Online QSAR Modelling Hackathon by Easy Access to Jaqpot: Deploy your model as a web service in a few minutes

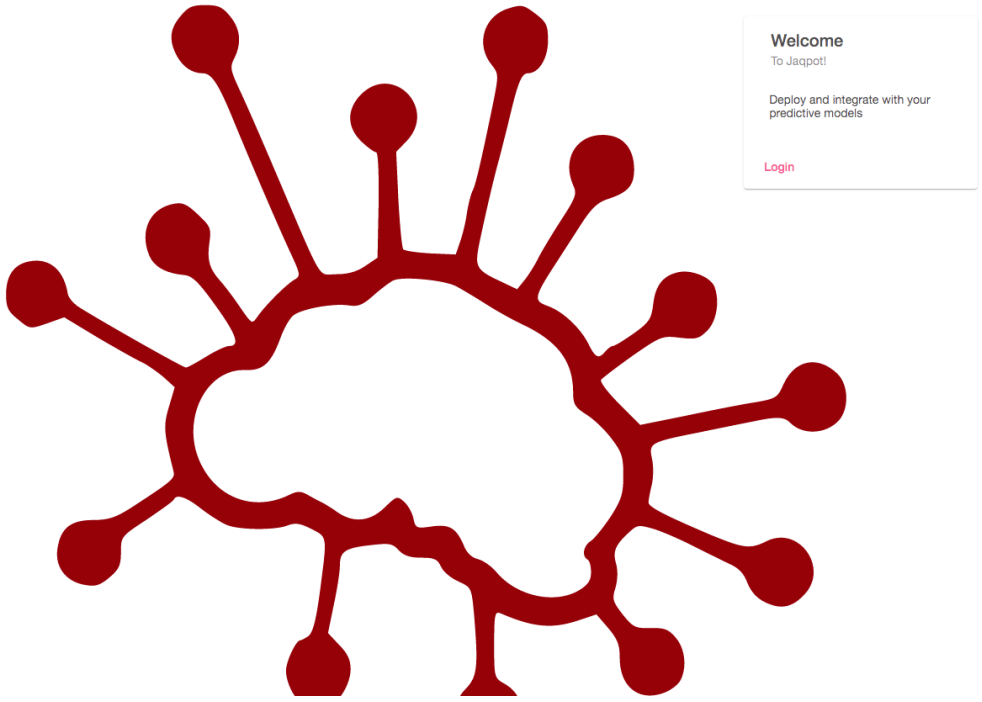
Tue, Apr 13, 2021 3:00-4:30 PM CEST

Briefly

- You will learn how a local QSAR model can be transformed to an online QSAR model web service, no installations required
- Prerequisites:
  - For access to Jaqpot, you will need a Jaqpot account
  - For access to Google collab you will need a Google account

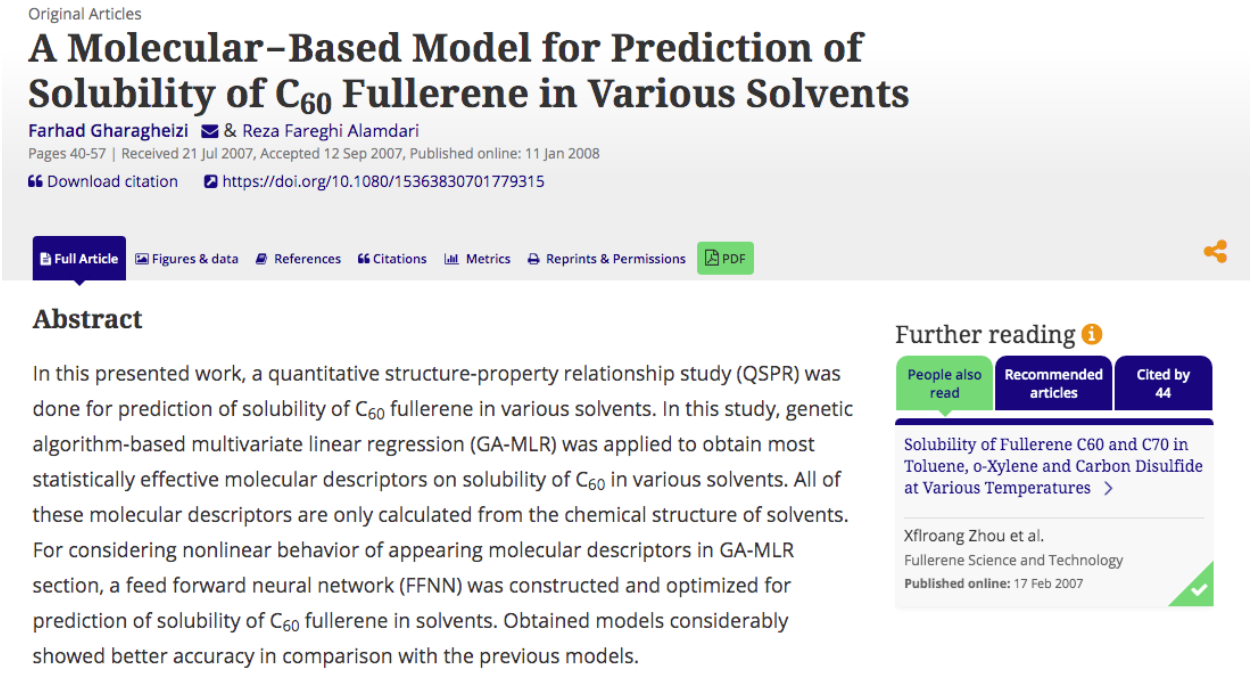
Introduction

Jaqpot 5 (https://app.jaqpot.org) is a powerful and versatile platform for toxicological in silico predictions, allowing you to deploy machine learning models incredibly easy and make them available to the community as web services.



Problem studied

In this tutorial we will demonstrate how to reproduce and publish in Jaqpot 5, a Linear nanoQSAR model predicting Solubility of C60 Fullerene in Various Solvents, with just a few lines of code in a Colab notebook. The model has been originally presented in the following publication: Farhad Gharagheizi & Reza Fareghi Alamdari (2008) *A Molecular-Based Model for Prediction of Solubility of C60 Fullerene in Various Solvents.* Fullerenes, Nanotubes and Carbon Nanostructures, 16:1, 40-57, DOI: 10.1080/15363830701779315. https://www.tandfonline.com/doi/full/10.1080/15363830701779315



Original Articles

A Molecular–Based Model for Prediction of Solubility of C_{60} Fullerene in Various Solvents

Farhad Gharagheizi & Reza Fareghi Alamdari

Pages 40-57 | Received 21 Jul 2007, Accepted 12 Sep 2007, Published online: 11 Jan 2008

Download citation https://doi.org/10.1080/15363830701779315

Full Article Figures & data References Citations Metrics Reprints & Permissions PDF

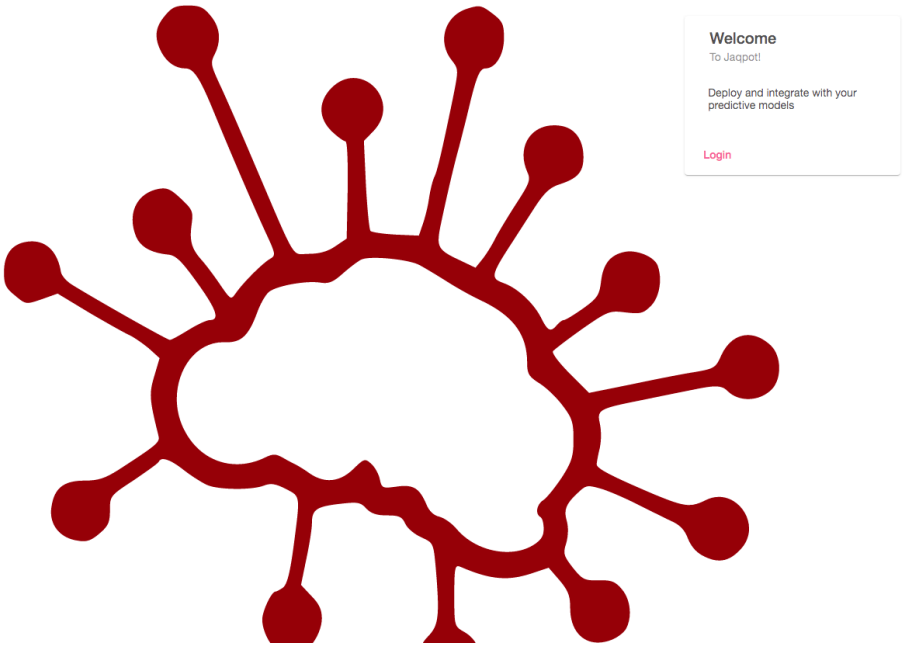
Abstract

In this presented work, a quantitative structure-property relationship study (QSPR) was done for prediction of solubility of C_{60} fullerene in various solvents. In this study, genetic algorithm-based multivariate linear regression (GA-MLR) was applied to obtain most statistically effective molecular descriptors on solubility of C_{60} in various solvents. All of these molecular descriptors are only calculated from the chemical structure of solvents. For considering nonlinear behavior of appearing molecular descriptors in GA-MLR section, a feed forward neural network (FFNN) was constructed and optimized for prediction of solubility of C_{60} fullerene in solvents. Obtained models considerably showed better accuracy in comparison with the previous models.

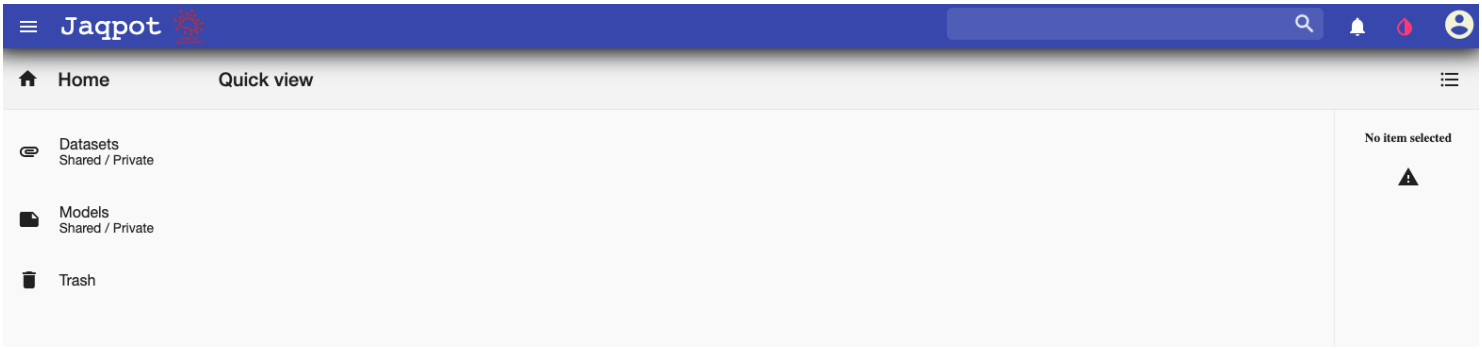
1. Accessing Jaqpot

In order to access Jaqpot, you need a Jaqpot account. For the purposes of this workshop, please avoid using the Google/Github option, to enable a more uniform approach among participants. If you are a new user, you can create a Jaqpot user account following the instructions provided in a separate document.

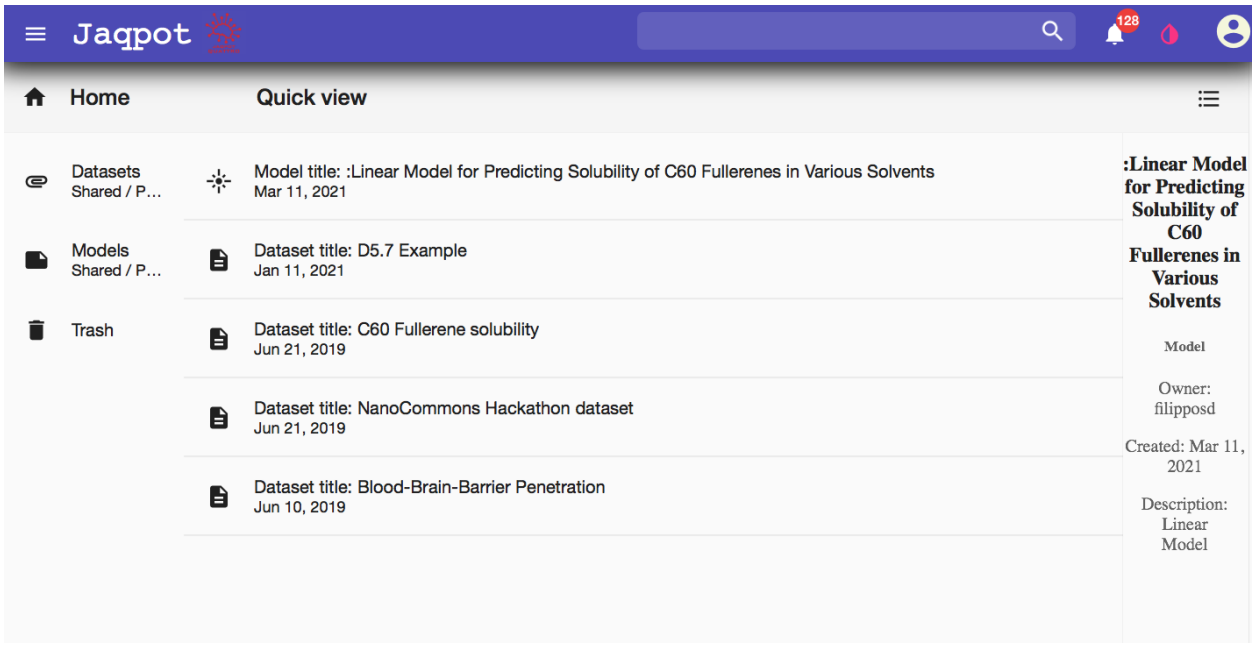
First click on **Login** on the welcome page:



You are then redirected to your Jaqpot homepage, which gives you access to your models and datasets.



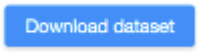
Currently you don't have any models or datasets. A populated homepage looks like this:



Each time you access Jaqpot on Google Collab (as we will see in a subsequent step) you need to identify yourself, in order to be granted access to your private resources by providing your username and password.

2. Retrieving the dataset from NanoPharos

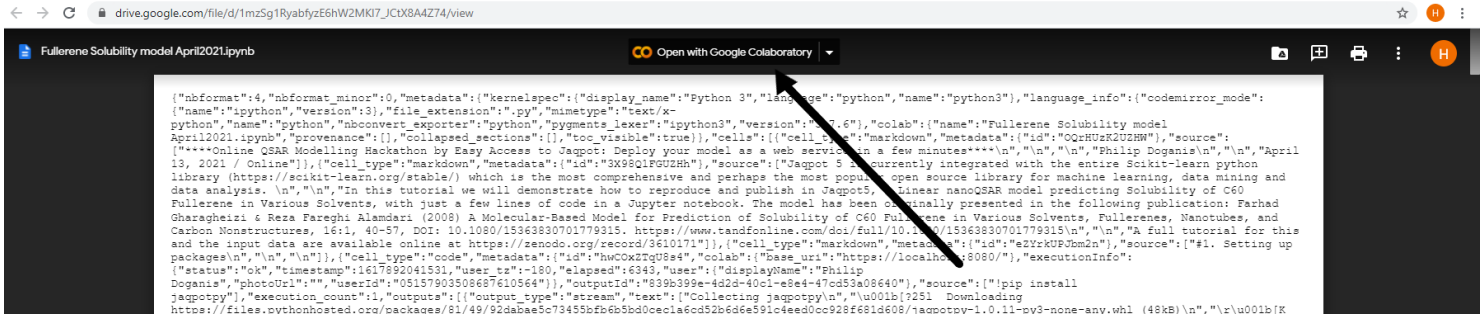
The dataset can be retrieved in ready-to-model format from the NanoPharos database, which is accessible at: https://db.nanopharos.eu/Queries/Datasets.zul

Please select the **Solubility of C60 Fullerene in Various Solvents** entry from the drop-down menu on the left and click the  button. An Excel file named '**70_model_reduced.xlsx**', will be downloaded and stored in your local computer.

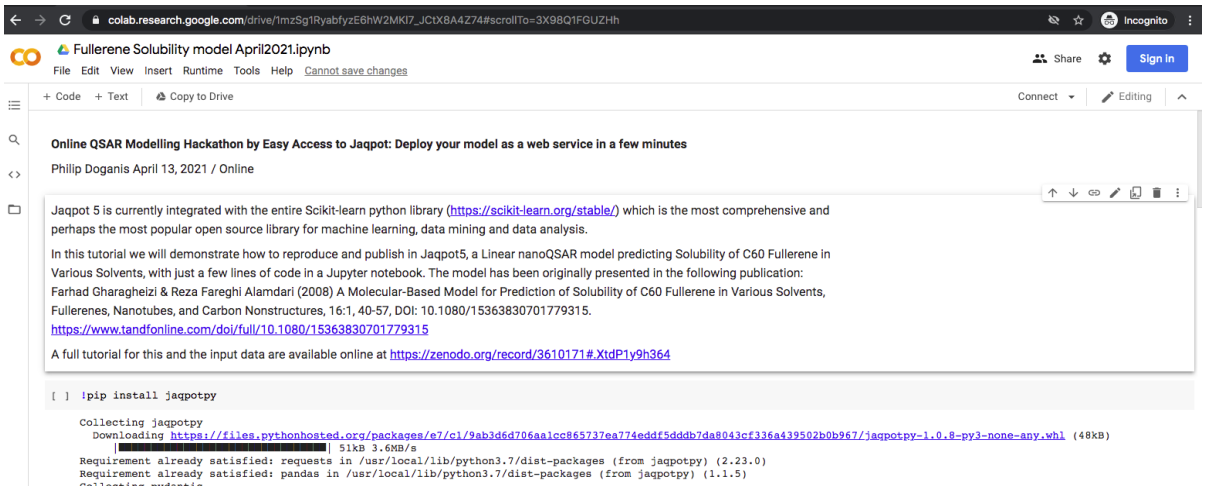
3. Accessing the Google colab notebook

The Google colab platform allows users to run code in the Python programming language without needing to perform any installation in their local machines. A Google colab notebook that performs all the calculations and contains all the commands for model creation and deployment has been prepared for you and is available at:
https://drive.google.com/file/d/1mzSg1RyabfyzE6hW2MKl7_JCtX8A4Z74/view?usp=sharing

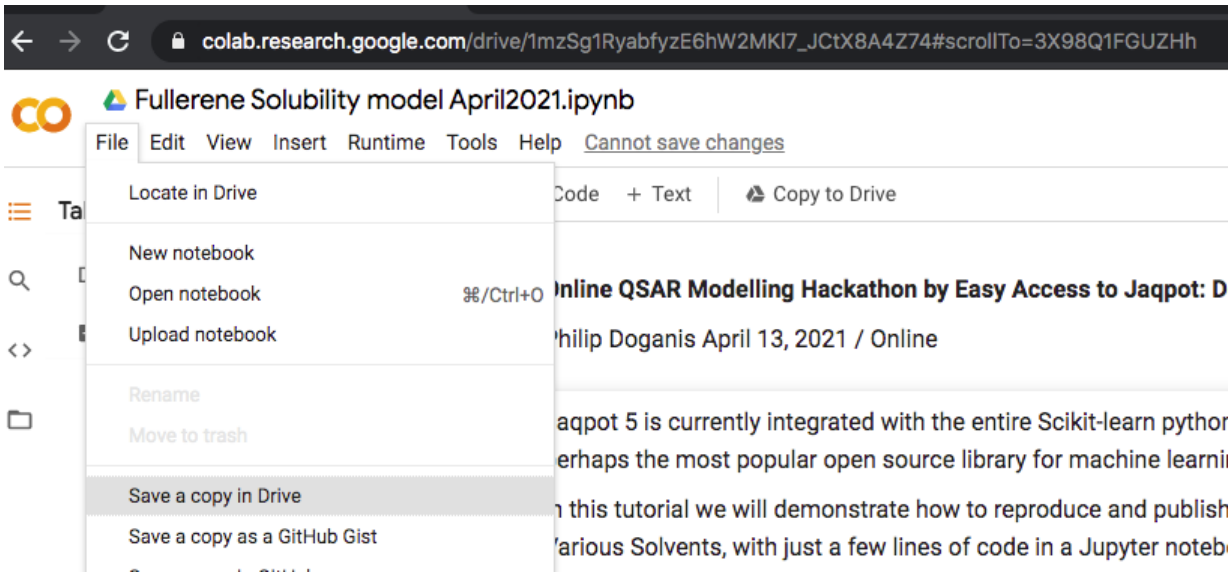
If a screen like this appears, please click on "Open with Google Colaboratory"



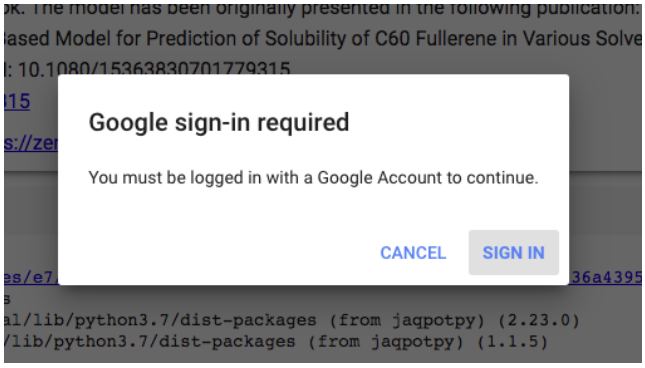
The Google colab notebook starting page looks like that:



In order to be able to use the notebook and make any changes, please save a copy of the notebook in the Drive by clicking the menu as shown below:



This can be done seamlessly when you are logged in as a Google user, if however you receive the message below you need to log in.



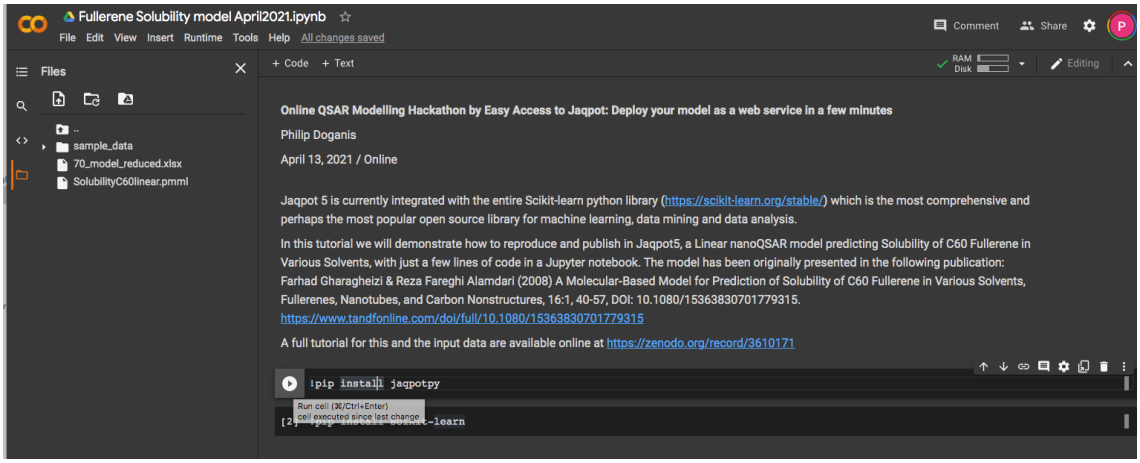
Now you have a local copy of the notebook that you modify and run. This is in your private space managed by Google.

4. Running the colab notebook to create the model web service on Jaqpot

The notebook contains both text with comments, and cells containing python commands that will handle the dataset, make the necessary calculations and deploy the model as a web service. Green text in code cells are comments and are not executable.

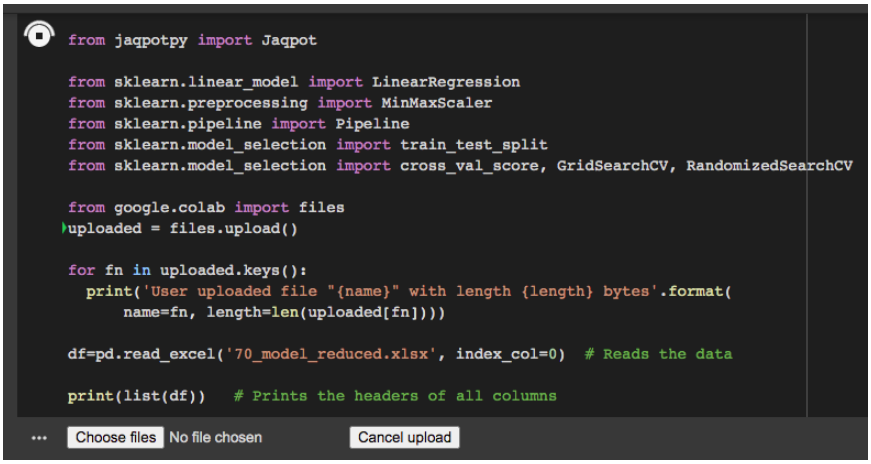
1. Setting up packages

In order to execute code on the collab notebook you can go to the left of each cell and hit the  icon. You see the outcome of the cell expanding right below the cell.
First , you will set up the Jaqpotpy, scikit-learn and pandas packages, so they can be used within the notebook:

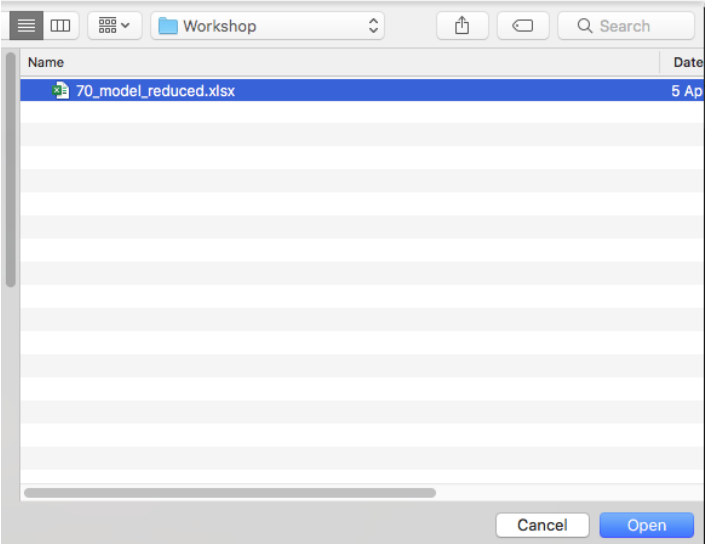


2. Uploading the data to the colab notebook

In this step, you will upload the dataset retrieved from NanoPharos. Executing the code in this cell generates a  button:



Clicking the button leads to a file browser window, where you can locate the file to be uploaded, which here is the 70_model_reduced.xlsx file.



3. Viewing and Preprocessing the dataset

In this step, you will:

- take a first look at the dataset
- generate descriptive statistics
- split the dataset randomly into training and test sets consisting of 75% and 25% of the data respectively
- generate the pipeline that will do:
    - preprocessing: transform features by scaling each feature to a given range using the MinMaxScaler function: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
    - modelling: The linear regression algorithm will be applied to the training dataset to generate the model

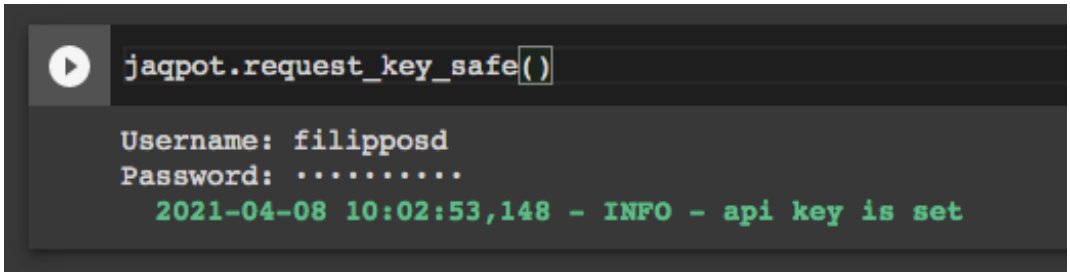
4. Training the model

Here you will:

- train the model
- print model performance metrics on the training, test dataset and on the total dataset
- perform a 5-fold cross validation test (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html).

5. Getting authentication to access Jaqpot

Please provide your username, press enter and then provide your password in the prompt that emerges.



6. Deploying the model

You will now deploy the model using the ***jaqpot.deploy_sklearn*** function with the following arguments:

Pipelinelinear	the pipeline to be used for preprocessing the data and the algorithm to be applied.
Xall	Dataset with the input (known) features.
Yall	Dataset with the output feature (the feature we want to predict with the model).
title="Solubility of C60 Fullerenes in Various Solvents- USERNAME"	Title of the model, as it will appear in the Jaqpot user interface. **Please change this by adding your username or any other character string that can distinguish your model and avoid multiple entries with the same model name.**
description="Description"	A short description intended to inform users about the model.
model_meta=True	By choosing True we enable listing of automatically generated model metadata (version of sklearn used, details on pipeline, preprocessing transformations and algorithm used for model). False disables this.
doa=X_train	Providing the dataset here enables Domain of Applicability calculations based on the leverage method. False disables this.

The command with the arguments listed above is:

Please change USERNAME by any string of characters that can distinguish your model.

```
modelId=jaqpot.deploy_sklearn(pipelinelinear,Xall,Yall,title="Solubility of C60
Fullerenes in Various Solvents-USERNAME",description="description",
model_meta=True,doa=X_train)
```

1 command is needed to deploy the model into Jaqpot 5: https://www.jaqpot.org/docs/sklearn

(please note some additional commands to time this procedure)

```
[ ] import time
    start_time = time.time()

    #This command deploys the model on Jaqpot
    modelId=jaqpot.deploy_sklearn(pipelinelinear,Xall,Yall,title="Solubility of C60 Fullerenes in Various Solvents-USERNAME",description

    #The response is the unique Jaqpot model ID on which the model is hosted.

    print("---Your Jaqpot web service was created in %s seconds ---" % (time.time() - start_time))

    2021-04-07 21:43:42,827 - INFO - Model with id: 99vBdRvvD8NoiFA6lN17 created. Storing Domain of applicability
    2021-04-07 21:43:43,413 - INFO - Stored Domain of applicability. Visit the application to proceed
    ---Your Jaqpot web service was created in 1.2052156925201416 seconds ---
```

The web service has been created in Jaqpot and has a unique **modelURI** of the following form: https://app.jaqpot.org/model/99vBdRvvD8NoiFA6lN17 .

```
[ ] modelURI='https://app.jaqpot.org/model/'+modelId
    print("You can use your model here or over the User Interface at: %s " % modelURI)

    You can use your model here or over the User Interface at: https://app.jaqpot.org/model/99vBdRvvD8NoiFA6lN17
```

Now your QSAR model is a web service!

You can access your model through the Jaqpot interface

https://app.jaqpot.org

ADDITIONAL STEPS

After completing the steps above, you can take a look at the following:

1. Accessing your model through the colab notebook
2. Share the model to Organisation
3. Adding meta information to your model: QMRF report
4. Test and use your Jaqpot model
5. More information on the Jaqpot platform

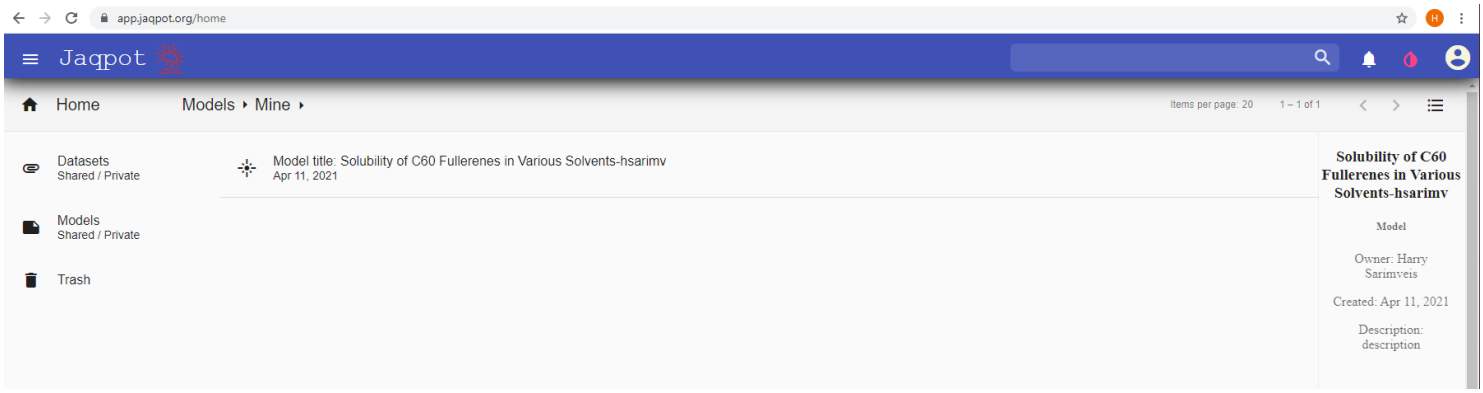
1. Access the model through the colab notebook

You can use the **jaqpot.predict** function in your colab notebook providing the dataset name (for example **Xall**) and the **`modelID`** as arguments in order to get predictions from the model, which are stored in the predictions variable:

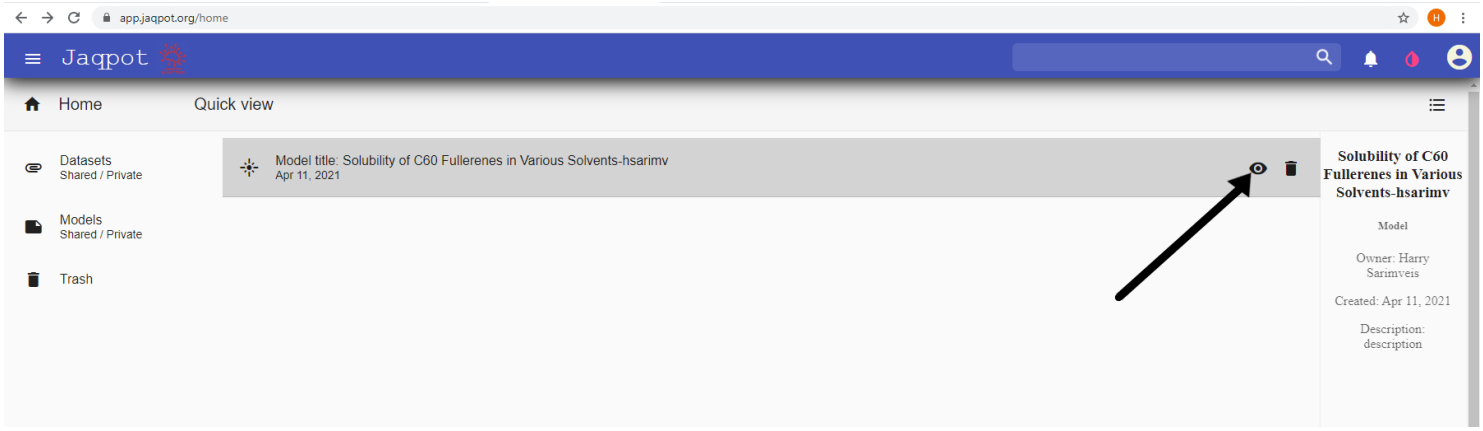
```
[ ] predictions = jaqpot.predict(Xall, modelId)

    2021-04-07 21:44:14,731 - INFO - completed 10.0
    2021-04-07 21:44:16,281 - INFO - completed 100.0

[ ] predictions

(                    piPC03  ATS1m  More23e    H1m  logS Exp.  Seigp
 pentane              1.099  1.609   -0.922  0.148  -5.818155    0.0
 hexane               1.386  1.792   -1.156  0.173  -5.419509    0.0
 octane               1.792  2.079   -1.656  0.208  -4.862671    0.0
 iso-octane           1.792  2.079   -1.529  0.242  -4.839394    0.0
 decane               2.079  2.303   -2.151  0.233  -4.479601    0.0
 ...                    ...    ...      ...    ...        ...    ...
 pyridine             3.056  1.992   -0.446  0.394  -4.148183   -0.6
 quinoline            4.123  2.512   -0.750  0.591  -2.817290   -0.6
 aniline              3.248  2.100   -0.504  0.351  -3.839706   -0.6
 N-methylaniline      3.359  2.234   -0.462  0.407  -3.556255   -0.6
 N.N-dimethylaniline  3.458  2.351   -0.435  0.399  -3.278345   -0.6

[124 rows x 6 columns], 'logS Exp.')
```

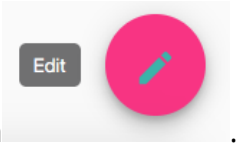
2. Share your model with the WorkshopApril2021 group

Access your model through the Jaqpot application: https://app.jaqpot.org:

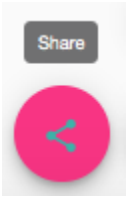


Select your model and click on the “View” icon

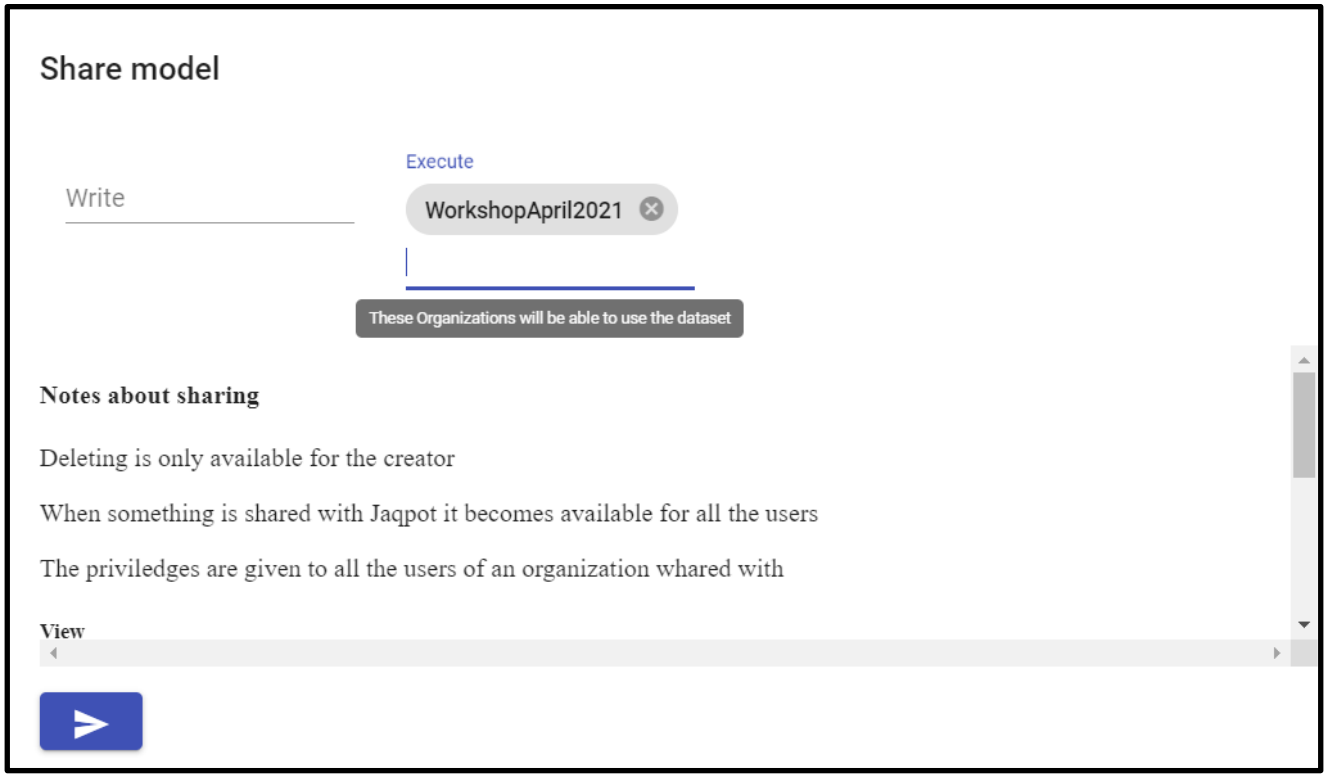


At the bottom right of the model page, click on the edit button 

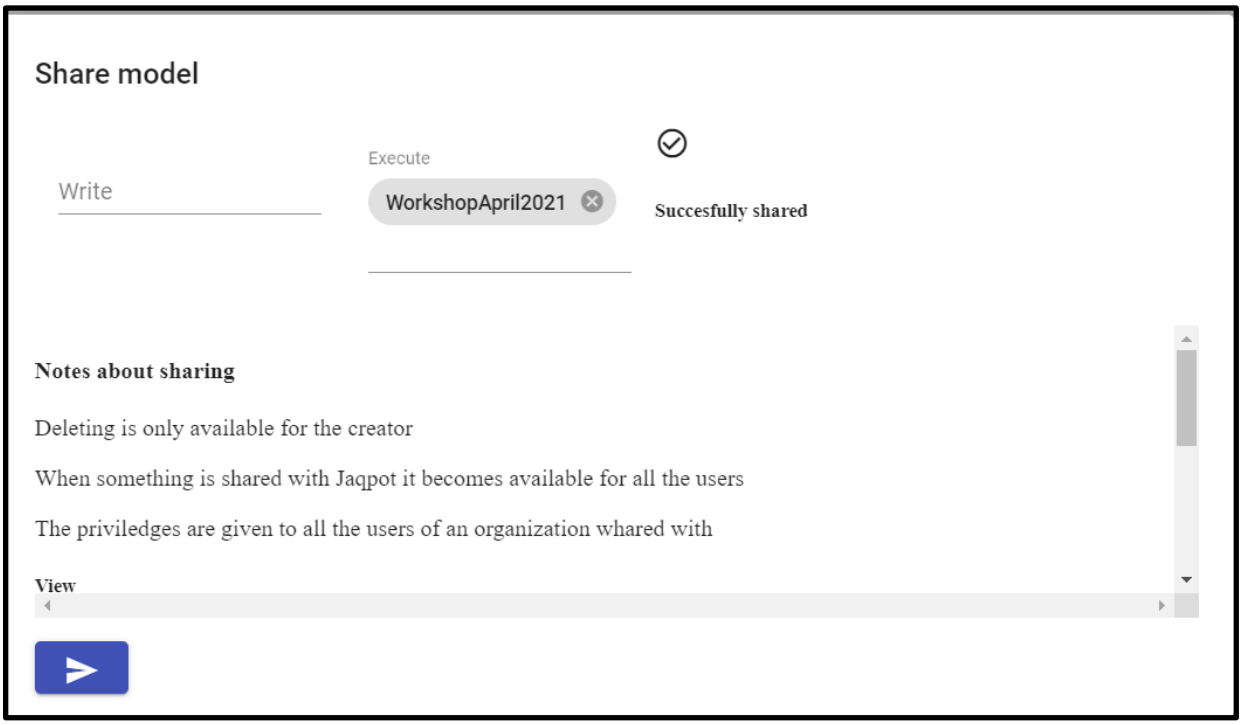
.

Clicking the Share button 

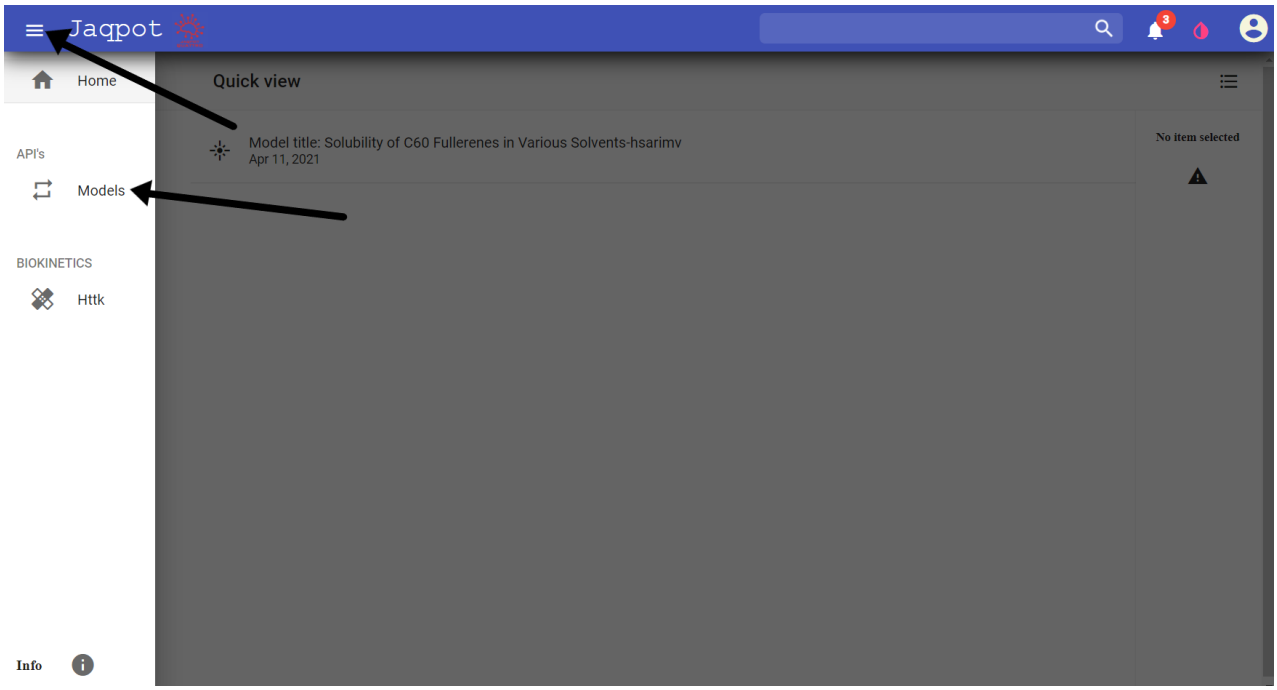
gives you the option to share the model with organisations of choice. Please select the WorkshopApril2021 group and give Execute rights only:



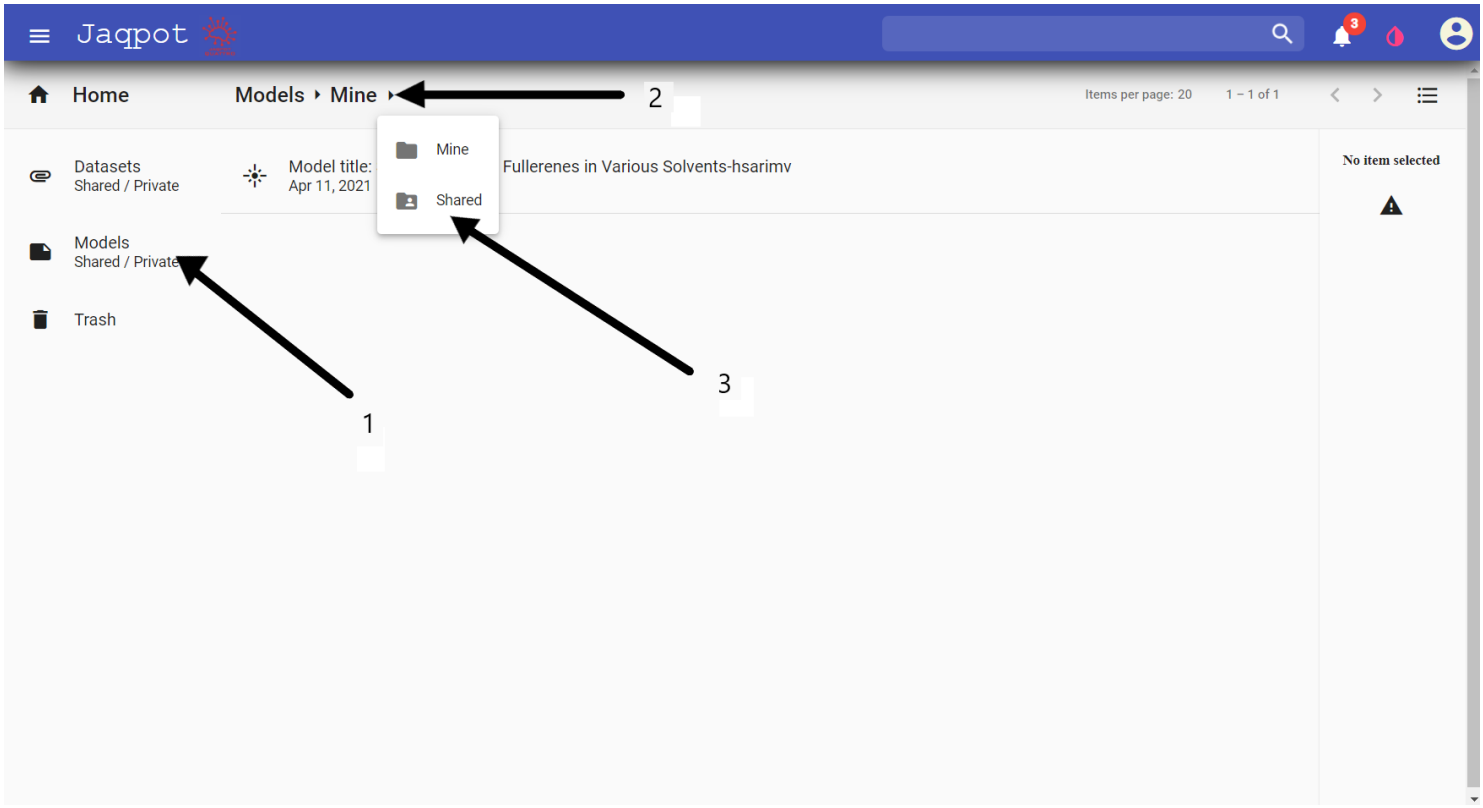
Adding an organisation to share with, activates the Share button, turning it from grey  to blue  . Clicking this button causes the “Successfully shared” message to appear:



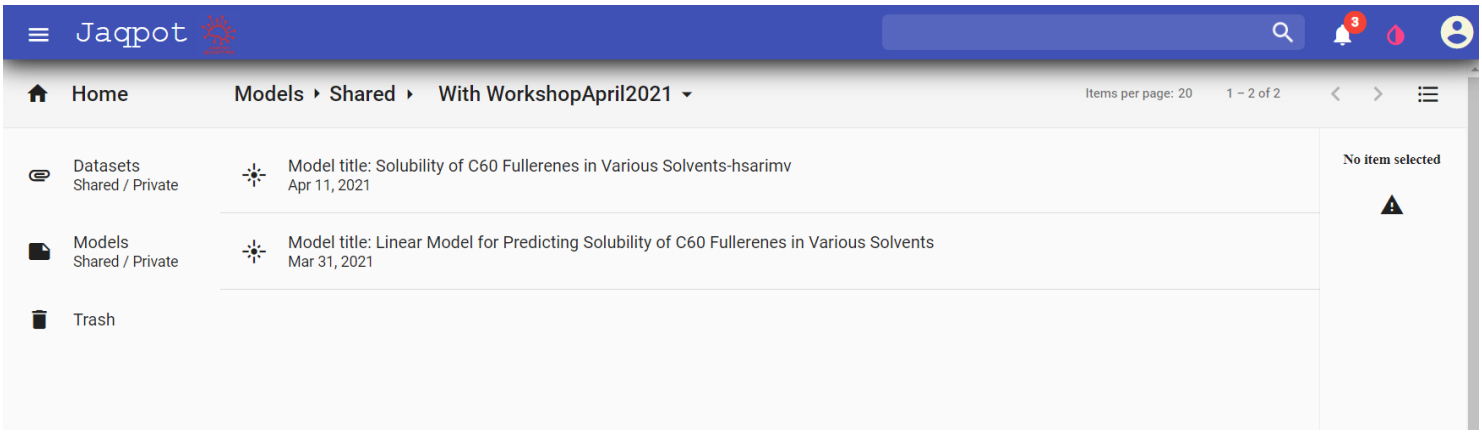
In order to view the shared models, please click on the menu button and select Models:



Select Models [1], click the triangle icon next to **Mine** [2] and select **Shared** [3]

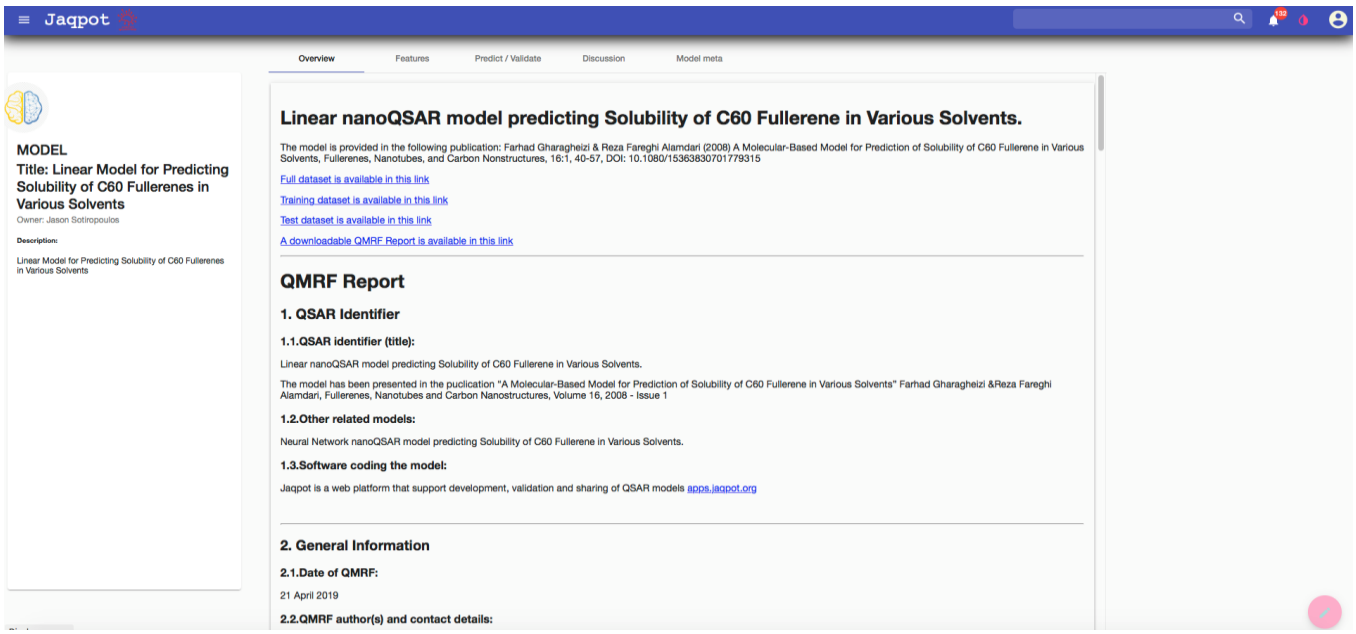


Select the **WorkshopApril2021** organisation. You are directed to a page containing all models shared through the **WorkshopApril2021** organisation.

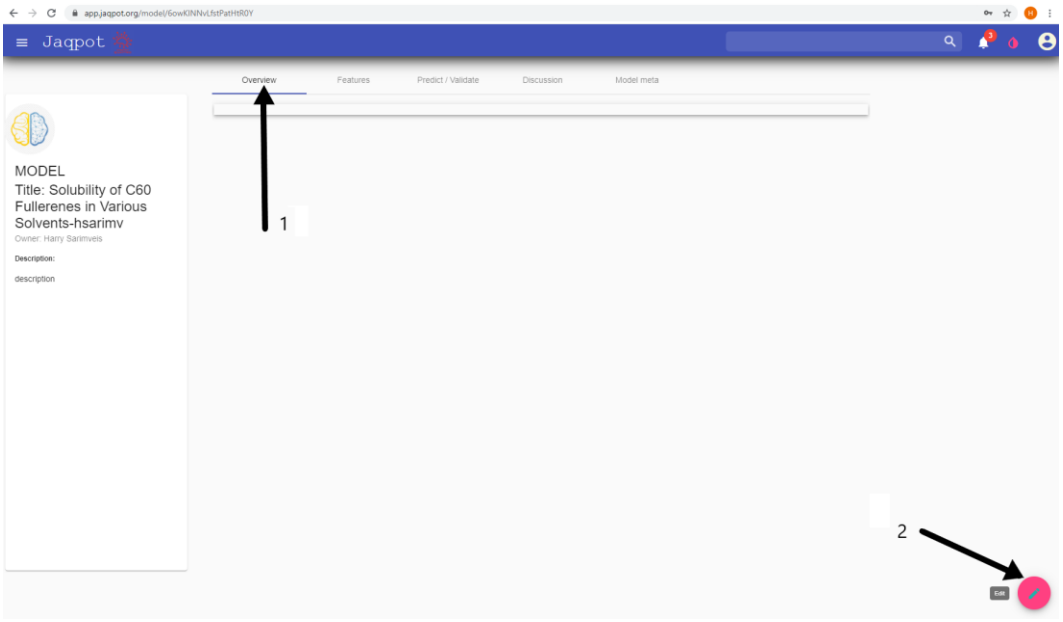


3. Adding meta information to your model

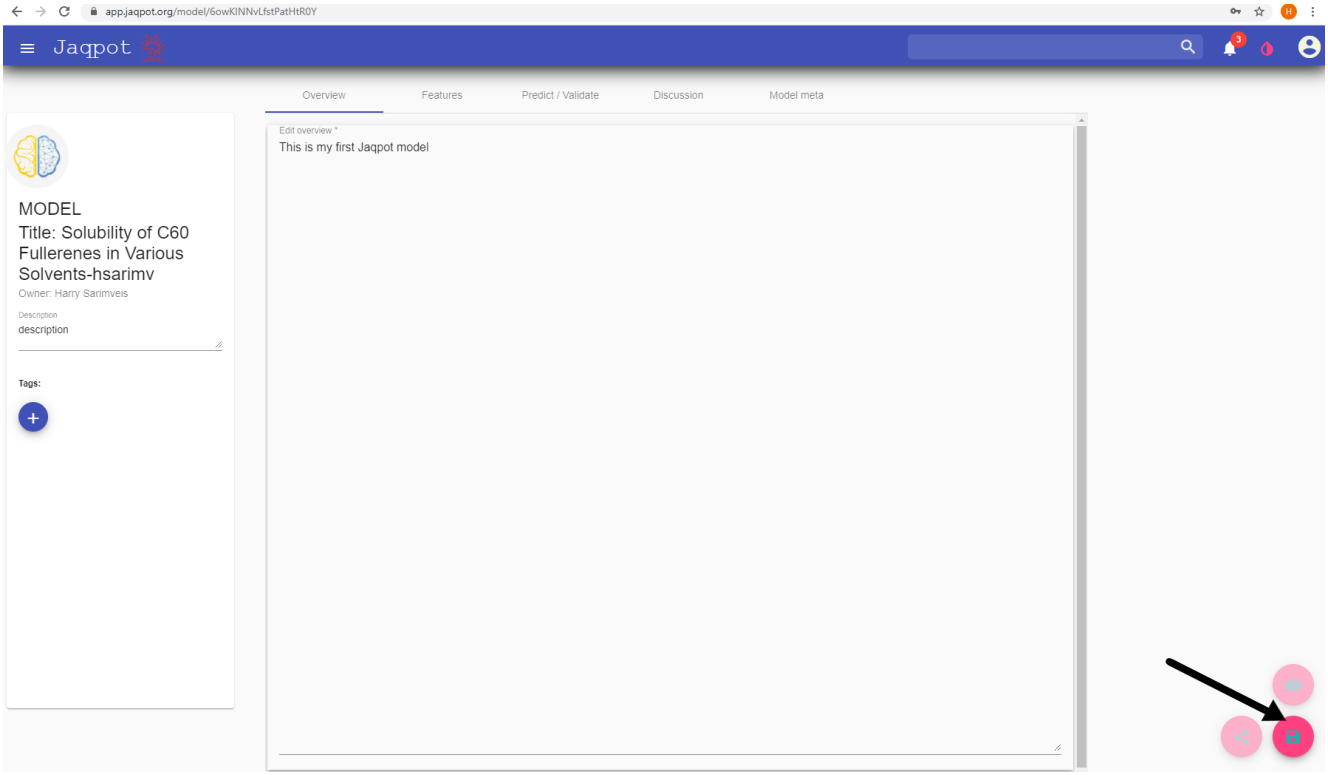
You can add any other information about the model over the user interface at https://app.jaqpot.org/ (for example detailed description, standard reports like Quantitative Model Reporting Format (QMRF), PMML representations, ontological annotations etc., descriptions of variables, ontological classes etc.). To demonstrate this, a fully documented model titled **"Linear Model for Predicting Solubility of C60 Fullerenes in Various Solvents"** is available at https://app.jaqpot.org/model/RqCRtRpY85kpbgGtsiXp and has been shared with the **WorkshopApril2021** organisation:



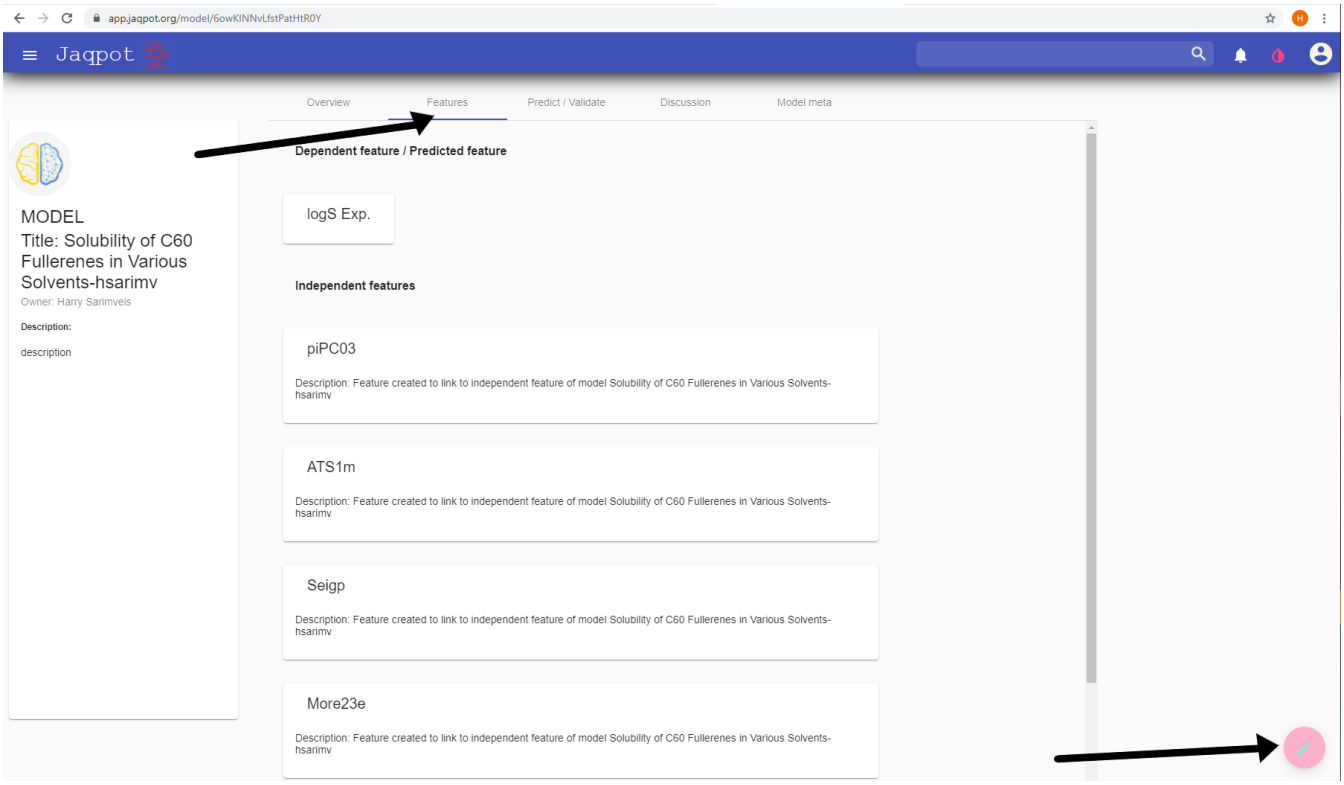
You can add information to your model by selecting the overview tab [1] and then clicking on the edit button.



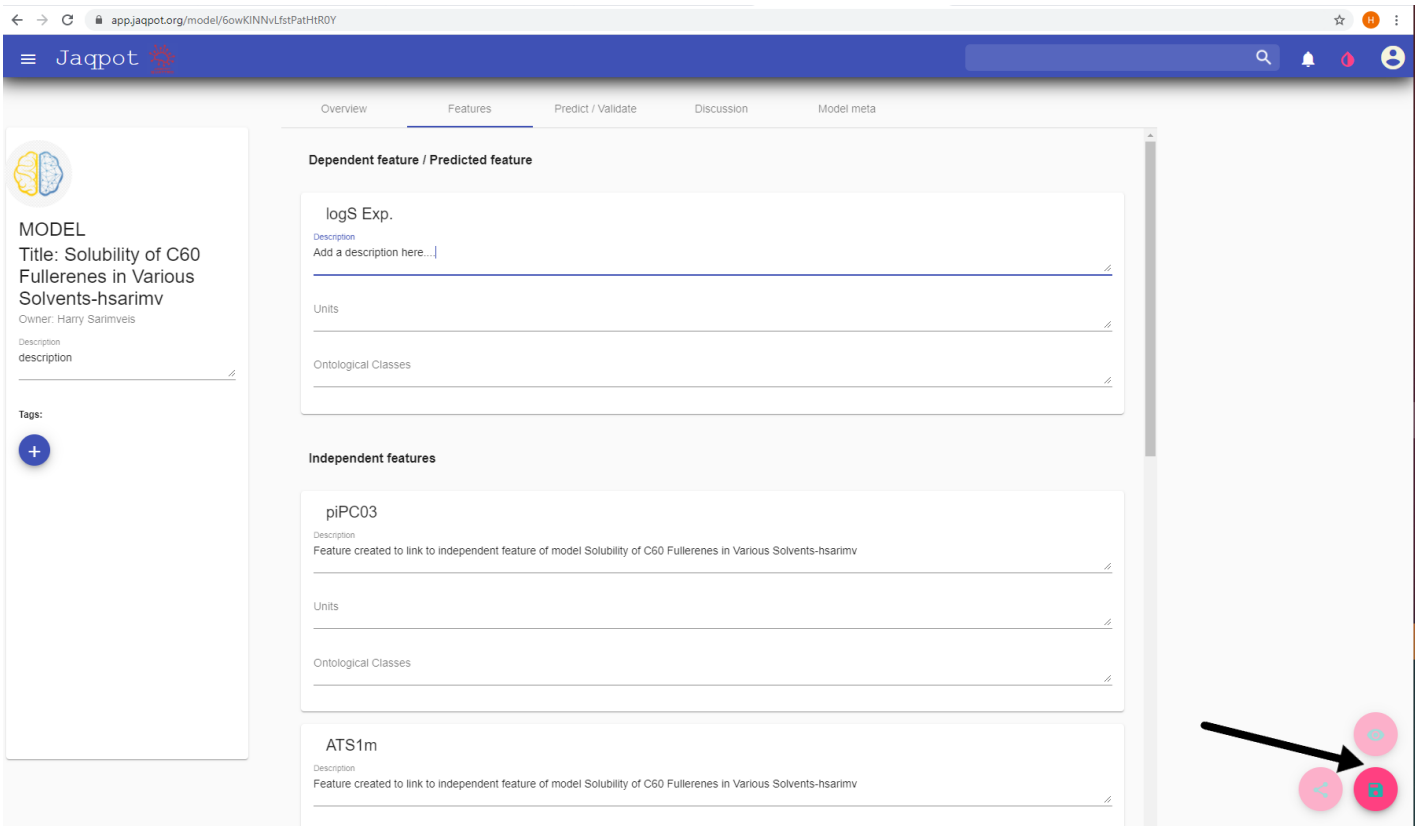
You can write any text (Markdown language is supported) and save it by clicking on the save button on the bottom right part of the page. You can return back and re-edit or extend the model description at any time.



Information about the variables used in the model (descriptions, units, ontological classes) can be added by selecting the “Features” tab and clicking on the Edit Button



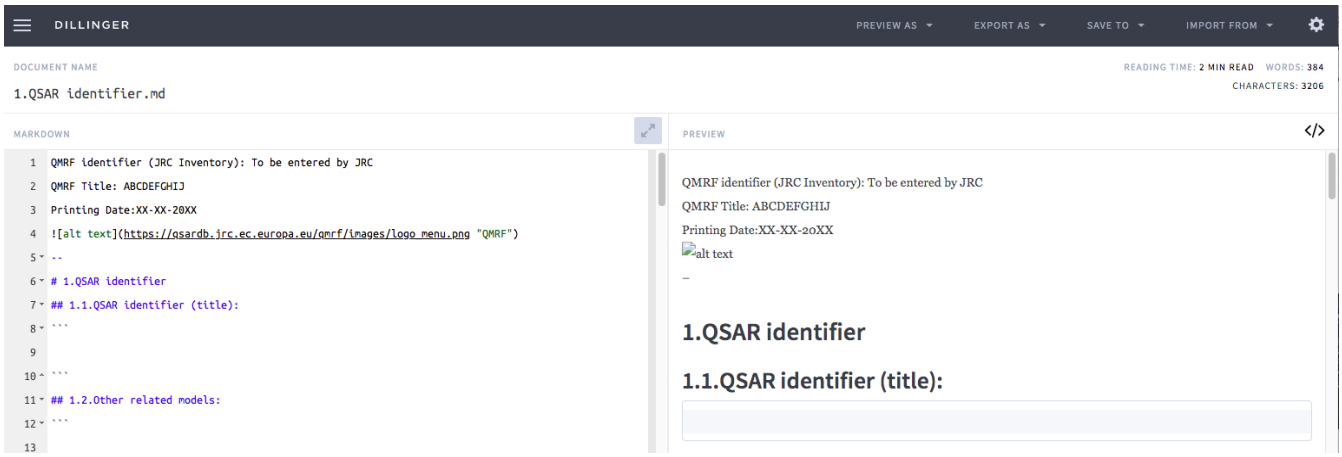
After entering the information, please click on the Save Button to store this information to the system. Descriptions can be updated and extended at any time.



To assist the users in adding editable versions of QMRF reports, we have created a QMRF markdown template, which can be downloaded at:
https://github.com/ntua-unit-of-control-and-informatics/QSAR-Models/blob/master/QMRF%20template.md .
by right clicking on the **Raw** button, as shown:

The user only needs to provide the necessary information under each section and the QMRF report is generated in an easy-to-read format.

There are online tools to edit MarkDown documents, such as: https://dillinger.io/



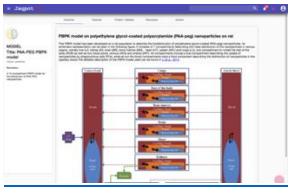
Alternatively, users can use a specialised QMRF editor for Windows at http://qmrf.sourceforge.net/.

4. Test and use your Jaqpot model

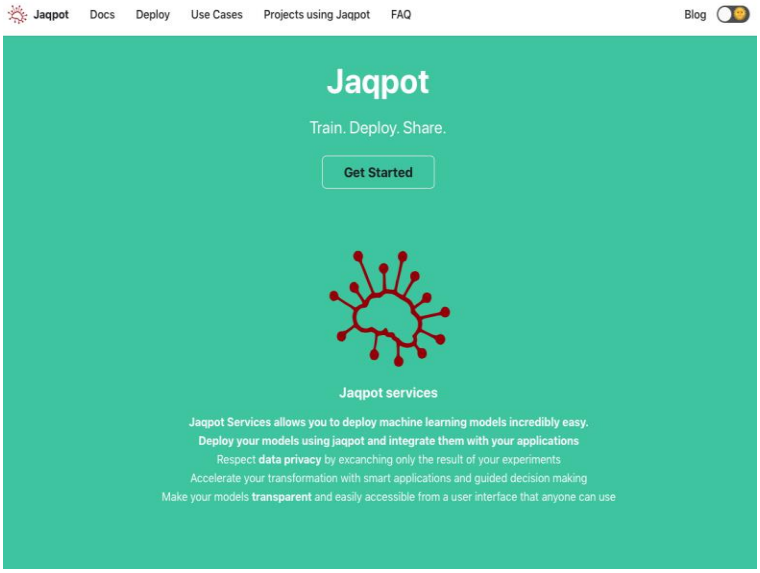
You can validate or use your model or any other Jaqpot model, by following the steps in the following tutorial:
Jaqpot 5: How to access and use an existing predictive model
The data should be provided in a csv file. A sample csv file can be downloaded by clicking on the following link:
https://zenodo.org/record/4671069/files/70_model_reduced.csv?download=1

5. More information on the Jaqpot platform

Jaqpot offers a suite of tools for QSAR, PBPK, image analysis, read-across methods:

			
https://app.jaqpot.org/ QSAR model	https://app.jaqpot.org/ PBPK model	https://nanoimage.jaqpot.org	toxflow.jaqpot.org

A comprehensive technical documentation is provided at https://www.jaqpot.org.



A collection of tutorials describing other Jaqpot functionalities is available on Zenodo:

1. Jaqpot 5 - User accounts
2. Jaqpot 5: How to manage and use organisations
3. Jaqpot 5: How to deploy a predictive model using the jaqpotpy library
4. Jaqpot 5: How to simulate biodistribution scenarios using custom PBPK models