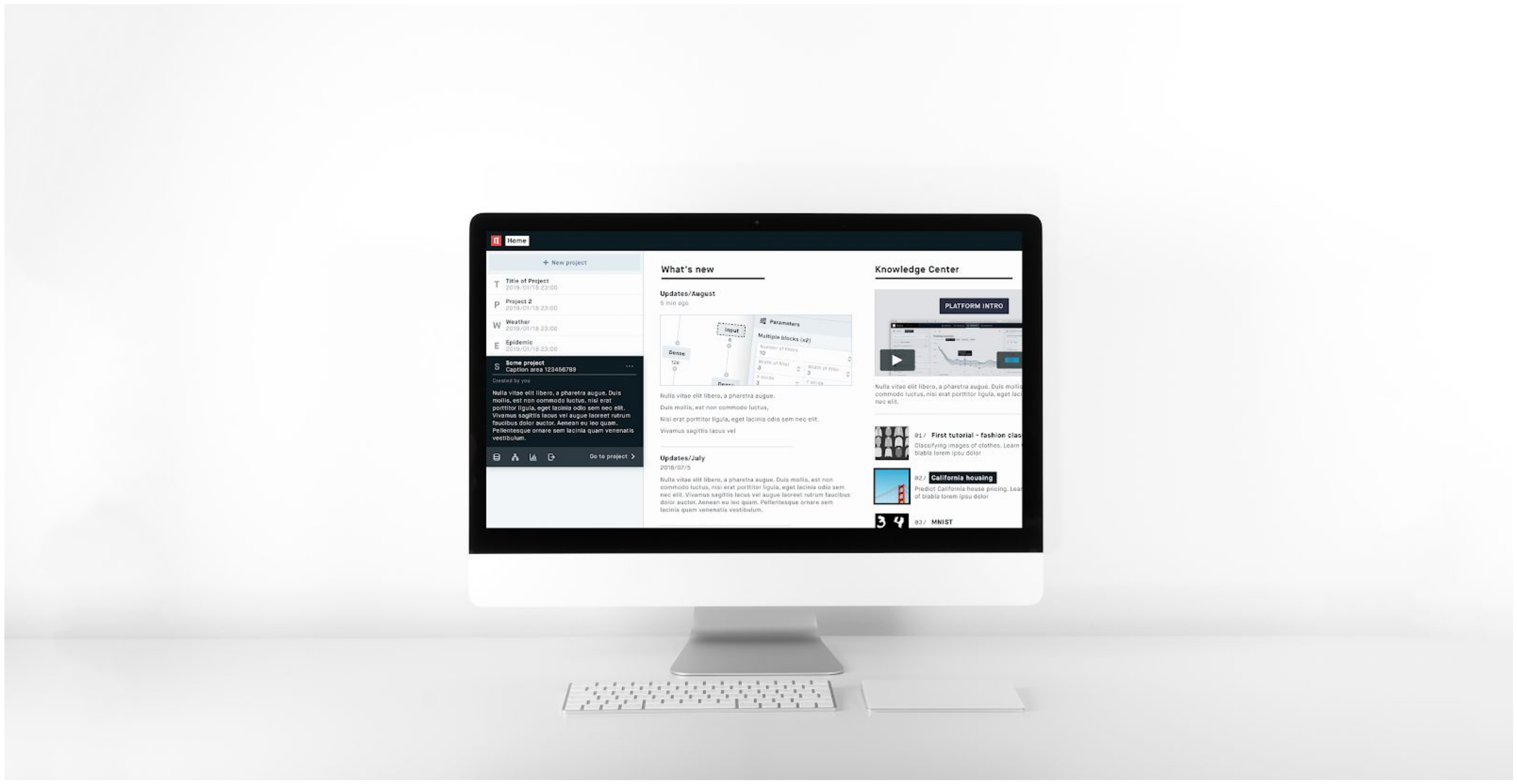
Peltarion operational AI platform description



2019-02-28

01/ Purpose

The purpose of this document is to provide a high-level introduction to the Peltarion Platform. The document presents a number of views of the platform and walks the reader through key parts of the graphical user interface (GUI). It describes the top-line functions and sections for the landing page, datasets view, modeling view, evaluation view and deployment view. For further information contact your Peltarion or partner contact.

02/ Platform introduction

The Peltarion operational AI platform is a single unified platform that is capable of executing a broad range of deep learning-based use cases and methodologies. The platform is cloud-based, with all tasks completed via the GUI. The components (sic views) have a consistent "look and feel" and are interoperable across the entire AI project "pipeline" – across data tasks, model creation, training, evaluation and deployment. These include tasks relating to data access and ingestion, data preprocessing and augmentation, interactive exploration and visualization, advanced modeling, model training and evaluation, deployment, project and resource management and collaboration.

03/ Landing page

Each landing page is unique to an organization. The user will see the landing page when logging into the Peltarion Platform, and will have access to the following resources:

- Projects
- GPU and cloud storage usage
- Running experiments list
- What's new
- Knowledge center.

The left section displays all the existing projects to which the user has access.

“Running experiments” in the upper middle section lists the experiments on the platform. The name, status and other relevant information are displayed for each experiment. The list displays all currently running experiments, as well as completed experiments (maximum 10) from the last 2 weeks.

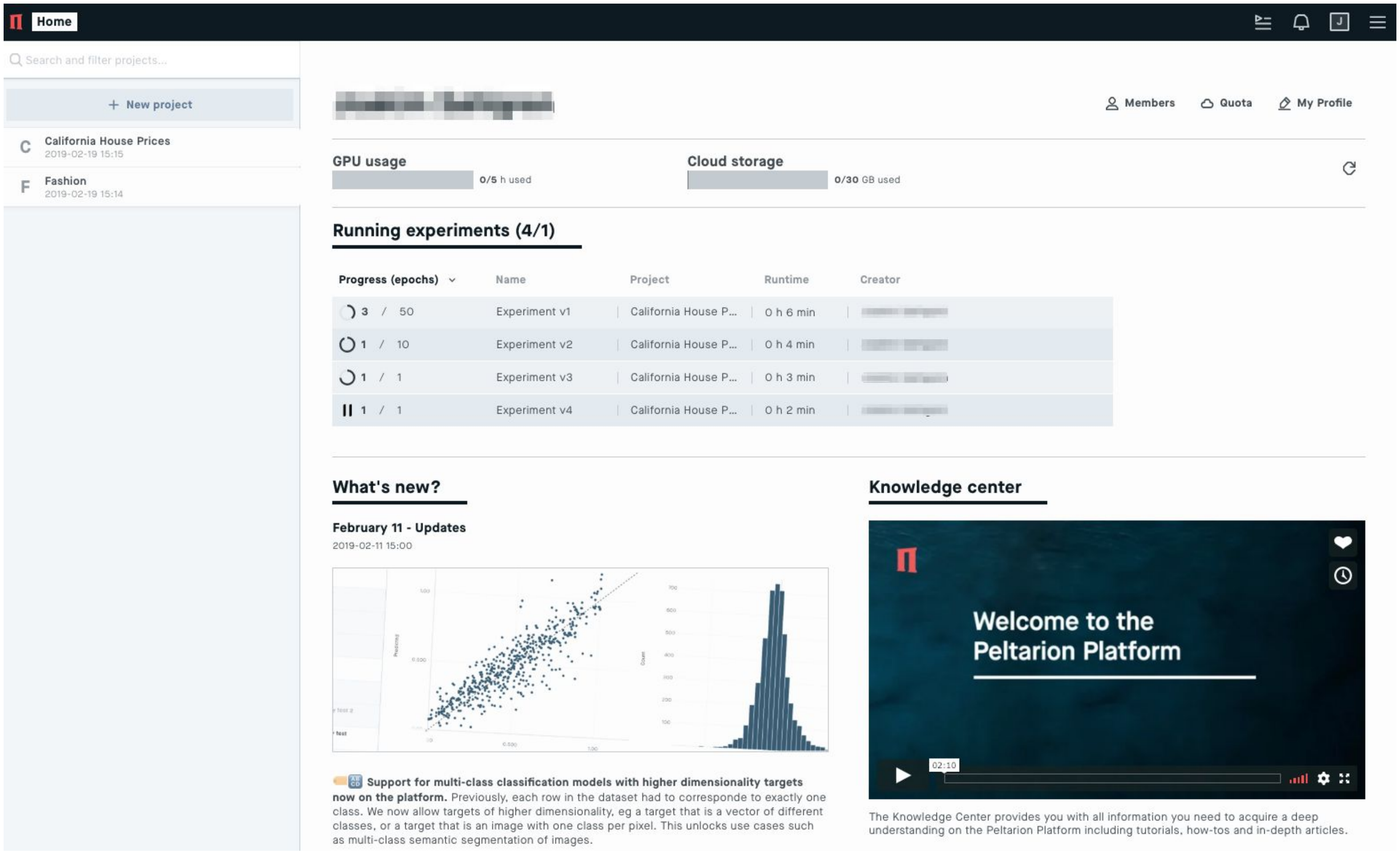
In case all the GPUs are busy, the new initiated jobs will appear in “Queued” status, and those recently completed or paused experiments will be listed as “Trained” experiments.

The “GPU usage” and “Cloud storage” at the top displays the available and remaining resources for the logged in user’s currently selected organization. If you select a project, you will get to see the actual GPU and storage usage as well as running experiments for this specific project.

“What’s new” section in the lower middle section gives updates regarding recent releases and modifications to the platform.

The right section is the window into the knowledge center. Various links allow access to tutorials and other information in the knowledge center.

In the right upper corner is a menu link with other useful links such as the knowledge center or support form.



C California House Prices 2019-02-19 15:15

F Fashion 2019-02-19 15:14

[redacted]

GPU usage 0/5 h used

Cloud storage 0/30 GB used

Running experiments (4/1)

Progress (epochs)	Name	Project	Runtime	Creator
3 / 50	Experiment v1	California House P...	0 h 6 min	[redacted]
1 / 10	Experiment v2	California House P...	0 h 4 min	[redacted]
1 / 1	Experiment v3	California House P...	0 h 3 min	[redacted]
1 / 1	Experiment v4	California House P...	0 h 2 min	[redacted]

What's new?

February 11 - Updates
2019-02-11 15:00

Support for multi-class classification models with higher dimensionality targets now on the platform. Previously, each row in the dataset had to corresponde to exactly one class. We now allow targets of higher dimensionality, eg a target that is a vector of different classes, or a target that is an image with one class per pixel. This unlocks use cases such as multi-class semantic segmentation of images.

Knowledge center

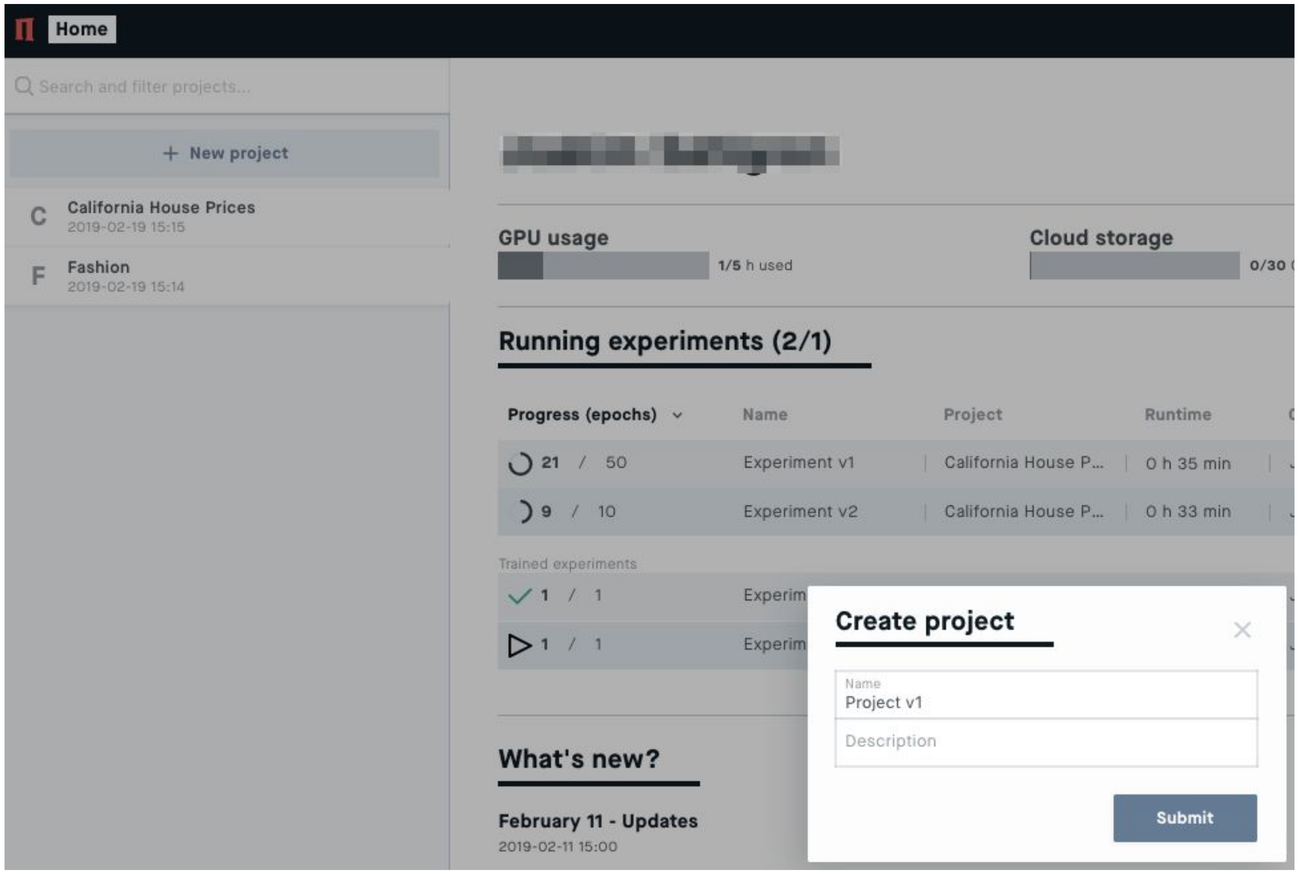
The Knowledge Center provides you with all information you need to acquire a deep understanding on the Peltarion Platform including tutorials, how-tos and in-depth articles.

Projects navigation (left section)

This section allows you to create a new project or select, edit or delete an existing project.

New project

To create a new project, click on "New project," and assign a name and description. After submission, you can begin uploading datasets and creating models.



C California House Prices 2019-02-19 15:15

F Fashion 2019-02-19 15:14

[redacted]

GPU usage 1/5 h used

Cloud storage 0/30

Running experiments (2/1)

Progress (epochs)	Name	Project	Runtime
21 / 50	Experiment v1	California House P...	0 h 35 min
9 / 10	Experiment v2	California House P...	0 h 33 min

Trained experiments

1 / 1	Experim		
1 / 1	Experim		

What's new?

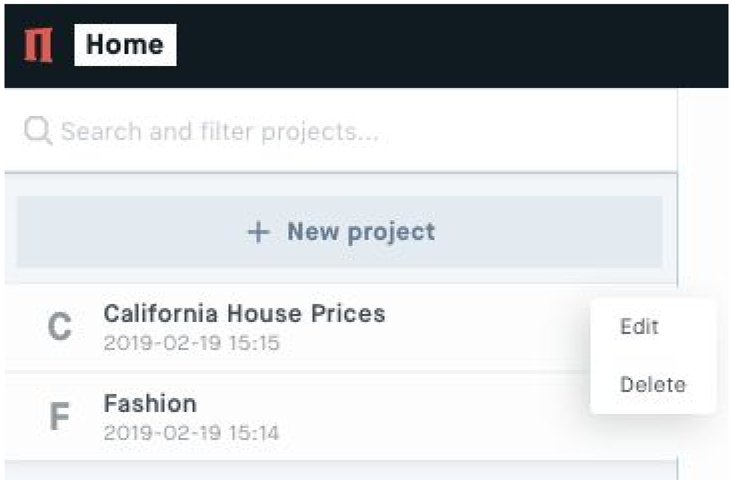
February 11 - Updates
2019-02-11 15:00

Create project

Name: Project v1

Description

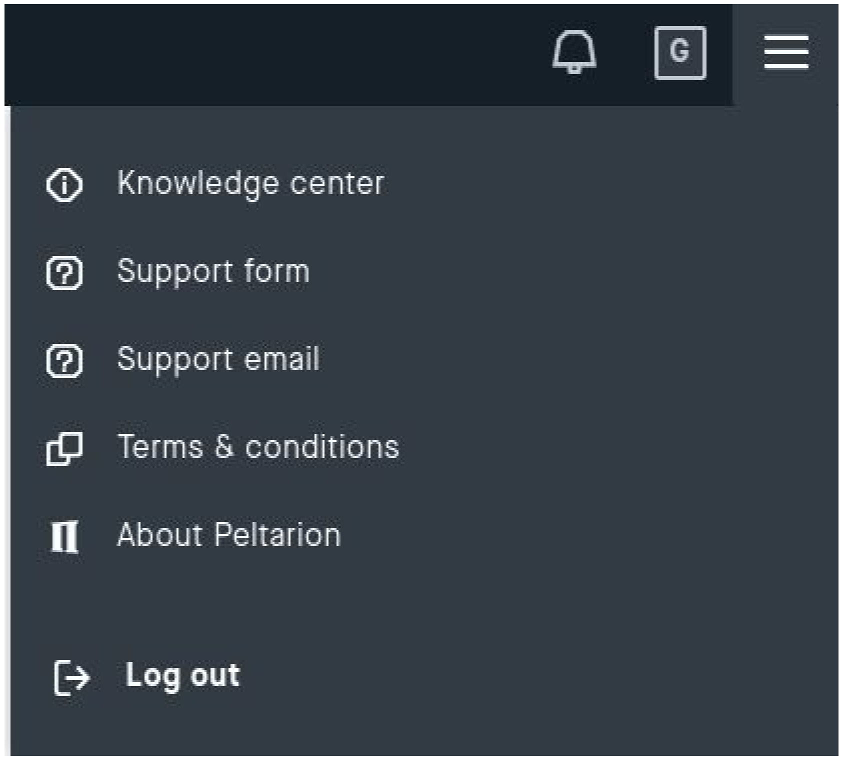
Edit and delete



Hovering over a specific project reveals an ellipsis (...). Clicking on it will open the Project options menu where you can edit the name and description of the project, or delete the project altogether.

You can also go directly to the dataset, modeling, evaluation and deployment views indicated by the icon on the bottom of the pop-up.

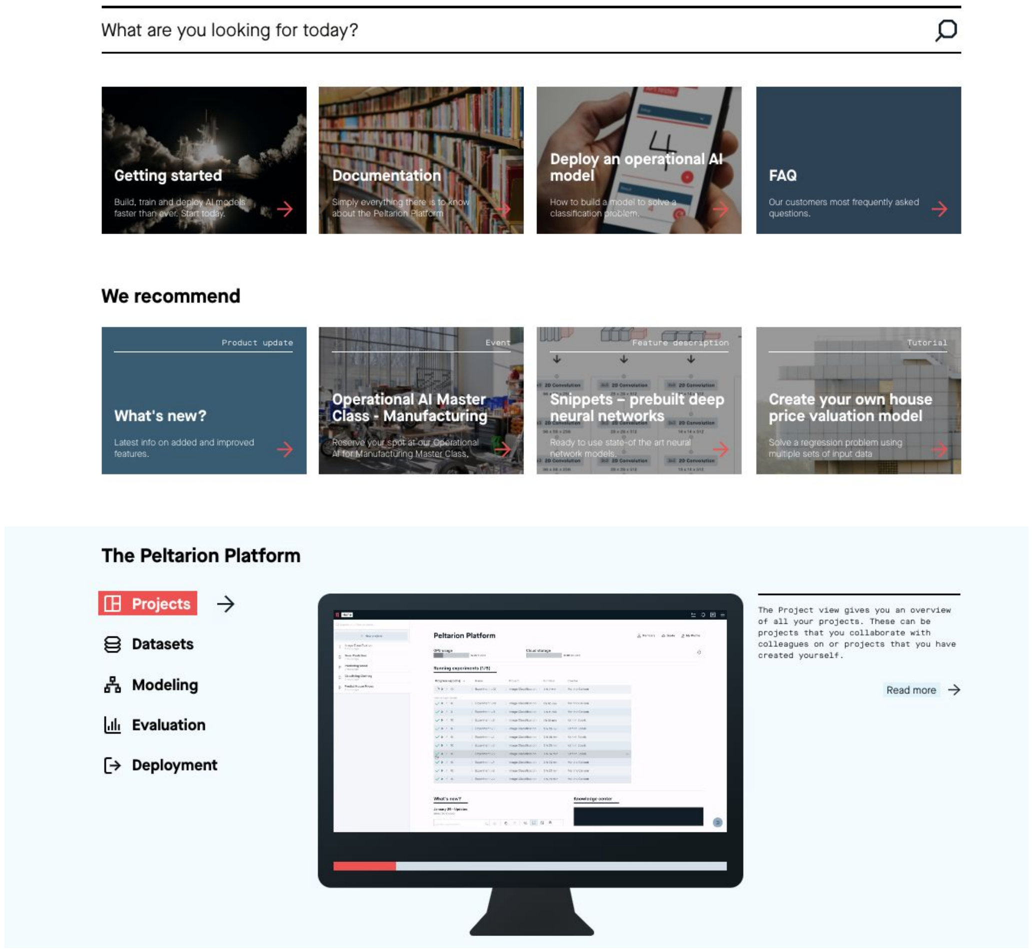
Menu (right upper corner)



- Notification button - provides access to system notifications. For example: Notifications are generated when a user is invited to an organization, accepts the invite, etc
- Organization button - allows access to settings (described further below)
- Knowledge center – Contains all information a user needs to understand the Peltarion Platform, e.g., tutorials, function descriptions, how-tos, etc.
- Support form – to submit feature requests or report an error
- Support email – to submit a support request (support@peltarion.com)
- Terms & conditions – of the Peltarion Platform
- About Peltarion

Knowledge center

The Knowledge center is a repository for all information a user needs to understand the Peltarion Platform, e.g., hands-on tutorials, FAQs and support for troubleshooting.



What are you looking for today?

Getting started
Build, train and deploy AI models faster than ever. Start today.

Documentation
Simply everything there is to know about the Peltarion Platform

Deploy an operational AI model
How to build a model to solve a classification problem.

FAQ
Our customers most frequently asked questions.

We recommend

Product update
What's new?
Latest info on added and improved features.

Event
Operational AI Master Class - Manufacturing
Reserve your spot at our Operational AI for Manufacturing Master Class.

Feature description
Snippets – prebuilt deep neural networks
Ready to use state-of the art neural network models.

Tutorial
Create your own house price valuation model
Solve a regression problem using multiple sets of input data

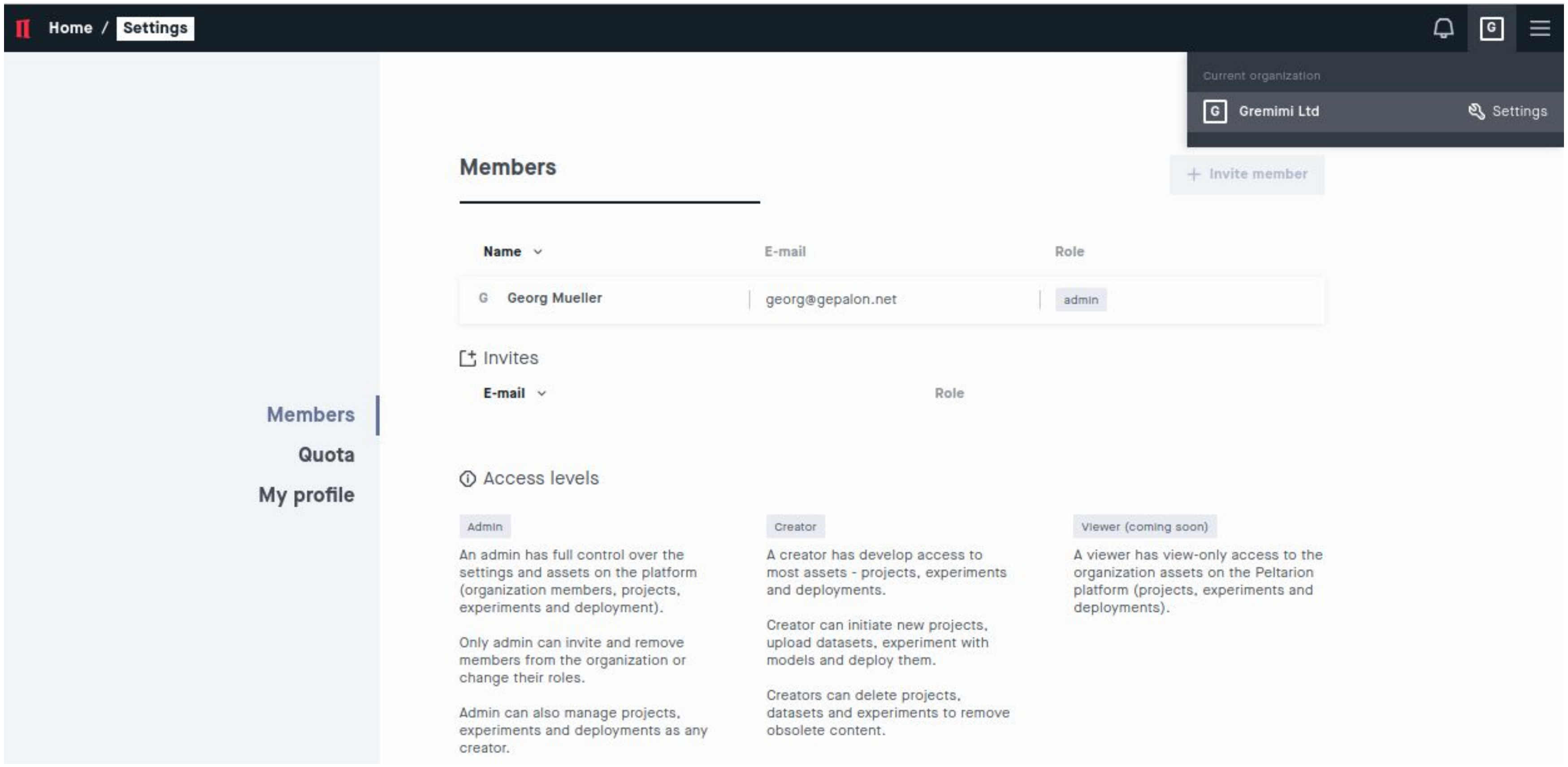
The Peltarion Platform

The Project view gives you an overview of all your projects. These can be projects that you collaborate with colleagues on or projects that you have created yourself.

Read more

Organization settings view

In the Organization settings view, other users can be invited to the solution and also set the access level e.g., Admin, Creator or Viewer. The user can also see the available quota for the system, e.g., available GPU time and memory.



Members

Name	E-mail	Role
G Georg Mueller	georg@gepalon.net	admin

Invites

E-mail	Role

Access levels

Admin

An admin has full control over the settings and assets on the platform (organization members, projects, experiments and deployment).

Only admin can invite and remove members from the organization or change their roles.

Admin can also manage projects, experiments and deployments as any creator.

Creator

A creator has develop access to most assets - projects, experiments and deployments.

Creator can initiate new projects, upload datasets, experiment with models and deploy them.

Creators can delete projects, datasets and experiments to remove obsolete content.

Viewer (coming soon)

A viewer has view-only access to the organization assets on the Peltarion platform (projects, experiments and deployments).

04/ Datasets view

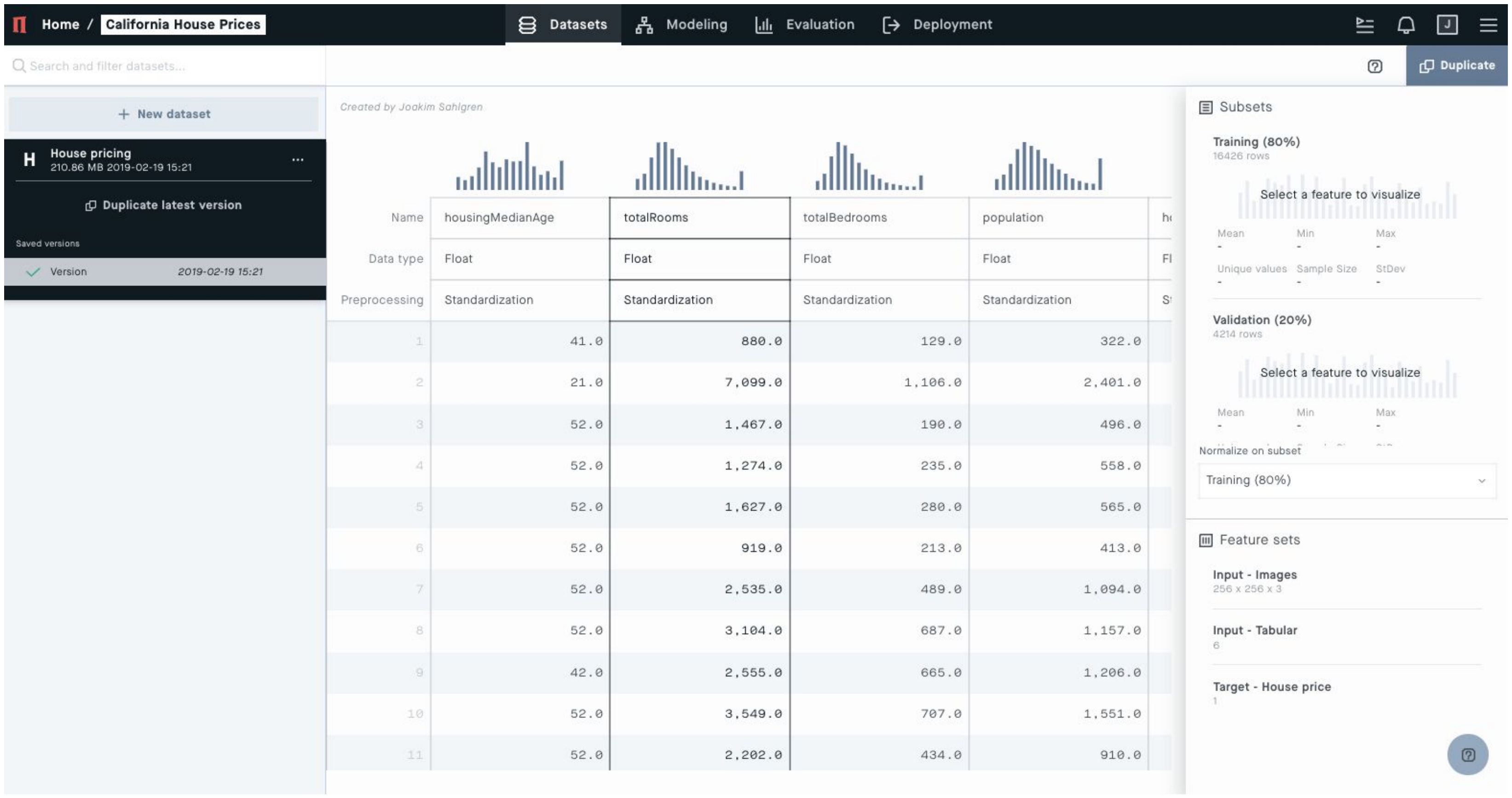
Introduction

The datasets area allows the user to import and manipulate data to run experiments. The view gives a full overview of all the datasets available to the user.

The datasets view gives access to information about the available datasets. You can upload your own datasets or access the ones that have been shared with you.

When uploading several files with many features, you can combine them into datasets to find out how or if things influence each other.

The dataset interface is split into three main sections: The list of available dataset versions on the left, the middle section where all the features of the selected dataset are presented with a small preview of examples, and the right side section where model input and target feature sets and training, as well as validation subsets, are managed.



Created by Joakim Sahlgren

Name	housingMedianAge	totalRooms	totalBedrooms	population
Data type	Float	Float	Float	Float
Preprocessing	Standardization	Standardization	Standardization	Standardization
1	41.0	880.0	129.0	322.0
2	21.0	7,099.0	1,106.0	2,401.0
3	52.0	1,467.0	190.0	496.0
4	52.0	1,274.0	235.0	558.0
5	52.0	1,627.0	280.0	565.0
6	52.0	919.0	213.0	413.0
7	52.0	2,535.0	489.0	1,094.0
8	52.0	3,104.0	687.0	1,157.0
9	42.0	2,555.0	665.0	1,206.0
10	52.0	3,549.0	707.0	1,551.0
11	52.0	2,202.0	434.0	910.0

Training (80%)
16426 rows

Select a feature to visualize

Mean	Min	Max
-	-	-
Unique values	**Sample Size**	**StDev**
-	-	-

Validation (20%)
4214 rows

Select a feature to visualize

Mean	Min	Max
-	-	-

Normalize on subset: Training (80%)

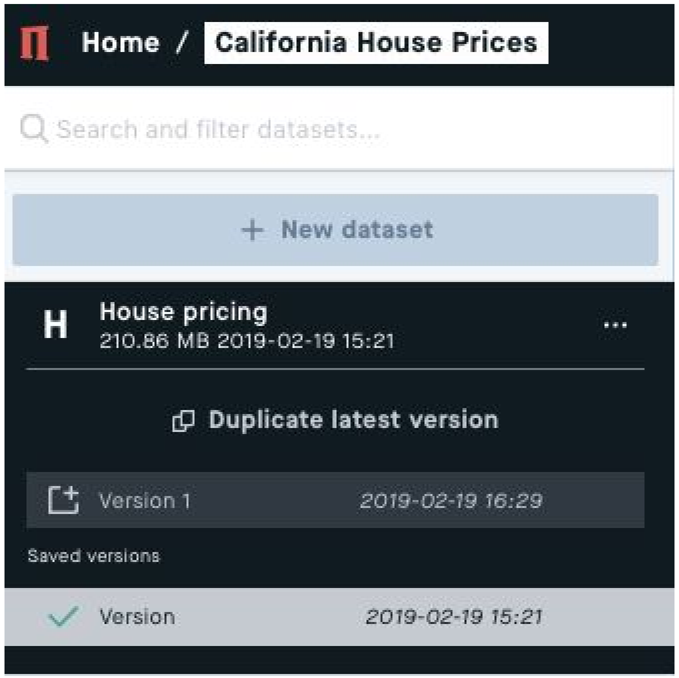
Input - Images
256 x 256 x 3

Input - Tabular
6

Target - House price
1

Dataset navigation (left section)

In this section, a new dataset can be created or an existing dataset duplicated.

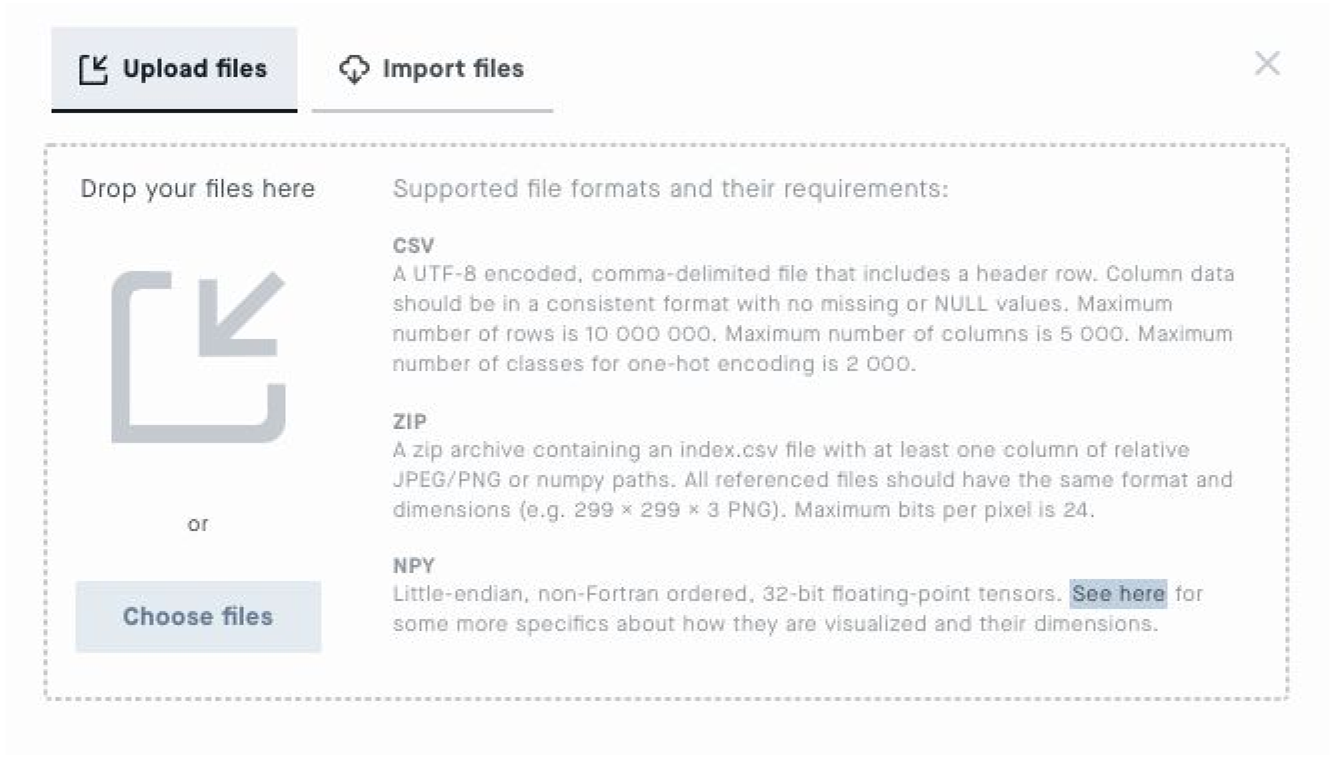

Home / California House Prices
Search and filter datasets...
+ New dataset
H House pricing
210.86 MB 2019-02-19 15:21
Duplicate latest version
Version 1 2019-02-19 16:29
Saved versions
Version 2019-02-19 15:21

Import a file or a data source

To import a file or a bundle of files you can either upload it or import via a URL.

Upload files

In the upload files tab, you can either drag and drop the file(s) to a drop area or click "Choose files" to upload the file(s) from your computer.



Supported file formats and their requirements:

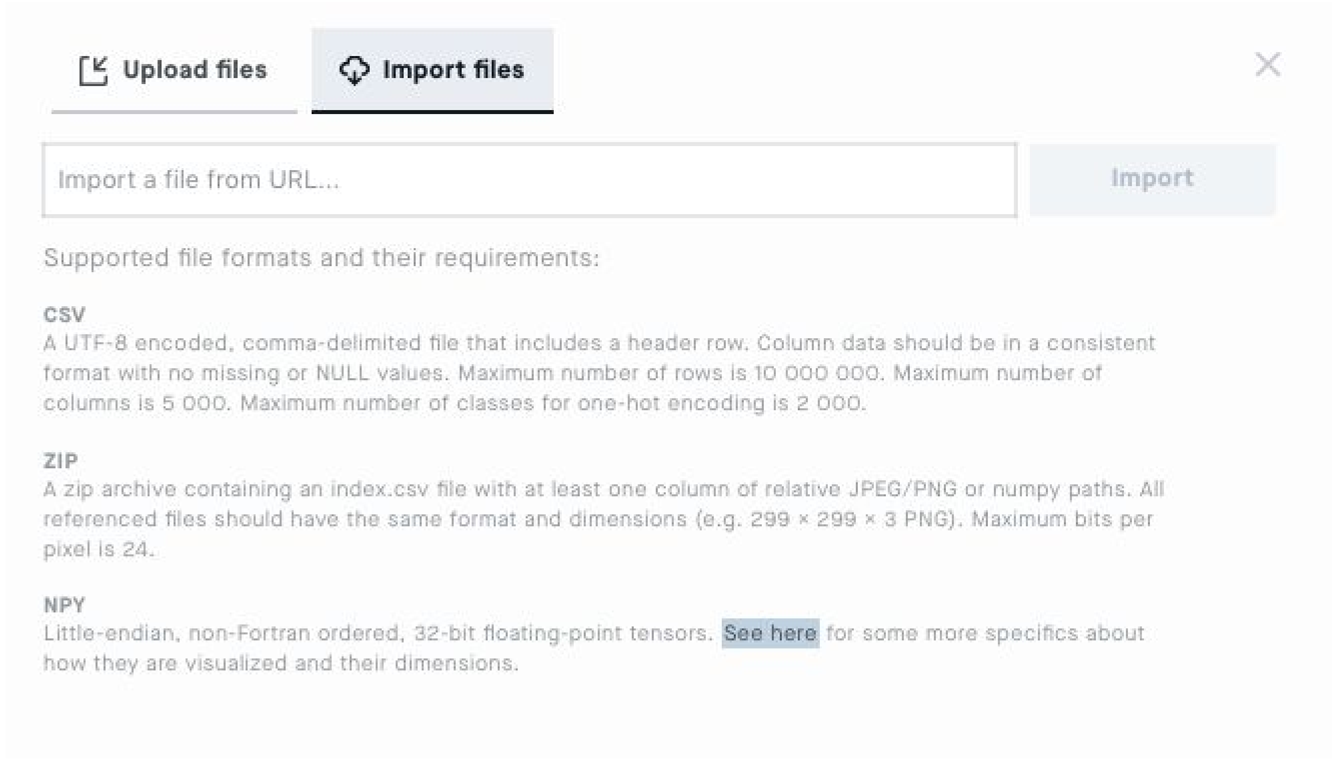
CSV
A UTF-8 encoded, comma-delimited file that includes a header row. Column data should be in a consistent format with no missing or NULL values. Maximum number of rows is 10 000 000. Maximum number of columns is 5 000. Maximum number of classes for one-hot encoding is 2 000.

ZIP
A zip archive containing an index.csv file with at least one column of relative JPEG/PNG or numpy paths. All referenced files should have the same format and dimensions (e.g. 299 × 299 × 3 PNG). Maximum bits per pixel is 24.

NPY
Little-endian, non-Fortran ordered, 32-bit floating-point tensors. See here for some more specifics about how they are visualized and their dimensions.

Import files

In the import files tab, enter the URL to a public data file in the URL field. This is recommended for files greater than 2GB.



Supported file formats and their requirements:

CSV
A UTF-8 encoded, comma-delimited file that includes a header row. Column data should be in a consistent format with no missing or NULL values. Maximum number of rows is 10 000 000. Maximum number of columns is 5 000. Maximum number of classes for one-hot encoding is 2 000.

ZIP
A zip archive containing an index.csv file with at least one column of relative JPEG/PNG or numpy paths. All referenced files should have the same format and dimensions (e.g. 299 × 299 × 3 PNG). Maximum bits per pixel is 24.

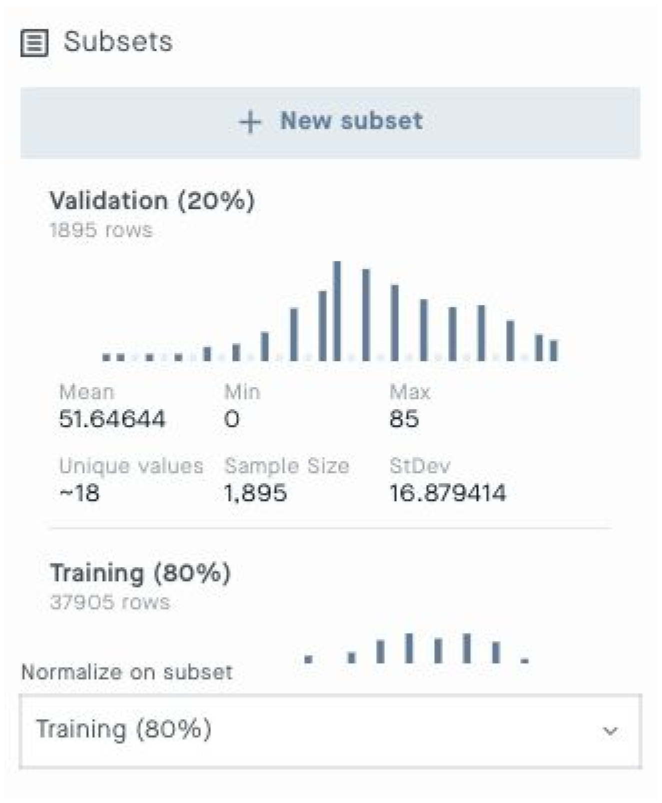
NPY
Little-endian, non-Fortran ordered, 32-bit floating-point tensors. See here for some more specifics about how they are visualized and their dimensions.

Prepare a dataset for use in experiments (right section)

Before you use a dataset in an experiment, you need to prepare the dataset. This preprocessing means that you define:

- Subsets for training and validation.
- Feature sets that include one or more features you want to treat in the same way during modeling.

For example, in a prediction project, create an input set for the independent features to train the model, and a target set of the variable to be predicted.



Validation (20%)
1895 rows

Mean	Min	Max
51.64644	0	85
Unique values	**Sample Size**	**StDev**
~18	1,895	16.879414

Training (80%)
37905 rows

Normalize on subset: Training (80%)

Subset for training and validating

A subset enables you to define a set of examples in the dataset to be used in a specific phase of the modeling process, i.e., training or validation. An example is one row in the dataset matrix.

When importing the datasets to the platform, we create a default 80% training and 20% validation data subset for you by shuffling and partitioning the examples randomly. The % values can be changed from the default, e.g., to 70% training and 30% validation.

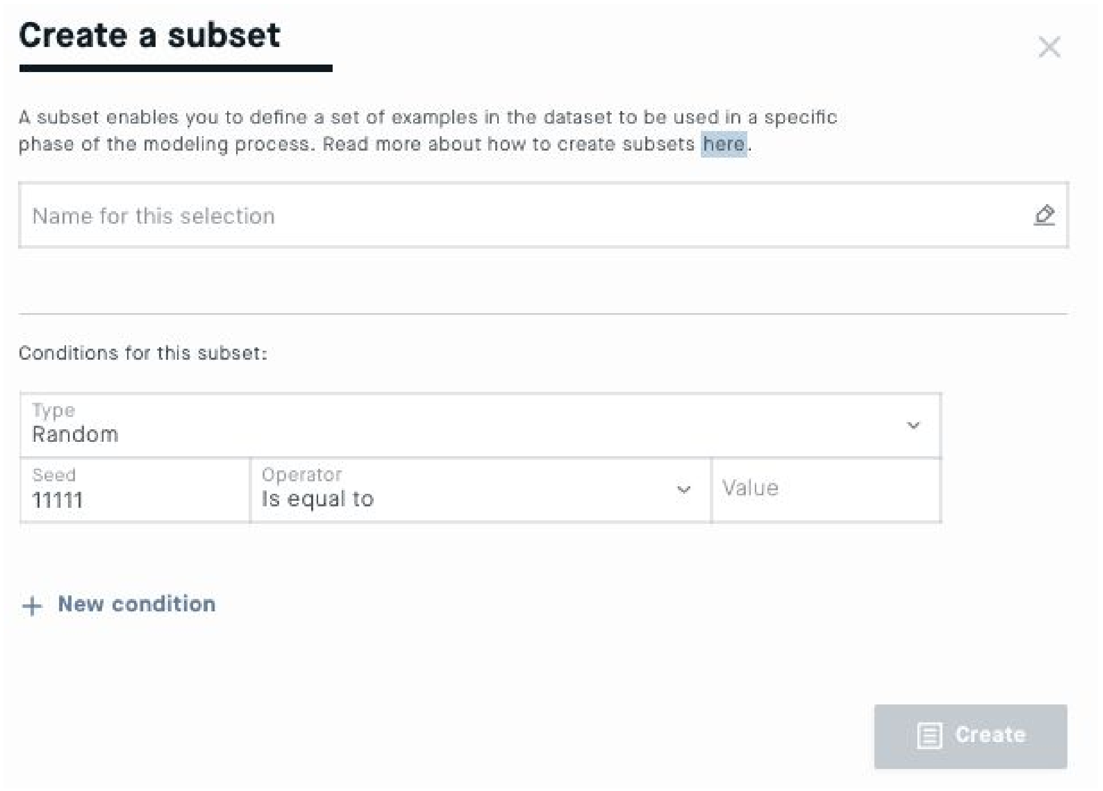
Narrow down a large dataset

You can also create subsets if you want to narrow down a dataset. This is useful when your dataset is very large and you want to use a smaller dataset to train your model.

Create subsets

To create a new subset, click on the upper right "+ New subset."

The following window will appear where you can set the condition for this subset.



Create a subset

A subset enables you to define a set of examples in the dataset to be used in a specific phase of the modeling process. Read more about how to create subsets here.

Name for this selection

Conditions for this subset:

Type: Random

Seed: 11111

Operator: Is equal to

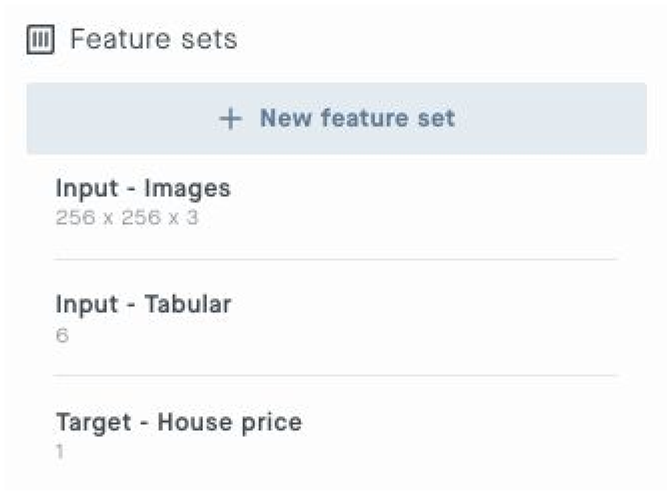
Value

This can be a random rule subset or a conditional rule subset.

When you select a feature, bar charts illustrating the distribution of values within the subsets are displayed. If you hover over a subset set, you'll see its definition rules.

Create feature sets

A feature set is a set of one or more features that you want to treat in the same way during modeling. Features are the columns in the dataset matrix. Every input or target block in the modeling window must be assigned a feature set.



Input - Images
256 x 256 x 3

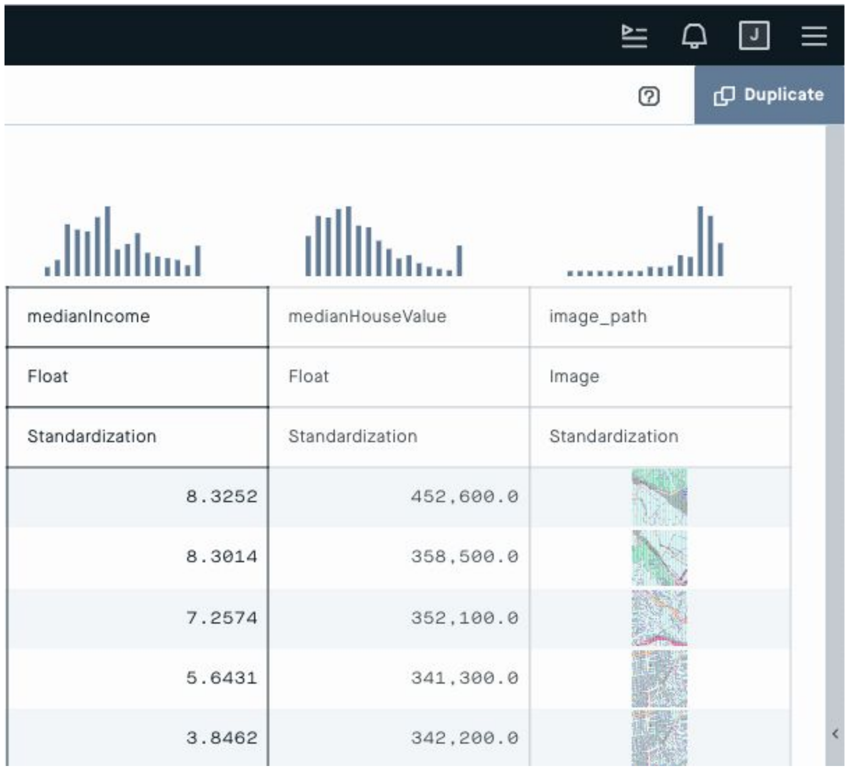
Input - Tabular
6

Target - House price
1

Features that you include in a feature set must have the same shape, e.g., identical number and size of dimensions. For instance, it would not be possible to create a feature set that includes 28x28x1 pixel images and scalar values.

If you hover over a feature set, you'll see a list of included features and the shape.

Dataset area (middle section)



medianIncome	medianHouseValue	image_path
Float	Float	Image
Standardization	Standardization	Standardization
8.3252	452,600.0	
8.3014	358,500.0	
7.2574	352,100.0	
5.6431	341,300.0	
3.8462	342,200.0	

In this area, all of the features and data is displayed. The data type for a specific column can be changed so it represents the correct type. Each type can have a different feature encoding setting, as seen below.

Data types:

- Float
- Float tensor
- Image
- Integer
- String

Feature Preprocessing:

- Standardization
- Min-max normalize
- One-hot encoded
- No Preprocessing

More details can be found here in the knowledge center.

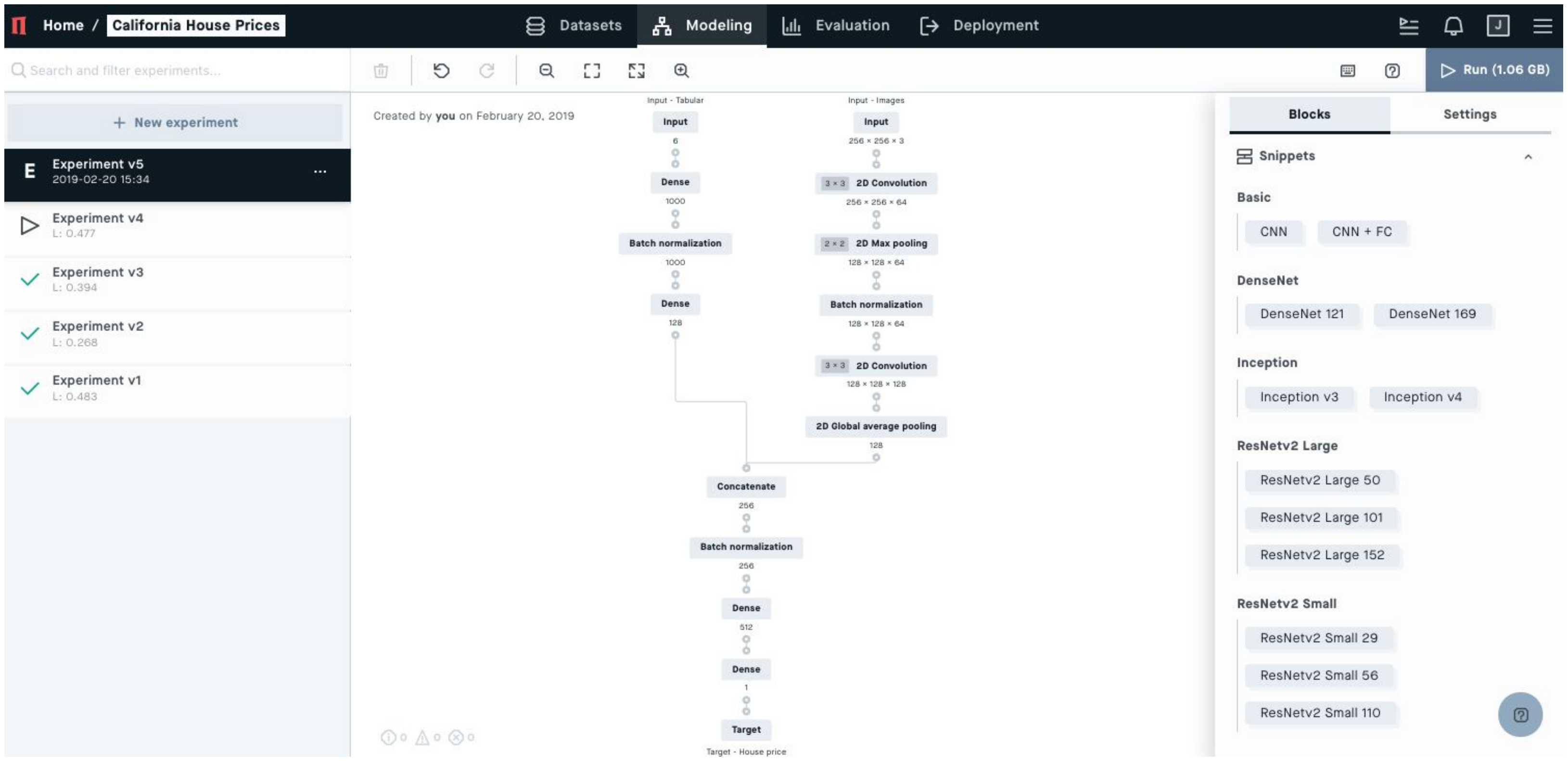
05/ Modeling view

Introduction

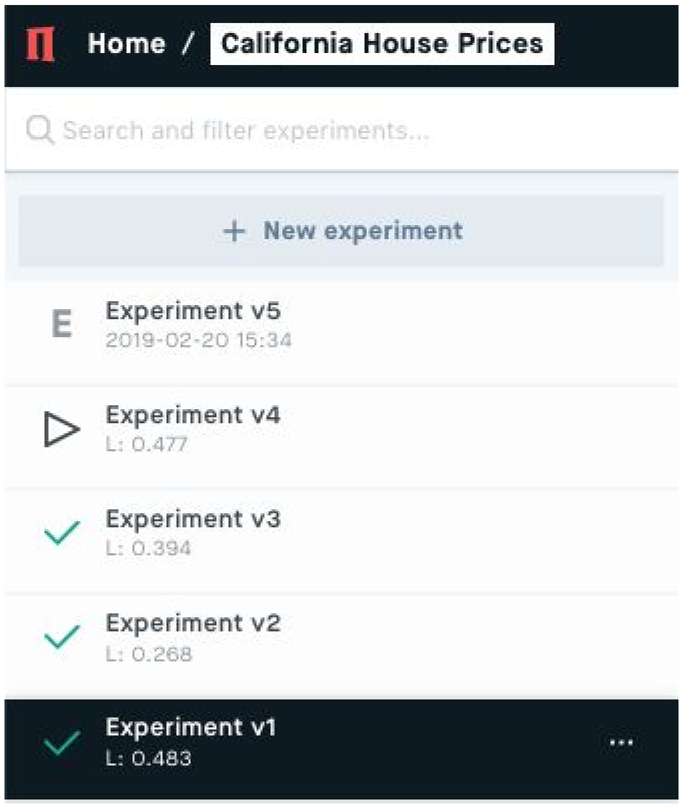
The modeling view allows the user to build an AI deep learning model in one place. You can start from scratch or select from a series of preloaded snippets.

Build a model yourself or work with a colleague on the same one. There are no limits to how you can build the model – it's as easy to make a simple one as it is to make a complex, multi-layered one. There is no coding required, just drag and drop to build your model. No coding means it's simpler and faster as well as fewer bugs.

There are three sections in the modeling view The left section is the experiments navigation section, where all existing model experiments can be viewed and loaded. The middle section is the modeling canvas area, which displays the model in a graphical way by showing the building blocks. The right area consists of the blocks, parameters, snippets and settings.


Home / California House Prices
Datasets
Modeling
Evaluation
Deployment
Search and filter experiments...
Run (1.06 GB)
+ New experiment
Experiment v5
2019-02-20 15:34
Experiment v4
L: 0.477
Experiment v3
L: 0.394
Experiment v2
L: 0.268
Experiment v1
L: 0.483
Created by you on February 20, 2019
Input - Tabular
Input
6
Dense
1000
Batch normalization
1000
Dense
128
Input - Images
Input
256 × 256 × 3
3 × 3 2D Convolution
256 × 256 × 64
2 × 2 2D Max pooling
128 × 128 × 64
Batch normalization
128 × 128 × 64
3 × 3 2D Convolution
128 × 128 × 128
2D Global average pooling
128
Concatenate
256
Batch normalization
256
Dense
512
Dense
1
Target
Target - House price
Blocks
Settings
Snippets
Basic
CNN
CNN + FC
DenseNet
DenseNet 121
DenseNet 169
Inception
Inception v3
Inception v4
ResNetv2 Large
ResNetv2 Large 50
ResNetv2 Large 101
ResNetv2 Large 152
ResNetv2 Small
ResNetv2 Small 29
ResNetv2 Small 56
ResNetv2 Small 110

Experiments navigation (left section)



In this section, you can select the experiments you want to view and model.

You can also duplicate, rename or delete an experiment. If an experiment is running you can also pause it.

Search and filter for experiments

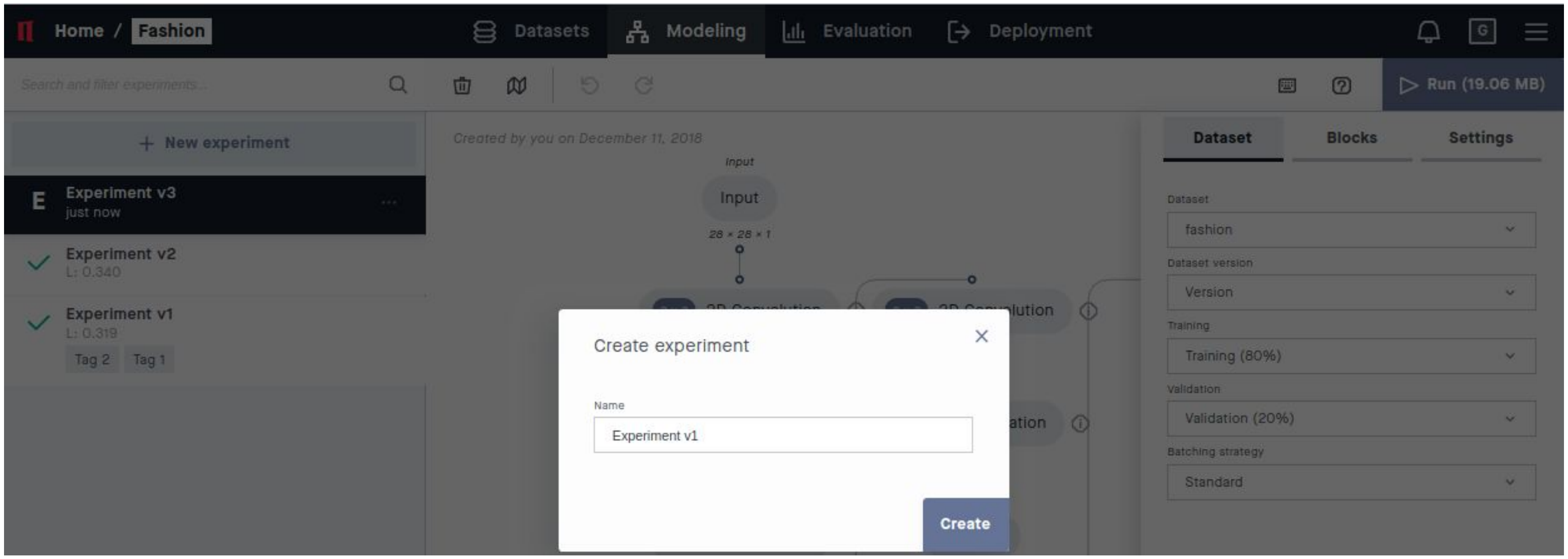
You can do a nested search and filter experiments based on:

- Created by
- Experiment status
- Name
- Loss function
- User-defined tags

You can also download an experiment as a Keras-compatible .h5 file.

The left pane lets you keep track of your experiments and launch new ones. Previously run and currently running experiments are locked; if you wish to change and tweak a locked experiment, you have to duplicate it. A user can only modify their own experiments.

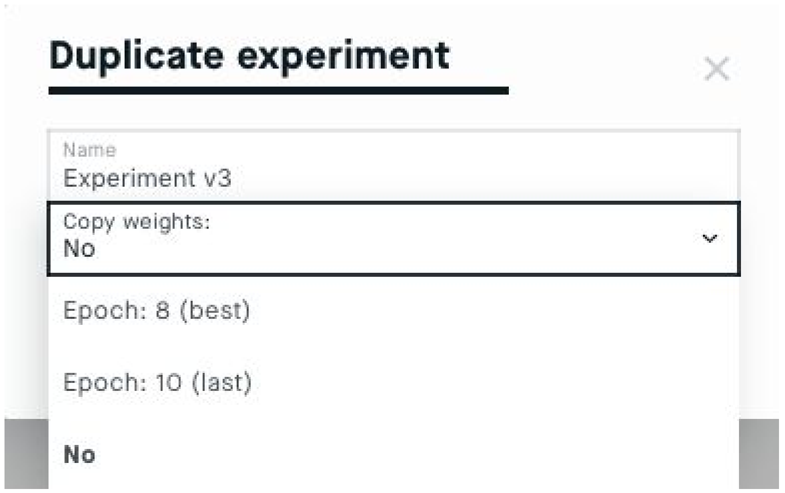
To build a new model, click on "New experiment" and give the new experiment a name.



The canvas is now ready to create a new model. This can be achieved by either selecting individual blocks, or by or by selecting premade parts of models called snippets. These snippets can help reduce the time it takes to build a model.

Duplicating an experiment

To continue to build on an existing experiment, click the ellipsis (...) to open the Experiment options menu and then select "Duplicate."
Name the new experiment and then decide if and from which checkpoint you want to copy the weights.



Duplicate experiment

Name
Experiment v3

Copy weights:
No

Transfer learning

Copy with weights can be used if you want to transfer knowledge from one of your models to another model. In machine learning terms, this is called "transfer learning," and can be useful if you have trained a complicated neural network on a large dataset and want to reuse parts of that knowledge to solve other related problems.

Say, for example, you want to learn if an image depicts a car or a truck. First, you need to train your model to learn that a car has wheels. While the model was never given information about wheels specifically, it was inferred through lots of pictures of cars and the label "car." The model represents this information in the weights that it learns.

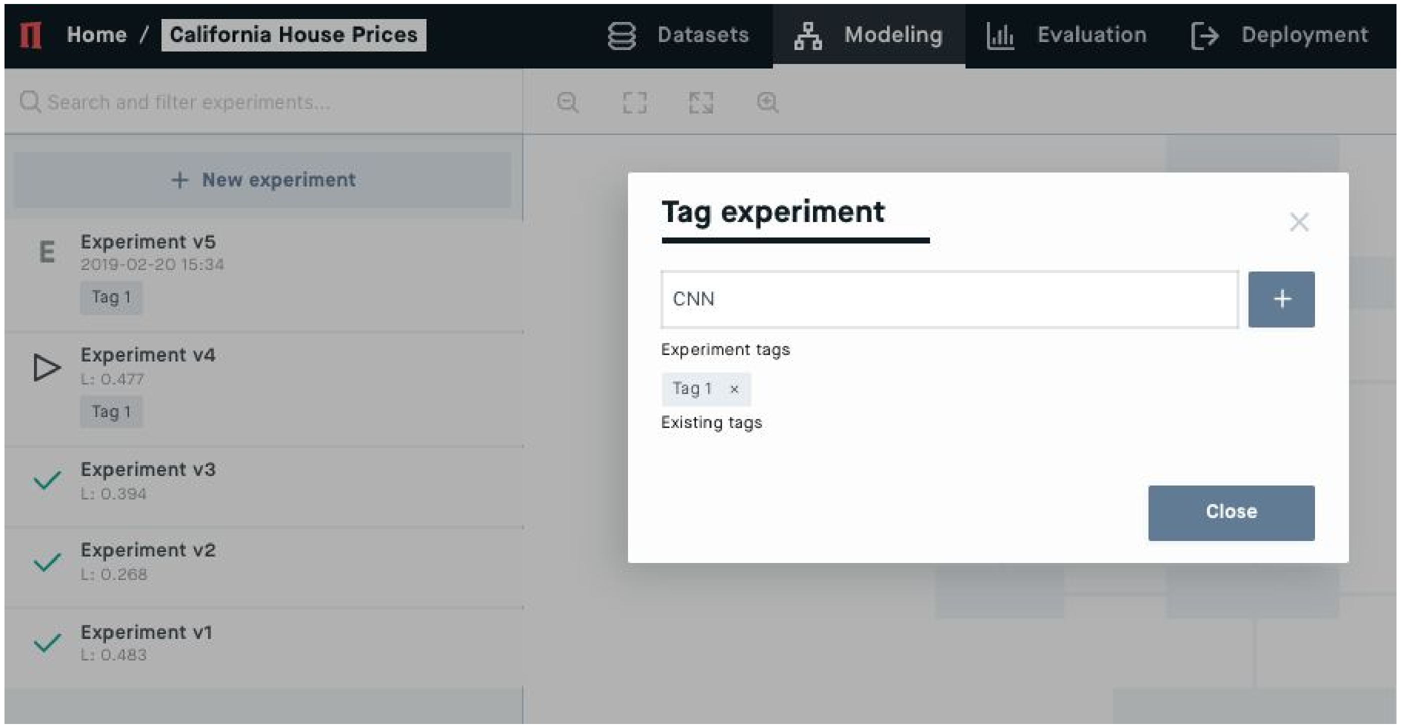
The model that has been duplicated with the copied weights can now easily be trained to recognize trucks since they also have wheels. Your new model could also learn the difference between a car and a truck without needing to be trained on cars.

Note that is also possible to copy individual blocks and their weights from a model.

Experiment tagging

Tagging is used to arbitrarily group and ungroup experiments in the Experiments list in both the Modeling and Evaluation views. This function is very useful not only for more easily locating experiments through the search field, but also helps facilitate custom comparisons in the Evaluation view.

To add, modify or delete a tag click the ellipsis (...) to open the Experiment options menu and then “Edit tags”. The “Tag experiment” pop-up will open:



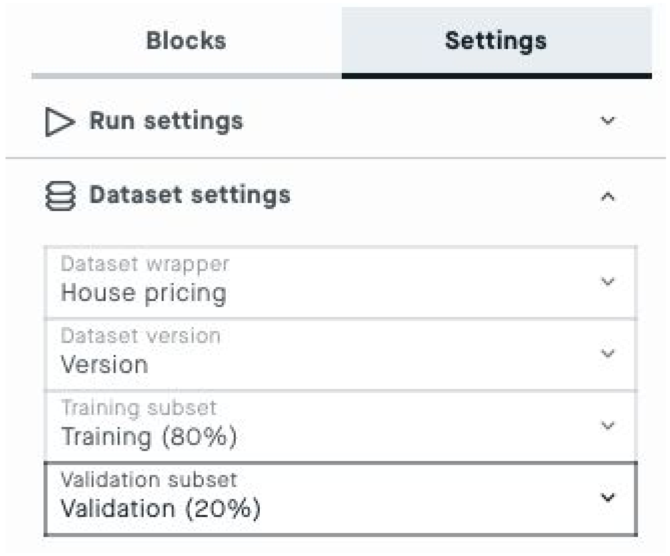
- To add a tag, fill in the input field and click on the Plus button.
- To remove an existing, tag click on the tag listed under Experiment tags.
- To add an existing tag, click on the tag listed under Existing tags.

Note: If an experiment is duplicated then the tags are also copied over to the new experiment.

Blocks, and Settings (right section)

In this section, you can select a version of a dataset, select blocks and snippets, and update optimizer and compiler options.

Dataset settings



Dataset wrapper: House pricing
Dataset version: Version
Training subset: Training (80%)
Validation subset: Validation (20%)

When you have created or duplicated an experiment, it is recommended that you first select a dataset, dataset version, training subset and validation subset.

When you add building blocks to your model. The parameters of each block will be validated against the shape of the features in your selected dataset. This facilitates troubleshooting while you are building your model.

By adding tags to your experiments, you can also compare the performance of models trained on specific versions or subsets.

Note that when you create a new experiment, the Platform will automatically select the latest dataset and version. Make sure to change this if you want to train the model on a different version.

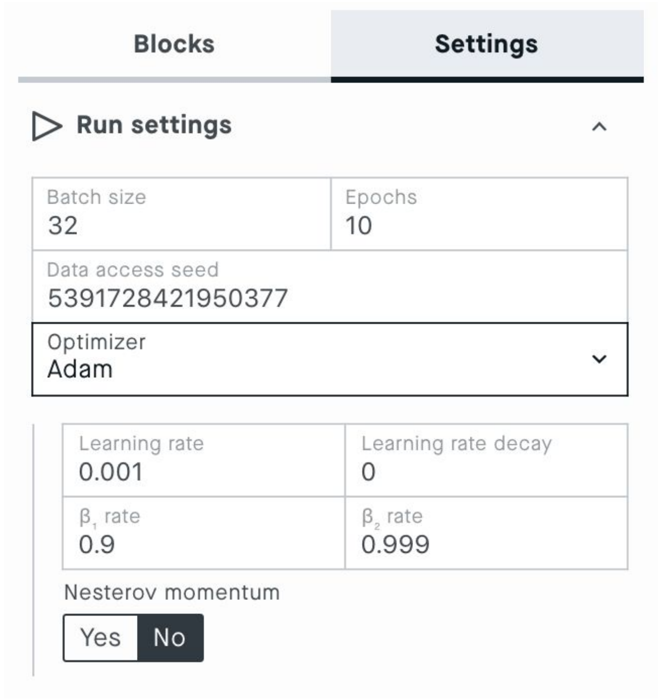
Run settings

To achieve a well-performing model, you'll typically need to update the parameters in the "Run settings".

- Batch size - The number of training examples in one iteration
- Epochs – An epoch is when all training data has run through the the model once; you might get a better result if you let the experiment run for more epochs
- Data access seed - Random integer to shuffle the data
- Optimizer – This is how the system optimizes the loss with respect to the weights of the network

Several different optimizers are available, for example, Adam, SGD, and RMSprop. The following parameters are common for all optimizers:

- Learning rate – a high learning rate means faster adaptation but at the cost of higher variance in the training; a low learning rate means that you might get stuck at a local minimum
- Learning rate decay

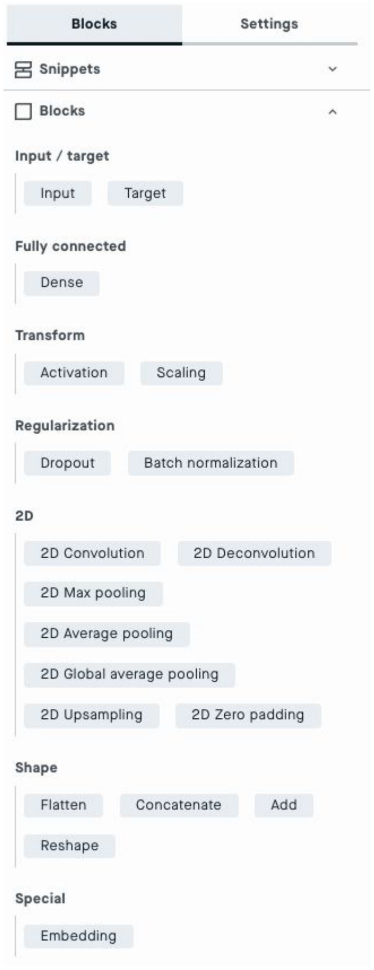


Batch size 32
Epochs 10
Data access seed 5391728421950377
Optimizer Adam
Learning rate 0.001
Learning rate decay 0
β_1 rate 0.9
β_2 rate 0.999
Nesterov momentum Yes [✓ No]

Blocks and parameters

Blocks and parameters go hand in hand. Each block has a series of parameters which have to be set. The following blocks are available:

- Input / Output
    - Input
    - Target
- Fully Connected
    - Dense
- Transform
    - Activation
    - Scaling
- Regularization
    - Dropout
    - Batch normalization
- 2D
    - 2D Convolution
    - 2D Deconvolution
    - 2D Max pooling
    - 2D Average pooling
    - 2D Global average pooling
    - 2D Upsampling
    - 2D Zero padding
- Shape
    - Flatten
    - Concatenate
    - Add
    - Reshape
- Special
    - Embedding



Activation functions

The activation function calculates what value a node should give as an output.

The activation function you choose depends on your model and what you want to achieve. You want the activation function to have a linear or almost linear part since this is more cost-effective to compute. However, the nonlinear part is what gives the multi-layer network more power than a single layer network.

The following activation functions are available on the Peltarion Platform:

- Linear
- ReLU
- Sigmoid
- Softmax
- TanH

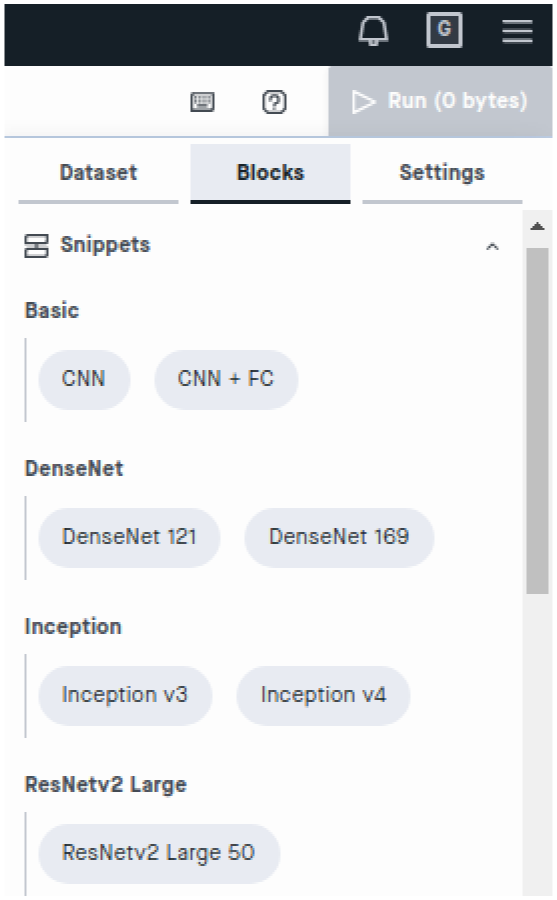
Loss functions

Loss indicates the magnitude of error your model made on its prediction. It's a method of evaluating how well your algorithm models your dataset. If your predictions are totally off, your loss function will output a higher number. If they're pretty good, it'll output a lower one. The following loss functions are available on the platform:

- Binary crossentropy
- Categorical crossentropy
- Mean absolute error
- Mean squared error
- Mean squared logarithmic error
- Poisson
- Squared hinge

Snippets

Snippets are predesigned models. These models can be modified as required by adding or deleting blocks and changing the parameters accordingly. The following snippets are currently available:

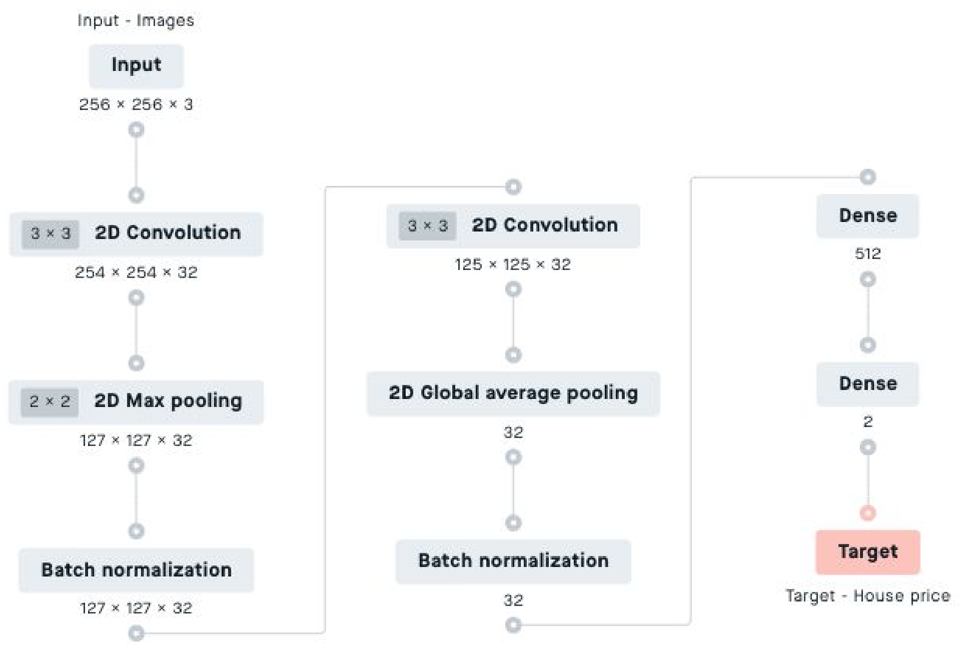


- Basic
  - CNN
  - CNN + FC
- DenseNet
  - DenseNet 121
  - DenseNet 169
- Inception
  - Inception v3
  - Inception v4
- ResNetv2 Large
  - ResNetv2 Large 50
  - ResNetv2 Large 101
  - ResNetv2 Large 152
- ResNetv2 Smal
  - ResNetv2 Small 29
  - ResNetv2 Small 56
  - ResNetv2 Small 110
- VGG
  - VGG16
  - VGG19
- Segmentation
  - Tiramisu
  - U-Net

Please note that currently the snippets are not pre-trained and training a deep network on a large set of data may consume a significant amount of GPU power. Detailed documentation on snippets can be found on the knowledge center.

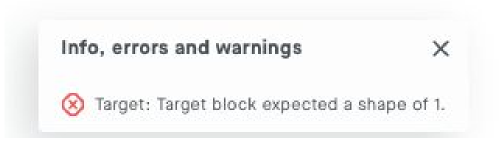
Modeling canvas (middle section)

The modeling canvas area is where existing models can be viewed, new models can be created or models can be modified after duplication. A model which has been locked can only be viewed. If you want to modify an existing model it needs to be duplicated.



If a part of your model is missing or incorrect, the relevant blocks will be highlighted.

All errors messages, information messages and warnings are shown in the Information center pop-up in the Modeling view bottom left corner. Click on a message and you'll be guided directly to the problematic area.



Target: Target block expected a shape of 1.

Controls to navigate the Modeling canvas

If a model extends beyond the visible part of the canvas, you can pan to bring it into view. You may also use the zoom controls to view a small or larger section of the model,

Zoom to fit. This will expand the model to fit the whole Modeling canvas.

Zoom to 100%. Click this icon to zoom into the center of the part of the model that is shown in the Modeling canvas.

Zoom out.

Zoom in.

Viewing and modifying parameters of blocks

To check which parameters have been set for a specific block, go into the model and click on the block. You'll then be able to view the parameters on the right pane. If the model is locked, you can only view these parameters. If the model is duplicated, all parameters can be modified.

You can view the settings that are common to multiple blocks by shift-clicking the blocks.

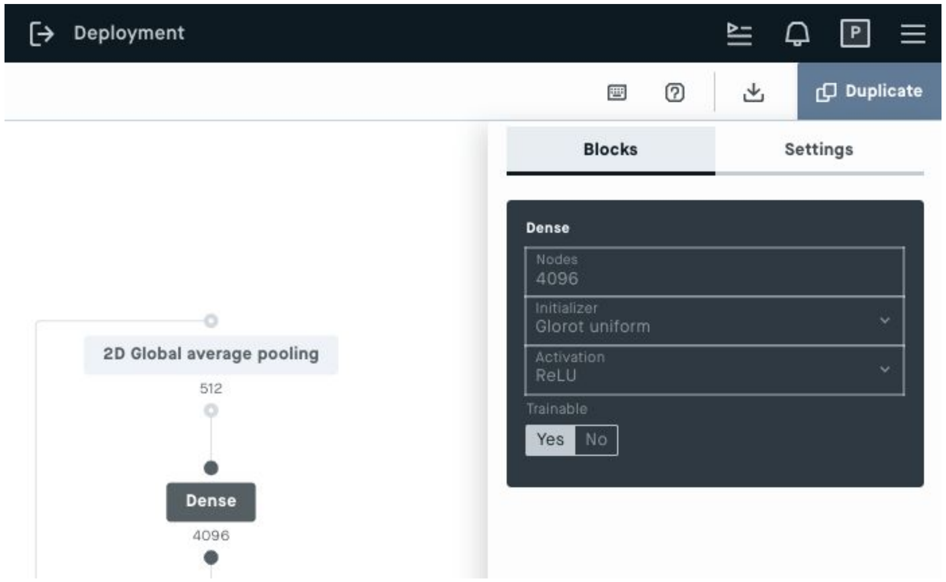
Add blocks to an existing model

Select an unconnected block on the canvas and then select the new block that you want to add in the right section. The new block will appear under the previous block on the canvas. A connection between the two blocks is created automatically.

If a block has not been selected on the canvas the new block will appear at the bottom of the model. Drag the block to where you want it in the model. Connect the new block to another block by using hold-and-drag between the connecting points.

If needed, remove a connector or block by selecting it and then pushing the delete or backward key on the keyboard or clicking the "Delete selection" button in the GUI.

Select the newly added block. In the parameter section, set the parameters for the block.



Dense

Nodes
4096

Initializer
Glorot uniform

Activation
ReLU

Trainable
[✓ Yes] No

Edit parameters

When selecting a block on the Modeling canvas, you can adjust the block parameters in the Blocks tab.

When you select more than one block, you can change their common settings.

Copy blocks with weights

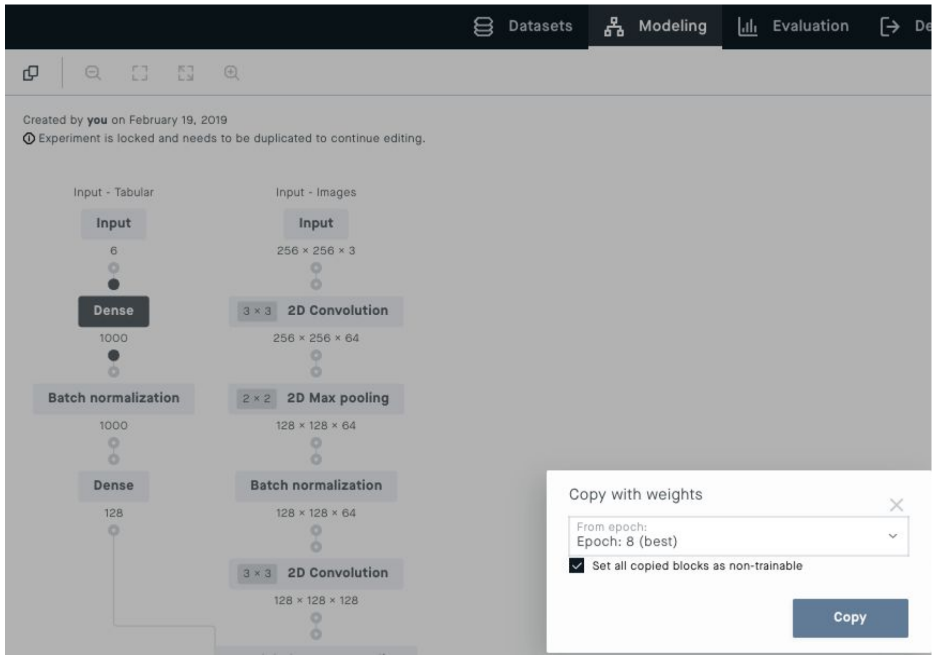
Select the blocks you want to copy (i.e., from another experiment that has been run).

Click “Copy with weights” in the left upper corner of the canvas area.

Select from which epoch you want to copy the weights and if you want to set the blocks as non-trainable.

Non-trainable means that the blocks’ weights will not be updated during the training of the new experiment. This is a good choice the first time you copy blocks.

Trainable means that the blocks can be trained with, for example, new data; that is, the weights associated with the blocks will be updated during training of the new experiment.



Created by you on February 19, 2019
Experiment is locked and needs to be duplicated to continue editing.

Copy with weights
From epoch: Epoch: 8 (best)
☑ Set all copied blocks as non-trainable

It is recommended you start with non-trainable, otherwise, there is a risk of over-fitting, i.e., the experiment is too closely fitted to a limited set of data points. You can also manually change the trainable setting for some blocks in the parameters section.

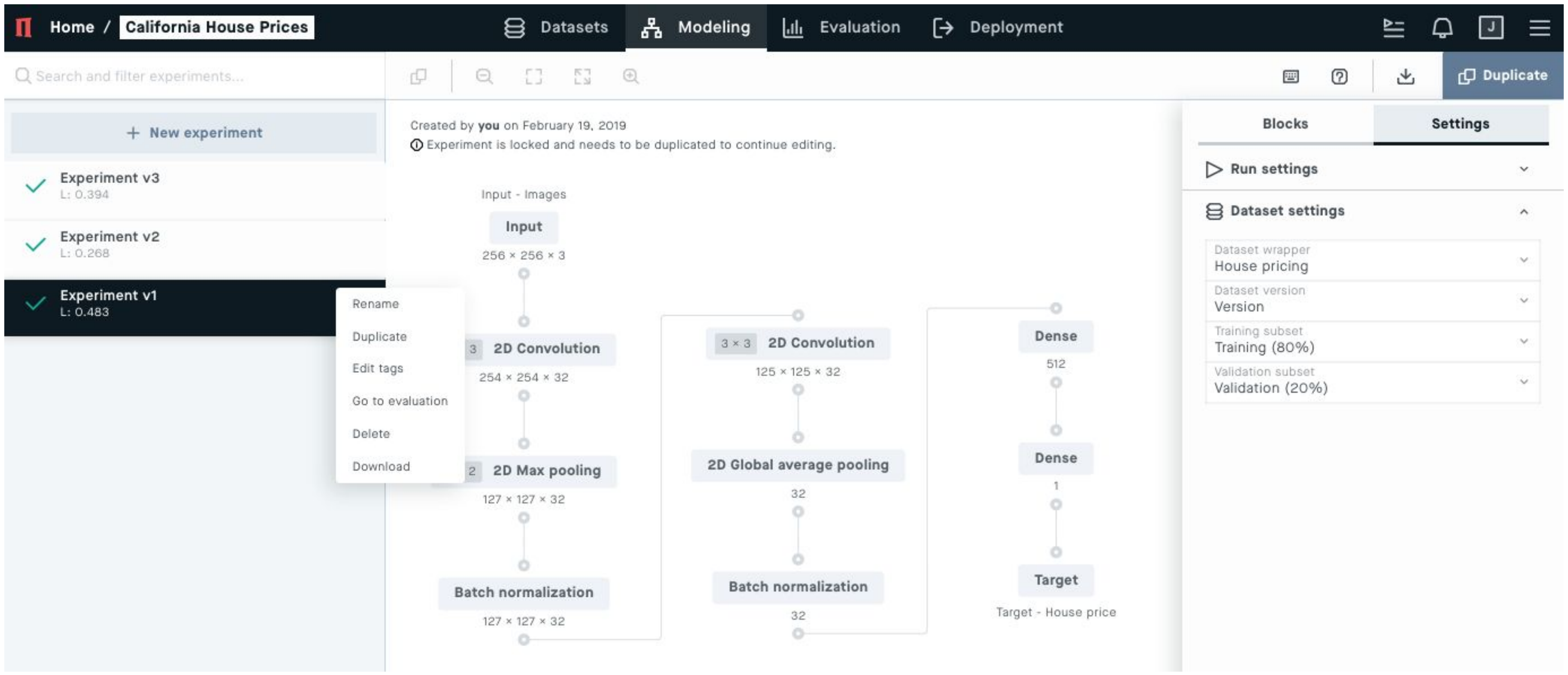
Be aware of the origins of the weights when copying and reusing blocks with weights. There is nothing in the platform preventing some or all of the samples used to train the weights in the validation set of the new model. As a result, your model may show that it generalizes better than it actually does.

Click “Copy.” and then paste the copied blocks in a new experiment.

Download and deploy an .h5 model with weights

There are two options to download and deploy the trained model in the modeling view, One option is to navigate to the Modeling view and press on the “download model button” in the upper right corner.

The second option is to click the ellipsis (...) next to a selected experiment and then “Download”.



Created by you on February 19, 2019

Experiment is locked and needs to be duplicated to continue editing.

Dataset wrapper: House pricing

Dataset version: Version

Training subset: Training (80%)

Validation subset: Validation (20%)

This will download the trained model with weights from the epoch with the best validation loss as an .h5 file for deployment in Keras-based Python programs.

Note: This solution does not currently take pre/post processing into account, which means that any normalization or one-hot encoding will not be part of this model. Currently, these operations and the metadata used to apply them is not exposed. Therefore, for users that rely on deploying .h5 files from their platform in their own systems, we recommend they do all preprocessing, such as normalization and one-hot encoding, outside the platform.

When downloading the .h5 file, make sure to check that the right Keras version is used to load the model and run predictions. An example of how you can load an .h5 model is explained in the Keras documentation.

06/ Evaluation view

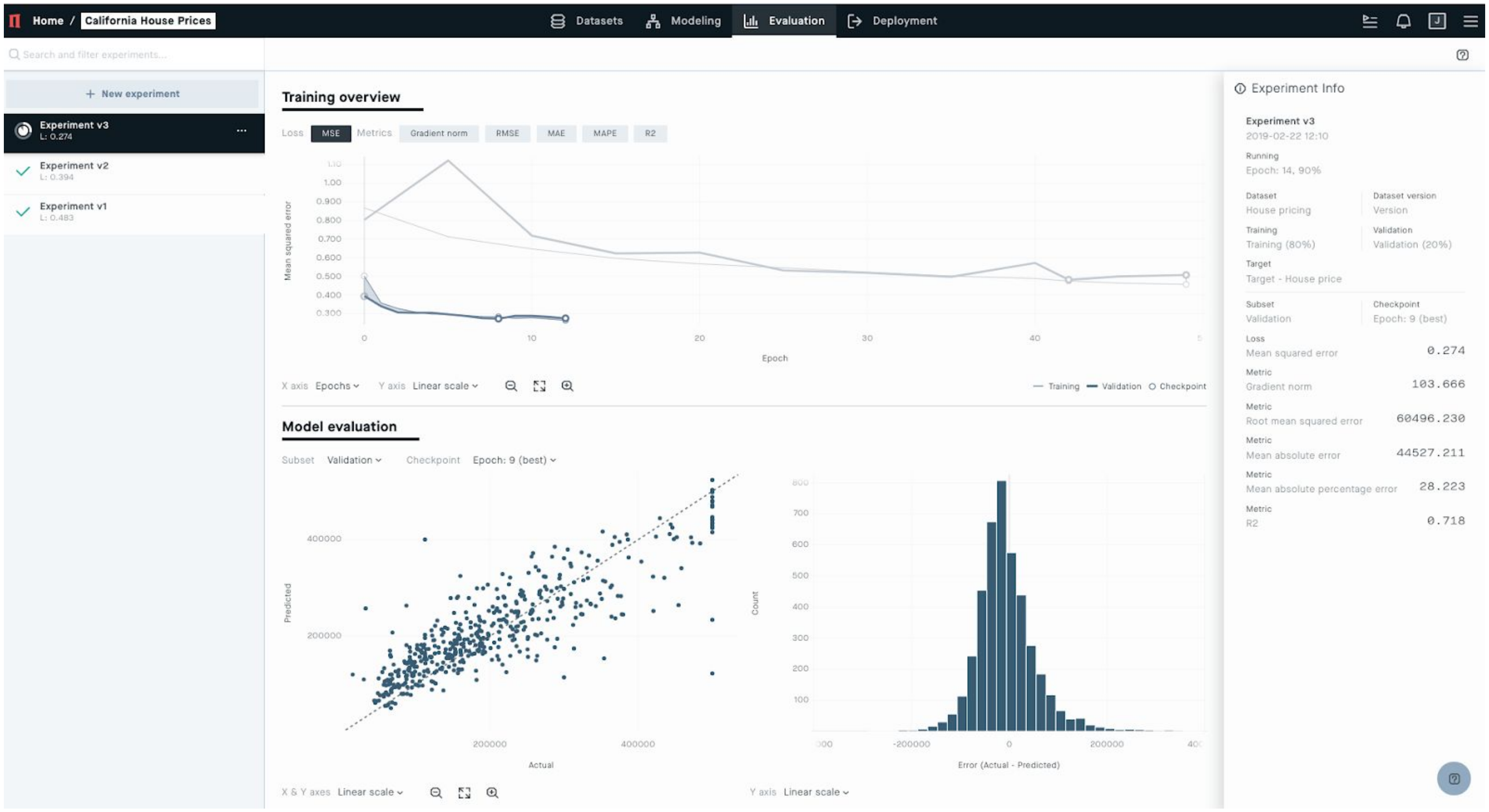
Introduction

In the evaluation view, you can see in real time how the AI model is performing as it's learning from the data.

You can verify if your AI model is doing what you want it to do. If not, you can stop the experiment, go back to the modeling view, tweak it in a few clicks and run it again. You can see the results of multiple experiments at the same time, so it's easy to compare different models.

The evaluation view lets you analyze in real time if the experiment is going in the right direction or not. It's quick and easy to compare experiments to find which one performs best.

The graph view will look different depending on the problem to be addressed. The first screenshot below shows the evaluation for a typical regression problem displaying a scatter plot, and the second screenshot is a classification problem displaying a confusion matrix.



Experiment v3
L: 0.274

Experiment v2
L: 0.394

Experiment v1
L: 0.483

Subset: Validation
Checkpoint: Epoch: 9 (best)

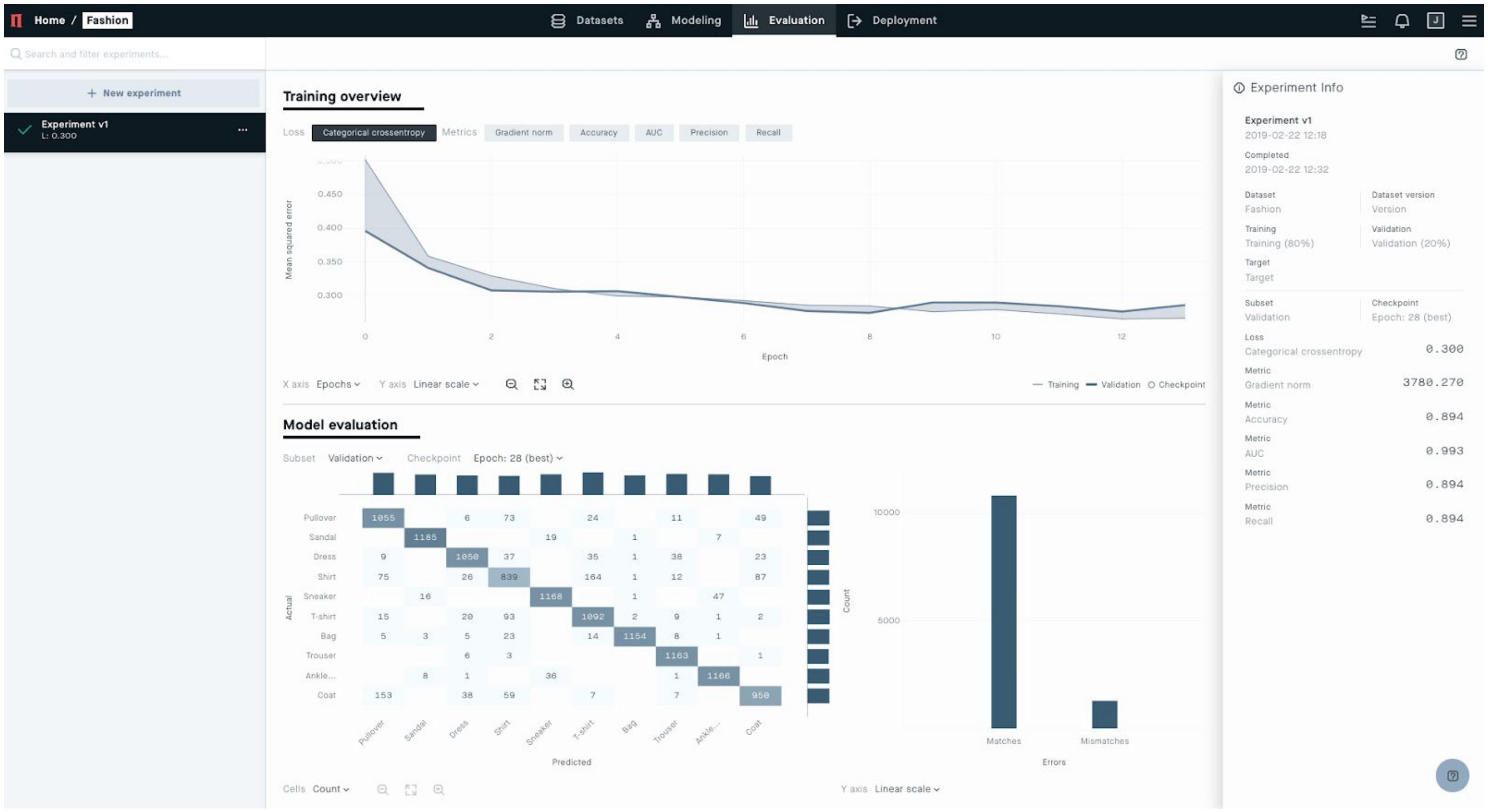
Experiment Info

Experiment v3
2019-02-22 12:10

Running
Epoch: 14, 90%

Dataset House pricing	Dataset version Version
Training Training (80%)	Validation Validation (20%)
Target Target - House price	
Subset Validation	Checkpoint Epoch: 9 (best)

Loss Mean squared error	0.274
Metric Gradient norm	103.666
Metric Root mean squared error	60496.230
Metric Mean absolute error	44527.211
Metric Mean absolute percentage error	28.223
Metric R2	0.718



Actual \ Predicted	Pullover	Sandal	Dress	Shirt	Sneaker	T-shirt	Bag	Trouser	Ankle...	Coat
Pullover	1055		6	73		24		11		49
Sandal		1185			19		1		7	
Dress	9		1050	37		35	1	38		23
Shirt	75		26	839		164	1	12		87
Sneaker		16			1168		1		47	
T-shirt	15		20	93		1092	2	9	1	2
Bag	5	3	5	23		14	1154	8	1	
Trouser			6	3				1163		1
Ankle...		8	1		36			1	1166	
Coat	153		38	59		7		7		950

Experiment Info

Experiment v1
2019-02-22 12:18

Completed
2019-02-22 12:32

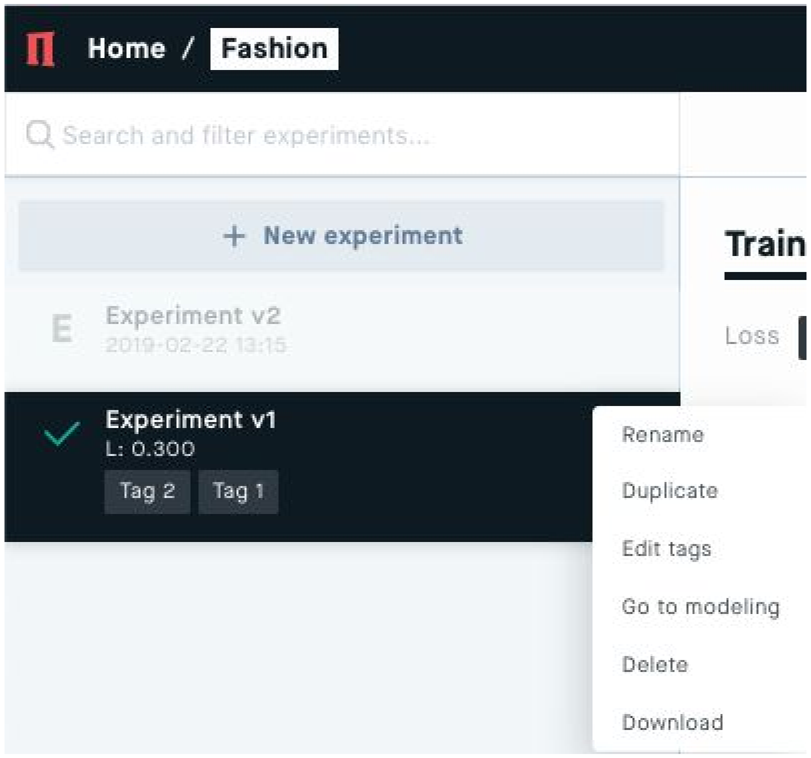
Dataset: Fashion	Dataset version: Version
Training: Training (80%)	Validation: Validation (20%)
Target: Target	
Subset: Validation	Checkpoint: Epoch: 28 (best)

Loss	Categorical crossentropy	0.300
Metric	Gradient norm	3780.270
Metric	Accuracy	0.894
Metric	AUC	0.993
Metric	Precision	0.894
Metric	Recall	0.894

Experiments navigation (left section)

In this section, you can select which experiments you want to view.

You can also duplicate, rename or delete an experiment and edit tags. If an experiment is running you can also pause it.



Search and filter for experiments

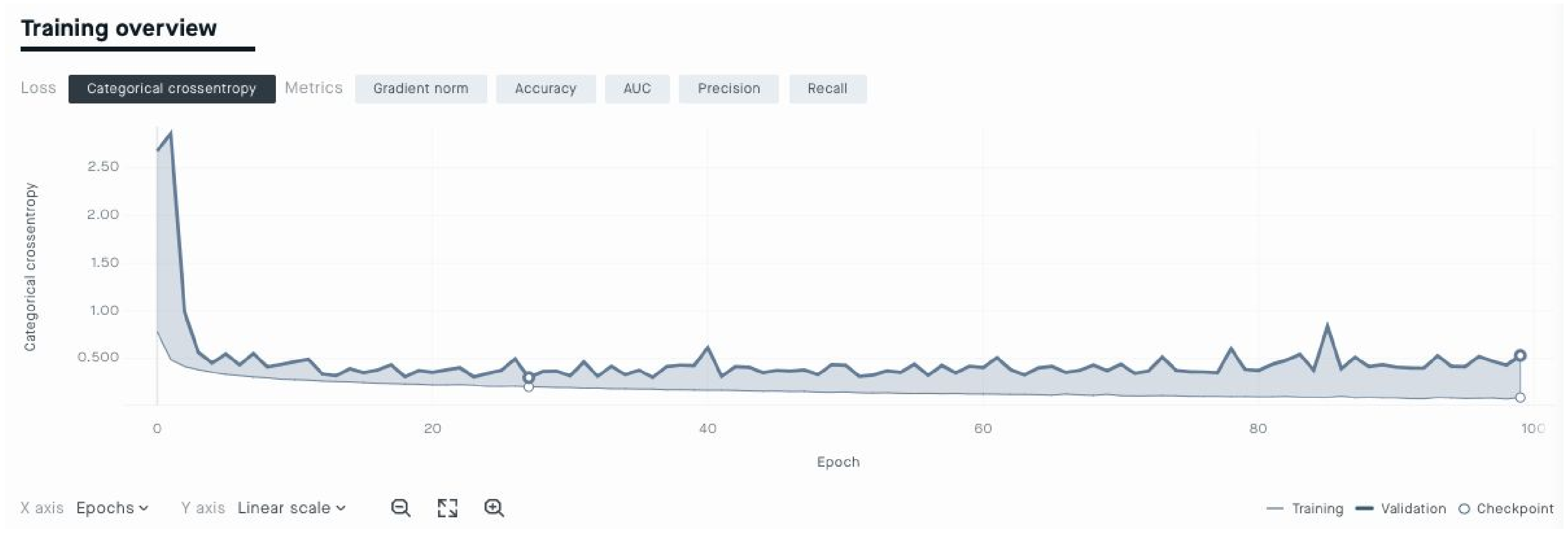
You can do a nested search and filter experiments based on:

- Created by
- Experiment status
- Name
- Loss function

You can also download an experiment as a Keras-compatible .h5 file for running a forward pass.

Evaluation canvas (middle section)

The evaluation canvas contains a training overview and model evaluation graphs. The evaluation overview graphs will look different for regression and classification problems. The performance metrics calculated during training and shown on the training overview graph depend on the problem type (regression or classification metrics).



In the training overview, you can select different settings for the X axis and Y axis.

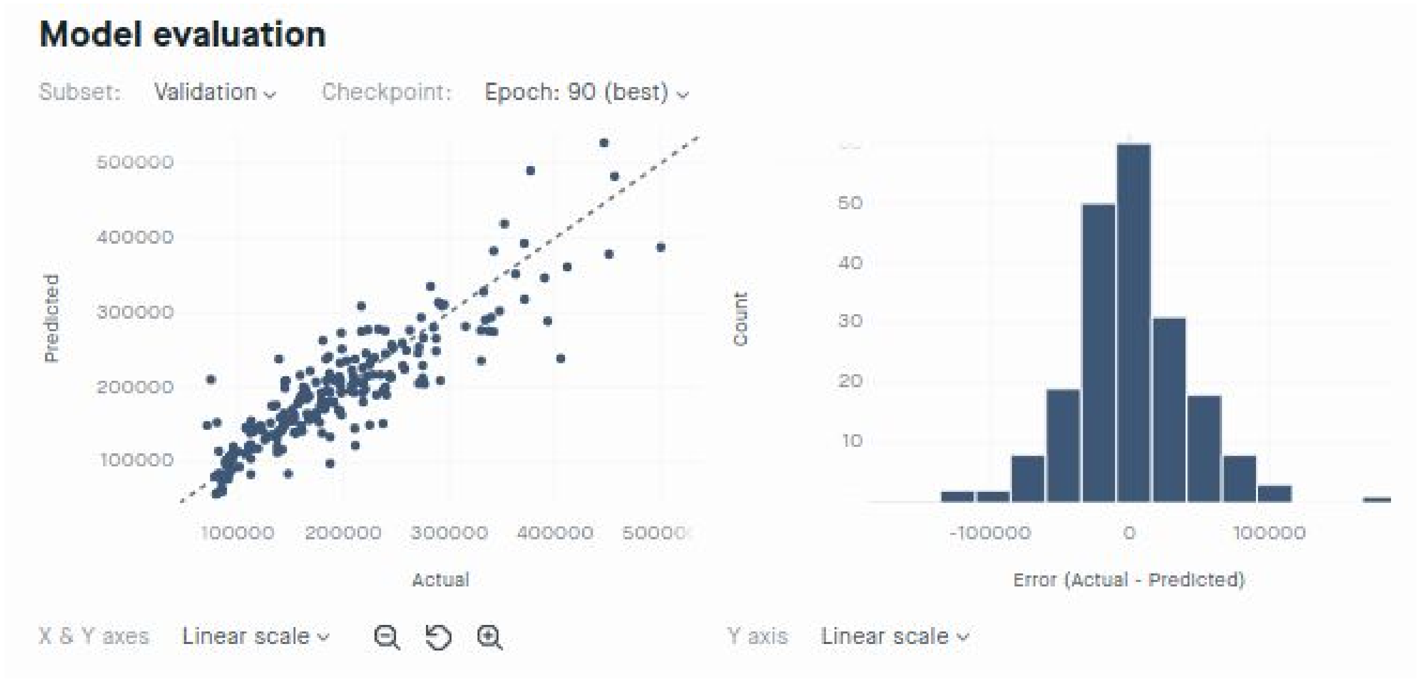
X-axis: Wall time, Epoch, Relative time

Y-axis: Linear scale or Logarithmic scale

Regression problem

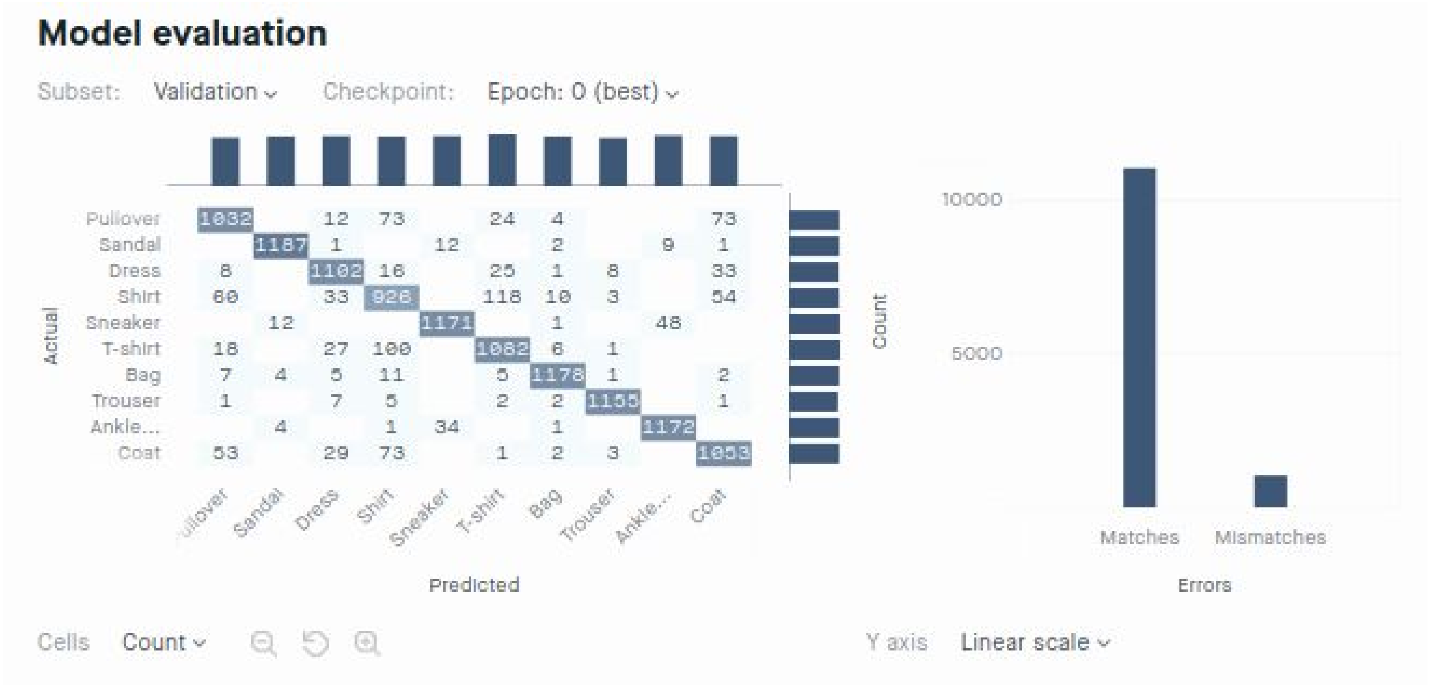
The model evaluation view allows for zoom in by clicking on the prediction scatter plot and also changes the scales on the axes. The scatter plot displays the predictions vs. actuals as well as the line of identity that helps the user easily understand how well the data points are grouped around zero, which is an easy indication of the model's quality.

X-axis and Y-axis on predicted/actual graph: Linear scale or Logarithmic scale.
Y-axis on count/error graph: Linear scale or Logarithmic scale.



Classification problem

The classification problem displays a confusion matrix. It is used to see how well a system does classification. The diagonal shows correct predictions, everything outside this diagonal are errors. In a perfect classification, you'll have 100% of the diagonal going from top left to bottom right. The confusion matrix also provides an easy view on total predictions, precision and recall numbers which provides a way to understand the outcomes between what the model should have predicted and what was actually predicted. In the case of high-level of predicted classes, the confusion matrix offers an easy way to zoom into specific classes predictions.



Model evaluation

Subset: Validation   Checkpoint: Epoch: 0 (best)

Actual \ Predicted	Pullover	Sandal	Dress	Shirt	Sneaker	T-shirt	Bag	Trouser	Ankle...	Coat
Pullover	1032		12	73		24	4			73
Sandal		1187	1		12		2		9	1
Dress	8		1102	16		25	1	8		33
Shirt	60		33	926		118	10	3		54
Sneaker		12			1171		1		48	
T-shirt	18		27	100		1082	6	1		
Bag	7	4	5	11		5	1178	1		2
Trouser	1		7	5		2	2	1155		1
Ankle...		4		1	34		1		1172	
Coat	53		29	73		1	2	3		1053

Cells: Count   Y axis: Linear scale

Visualizations for higher dimensionality targets

The scatter plot and confusion matrix can visualize multi-dimensional targets. For example a vector of numeric values or a target image with one class per pixel.

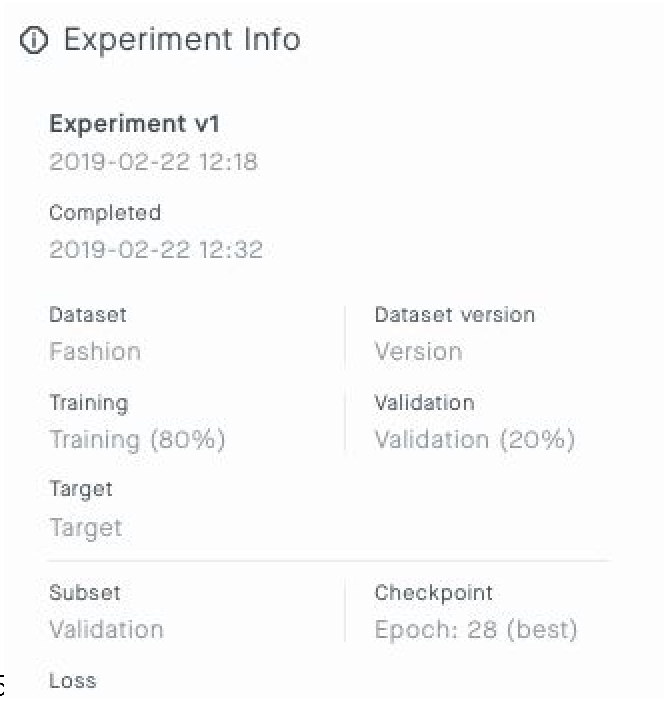
In the case of a classification problem with a multi-dimensional target, the confusion matrix is sampled to a maximum of 500 000 values. For a multi-dimensional target, each value in the confusion matrix corresponds to a vector element, or to a pixel in the target image. This means that the total number of values in the confusion matrix will be many more than the number of samples in the dataset.

In the case of a regression problem with a multi-dimensional target, each dot in the scatter plot represents an element in the target vector or a pixel value in the target image. For visibility reasons, the scatter plot is sampled to show a maximum of 500 data points. The error distribution plot is based on 5000 sampled values.

Experiment info (right section)

The experiment info section displays general information in regard to the progress of an experiment as well as relevant metrics.

The metrics displayed will be different for a regression problem compared to a classification problem.



Experiment Info

Experiment v1
2019-02-22 12:18

Completed
2019-02-22 12:32

Dataset Fashion	Dataset version Version
Training Training (80%)	Validation Validation (20%)
Target Target	
Subset Validation	Checkpoint Epoch: 28 (best)
Loss	

Compare experiments in the evaluation view

When you have run more than one experiment it's easy to visually evaluate and compare the performance and results of your experiments. As long as you keep the same loss function or your model has the same performance metrics, you can compare the results.

In the evaluation view, select the experiment you want to compare in the experiments sidebar. The graph will show metric performance lines for all available experiments from the list that use the same metric. If you want to minimize the number of experiments shown on the graph, use the tags, search and filtering capabilities to list only experiments that are relevant for comparison.

If you notice that your experiment loss and accuracy metrics were continuously improving during the training until all epochs were done, you might want to duplicate the experiment with the weights from the latest epoch and train it further. When including the weights from the previous well-performing training, the model may continue improving without having to restart the computing from the beginning.

Alternatively, if you notice that the loss and accuracy curves for validation data are no longer improving over the last epochs, you may want to stop the training as this may be the peak of how good this model can perform. Or, if you notice that the training and validation curves are drifting significantly apart, while training error may still improve but validation error is increasing, it may also be a good time to stop the training, to avoid excess resource waste as the model is probably already over-fitting and not generalizing well enough. If the loss looks jagged, try to either increase the batch size or decrease the learning rate. If the loss looks too smooth, be bold and decrease the batch size to see what happens.

07/ Deployment view

Introduction

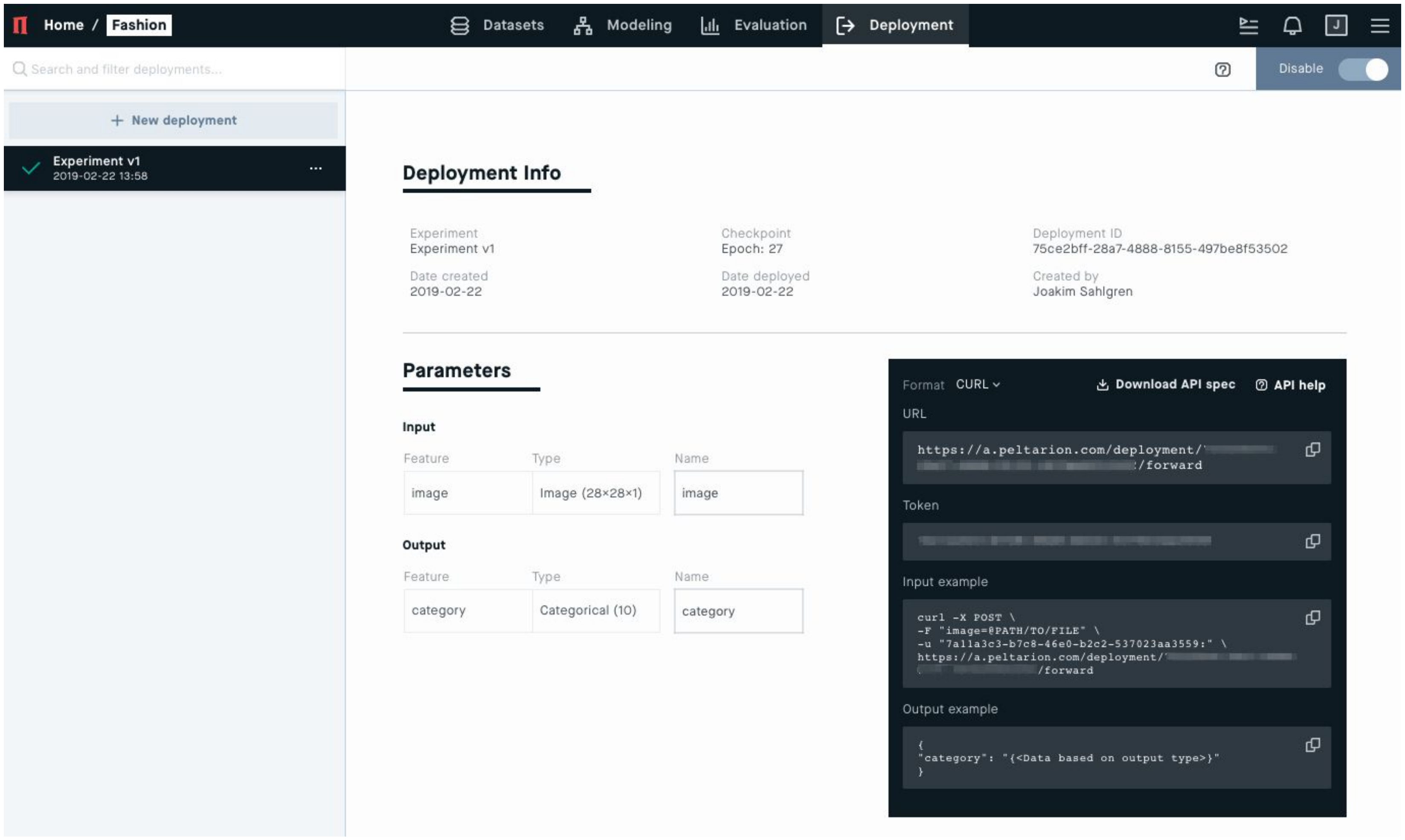
When a model has been built and trained it is time to deploy.
Currently, the solution supports two ways of deploying a model:

- Deploy as API in the deployment view
- Download the model as .h5 (Note: this function is accessed from the modeling view and explained in the modeling section)

The API uses the HTTP POST method (over HTTPS) for requests and the responses are returned in a JSON-encoded format. Since it is based on a REST architectural style, you can implement it in any language, e.g., Python, PHP, Node.js, Java etc.

Both single requests, as well as batch requests, can be sent to the API.

The CURL-example in the Deployment view is a good way to test out the deployment functionality. Select “CURL” to expose the information and example code needed to send requests to the model.



Deployment Info

Experiment Experiment v1	Checkpoint Epoch: 27	Deployment ID 75ce2bff-28a7-4888-8155-497be8f53502
Date created 2019-02-22	Date deployed 2019-02-22	Created by Joakim Sahlgren

Parameters

Input

Feature	Type	Name
image	Image (28×28×1)	image

Output

Feature	Type	Name
category	Categorical (10)	category

Format CURL

URL

```
https://a.peltarion.com/deployment/[redacted]
[redacted]/forward
```

Token

```
[redacted]
```

Input example

```
curl -X POST \
-F "image=@PATH/TO/FILE" \
-u "7a11a3c3-b7c8-46e0-b2c2-537023aa3559:" \
https://a.peltarion.com/deployment/[redacted]
[redacted]/forward
```

Output example

```
{
"category": "{<Data based on output type>}"
}
```

Deploy to API

To create a model serving API, go to the deployment view and click “New deployment”. In the pop-up, select which experiment and checkpoint should be used as deployment.

When the experiment checkpoint and parameter names are set, the deployment can be enabled with just one click on the Enable deployment toggle in the right upper corner. Once obsolete, disable the deployment to save resources.

If you click the Download API spec button the Platform will generate and download a JSON-file with an OpenAPI specification of the API. The file includes information such as endpoint and schemas for input and output including examples. The file format follows the OpenAPI specification version 3.0. A standard, programming language-agnostic interface description for REST APIs.

Use the API

The API supports basic access authentication and OAuth2.0 bearer token usage.

Each request to the API must contain the authentication token and the URL to deployment

The payload of the HTTP request body depends on the type and number of features that the model was trained on. For instance, if the input feature to the model is a PNG image with the dimensions 32x32, the request body must contain an identically sized image.

The input feature values can be encoded as multipart/form-data or as application/json. Batch requests need to be JSON-formatted.

08/ Platform limitations

The purpose of this section is to provide the known limitations as well as the system requirements of the Peltarion Platform. The platform is rapidly evolving, so the reader should check with their Peltarion contact to verify the status of the limitation and roadmap cover. In addition, Peltarion provides assets (code, solution, etc.) and workarounds for certain tasks which can be supported on the platform.

Should a user experience limitations that are not listed in this document, it would be advisable to get in contact with Peltarion.

The limitations described below are as of November 2018.

Data format and dataset restrictions

Overall restrictions

The dataset size is recommended to be less than 200GB. Larger datasets may work but the upload cannot be fully done in the background as the browser window has to remain open. This could restrict the uploading of larger files.

Dataset size for uploads by drag & drop into the browser window is limited to 4GB due to web browser limitations.

Supported file formats

Image archives (.zip with .csv + .png/jpeg/npy-files)

A .zip file, containing a csv file for referencing and adding metadata to a series of images is supported.

Restrictions:

The system supports Grayscale, RGB and RGBA PNG files. Indexed PNGs are supported (like PNG-8), but there is no support for working directly on the index position, i.e., the indexed colors will always be used in training and not the index position.

The system supports RGB JPG files (i.e., CMYK files are not supported).

Recommended maximum pixel size: 299x299 pixels.

CSV files (.csv)

Table data in CSV-files are interpreted as either float-, integer- or string types. Strings will be one-hot encoded.

The CSV file must contain a constant number of columns.

The data needs to be well-formatted, e.g., the data type in each column is consistent and contains no empty or null values.

The system supports UTF-8 encoding.

Restrictions:

Note: These are indicative numbers, e.g., 10 000 000 rows with 1 000 columns with different one-hot encodings may not work.

One-hot encoding categories	2 000
Maximum number of rows	10 000 000
Maximum number of columns	5 000

NumPy files (.npy)

The NumPy file format is used as a general carrier for raw tensor data. Only little-endian format is supported. Fortran order is not supported.

Resources

Resources cannot be shared, exported or imported between projects. To use the same datasets or models in two projects they need to be uploaded and recreated in both.

User account limitations and organization management

Sign-in is only supported through a Peltarion account.

Organization quota management (upscale, downscale) and payments are only handled through sales@peltarion.com.

Adding, removing and inviting new collaborators to the organization is managed by the customer. Admin rights are needed to perform those actions. Max allowed accounts depends on the purchased quota plan. Please see the link to organization settings view for more information and get in contact with sales@peltation.com for the admin guide.

Browser limitations

Only Chrome browser version 66.0 upwards is supported.

Resources and running jobs

Note that training can be resumed after an experiment has been stopped if the experiment has trained less than the maximum number of epochs.

The system only supports training on GPUs. Deployment API forward-passes run on CPUs.

The platform can run multiple experiment training jobs simultaneously, but with one GPU per job.

The platform will show the amount of storage used and the number of GPU hours consumed. The platform will not throttle the over-usage of quotas but Peltarion will monitor and contact the customers that exceed their quota.

Deployment

There is no dedicated support for real-time applications (i.e., no response times under 1 second).

Peltarion does not support GPU-based deployment.

Peltarion does not support deployment using dedicated instances.

The system does not support the saving of data being fed through the deployment system for future retraining and the saving of target results.

Mini-batches are supported (i.e., all samples in the request will form one batch). The maximum number of samples in a batch will depend on sample size and model size.
Recommendation: keep the number of rows less than the number of rows in each training batch.

The system only supports models with one target block.

The system does not support channel or bit-depth conversion of images. The data format used in training needs to also be used in deployment. For example, if one channel PNGs were used during training, then it will not be possible to input three channel images to the deployed service.

Peltarion AB
Holländargatan 17
SE-111 60 Stockholm
Sweden

Peltarion provides an operational AI platform for producing real-world AI applications at scale and at speed. It's user-friendly, intuitive and end-to-end.

Over 300 companies and organizations have used Peltarion's AI technology including NASA, Tesla, iZettle, General Electric, Dell, BMW, Deutsche Bank, Lloyds Banking Group, and the Universities of Harvard, MIT and Oxford. Peltarion was founded in Stockholm, Sweden in 2004.

PELTARION