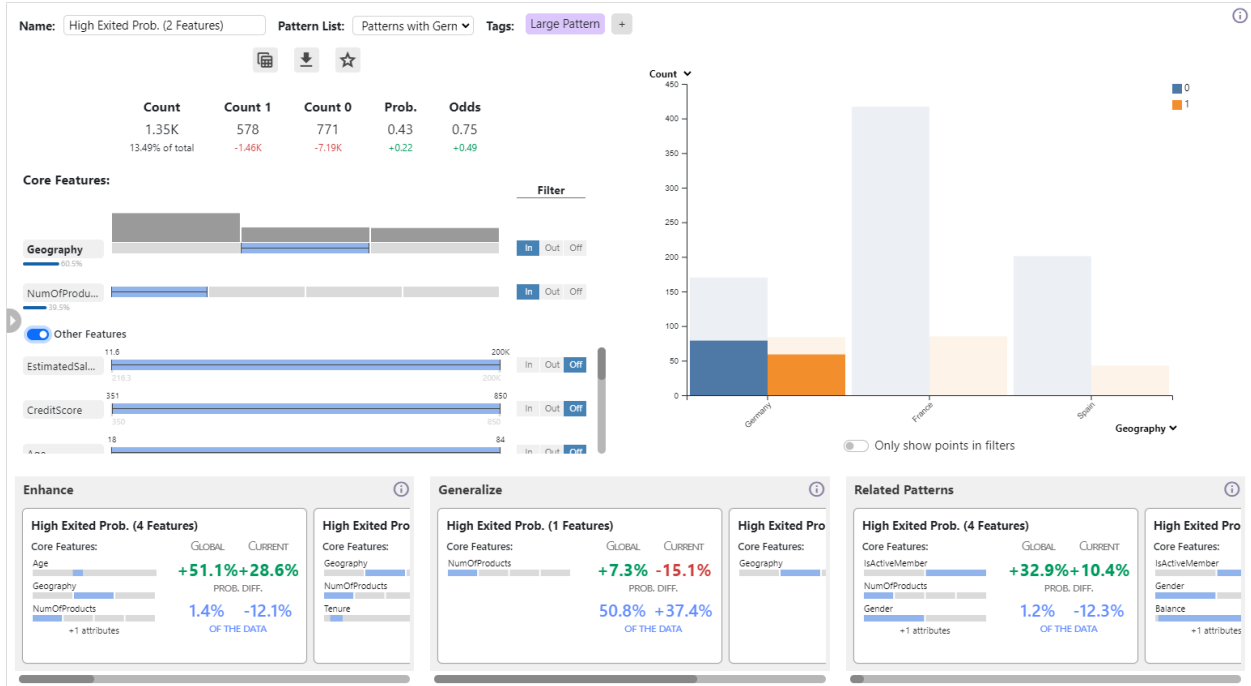


Buttons: The card peek and detail views have three buttons shown in the image below. The first button allows the user to open a data table that contains all the data points that belong to a pattern. The second button allows the user to download this pattern. And the third star-shaped pattern allows the user to ‘star’ a pattern which adds it to the output of the pattern browser action. The output can be used by other actions such as the what-if tool.

Detail View Pattern Lists: The pattern detail contains three additional horizontal lists that contain other patterns that were mined. An example is shown in the image below with the lists toward the bottom of the interface. The first list is the ‘Enhance’ list which contains patterns that are enhancements of the current pattern. An enhanced pattern is one which contains the same core features as the current pattern and its ranges fall within the ranges of the current pattern, and it can also have additional (enhancing) core features. The second list is the ‘Generalize’ list which contains patterns that are generalizations of the current pattern. A generalized pattern is one which contains the same core features or a subset of the core features of the current pattern and its ranges encompass the ranges of the current pattern.

The third list is the ‘Related Patterns’ list which contains patterns that are related to the current pattern based on the data points they contain. Related patterns are patterns

that may not have the same core features or core feature ranges but contain a significant amount of identical data points i.e. they share a significant number of data points.



AK Visualizer Action



The visual explorer action allows you to visually explore data in the AK analyst. This action can connect to any action that outputs a data table. The action currently enables two visualizations - scatterplot and line chart.

Scatterplot

The scatterplot can visualize up to four variables simultaneously. To use the scatterplot, select it as the base chart and then configure the parameters as shown in the figure below. The parameters that can be selected are:

X Attribute - The attribute to which the x-axis maps.

Y Attribute - The attribute to which the y-axis maps.

Color - The color of the scatterplot points. Note this will be overridden if a color attribute is selected.

Radius Attribute - The attribute to which the radius size maps. This value must be numerical.

Radius Size - The radius of the scatter plot points.

Radius Size (Max) - The maximum radius of the scatter plot points. This is only used if the radius attribute is selected and it must be larger than radius size.

Color Attribute - The attribute to which the color of points maps. If a numerical value is selected a single color scale is used ranging from white to that color. If a nominal variable is selected then a multi-color categorical color scale is selected.

CONFIGURATOR

Base Chart: Scatterplot

Scatterplot Options

X Attribute: price

Y Attribute: return

Color: [Blue color swatch]

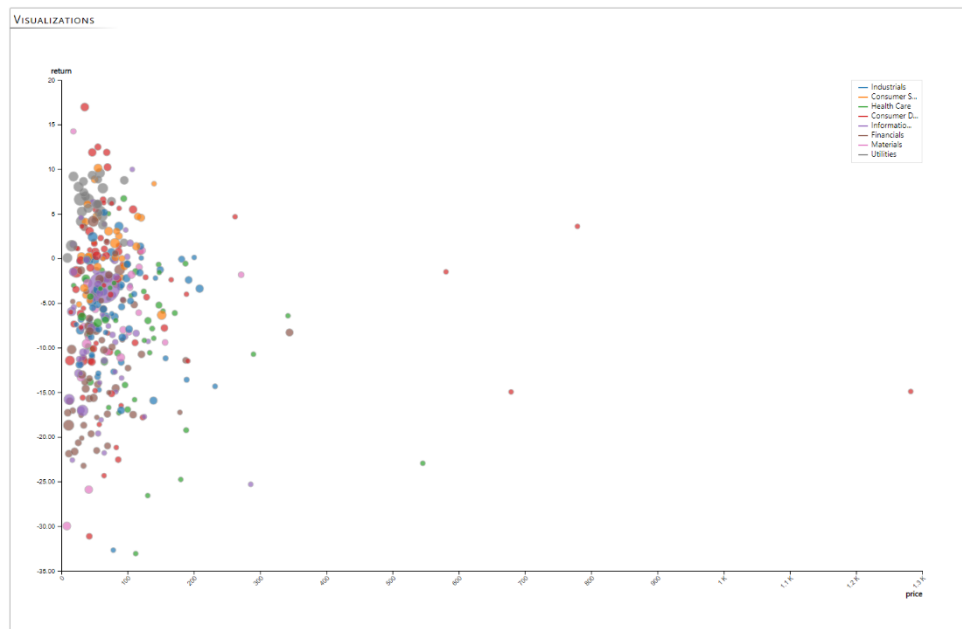
Radius Attribute: dividend

Radius Size: 4

Radius Size (Max): 25

Color Attribute: sector

Example: An example of a scatterplot created with this action is shown below. Numerical attributes “price” and “return” have been mapped to the x-axis and y-axis respectively. Most of the items are concentrated to the lower end of the “price” range while the “return” is varying more. The numerical attribute “dividend” has been mapped to the radius of each point/circle – larger circles indicate bigger dividends. And color has been mapped to the nominal attribute “sector” which indicates which industry sector each data item (stock in this case) belongs to.



Line Chart

The line chart visualizes multiple attributes that change with a time attribute or something similar with lines. To use the line chart, select it as the base chart and then configure the parameters as shown in the figure below. The parameters that can be selected are:

X Attribute - The attribute to which the x-axis maps. For a line chart, this is most often a variable representing time or similar.

Line - You can add multiple lines with the following parameters for each line.

Y Attribute - The attribute to which the y-axis maps.

Color - The color of the line.

Lower Bound Attribute - The attribute which serves as a lower bound for the y-attribute.

Upper Bound Attribute - The attribute which serves as an upper bound for the y-attribute. When both the upper and lower are selected an interval will be highlighted around the line based on their values.

Marker Type - The shape assigned to a marker along the line.

Condition Join - The logical operator used to join multiple conditions that determine if a marker is plotted.

Condition - The condition used to determine whether a marker is plotted. The condition consists of two attribute names and a logical operator. For example, $x > y$ where x and y are attributes in the data.

CONFIGURATOR

Base Chart: Line Chart

Line Chart Options

X Attribute: date

Line 1

Y Attribute: Close_Price_qyld

Lower Bound Attribute: None

Upper Bound Attribute: None

Marker Type: Square

Condition Join: OR

Condition: < Close_P

Condition: > Close_P

Add Condition

Line 2

Y Attribute: Close_Price_qyld

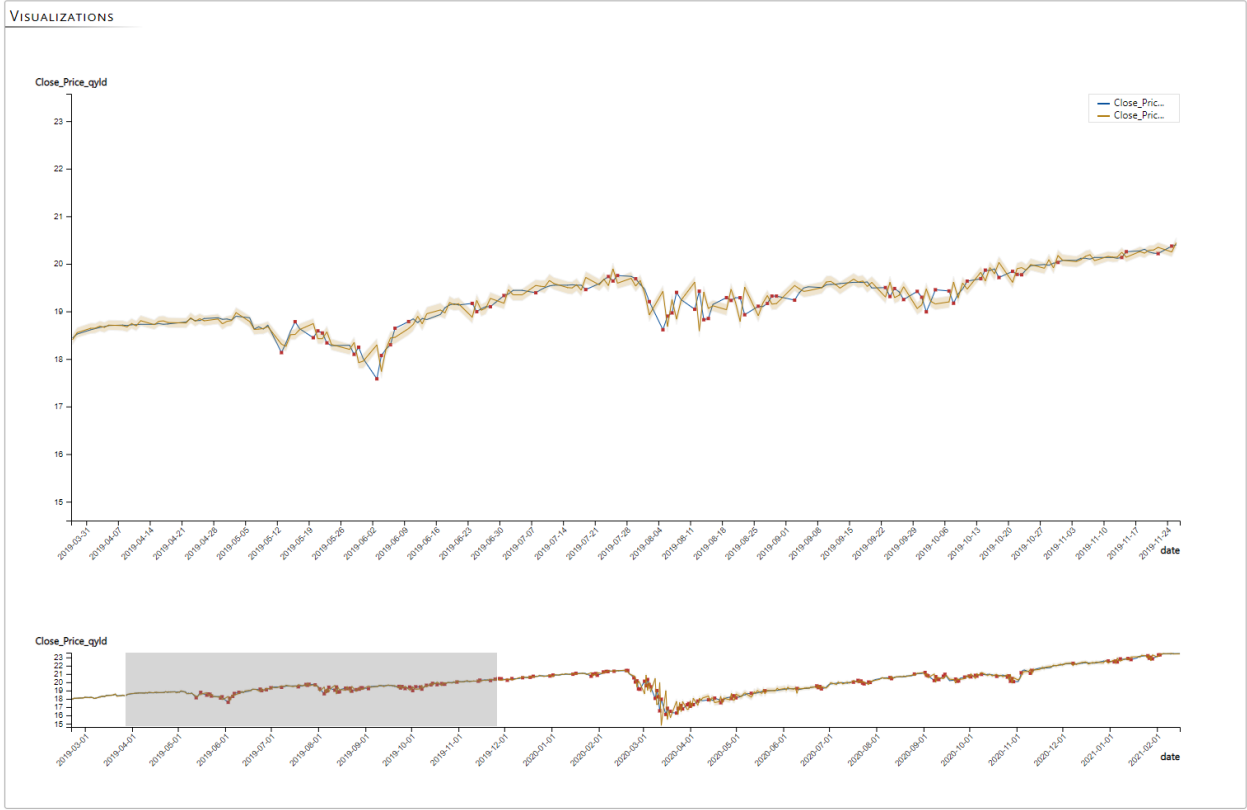
Lower Bound Attribute: Close_Price_qyld_ci_lb

Upper Bound Attribute: Close_Price_qyld_ci_ut

Marker Type: None

Add Y Attribute

Example: An example of a line chart created with this action is shown below. Here dates have been mapped to the x-axis and a close price of a fund qyld (Close_price_qyld) has been mapped to the y-axis. Two lines have been added to this plot – the actual close price (blue line) and a predicted close price (orange). Additionally, a confidence interval for the predicted close price has been added using the lower and upper bound attribute fields and is displayed as an orange halo. Markers have also been added using the marker and condition configuration parameters. These markers are red squares indicating when the actual price broke the confidence interval i.e. the price was lower or higher than the lower bound or upper bound respectively.



Bar Chart

X Attribute - The attribute to which the x-axis maps. For a bar chart, this is a nominal attribute.

Y Attribute - The attribute to which the y-axis maps. By default, it maps to None otherwise a numerical attribute can be mapped to the y-axis.

Y Agg. Function - The method by which the y-attribute is aggregated for each category of the nominal attribute on the x-axis. When the y-attribute is None, the options are count or probability which is the percentage of the data set the category covers. When the y-attribute is a numeric attribute, the options are mean, median, min and max.

Color - The color of the bar.

Color Attribute - The attribute that maps to different color bars that further divide a nominal attribute on the x-axis. This can only be a nominal attribute. When this attribute is set, it will create a grouped bar chart. Each nominal attribute along the x-axis will have N bars where N is the number of categories present in the color attribute

CONFIGURATOR i

Base Chart Bar Chart ▾

Bar Chart Options

X Attribute Geography ▾

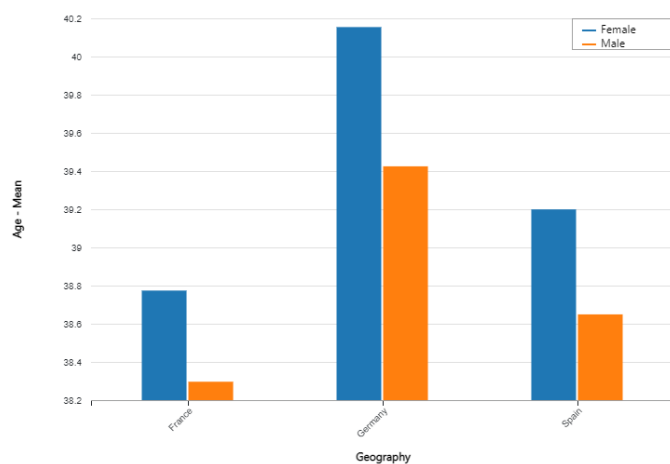
Y Attribute Age ▾

Y Agg. Function Mean ▾

Color

Color Attribute Gender ▾

Example: An example of a bar chart created with this action is shown on the right. Here attribute geography is mapped to the x-axis and age is mapped to the y-axis. The y-aggregation function is set to mean and the color attribute is set to gender. Thus the bar chart shows the mean or average age of females (blue) and males (orange) for 3 different countries.



Pie Chart

Slice Attribute - The nominal attribute that maps to the slices or segments of the pie chart. Each category in the attribute will have its own slice.

Angle Mapping - The value to which the angle maps. By default it maps to count i.e. the number of items in each category. It can be changed to equal size so that each slice has the same angle.

Radius Attribute - The attribute to which the radius of the slice maps. By default it is None. It can be changed to a numerical attribute. In this case, the radius of each slice will be determined by the aggregated numerical value of the items in that slice. This turns the chart into a circular bar-like chart.

Radius Agg. Function - The method by which the radius attribute is aggregated. When the radius attribute is None, the options are disabled and a fixed radius is used. When the radius attribute is set, the options are mean, median, min and max.

Inner Radius - The radius of the hole in the center of the pie chart. When the value is greater than 0, the chart becomes a donut chart.

Outer Radius - The radius of the pie chart. It is the distance from the center of the chart to the outer edge of the slices or the largest/longest slice when the radius attribute is set.

Example: Two examples of pie-like charts created with this action is shown on the right. The first is a pie chart showing two categories - female and male - as parts of a whole. We see that there are a little more males (orange) as compared to females (blue) with the counts mapping to the angle. The second chart is a circular bar chart here the categories -1, 2, 3, and 4 - are mapped to slices, the angle mapping is set to equal size so angle does not represent any data values, and the radius attribute is set to Age which is aggregated to the mean Age for each

CONFIGURATOR ⓘ

Base Chart Pie Chart ▾

Pie Chart Options

Slice Attribute NumOfProducts ▾

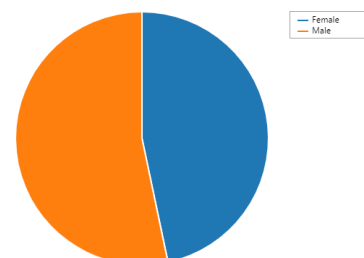
Angle Mapping Equal Size ▾

Radius Attribute Age ▾

Radius Agg. Function Mean ▾

Inner Radius 30

Outer Radius 175



category. Category 4 (red) has the highest average age and category 2 (orange) has the lowest average age.

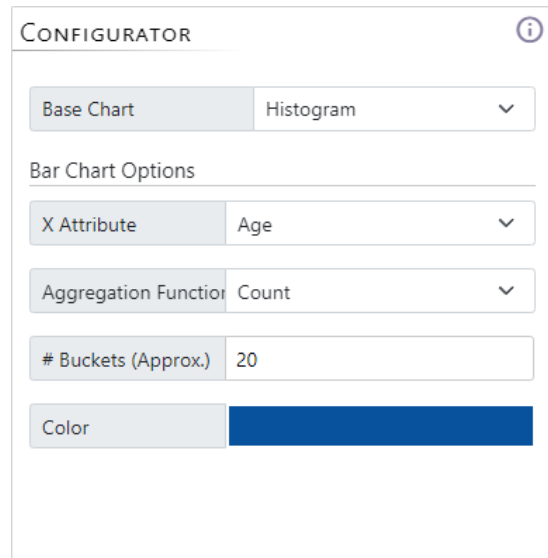
Histogram

X Attribute - The attribute to which the x-axis maps. For a histogram, this is a numerical attribute.

Aggregation Function - The method by which the y-axis is aggregated for each bucket created for the attribute on the x-axis. The methods are count and probability. The count method shows the number of data items in a bucket while the probability method shows the percentage of data items in a bucket versus all buckets.

Buckets (Approx.) - The approximate number of buckets the x-attribute range will be divided into. The value is approximate as internal algorithms will determine an appropriate value such that bucket edges are cleaner numbers.

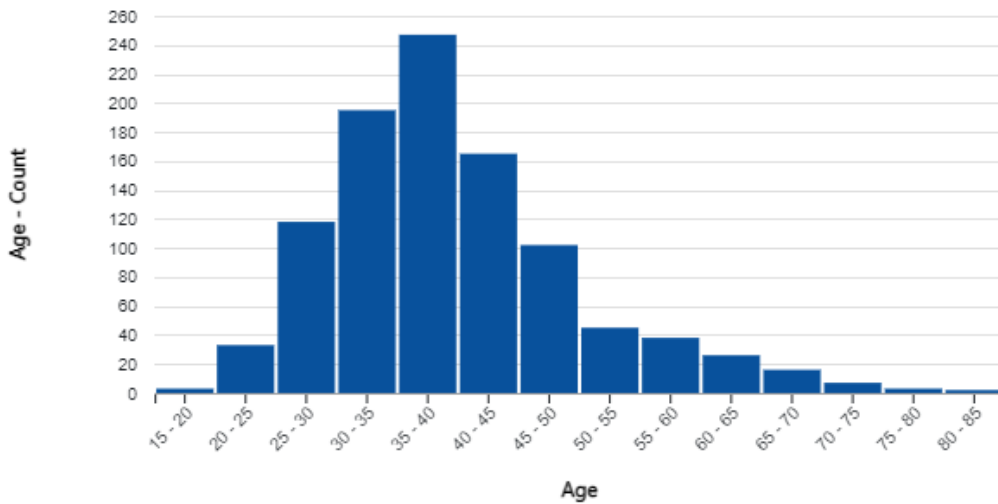
Color - The color of the bars.



The image shows a 'CONFIGURATOR' window for a histogram. It has a title bar with an information icon. Below the title bar, there are several configuration options:

- Base Chart:** A dropdown menu set to 'Histogram'.
- Bar Chart Options:** A section header.
- X Attribute:** A dropdown menu set to 'Age'.
- Aggregation Function:** A dropdown menu set to 'Count'.
- # Buckets (Approx.):** A text input field containing the value '20'.
- Color:** A color selection area with a blue bar.

Example: An example of a histogram created with this action is shown below. Here attribute Age is mapped to the x-axis and its count is shown along the y-axis. The #Buckets was set to 20 but the internal method determined 14 was appropriate so that each bucket covers five years with the first bucket covering the Age range 15-20 and the last bucket covering the range 80-85.



Balloon Chart

X Attribute - The attribute to which the x-axis maps. For a balloon chart, this is a nominal attribute.

Y Attribute - The attribute to which the y-axis maps. For a balloon chart, this is a nominal attribute.

Color - The color of the balloons or circles.

Max Radius - The radius of the largest balloon or circle.

CONFIGURATOR i

Base Chart Balloon Chart ▾

Balloon Chart Options

X Attribute Geography ▾

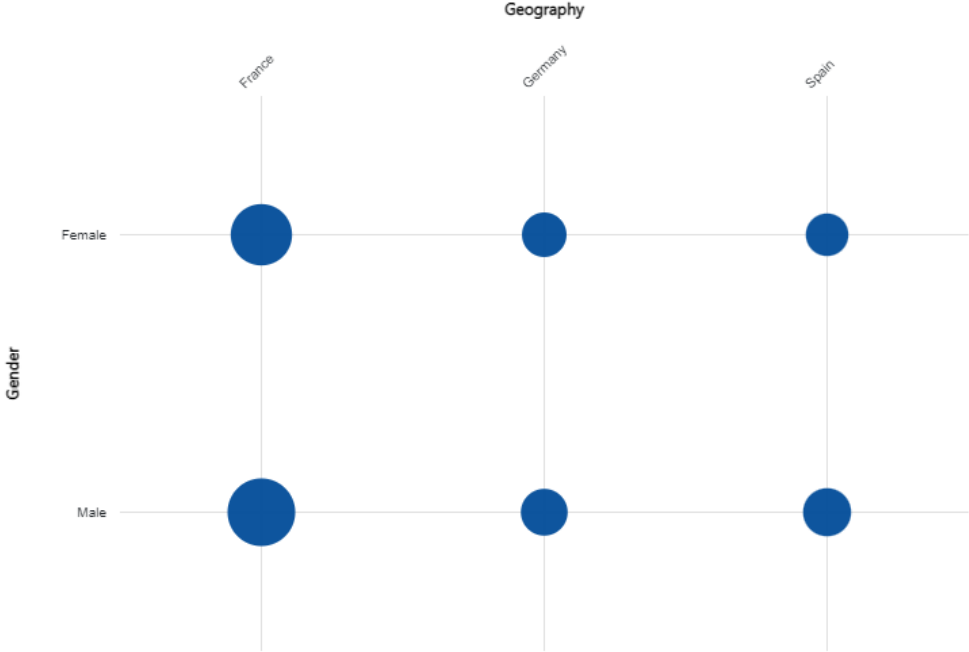
Y Attribute Gender ▾

Color

Max Radius 25

Example: An example of a balloon chart created with this action is shown below. Here the x-axis is set to Geography, the y-axis is set to Gender, the color is set to blue, and

the max radius is set to 25. Since there are only 2 genders and 3 countries present in the data we have 6 circles representing the data items for each gender-country combination. The largest circle representing males in France has a radius of 25.

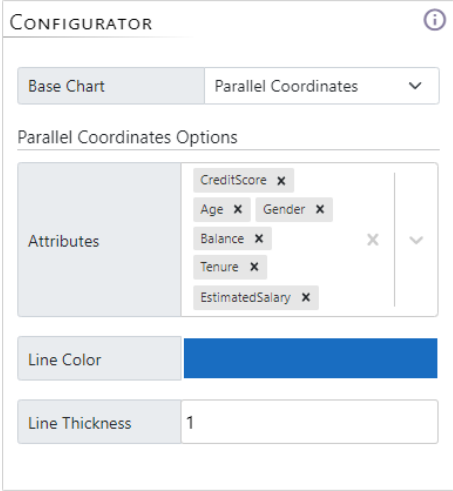


Parallel Coordinates

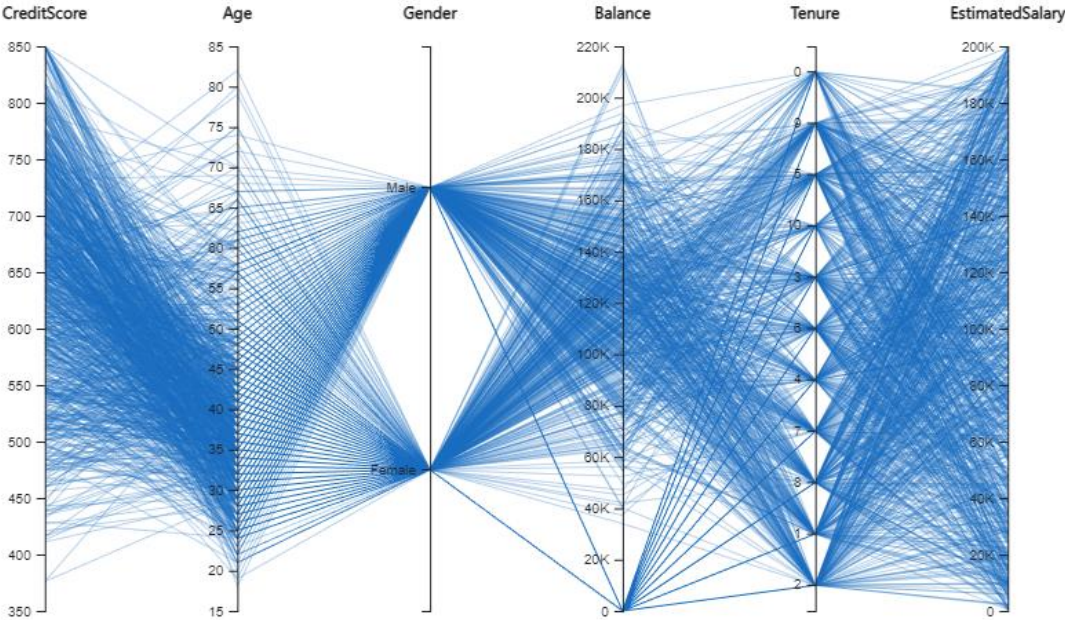
Attributes - The list of attributes for the parallel coordinate chart. A parallel axis will be created for each attribute. There must be at least 2 attributes to create a valid chart. Axes are ordered based on the order in which they are selected here.

Line Color - The color of the lines.

Line Thickness - The thickness of the lines.



Example: An example of a parallel coordinate chart created with this action is shown below. Here six attributes are selected - CreditScore, Age, Gender, Tenure, and Age, and EstimatedSalary - creating six axes. The color is set to blue and line thickness is set to 1.



Correlation Plot

Attributes - The list of attributes for a correlation plot. The attributes must all be numerical. An NxN grid will be created for N attributes.

Positive Corr. Color - The color for a correlation value of 1.

Negative Corr. Color - The color for a correlation value of -1.

CONFIGURATOR ⓘ

Base Chart: Correlation Plot ▼

Parallel Coordinates Options

Attributes: CreditScore ×, Age ×, Balance ×, EstimatedSalary × × ▼

Positive Corr. Color:

Negative Corr. Color:

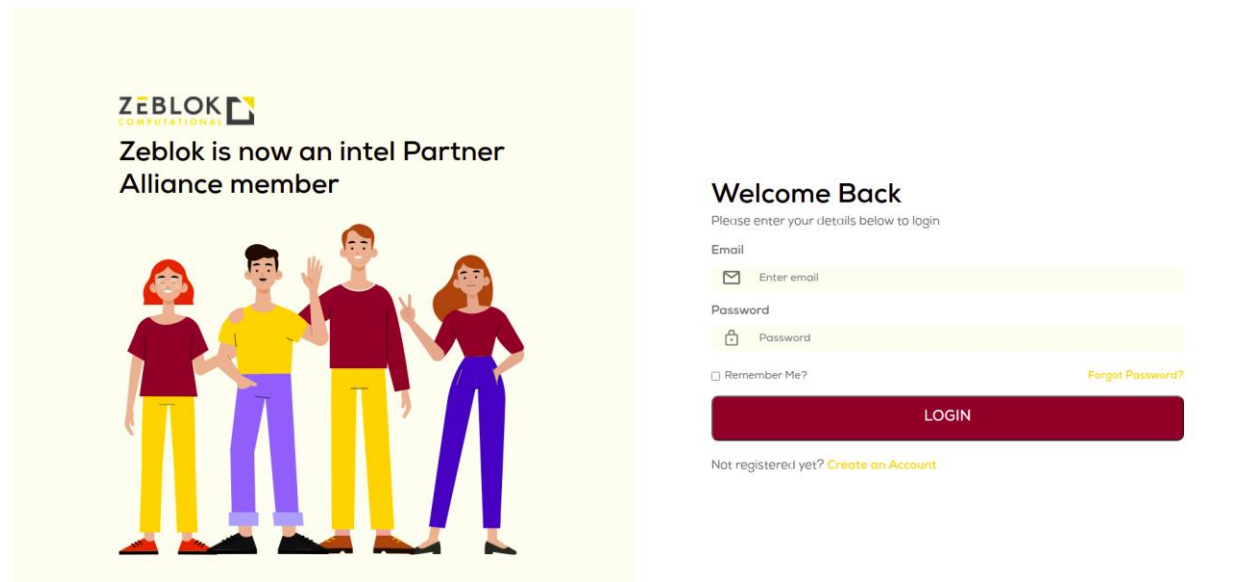
Example: An example of a parallel coordinate chart created with this action is shown below. Here six attributes are selected - EstimatedSalary, Balance, Age, and CreditScore, Age, and - creating six axes. The color for positive correlation is set to blue and the color for negative correlation is set to red.



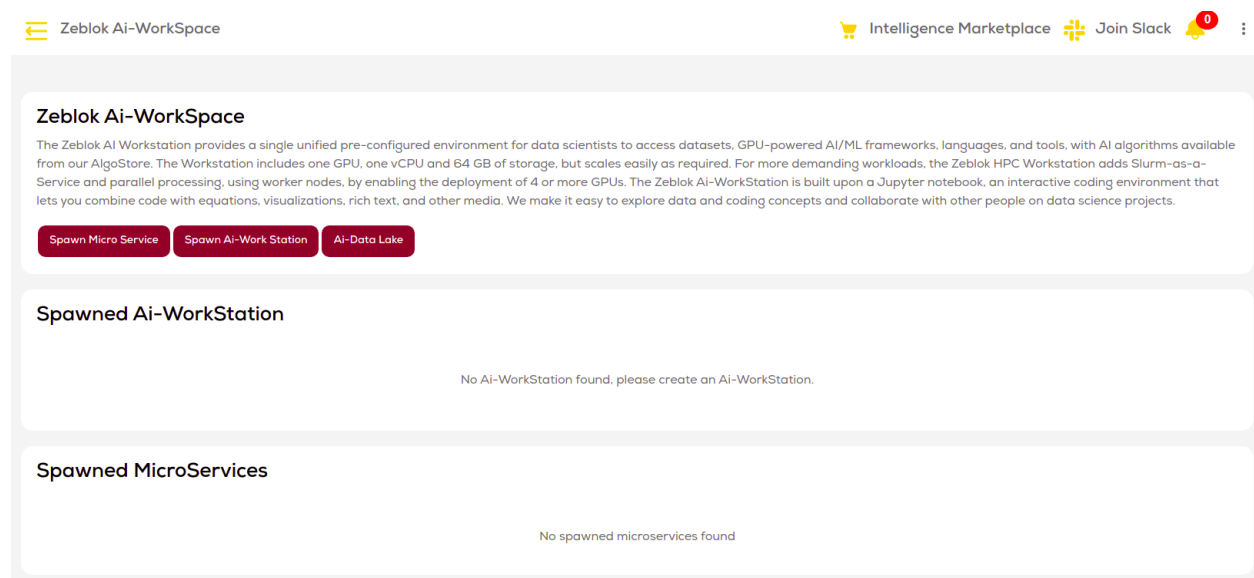
Launching the Workspace

The following instructions are a step-by-step guide for launching the AI Rover workstation through the Zeblok computational AI platform.

Login to the Zeblok computational platform at <https://app.zbl-aws.zeblok.com/>











On the homepage click on the “Spawn Micro Service” button.



Select Akai Kaeru Explainable AI MicroService

Select Your MicroService — Select Your DataCenter — Select Your Plan — Select Your Namespace — Configure Your MicroService




Search...

 mongoDB testing_1: 1649446540 Select MicroService	 Akai Kaeru: AK Analyst Select MicroService	 OVMS Select MicroService	 jupyter HPC Select MicroService
 intel Open MPI, Horovod*, and Jupyter* Notebook Select MicroService	 intel TensorFlow* & oneDNN with Jupyter* Notebook Select MicroService	 OpenVINO DEEP LEARNING WORKBENCH Import a model Perform baseline inference Intel DL workbench Select MicroService	 mongoDB MongoDB Select MicroService

Select Zeblok AWS Ohio DataCenter

Select Your MicroService — Select Your DataCenter — Select Your Plan — Select Your Namespace — Configure Your MicroService

Select Your MicroService (Akai Kaeru Explainable AI)

 amazon web services Zeblok AWS Ohio Ohio, US Category: Enterprise Select DataCenter	 Zeblok Azure Redmond, US Category: Enterprise Select DataCenter	 Advantech Oregon, US Category: Edge Select DataCenter
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Select Plan


Select Your MicroService.....
Aka Kearu Explainable AI


Select Your DataCenter.....
Zeblok AWS Ohio

Select Your Plan.....

Select Your Namespace.....

Configure Your MicroSer.....



C1-0-1-1-1-MS 

0 GPU

1vCPU

1 GB RAM

1 GB Storage

Select Plan

Select Namespace as akaikaerupublicaws

Select Your MicroService.....
Aka Kearu Explainable AI

Select Your DataCenter.....
Zeblok AWS Ohio

Select Your Plan.....
C1-0-1-1-1-MS

Select Your Namespace.....
akaikaerupublicaws

Configure Your MicroSer.....

Select Namespace

akaikaerupublicaws

Click Next

Select Your MicroService.....
Aka Kearu Explainable AI

Select Your DataCenter.....
Zeblok AWS Ohio

Select Your Plan.....
C1-0-1-1-1-MS

Select Your Namespace.....
akaikaerupublicaws

Configure Your MicroSer.....

Ports

5000

Arguments

Example: --config_path=/models/name:--port=9000

Environment Variables

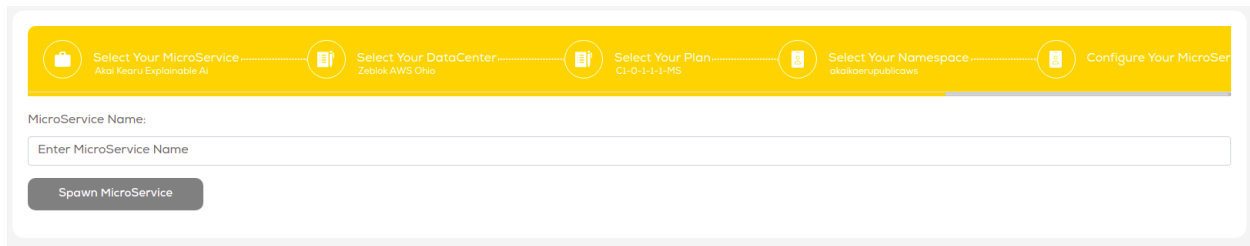
Example: FILE_UPLOAD_PATH=/public/uploads:AWS_KEY=xxctedgagj

Command

If your run command looks like this: ["bash", "-c", "token='password' "] enter bash, -c, token = "password"

NEXT

Enter a descriptive name and click “Spawn MicroService”



Select Your MicroService..... Akai Kearu Explainable AI

Select Your DataCenter..... Zeblok AWS Ohio

Select Your Plan..... C1-0-1-1-1-MS


Select Your Namespace..... akaikearupublicaws

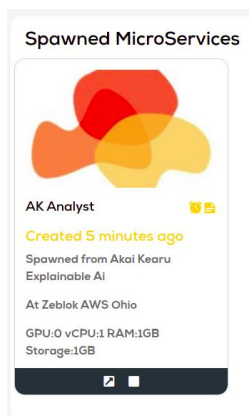
Configure Your MicroSer

MicroService Name:


Enter MicroService Name


Spawn MicroService

Back on the homepage you will see the spawned MicroService. Click  to open.



Spawned MicroServices





AK Analyst 

Created 5 minutes ago

Spawned from Akai Kearu Explainable AI

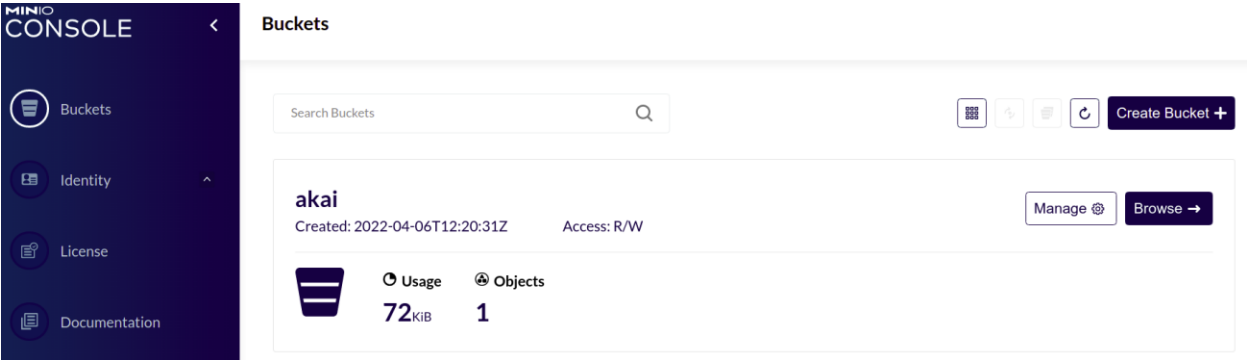
At Zeblok AWS Ohio

GPU:0 vCPU:1 RAM:1GB Storage:1GB

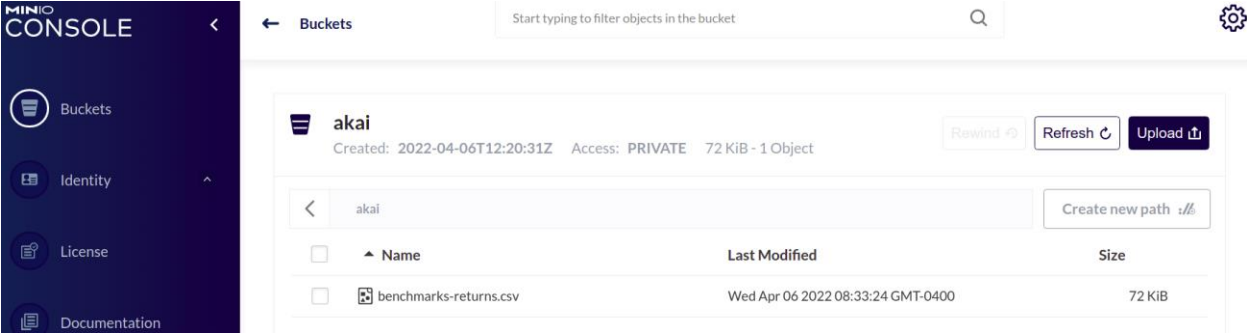
 

Useful Tips

Accessing the DataLake: Clicking on the “AI-Data Lake” button on the Zeblok home page will bring up the MinIO console containing an s3 bucket available for storing large data files.



Click “Browse” to browse the contents of the s3 bucket.



Finally, click “Upload” to upload a file to the bucket. Within the AK Analyst, you can load data from the DataLake through the “Load Data Lake” action. The IP address to use is `datalake.zbl-aws.zeblok.com:9000`. The image on the right shows the credentials to access the data in the bucket shown above.

