

## Group 31 - feedback for final project resit

### PRELIMINARIES

1. **WORK ON LINUX OR WSL**
  2. Communicate to the lecturer your GitHub usernames. You will be receiving an invitation to join a new GitHub repo as collaborators
  3. On a local machine, delete the .git folder from the project files
    - a. `rm -r .git`
  4. Open the terminal on the folder, type `git init`. This will create another git repository detached from the previous one
  5. In the same terminal, type `git remote add origin https://github.com/rug-oop-2024/group-31-resit.git`
  6. Add every file to the new repo `git add -A`
  7. Commit `git commit -m "First commit of resit"`
  8. Push `git push origin master`. If branch master does not exist, replace “master” with “main”
- Dataset:
    - No dataset is selectable. You get a message like in the picture below

Iris dataset was registered

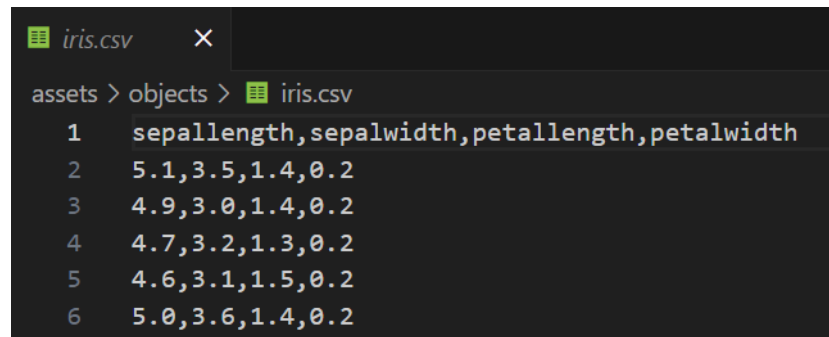
Select a Dataset:

No options to select.

You should let the user import any dataset from a csv file

- Provide a summary of the dataset in this page, like you do in the modelling page
- Allow the user to save the dataset as artifact with a custom name and version id
- The Datasets page works correctly
- Notice that the correct iris dataset has 5 columns - we updated the dataset
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
  - Variety (categorical variable)

In your case, it seems that your csv file is missing a column



	sepal.length	sepal.width	petal.length	petal.width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2

- Modeling:

- From the target selection, remove features which have already been added as input features - we made another variable called available\_features to keep features which are not in the input

## Step 2: Feature Selection

Select input features

sepal.length x

petal.length x

Select target feature:

petal.width

sepal.length

sepal.width

petal.length

petal.width

- Allow the user to choose multiple metrics, using the same structure you used for the input features selection - this was done with an st.multiselect

## Step 5: Select Metrics

Select Metric:

mean\_absolute\_error

mean\_squared\_error

mean\_absolute\_error

root\_mean\_squared\_error

- The metrics are not shown. It may just be a problem with the printing - yes we were taking metrics as a string

Target Feature: petalwidth

Split Ratio: 0.31

**AttributeError: 'MeanAbsoluteError' object has no attribute 'name'**

Traceback:

```
File "/root/miniconda3/envs/oop/lib/python3.10/site-packages/streamlit/runtime
result = func()

File "/root/miniconda3/envs/oop/lib/python3.10/site-packages/streamlit/runtime
exec(code, module.__dict__)

File "/root/Documents/OOP/final_project/oop-24-25-final-project-31/app/pages/2
main()

File "/root/Documents/OOP/final_project/oop-24-25-final-project-31/app/pages/2
pipeline = page.summary(model, dataset,

File "/root/Documents/OOP/final_project/oop-24-25-final-project-31/app/pages/2
st.write("Metrics:", [metric.name for metric in metrics])

File "/root/Documents/OOP/final_project/oop-24-25-final-project-31/app/pages/2
st.write("Metrics:", [metric.name for metric in metrics])
```

As an added step, please verify the metrics with a rule of thumb:

- MSE and MAE on Iris should possibly be  $>0$  and  $<1$
- $R^2$  should be  $>0$  and  $<1$ , at least for linear regression
- Accuracy should be  $> 0.5$  and  $< 1$  for classification tasks
- Check that classification tasks are functional after you add a dataset with a categorical feature
- Deployment
  - Save pipeline as artifact and implement deployment
  - Deployment should happen on a dataset whose feature names correspond [at least] on the input features.
  - Show predictions on target feature
  - In addition, allow computation of metrics on predictions if the imported dataset has a feature name with the same name as the target feature in the pipeline - we get an error stating that the model is not fitted when we try to show predictions. This is probably due to saving but we can not find the problem.
- Style
  - Be careful at private vs public attribute - we checked again and modified a method in Artifact
  - Why would the Model take another Model as an argument of the constructor? You probably mean BaseEstimator from sklearn, Why would you want to be able to switch the 'type' of the model? Each model is either a classifier or a regressor. - we could not implement this
  - The setter for 'parameters' is useless because even if it is used, the underlying sklearn model remains unchanged. Remove it.

- All concrete implementations of model have the same `fit()` and `predict()` methods. To avoid the code duplication, you could create an intermediate class, say `SklearnWrapperModel` that implements these methods. Then, the concrete children would only need to correctly instantiate `self._model`. All models have a `self.model` and a `self._model` - we built a class that does this

## SUBMISSION

- Submit the project on GitHub AND the dedicated resit assignment on Brightspace by Jan 26th
- Add to GitHub a copy of this document in which you briefly explain how the feedback has been implemented
- We believe the Datasets and Modelling pages work correctly, and we also implemented the Deployment page. However, we think there is a problem with the saving of the pipeline that stops us from using deployment even though the whole functionality of the page is written.