Data Context Map Quick Start Guide.
version 1.1 (*Beta*)

Eric Papenhausen (epapenha@akaikaeru.com)
Klaus Mueller (mueller@akaikaeru.com)
https://akaikaeru.com

March 23, 2021

Introduction

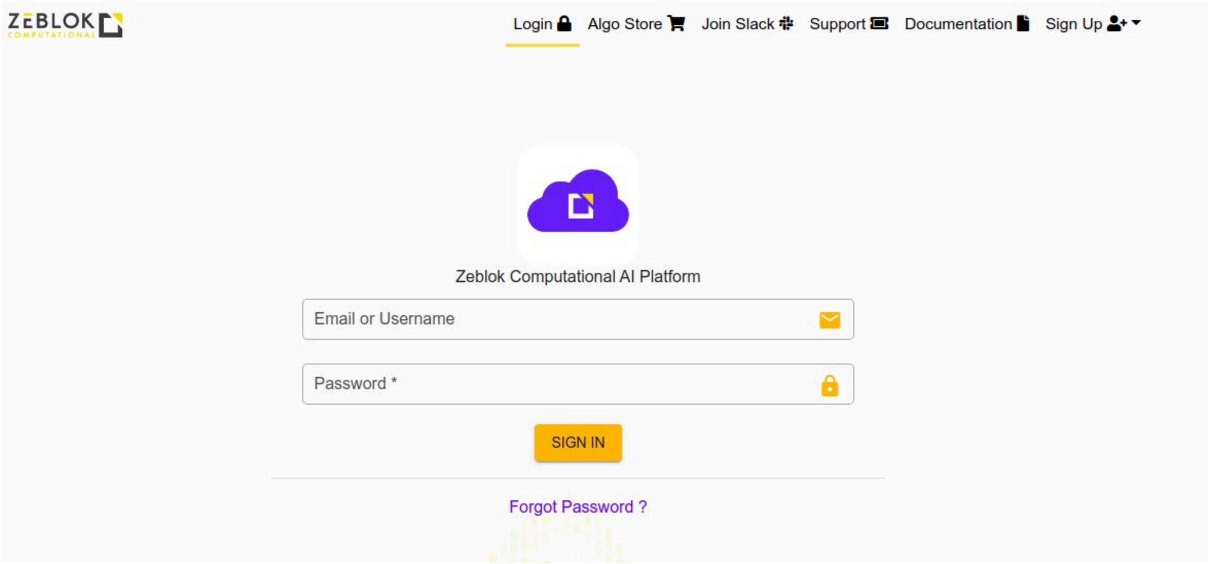
The AI rover finds and visualizes statistically significant and temporally consistent patterns. For a data set consisting of stocks, for example, these patterns represent patterns of stock behavior (e.g. price/book ratio $<$ 1) that are associated with unusually high or low returns. In the case of a data set with a temporal component (e.g. stocks in the NYSE over time), the patterns found are temporally consistent (i.e. they are consistently high/low over time). The AI rover is implemented as an extension to Jupyter lab to allow for easy integration with existing data science workflows and to allow insights found within the AI rover to be exported and analyzed further.

Contents

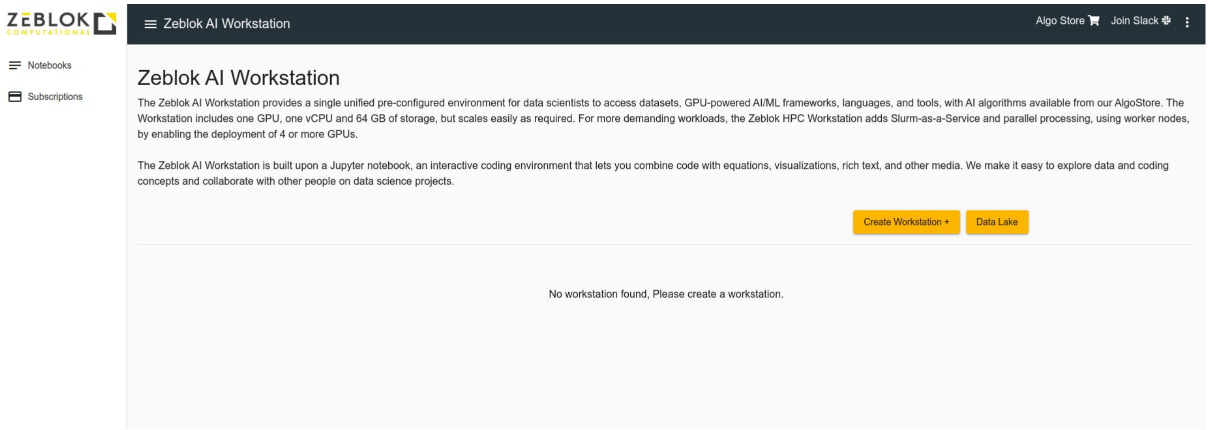
1 Launching a Workstation

The following instructions will help guide you in launching the AI Rover workstation through the Zeblok computational AI platform. This guide assumes that you have an account set up with Zeblok.

1. Login into the Zeblok computational AI platform.



2. This will bring up the home page. Click on the "Create Workstation" button.



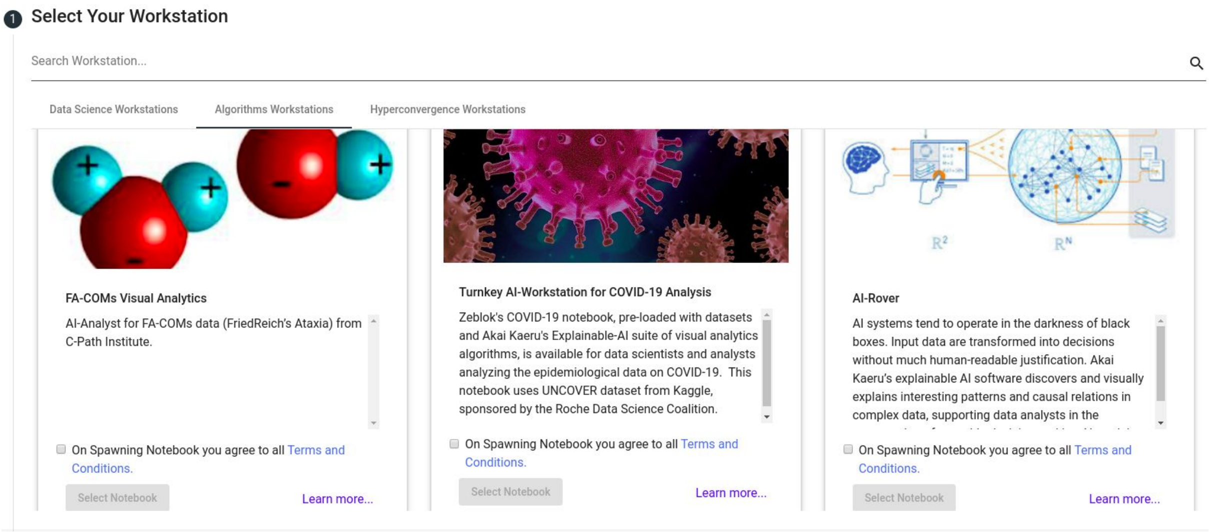
Zeblok AI Workstation

The Zeblok AI Workstation provides a single unified pre-configured environment for data scientists to access datasets, GPU-powered AI/ML frameworks, languages, and tools, with AI algorithms available from our AlgoStore. The Workstation includes one GPU, one vCPU and 64 GB of storage, but scales easily as required. For more demanding workloads, the Zeblok HPC Workstation adds Slurm-as-a-Service and parallel processing, using worker nodes, by enabling the deployment of 4 or more GPUs.

The Zeblok AI Workstation is built upon a Jupyter notebook, an interactive coding environment that lets you combine code with equations, visualizations, rich text, and other media. We make it easy to explore data and coding concepts and collaborate with other people on data science projects.

No workstation found, Please create a workstation.

3. Select the "Algorithms Workstation" tab and scroll down to the AI Rover workstation. Accept the terms and conditions and click on the "Select Notebook" button.



1 Select Your Workstation

Search Workstation...

FA-COMs Visual Analytics

AI-Analyst for FA-COMs data (FriedReich's Ataxia) from C-Path Institute.

□ On Spawning Notebook you agree to all Terms and Conditions.

Select Notebook   Learn more...

Turnkey AI-Workstation for COVID-19 Analysis

Zeblok's COVID-19 notebook, pre-loaded with datasets and Akai Kaeru's Explainable-AI suite of visual analytics algorithms, is available for data scientists and analysts analyzing the epidemiological data on COVID-19. This notebook uses UNCOVER dataset from Kaggle, sponsored by the Roche Data Science Coalition.

□ On Spawning Notebook you agree to all Terms and Conditions.

Select Notebook   Learn more...

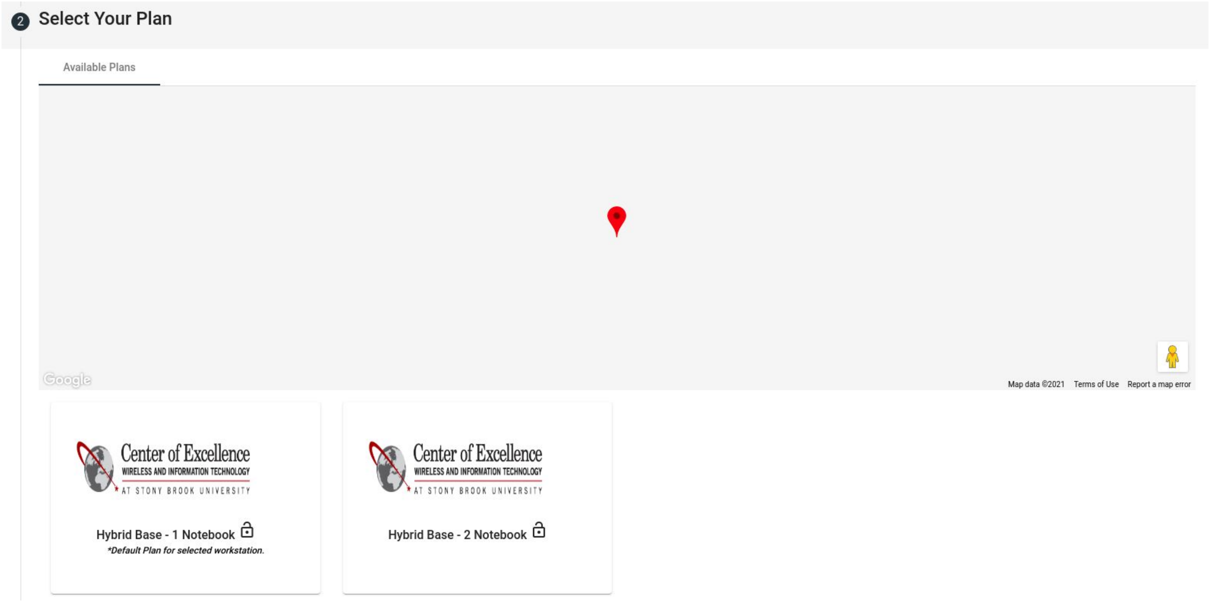
AI-Rover

AI systems tend to operate in the darkness of black boxes. Input data are transformed into decisions without much human-readable justification. Akai Kaeru's explainable AI software discovers and visually explains interesting patterns and causal relations in complex data, supporting data analysts in the

□ On Spawning Notebook you agree to all Terms and Conditions.

Select Notebook   Learn more...

4. On the "Select Plan" page, you can select a plan that works best for the current project. Here you can determine what computational resources and needed for the project.



2 Select Your Plan

Available Plans

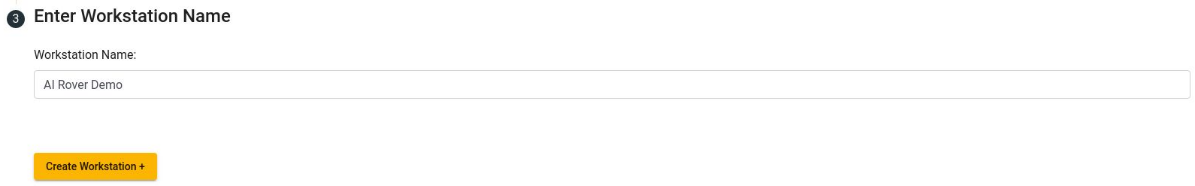
Google
Map data ©2021
Terms of Use
Report a map error
Center of Excellence
WIRELESS AND INFORMATION TECHNOLOGY
AT STONY BROOK UNIVERSITY
Center of Excellence
WIRELESS AND INFORMATION TECHNOLOGY
AT STONY BROOK UNIVERSITY

Hybrid Base - 1 Notebook

**Default Plan for selected workstation.*

Hybrid Base - 2 Notebook

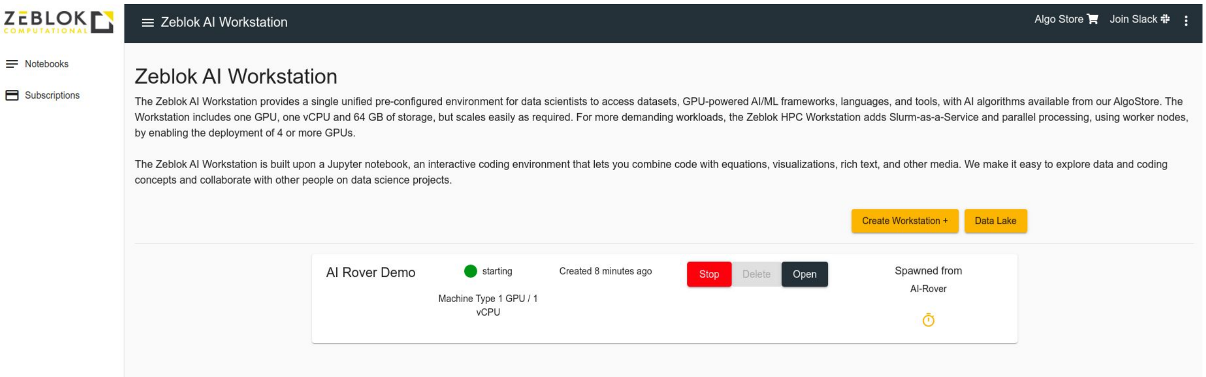
5. Enter a descriptive name for the project and click "Create Workstation".



③ Enter Workstation Name

Workstation Name: AI Rover Demo

6. You will be redirected to the homepage for managing your workstations. From here you can start/stop the workstation or select "Open" to begin working with the AI Rover.



Zeblok AI Workstation

The Zeblok AI Workstation provides a single unified pre-configured environment for data scientists to access datasets, GPU-powered AI/ML frameworks, languages, and tools, with AI algorithms available from our AlgoStore. The Workstation includes one GPU, one vCPU and 64 GB of storage, but scales easily as required. For more demanding workloads, the Zeblok HPC Workstation adds Slurm-as-a-Service and parallel processing, using worker nodes, by enabling the deployment of 4 or more GPUs.

The Zeblok AI Workstation is built upon a Jupyter notebook, an interactive coding environment that lets you combine code with equations, visualizations, rich text, and other media. We make it easy to explore data and coding concepts and collaborate with other people on data science projects.

AI Rover Demo ● starting Created 8 minutes ago

Machine Type 1 GPU / 1 vCPU

Spawned from AI-Rover

2 Overview of Features

2.1 Getting it to Work

There are three sample Jupyter notebooks that show how to operate the data context map. These are in the sample folder in the data_context_map directory. There are two modules that can be imported within a Jupyter notebook cell. The first is the pattern_miner module:

```
import data_context_map.pattern_miner as pm
```

This is needed for the multi-variate pattern miner. It is initiated by calling:

```
out = pm.AKMiner(df, target)
```

where df is a pandas dataframe and target is the target attribute of interest. The data context map visual interface is then rendered by calling:

```
out.render()
```

The second is the correlation_miner module:

```
import data_context_map.correlation_miner as cm
```

It is initiated by calling:

```
ctab = cm.CorrelationTable(df, target)
```

where df is a pandas dataframe and target is the target attribute of interest. The visual interface is then rendered by calling:

```
ctab.render()
```

2.2 Pattern Mining

The AI rover finds and visualizes groups of data points called patterns. Patterns are hypercubes of the form ’attribute’ (<,>,=) ’value’ (e.g. price/book < 1 and sector = Financial). Two types of pattern mining are supported within the data context map – numeric and binary. Numeric pattern mining is used when the target variable is continuous (e.g. returns). Binary pattern mining is used when the target variable is a 0 or 1 indicator (e.g. defaulting on a loan).

In both cases, patterns are found to be "interesting" when the pattern satisfies the following conditions:

1. The target variable within the pattern is statistically significantly higher or lower than the rest of the data set.
2. The effect size is large (i.e. higher than some predefined threshold)
3. The size of the pattern is large (i.e. higher than some predefined threshold)

2.2.1 Numerical

The statistical test performed with numeric pattern mining is the non-parametric Mann-Whitney U test. This is a non-parametric test and so it makes no assumption about the distribution of the target variable. The effect size used is the common language effect size. This is a measure of the probability of an item selected from the pattern being higher / lower than an item selected from outside the pattern. For example, an effect size of 0.8 would indicate that if we were to randomly select one point within the pattern and one point outside the pattern, 80% of the time the point within the pattern will be higher. Negative effect size measures indicate the opposite relationship (e.g. -0.8 indicates that 80% of the time the point within the pattern is lower).

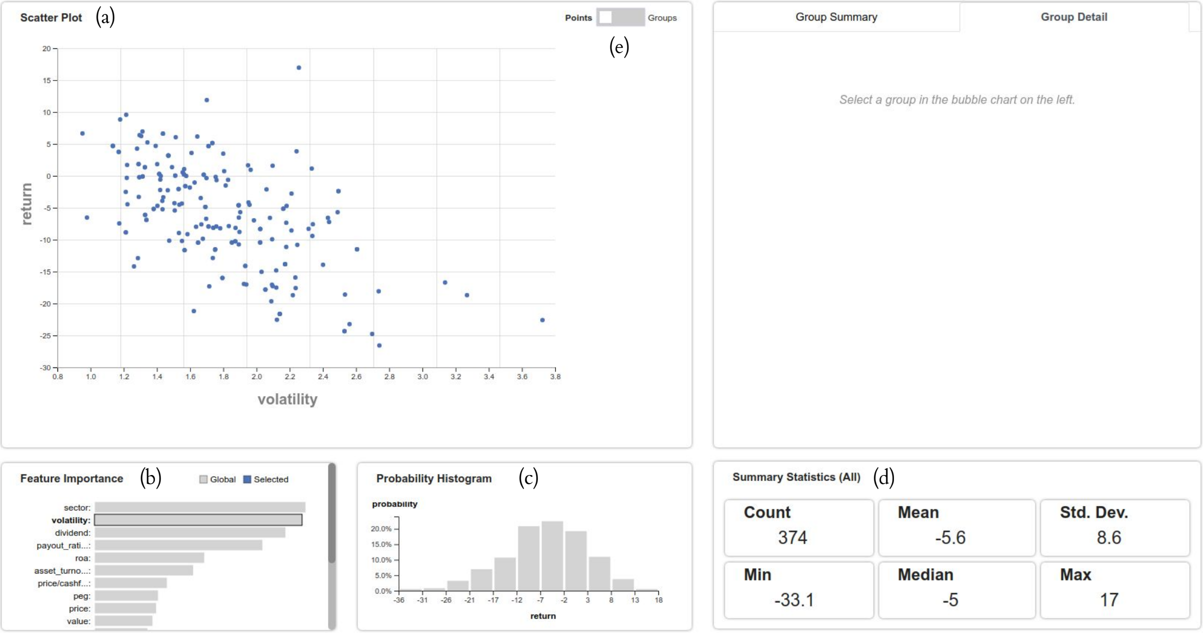
2.2.2 Binary

For pattern mining with a binary target variable, the target attribute is assumed to consist of 1’s and 0’s. The statistical test performed with binary pattern mining is the chi squared test for independence. This determines if the number of 1’s within the pattern is statistically higher than the overall data set. The effect size used is the odds ratio. An odds ratio of 2 indicates that the odds of the target being a 1 within the pattern increase by a factor of 2x compared to the overall data set.

2.3 AI Rover Interface

Figure 1 shows the starting state of the AI rover interface. The main plot (figure 1(a)) shows a scatter plot of `volatility` v.s. `return`. In this example, the target variable is `return`. The attribute of interest (AOI) is `volatility`. The AOI is a user selected feature and can be changed by clicking on a feature in the Feature Importance plot (b). This will change the x-axis of the main plot to the selected feature. By clicking on the switch in figure 1(e) the main plot will switch from a scatter plot to the group bubble chart.

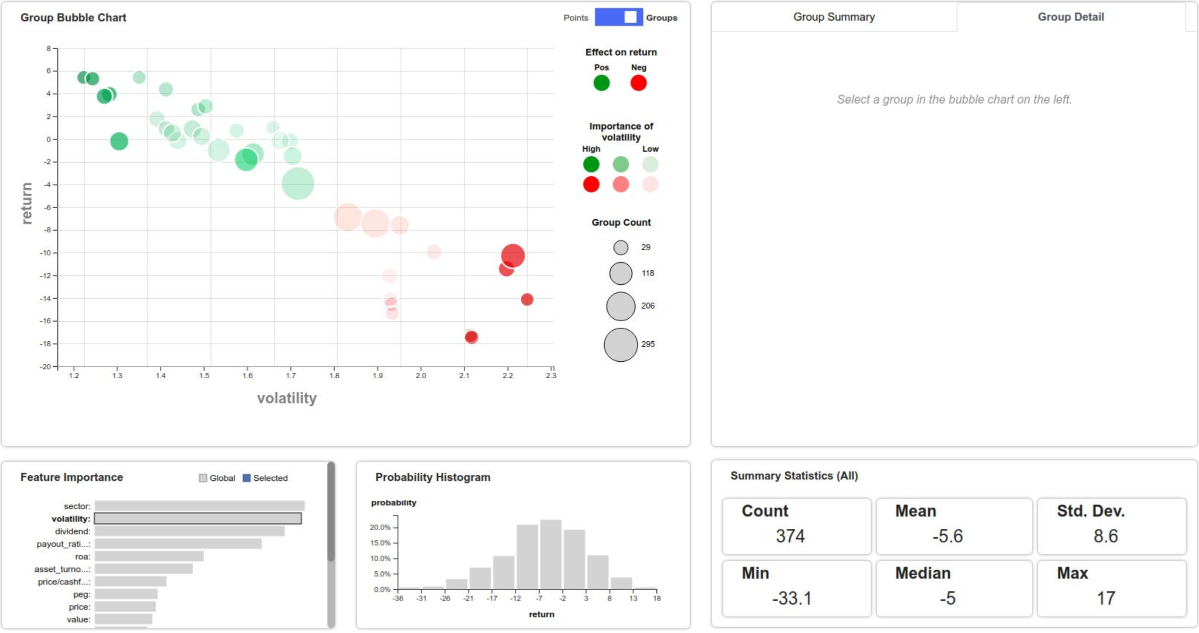
Figure 2 shows the interface after switching to the group bubble chart. The colored circles in the main plot represent groups of data points that are similar in some way and have an unusually low/high distribution of the target variable. The green (red) circles indicate that `return` is higher (lower) within these groups. The position of the circles is based on the median target and AOI values within the group. The opacity of the circles indicates how important the AOI is in defining the groups. In figure 2, for example, the circles at the extreme ends of `volatility` are opaque; indicating that `volatility` is an important feature for these groups. Conversely, the circles in the mid-range of `volatility` are more transparent. This indicates that `volatility` is not an important attribute for these groups (i.e. these groups are defined by other features). Finally, the size of the circle is based on the number of data points within the group.



Select a group in the bubble chart on the left.

Count	Mean	Std. Dev.
374	-5.6	8.6
Min	**Median**	**Max**
-33.1	-5	17

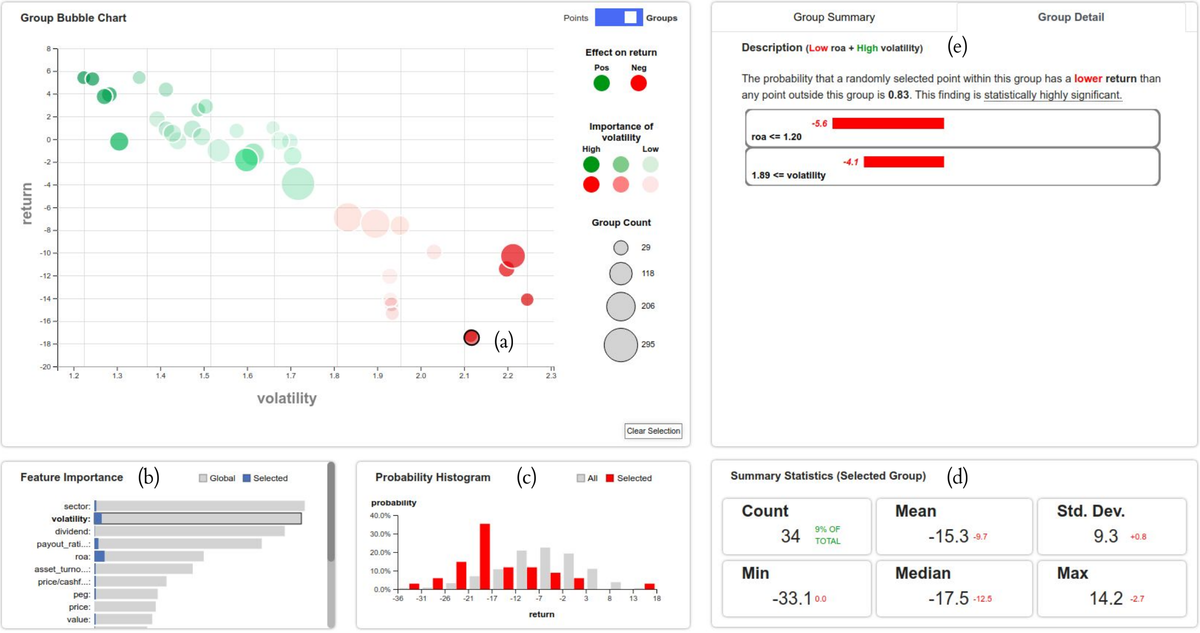
Figure 1: The initial view of the AI Rover interface. (a) Scatter plot plotting a selected attribute of interest (i.e. volatility) v.s. the target attribute (i.e. return). (b) Feature importance plot showing the relative predictive value of each of the features. (c) Probability histogram showing the distribution of the target variable (i.e. return). (d) Summary statistics for the target. (e) A switch for toggling between the scatter plot and the group bubble chart.



Select a group in the bubble chart on the left.

Count	Mean	Std. Dev.
374	-5.6	8.6
Min	**Median**	**Max**
-33.1	-5	17

Figure 2: The AI rover after a user switches to the group view.



Description (Low roa + High volatility)

The probability that a randomly selected point within this group has a **lower return** than any point outside this group is **0.83**. This finding is statistically highly significant.

-5.6
roa <= 1.20

-4.1
1.89 <= volatility

Summary Statistics (Selected Group)

Count	Mean	Std. Dev.
34 9% OF TOTAL	-15.3 -9.7	9.3 +0.8
Min	**Median**	**Max**
-33.1 0.0	-17.5 -12.5	14.2 -2.7

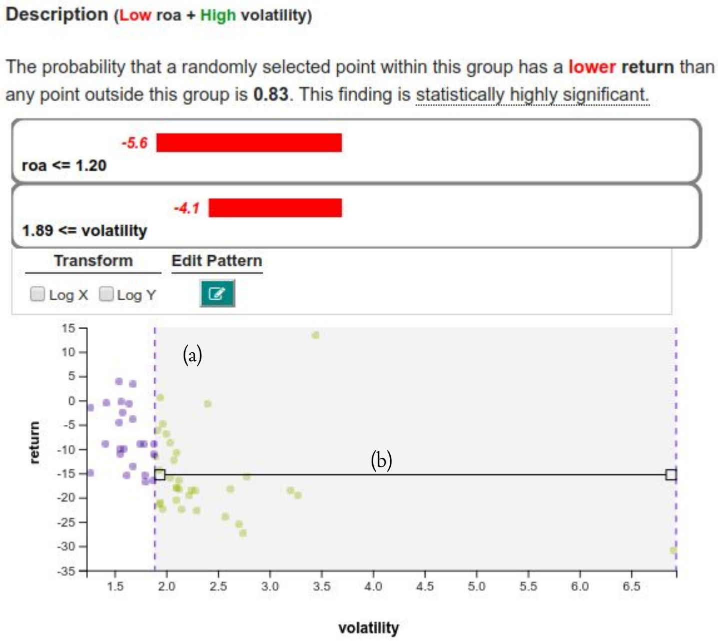
Figure 3: The AI Rover interface after clicking on a group (a).

Clicking on a circle in the main plot will update the interface to show more detail about the clicked group (see figure 3). The clicked group (figure 3(a)) is indicated by a black border. This allows the user to easily track its new position if she changes the AOI. Figure 3(b) shows the updated feature importance plot. The blue bars show how much the selected group contributes to each feature's global feature importance. In this example, we can see that the selected group contributes the most to `volatility` and `roa`. This also indicates that these two attributes are the most important for this group.

Figure 3(c) shows the updated probability histogram. The red bars show the distribution of `return` within the group while the gray bars show the overall distribution of `return`. The summary statistics view (d) are updated to show the summary statistics of the selected group. This also includes colored text which shows the difference between the overall dataset and the selected group (e.g. the difference between mean `return` of the selected group and overall mean is -9.7).

Figure 3(e) shows the group detail panel. This view contains specific information about how the selected group is defined as well as its effect on the target attribute. This includes a description of the criteria for this group (i.e. the group is defined by data points with low `roa` and high `volatility`). Under the description is text which describes the effect of this group. Under the text are two white boxes which contain a quantitative description of the selected group. This group is defined by data points where `roa` ≤ 1.2 and `volatility` ≥ 1.89. The red bars associated with this description show the individual effect (i.e. using shapely values) of each of these constraints. In this case, the low `roa` constraint reduces the `return` by -5.6, while the high `volatility` constraint reduces the `return` by -4.1.

A scatter plot is associated with each constraint in the group detail view. Clicking on one of the white boxes will bring up the scatter plot (see figure 4). The scatter plot for a clicked box only contains points that satisfy all of the constraints above it in the group detail view. In figure 4, for example, the scatter plot only shows the points whose `roa` ≤ 1.20. Conversely, the scatter plot associated with the first white box (i.e. the `roa` constraint) will contain all points in the data set. More generally, the first constraint in the group detail view will always contain the full



Description (**Low** roa + **High** volatility)

The probability that a randomly selected point within this group has a **lower return** than any point outside this group is **0.83**. This finding is statistically highly significant.

-5.6 **roa <= 1.20**

-4.1 **1.89 <= volatility**

Transform | Edit Pattern

□ Log X □ Log Y

Figure 4: The scatter plot after clicking on `volatility`. The blue points represent the data points that fall outside the group, while the yellow points indicate the points inside the group. The transparent box (a) marks all the points that satisfy the constraint `volatility` ≥ 1.89. The black line (b) shows the average return for this group.

scatter plot.

The group summary tab in figure 5 and figure 6 shows all feature's distributions for the selected group. Each feature is divided into 3 bins – low, medium and high. Each bin is then colored from white to blue based on the number of points within the bin. The first row in figure 6, for example, shows that the selected group has a low `roa`. The second row indicates that the selected group has high `volatility`. This makes sense since we know from the group detail view, that these are the defining characteristics of this group.

On the right we see a column labeled "Effect on return". This column provides the ability to do a counter-factual analysis. For example, if we were to add `sector` as a defining characteristic for this subgroup, it would only reduce the average `return` in this group by -1.1. This allows us to answer the question, "What if this group were defined in part by `sector`." In this case, once we know `roa` and `volatility`, the other features do not tell us much.

2.4 Correlation Miner

In addition to identifying groups where the distribution of some target attribute is unusually high / low, we also identify patterns where the correlation between two attributes is high. The correlation miner identifies sub-spaces where the correlation between one attribute and a target attribute is unusually high. This is then presented as a correlation table as seen in figure 7. Clicking on the "Scatter Plot" drop-down shows the correlation pattern control interface (see figure 8).

An interesting consequence of the correlation miner is that it can identify conflicting correlations in different sub-spaces. In figure 7, for example, although `asset_turnover` is positively correlated with `fut_return`, the correlation miner finds a group of data points where there is a moderate negative correlation. The ability to separate regions of positive and negative correlations is a key strength of the correlation miner.

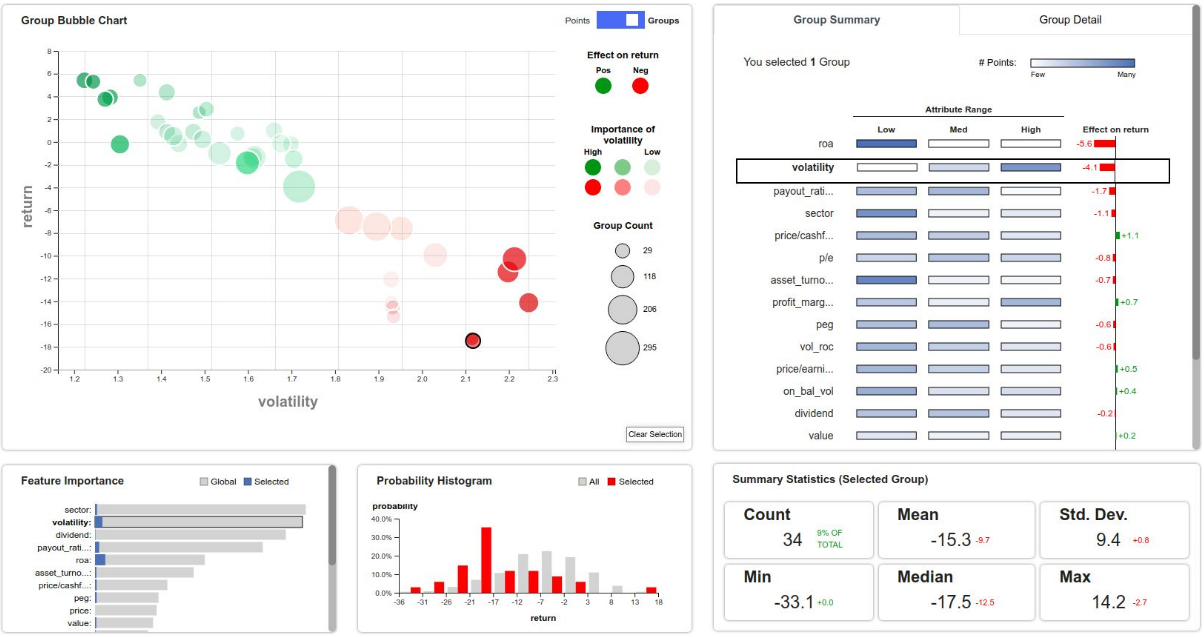


Figure 5: The AI Rover interface after clicking on the "Group Summary" tab.

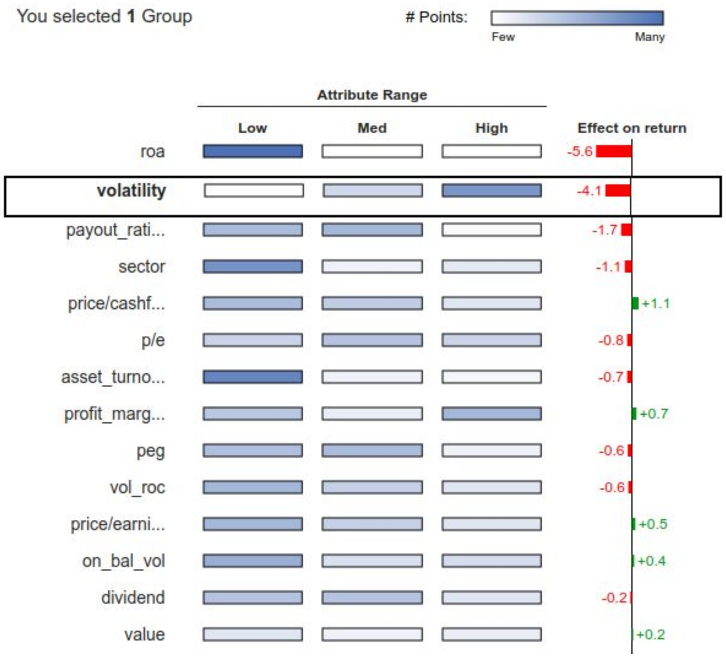
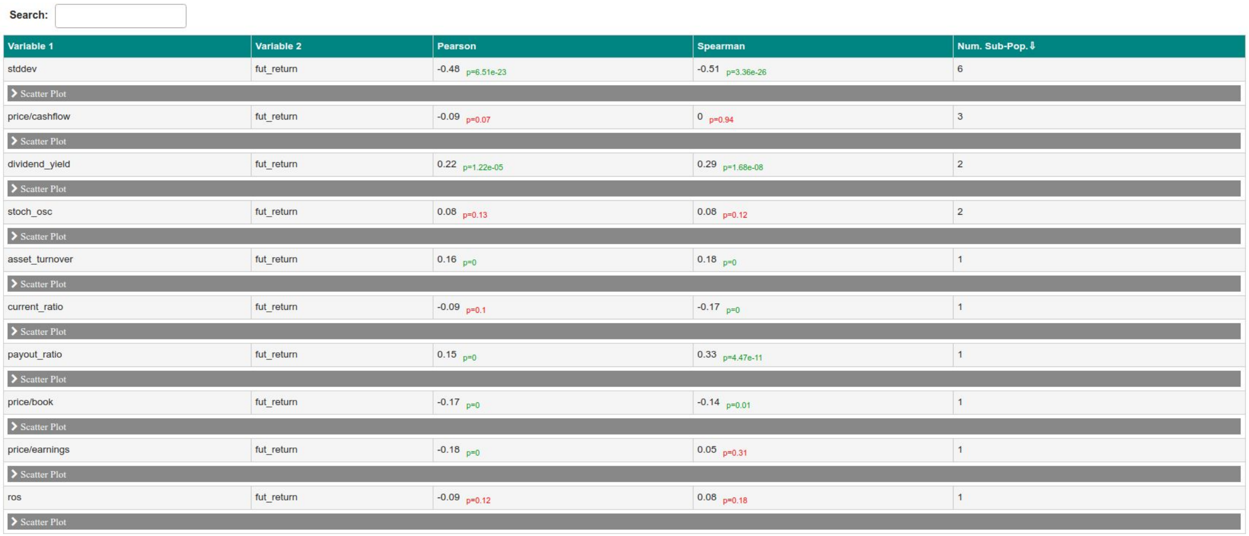


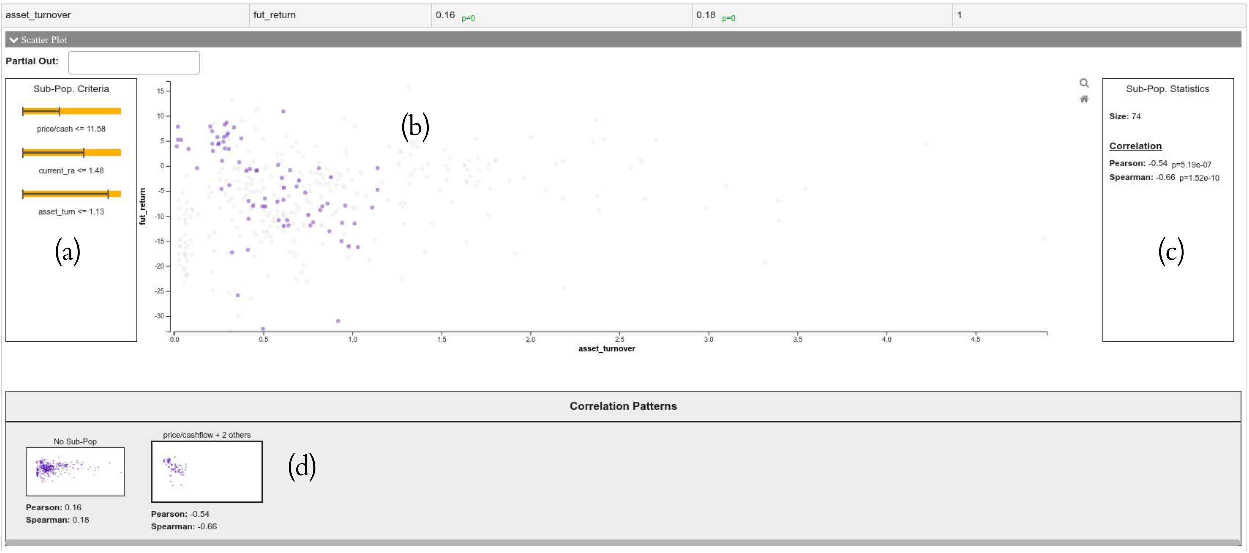
Figure 6: The group summary tab.



Search:

Variable 1	Variable 2	Pearson	Spearman	Num. Sub-Pop.⇩
stddev	fut_return	-0.48 p=6.51e-23	-0.51 p=3.36e-26	6
› Scatter Plot				
price/cashflow	fut_return	-0.09 p=0.07	0 p=0.94	3
› Scatter Plot				
dividend_yield	fut_return	0.22 p=1.22e-05	0.29 p=1.68e-08	2
› Scatter Plot				
stoch_osc	fut_return	0.08 p=0.13	0.08 p=0.12	2
› Scatter Plot				
asset_turnover	fut_return	0.16 p=0	0.18 p=0	1
› Scatter Plot				
current_ratio	fut_return	-0.09 p=0.1	-0.17 p=0	1
› Scatter Plot				
payout_ratio	fut_return	0.15 p=0	0.33 p=4.47e-11	1
› Scatter Plot				
price/book	fut_return	-0.17 p=0	-0.14 p=0.01	1
› Scatter Plot				
price/earnings	fut_return	-0.18 p=0	0.05 p=0.31	1
› Scatter Plot				
ros	fut_return	-0.09 p=0.12	0.08 p=0.18	1
› Scatter Plot				

Figure 7: Correlation table showing Pearson and Spearman correlations between an attribute and the target attribute of interest.



asset_turnover	fut_return	0.16 p=0	0.18 p=0	1

Scatter Plot

Partial Out:

Sub-Pop. Criteria

price/cash <= 11.58

current_ra <= 1.48

asset_turn <= 1.13

Sub-Pop. Statistics

Size: 74

Correlation

Pearson: -0.54 p=5.19e-07

Spearman: -0.66 p=1.52e-10

Correlation Patterns

No Sub-Pop

Pearson: 0.16

Spearman: 0.18

price/cashflow + 2 others

Pearson: -0.54

Spearman: -0.66

Figure 8: Correlation Mining control interface showing a pattern where asset_turnover is negatively correlated with fut_returns. (a) shows the criteria that defines the selected pattern. (b) shows a scatter-plot stratified on the criteria in (a). (c) shows statistics for this pattern (i.e. size, Pearson and Spearman correlations). (d) Shows a small multiples display illustrating the correlation patterns mined by the software. Clicking a pattern in this view will update (a), (b), and (c) accordingly.

3 API

There are several data context map specific functions that a user calls within the Jupyter notebook interface. These are explained in more detail in this section.

3.1 AI Rover

```
data_context_map.pattern_miner.AKMiner(df, dependent, temporal=None, pattern_json=None, mine_type=’numeric’,
minsup=0.01, es_thresh=’auto’, max_pattern=None, max_depth=None, min_stable=None, ts_width=None,
train_range=None, license_path=None, lib_path=None, opt_bound=False, fdr=’fast’, holdout=3,
causal_rule=None, verbose=False)
```

Description:

Performs pattern mining and prepares the visualization to be rendered.

Parameters:

- **df** (*pandas dataframe*): The data set to mine.
- **dependent** (*str*): The column of the dataframe that acts as the dependent variable. This is the attribute to predict.
- **temporal** (*str*): The attribute to be treated as a time variable.
- **pattern_json** (*list*): A list of pre-computed patterns of the form `{attribute:{’lb’: v1, ’ub’: v2}}`. When this parameter is set, pattern mining will not occur and instead this list of patterns will be visualized.
- **mine_type** (*str*): The data type of the dependent attribute. Valid mine types are ’numeric’ and ’binary’.
- **minsup** (*float*): The minimum size threshold of a pattern as a percentage of the dataset size. Default is 1%.
- **es_thresh** (*float* or *dict*): The minimum effect size for a pattern to be considered ’interesting’. The effect size is the common language effect size for numeric mine_type (default is 0.6) and the odds ratio for binary or multiclass mine_types (default is 2). For multiclass mine_types, it can also be a dictionary mapping class ids (i.e. 0, 1, or 2) to its corresponding minimum effect size.
- **max_pattern** (*int*): Maximum number of patterns to mine.
- **max_depth** (*int*): Maximum complexity of a pattern. e.g. max_depth=2 indicates that no pattern containing more than 2 attributes will be returned.
- **min_stable** (*int*): The number of consecutive time steps for which a pattern must be ’interesting’ to be considered temporally consistent.
- **ts_width** (*int*): Width of the timestep. If ts_width = 1, then each unique value of the temporal attribute is a timestep. If ts_width > 1, then each time step is a range.
- **train_range** (*list*): List defining [start, end] of a training range along the temporal dimension under which to mine the patterns. If None, then patterns are mined from the full data set.
- **license_path** (*str*): File path to the license.txt license key.
- **lib_path** (*str*): File path to the libdcm.so (libdcm.dll) shared library.
- **opt_bound** (*bool*): If true, the bounds of the patterns are optimized throughout each level of the mining process.
- **fdr** (*str*): Method for controlling the false discovery rate during pattern mining (’fast’ or ’exhaustive’). The ’fast’ method is a greedy strategy, whereas the ’exhaustive’ looks at all significant patterns and removes false discoveries afterwards.
- **holdout** (*int*): Number of holdout sets to validate the patterns on.

- **causal_rule** (*str*): None or 'iptw'. If 'iptw' then the average treatment effect is determined using inverse probability of treatment weighting. Patterns without a significant treatment effect are pruned.
- **verbose** (*bool*): Boolean controlling verbose output.

Returns:

An `AKMiner` object.

3.2 `render`

`AKMiner.render()`

Description:

Draws the data context map list view to the screen.

3.3 get_pattern_json

`AKMiner.get_pattern_json()`

Description:

Returns the mined patterns as list of python dictionaries of the form `{attribute:{'lb': v1, 'ub': v2}}`.

3.4 `CorrelationTable`

`data_context_map.correlation_miner.CorrelationTable(df, dependent, precompute=False, minsup=0.01, es_thresh=None, license_path=None, lib_path=None, fdr='fast', holdout=3, alpha=0.05, max_depth=None, max_pattern=None, opt_bound=False, verbose=False)`

Description:

Performs correlation mining and prepares the visualization to be rendered.

Parameters:

- **df** (*pandas dataframe*): The data set to mine.
- **dependent** (*str*): The column of the dataframe that acts as the dependent variable. This is the attribute to predict.
- **precompute** (*bool*): If True, all correlation patterns are identified prior to showing the correlation table.
- **minsup** (*float*): The minimum size threshold of a pattern as a percentage of the dataset size. Default is 0.01 (i.e. 1%).
- **es_thresh** (*float*): The minimum effect size for a pattern to be considered 'interesting'. The effect size is measured as the Spearman correlation.
- **license_path** (*str*): File path to the license.txt license key.
- **lib_path** (*str*): File path to the libdcm.so (libdcm.dll) shared library.
- **fdr** (*str*): Method for controlling the false discovery rate during pattern mining ('fast' or 'exhaustive'). The 'fast' method is a greedy strategy, whereas the 'exhaustive' looks at all significant patterns and removes false discoveries afterwards.
- **holdout** (*int*): Number of holdout sets to validate the patterns on.
- **alpha** (*float*): P-value significance level.

- **max_depth** (*int*): Maximum complexity of a pattern. e.g. max_depth=2 indicates that no pattern containing more than 2 attributes will be returned.
- **max_pattern** (*int*): Maximum number of patterns to mine.
- **opt_bound** (*bool*): If true, the bounds of the patterns are optimized throughout each level of the mining process.
- **verbose** (*bool*): Boolean controlling verbose output.

Returns:

A `CorrelationTable` object.

3.5 `render`

`CorrelationTable.render()`

Description:

Draws the correlation table to the screen.