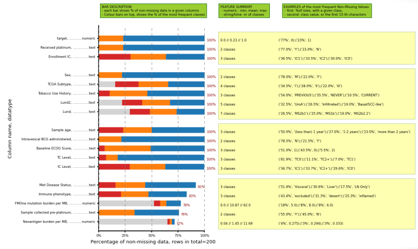
EDA & DEV.

PIPELINE

Notebook 1

- ❑ EDA
- ❑ Feature Selection
- ❑ Testing Pre-processing Steps



Patient Data

Tpm Data

Data

scikit learn

ML Models

External recourses

+

My recourses

PY class

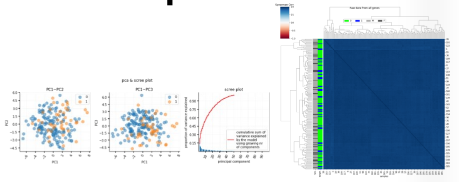
Data Frame Explorer

SkinDiagnosticAI

Notebook 2

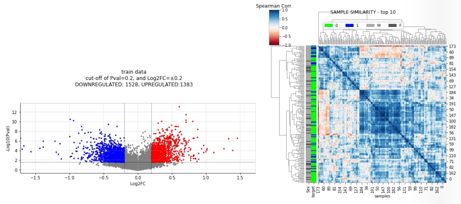
Development of:

- ❑ Custom trans-formers
- ❑ QC methods
- ❑ Differential Gene Expr
- ❑ Reports



Notebook 3

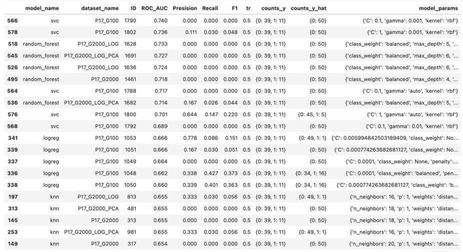
- ❑ Cleaning
- ❑ Outlier removal
- ❑ Scaling
- ❑ Differential Gene expression



Notebook 4

- ❑ Model Training
- ❑ Prediction saving
- ❑ GridSearch

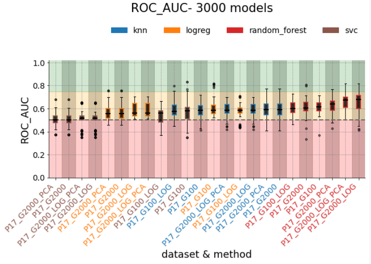
~3000 models



	model_name	dataset_name	ID	ROC_AUC	Presision	Recall	F1	tr	counts_y	counts_y_hat	model_params
566	svc	P17_G100	1790	0.740	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}
578	svc	P17_G100	1802	0.736	0.111	0.030	0.048	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
518	random_forest	P17_G2000_LOG	1628	0.733	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 5, '...
545	random_forest	P17_G2000_LOG_PCA	1691	0.727	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 6, '...
526	random_forest	P17_G2000_LOG	1636	0.724	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 6, '...
495	random_forest	P17_G2000	1461	0.718	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 4, '...
564	svc	P17_G100	1788	0.717	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 'auto', 'kernel': 'rbf'}
536	random_forest	P17_G2000_LOG_PCA	1682	0.714	0.167	0.026	0.044	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 5, '...
576	svc	P17_G100	1800	0.701	0.644	0.147	0.220	0.5	{0: 39, 1: 11}	{0: 45, 1: 5}	{'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
568	svc	P17_G100	1792	0.689	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'}
341	logreg	P17_G100	1053	0.666	0.778	0.086	0.151	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'C': 0.005994842503189409, 'class_weight': No...
339	logreg	P17_G100	1051	0.666	0.167	0.030	0.051	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.000774263682681127, 'class_weight': No...
337	logreg	P17_G100	1049	0.664	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.0001, 'class_weight': None, 'penalty':...
336	logreg	P17_G100	1048	0.662	0.338	0.427	0.373	0.5	{0: 39, 1: 11}	{0: 34, 1: 16}	{'C': 0.0001, 'class_weight': 'balanced', 'pen...
338	logreg	P17_G100	1050	0.660	0.339	0.401	0.363	0.5	{0: 39, 1: 11}	{0: 34, 1: 16}	{'C': 0.000774263682681127, 'class_weight': 'b...
197	knn	P17_G2000_LOG	813	0.655	0.333	0.030	0.056	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'n_neighbors': 16, 'p': 1, 'weights': 'distan...
313	knn	P17_G2000_PCA	481	0.655	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 18, 'p': 1, 'weights': 'distan...
145	knn	P17_G2000	313	0.655	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 18, 'p': 1, 'weights': 'distan...
253	knn	P17_G2000_LOG_PCA	981	0.655	0.333	0.030	0.056	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'n_neighbors': 16, 'p': 1, 'weights': 'distan...
149	knn	P17_G2000	317	0.654	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 20, 'p': 1, 'weights': 'distan...

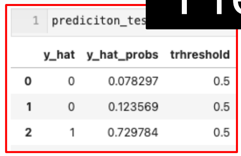
Notebook 5

- ❑ ROC analysis
- ❑ Model Selection
- ❑ Threshold calibration



Results

Predictions

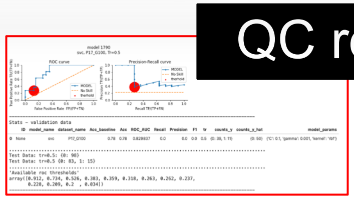


```
1  prediciton_tes
```

	y_hat	y_hat_probs	trhreshold
0	0	0.078297	0.5
1	0	0.123569	0.5
2	1	0.729784	0.5

In Ardigen/data/results

QC report



```
Stats - validation data
```

	ID	model_name	dataset_name	Acc_baseline	Acc	ROC_AUC	Recall	Presision	F1	tr	counts_y	counts_y_hat	model_params
0	None	svc	P17_G100	0.78	0.78	0.829837	0.0	0.0	0.0	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}

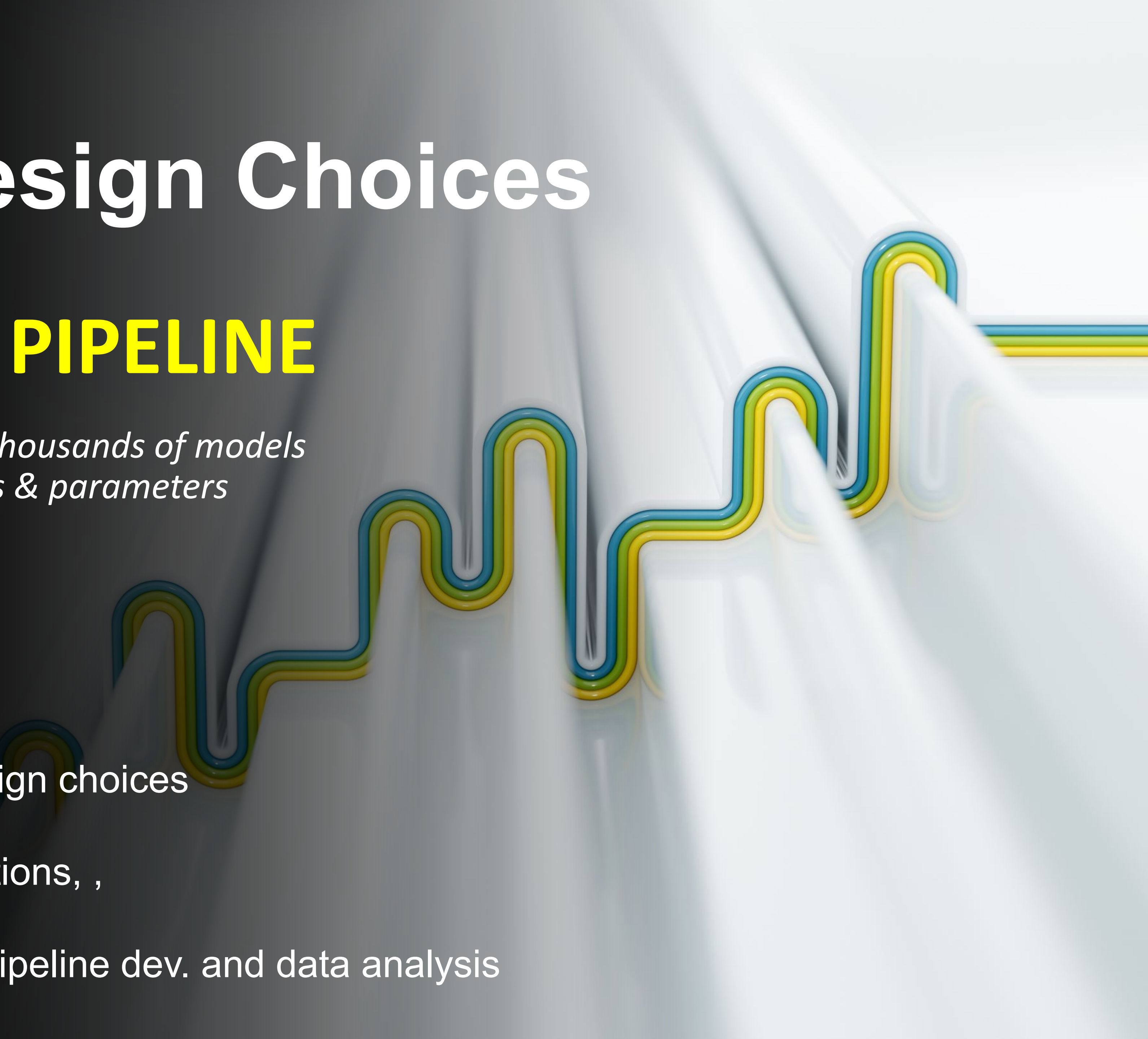
```
Test Data: tr=0.5: {0: 98}
Test Data: tr=0.5 {0: 83, 1: 15}
'Available roc thresholds'
array([0.912, 0.734, 0.526, 0.383, 0.359, 0.318, 0.263, 0.262, 0.237,
       0.228, 0.209, 0.2  , 0.034])
```

In Notebook 05

Early-Stage Design Choices

TO CREATE A PIPELINE

For running & selection of thousands of models
with different datasets & parameters

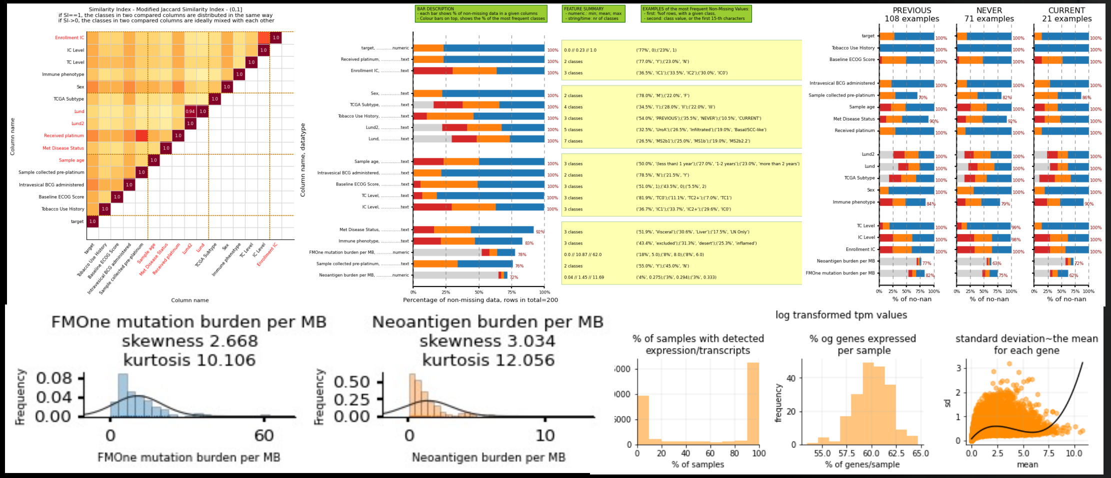


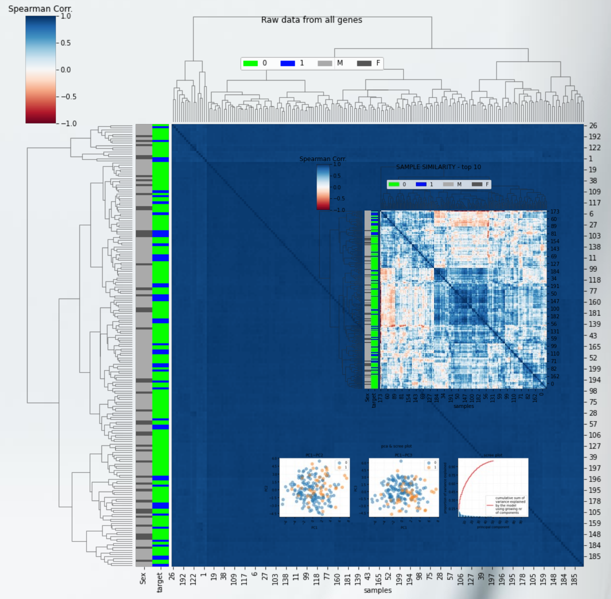
- Divide the work into smaller pieces
- Use parameters for controlling design choices
- Consistent naming conventions,
- Ability to add new steps and conditions, ,
- QC data collected at each step
- the same functions used in EDA, pipeline dev. and data analysis

Early-Stage Design Choices

MAKE IT USER FRIENDLY

- *Similar layout in all notebooks*
- *Informative names*
- *Help for all functions,*
- *Evaluation with plots and summary tables*





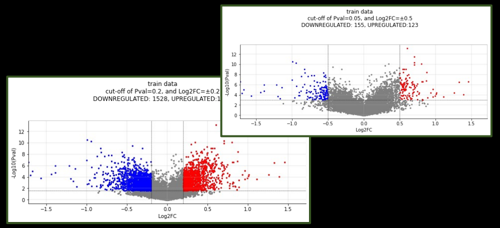
Early-Stage Design Choices

TO CREATE ALL THE FUNCTIONS

I used only most basic functions, from open source packages such as numpy, pandas, Matplotlib, and skleanr, to crate all functions, presented in this project

That includes:

- Custom transformers,
- QC reports,
- Pipeline for differential gene expression
- And more…

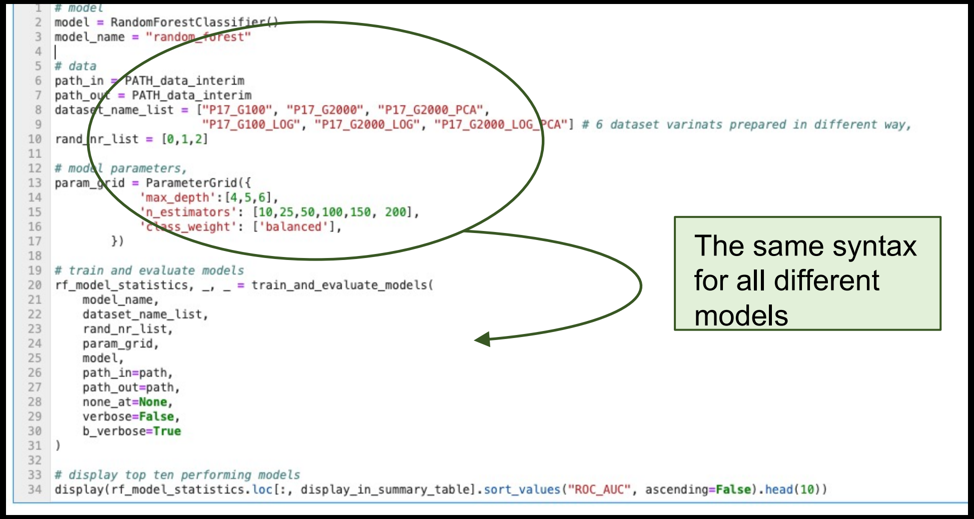

train data
cut-off of Pval=0.05, and Log2FC=±0.5
DOWNREGULATED: 155, UPREGULATED:123

train data
cut-off of Pval=0.2, and Log2FC=±0.2
DOWNREGULATED: 1528, UPREGULATED:1

Model Training

Notebook 04

I created large number of model (>3000) with 4 different techniques



```
# model
model = RandomForestClassifier()
model_name = "random_forest"

# data
path_in = PATH_data_interim
path_out = PATH_data_interim
dataset_name_list = ["P17_G100", "P17_G2000", "P17_G2000_PCA",
                     "P17_G100_LOG", "P17_G2000_LOG", "P17_G2000_LOG_PCA"] # 6 dataset varinats prepared in different way,
rand_nr_list = [0,1,2]

# model parameters,
param_grid = ParameterGrid({
            'max_depth':[4,5,6],
            'n_estimators': [10,25,50,100,150, 200],
            'class_weight': ['balanced'],
        })

# train and evaluate models
rf_model_statistics, _, _ = train_and_evaluate_models(
    model_name,
    dataset_name_list,
    rand_nr_list,
    param_grid,
    model,
    path_in=path,
    path_out=path,
    none_at=None,
    verbose=False,
    b_verbose=True
)

# display top ten performing models
display(rf_model_statistics.loc[:, display_in_summary_table].sort_values("ROC_AUC", ascending=False).head(10))
```

The same syntax for all different models

I USED FOUD DIFFERENT ALGORITHMS –

- **Logistic regression** – classic solution for binary classification problems
- **Random Forest** – with different tree nr, and depth
- **SVM** – to apply kernel trick, for samples mixed in feature space

and SIX DIFFERENT DATASETS with

- **different number of expressed differentially genes (~100, or ~2000)**
- **data from potential outliers or not**
- **I could use only patient data or only tpm data, (I had no time to do that but there is a simple parameter in the pipeline that allows that)**
- **Different scaling methods ….**
- **And many more choices that may be introduced and tested in Notebooks 2 and 3.**

Model Performance

Notebook 05

Because of Large number of models (~3000 in total)

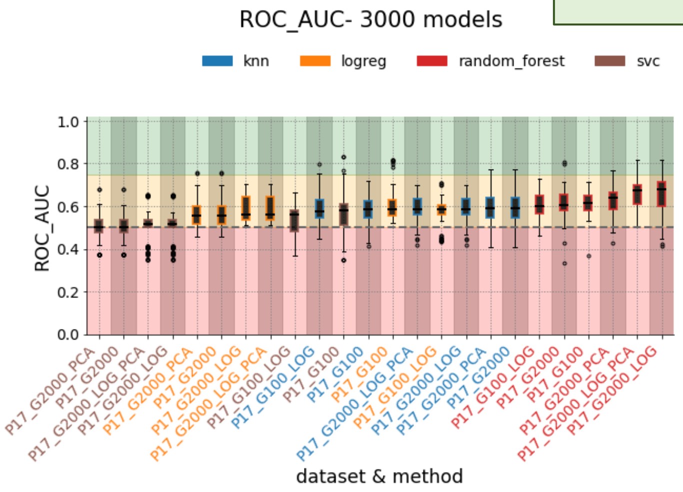
I provided three types of reports (plots and tables), to select and fine tune ML models in notebook 5

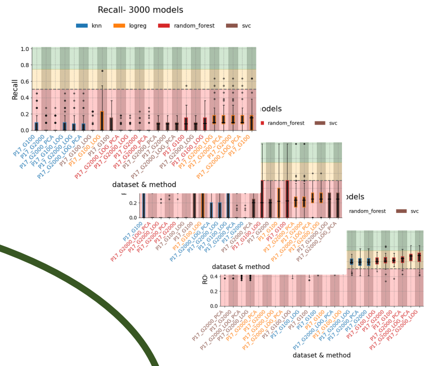
- **High Level Performance report:**
  - For tuning feature selection and data preprocessing pipeline
  - comparing methods, such as knn, svm, nn implemented
- **Intermediate Level Performance report:**
  - to select best model comparing methods,

  such as knn, svm, nn
- **Low High Level Performance report:**
  - Detailed examination of the best possible candidates, with large number of available statistics
  - hyperparameters, compare with similar models,
  - P threshold for classification
  - Plots, with ROC, PR curves,
  - Confusion matrices and more ….

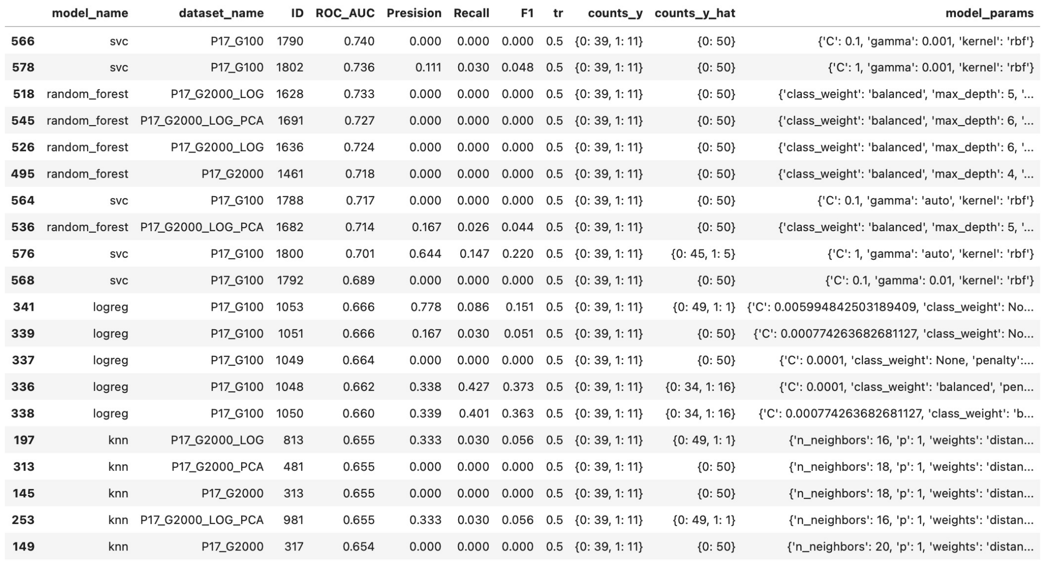
High Level

Compare any number of models on few plots





Intermediate Level



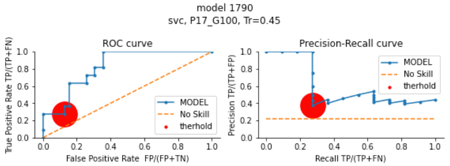
	model_name	dataset_name	ID	ROC_AUC	Presision	Recall	F1	tr	counts_y	counts_y_hat	model_params
566	svc	P17_G100	1790	0.740	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}
578	svc	P17_G100	1802	0.736	0.111	0.030	0.048	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
518	random_forest	P17_G2000_LOG	1628	0.733	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 5, '...
545	random_forest	P17_G2000_LOG_PCA	1691	0.727	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 6, '...
526	random_forest	P17_G2000_LOG	1636	0.724	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 6, '...
495	random_forest	P17_G2000	1461	0.718	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 4, '...
564	svc	P17_G100	1788	0.717	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 'auto', 'kernel': 'rbf'}
536	random_forest	P17_G2000_LOG_PCA	1682	0.714	0.167	0.026	0.044	0.5	{0: 39, 1: 11}	{0: 50}	{'class_weight': 'balanced', 'max_depth': 5, '...
576	svc	P17_G100	1800	0.701	0.644	0.147	0.220	0.5	{0: 39, 1: 11}	{0: 45, 1: 5}	{'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
568	svc	P17_G100	1792	0.689	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.1, 'gamma': 0.01, 'kernel': 'rbf'}
341	logreg	P17_G100	1053	0.666	0.778	0.086	0.151	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'C': 0.005994842503189409, 'class_weight': No...
339	logreg	P17_G100	1051	0.666	0.167	0.030	0.051	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.000774263682681127, 'class_weight': No...
337	logreg	P17_G100	1049	0.664	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'C': 0.0001, 'class_weight': None, 'penalty':...
336	logreg	P17_G100	1048	0.662	0.338	0.427	0.373	0.5	{0: 39, 1: 11}	{0: 34, 1: 16}	{'C': 0.0001, 'class_weight': 'balanced', 'pen...
338	logreg	P17_G100	1050	0.660	0.339	0.401	0.363	0.5	{0: 39, 1: 11}	{0: 34, 1: 16}	{'C': 0.000774263682681127, 'class_weight': 'b...
197	knn	P17_G2000_LOG	813	0.655	0.333	0.030	0.056	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'n_neighbors': 16, 'p': 1, 'weights': 'distan...
313	knn	P17_G2000_PCA	481	0.655	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 18, 'p': 1, 'weights': 'distan...
145	knn	P17_G2000	313	0.655	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 18, 'p': 1, 'weights': 'distan...
253	knn	P17_G2000_LOG_PCA	981	0.655	0.333	0.030	0.056	0.5	{0: 39, 1: 11}	{0: 49, 1: 1}	{'n_neighbors': 16, 'p': 1, 'weights': 'distan...
149	knn	P17_G2000	317	0.654	0.000	0.000	0.000	0.5	{0: 39, 1: 11}	{0: 50}	{'n_neighbors': 20, 'p': 1, 'weights': 'distan...

Select best candidate and compare with each other

- Consistent nomenclature
- And unique ID for each model

Low Level

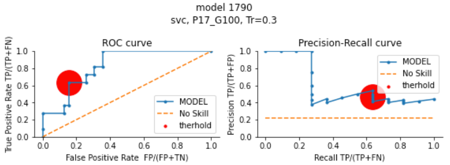
Fine tuning of model hyperparameters,



```
------------------------------------------------------------------------------------
Stats - validation data
```

	ID	model_name	dataset_name	Acc_baseline	Acc	ROC_AUC	Recall	Presision	F1	tr	counts_y	counts_y_hat	model_params
0	None	svc	P17_G100	0.78	0.74	0.829837	0.272727	0.375	0.315789	0.45	{0: 39, 1: 11}	{0: 42, 1: 8}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}

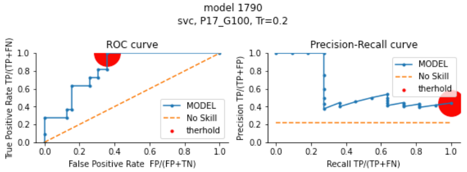
```
...........................................................................
Test Data: tr=0.5: {0: 98}
Test Data: tr=0.45 {0: 81, 1: 17}
...........................................................................
'Available roc thresholds'
array([0.912, 0.734, 0.526, 0.383, 0.359, 0.318, 0.263, 0.262, 0.237,
       0.228, 0.209, 0.2  , 0.034])
------------------------------------------------------------------------------------
```



```
------------------------------------------------------------------------------------
Stats - validation data
```

	ID	model_name	dataset_name	Acc_baseline	Acc	ROC_AUC	Recall	Presision	F1	tr	counts_y	counts_y_hat	model_params
0	None	svc	P17_G100	0.78	0.76	0.829837	0.636364	0.466667	0.538462	0.3	{0: 39, 1: 11}	{0: 35, 1: 15}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}

```
...........................................................................
Test Data: tr=0.5: {0: 98}
Test Data: tr=0.3 {0: 69, 1: 29}
...........................................................................
'Available roc thresholds'
array([0.912, 0.734, 0.526, 0.383, 0.359, 0.318, 0.263, 0.262, 0.237,
       0.228, 0.209, 0.2  , 0.034])
------------------------------------------------------------------------------------
```



```
------------------------------------------------------------------------------------
Stats - validation data
```

	ID	model_name	dataset_name	Acc_baseline	Acc	ROC_AUC	Recall	Presision	F1	tr	counts_y	counts_y_hat	model_params
0	None	svc	P17_G100	0.78	0.72	0.829837	1.0	0.44	0.611111	0.2	{0: 39, 1: 11}	{0: 25, 1: 25}	{'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}

```
...........................................................................
Test Data: tr=0.5: {0: 98}
Test Data: tr=0.2 {0: 60, 1: 38}
...........................................................................
'Available roc thresholds'
array([0.912, 0.734, 0.526, 0.383, 0.359, 0.318, 0.263, 0.262, 0.237,
       0.228, 0.209, 0.2  , 0.034])
------------------------------------------------------------------------------------
```