Integration of Meta Machine Learning in GenoML

Authors:

Eduardo Salmerón Castaño
Juan A. Botía

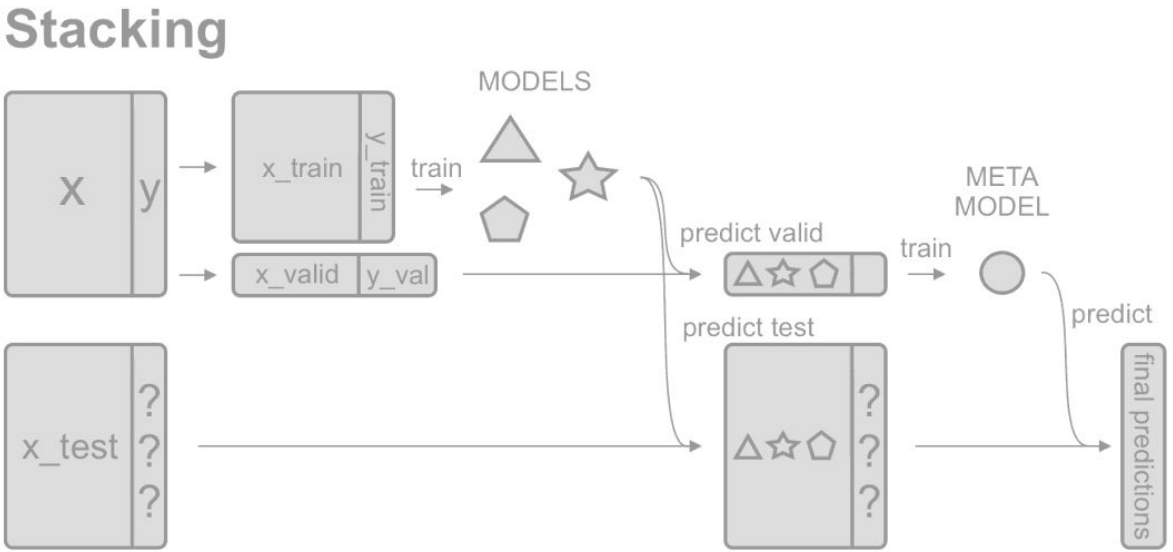
Index

1. Meta Machine Learning.
2. Code.
3. Using the tool.

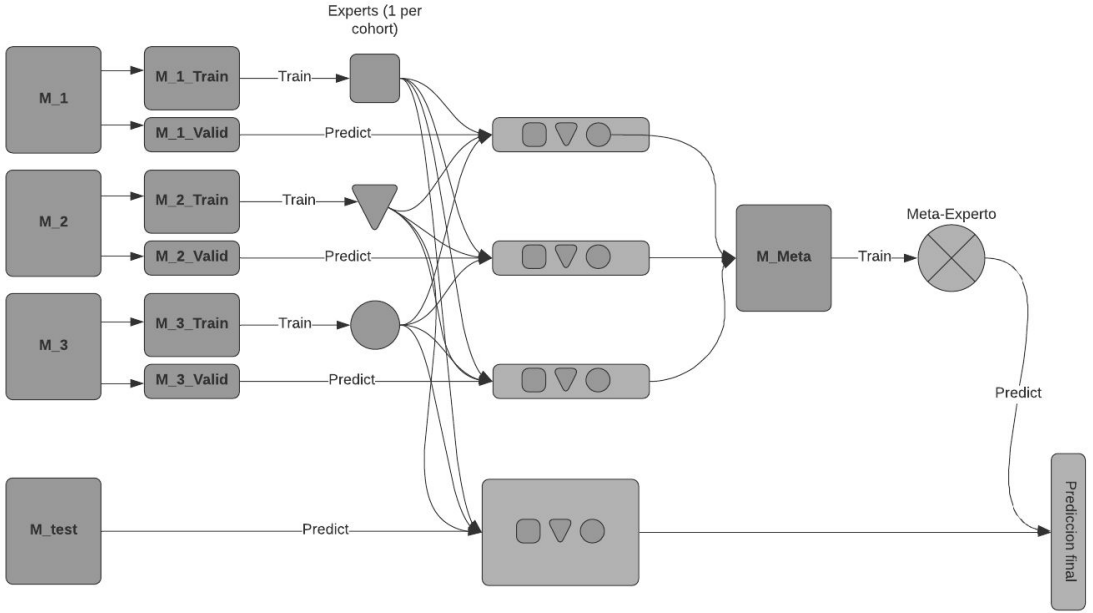
Index

1. **Meta Machine Learning.**
2. Code.
3. Using the tool.

Meta-ml as a basic algorithm in GenoML



Meta-ML as a multi-silo integration algorithm in GenoML



Loss of accuracy for using a multi-silo approach with nSmE

Algorithm	Mean	Confidence interval
QuadraticDiscriminantAnalysis	50.3%	48.3 - 52.3
RandomForestClassifier	56.7%	54.5 - 58.9
MML nSmE	62.5%	58.7 - 66.2
LogisticRegression	71.7%	71.7 - 71.7
SGDClassifier	72.1%	68.5 - 75.7

Index

1. Introducción.
2. **Code.**
3. Using the tool.

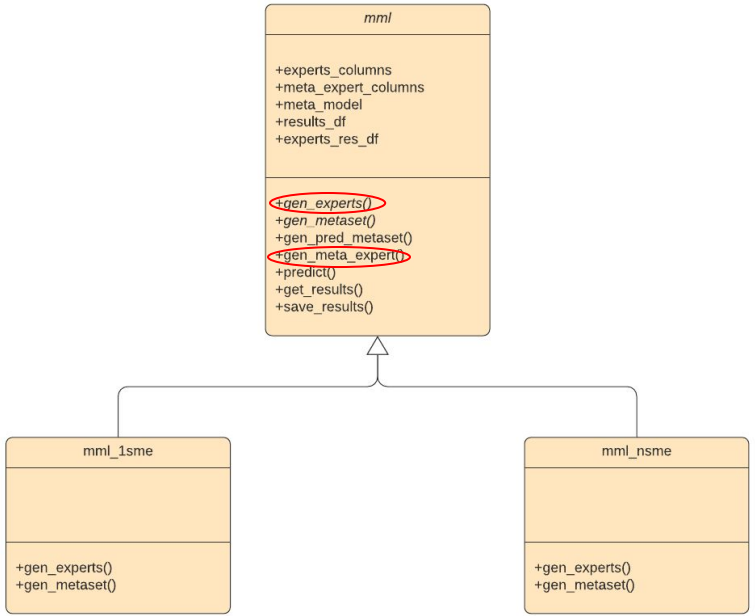
Code

Class hierarchy:

- Abstract class mml
- Classes inheriting from mml

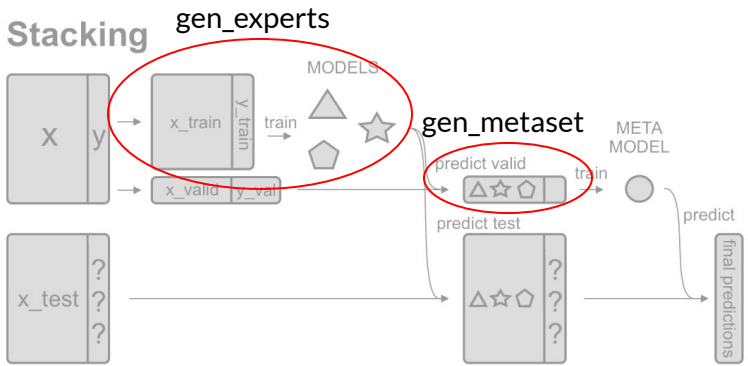
gen_experts and gen_meta_expert use GenoML base functionality.

https://github.com/edusalcas/TFG_MML_GenoML



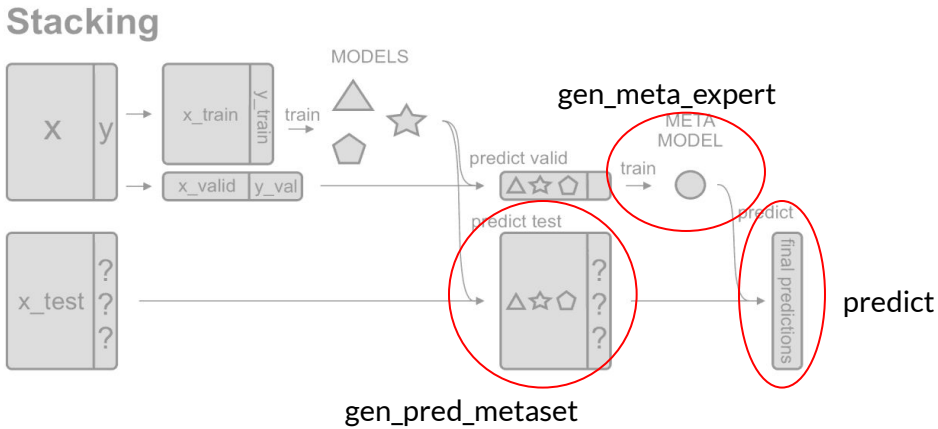
Code - mml abstract methods

```
68
69      @abstractmethod
70      def gen_experts(self,
71                      path,
72                      expert_names=None,
73                      prefix=None,
74                      metric_max=None,
75                      seed=None,
76                      metric_tune=None,
77                      max_tune=None,
78                      algs=None):
79          pass
80
81      @abstractmethod
82      def gen_metaset(self,
83                      data,
84                      experts_path,
85                      experts_names):
86          pass
87
```



Code - mml methods

```
87
88      def gen_pred_metaset(self, data, experts_path, experts_names, path, meta_prefix): ...
117
118     def gen_meta_expert(self, data, path, experts_names, prefix, metric_max, algs, seed, metric_tune, max_tune): ...
135
136     def predict(self, data, path, experts_names, meta_prefix, iteration=0): ...
152
```



Code - mml_1sme and mml_nsme methods

```
12
13  class MML_1SmE(MML):
14
15      def gen_experts(self, path, expert_names=None, prefix=None, metric_max=None, s
28
29      def gen_metaset(self, data, experts_path, experts_names): ...
49
```

```
12
13  class MML_nSmE(MML):
14
15      def gen_experts(self, path, expert_names=None, prefix=None, metric_max=None, se
28
29      def gen_metaset(self, data, experts_path, experts_names): ...
71
```

Index

1. Introducción.
2. Code.
3. **Using the tool.**

Workflow

1. Munge data.		1. Munging from GenoML.
2. Split train-test.		2. train_test_split from sklear
3. Generate level one experts.	---->	**3. gen_experts from our code**
4. Generate meta-expert.		**4. gen_meta_expert from our code**
5. Predict.		**5. predict from our code**

In points 3 and 4, we use train and tune functions from GenoML.

Example - Experiment with MML 1SmE

```
109
110     manager_mml = MML_1SmE(actual_path + prefix)
111
112     # Generate level 1 experts
113     manager_mml.gen_experts(path=actual_path,*
114                             expert_names=algorithms,*
115                             prefix=train_prefix,*
116                             metric_max=metric,*
117                             seed=seed,
118                             metric_tune=metric,*
119                             max_tune=tune_iterations)*
120
121     # Generate level 2 expert
122     manager_mml.gen_meta_expert(data=dataForML_test,
123                                 path=actual_path,*
124                                 experts_names=algorithms,
125                                 prefix=meta_prefix,*
126                                 metric_max=metric,*
127                                 algs=algorithms_names,*
128                                 seed=seed,
129                                 metric_tune=metric,*
130                                 max_tune=tune_iterations)*
131
132     # Predict test set
133     balacc = manager_mml.predict(data=test_df,
134                                  path=actual_path,
135                                  experts_names=algorithms,
136                                  meta_prefix=meta_prefix,
137                                  iteration=n)
```

*: Parameters needed for base GenoML functions
*: Parameters needed for modified GenoML Train Function.

UNIVERSIDAD DE MURCIA

End

Thanks for your attention