Preliminary models

Álvaro Román Gómez

5/1/23

Table of contents

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pycaret.classification import *

# IMPORT CUSTOM MODULES
import sys

sys.path.append("../src")
import utils.stats as st
```

```
# DIRECTORIES
input_path = "../data/processed/"
train_path = "../data/processed/train_data/"
test_path = "../data/processed/test_data/"
# FILES
# MOLECULAR DESCRIPTORS
molecular_descriptors_training_file = "molecular_descriptors_training.csv"
molecular_descriptors_test_file = "molecular_descriptors_test.csv"
# MACCS KEYS
maccs_keys_training_file = "maccs_keys_training.csv"
maccs_keys_test_file = "maccs_keys_test.csv"
# ECFP4 FINGERPRINTS
ecfp4_fingerprints_training_file = "ecfp4_fingerprints_training.csv"
ecfp4_fingerprints_test_file = "ecfp4_fingerprints_test.csv"
```

```
# LOAD DATA
# MOLECULAR DESCRIPTORS
## TRAINING
molecular_descriptors_training = pd.read_csv(
    train_path + molecular_descriptors_training_file
)
X_training_molecular_descriptors = molecular_descriptors_training.drop(
    columns=["activity"]
)
Y_training_molecular_descriptors = molecular_descriptors_training["activity"]
## TEST
molecular_descriptors_test = pd.read_csv(test_path + molecular_descriptors_test_file)
X_test_molecular_descriptors = molecular_descriptors_test.drop(columns=["activity"])
Y_test_molecular_descriptors = molecular_descriptors_test["activity"]
# MACCS KEYS
## TRAINING
```

```
macc_keys_training = pd.read_csv(train_path + maccs_keys_training_file)
X_training_maccs_keys = macc_keys_training.drop(columns=["activity"])
Y_training_maccs_keys = macc_keys_training["activity"]
## TEST
macc_keys_test = pd.read_csv(test_path + maccs_keys_test_file)
X_test_maccs_keys = macc_keys_test.drop(columns=["activity"])
Y_test_maccs_keys = macc_keys_test["activity"]
# ECFP4 FINGERPRINTS
## TRAINING
ecfp4_fingerprints_training = pd.read_csv(train_path + ecfp4_fingerprints_training_f
X_training_ecfp4_fingerprints = ecfp4_fingerprints_training.drop(columns=["activity'
Y_training_ecfp4_fingerprints = ecfp4_fingerprints_training["activity"]
## TEST
ecfp4_fingerprints_test = pd.read_csv(test_path + ecfp4_fingerprints_test_file)
X_test_ecfp4_fingerprints = ecfp4_fingerprints_test.drop(columns=["activity"])
Y_test_ecfp4_fingerprints = ecfp4_fingerprints_test["activity"]
```

Chapter 1

PRELIMINARY MODELS

1.1 MOLECULAR DESCRIPTORS

```
# CREATE MODELS WITH PYCARET
molecular_descriptors_models = setup(
    data=molecular_descriptors_training,
    target="activity",
    test_data=molecular_descriptors_test,
    session_id=123,
)
```

Table 1.1

	Description	Value
0	Session id	123
1	Target	activity
2	Target type	Binary
3	Original data shape	(299, 87)
4	Transformed data shape	(419, 87)
5	Transformed train set shape	(299, 87)
6	Transformed test set shape	(120, 87)
7	Numeric features	86
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10

	Description	Value
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	3863

```
# COMPARE MODELS
best_model = compare_models(verbose=True)
```

```
<IPython.core.display.HTML object>
```

Table 1.2

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
lda	Linear Discriminant Analysis	0.6518	0.6510	0.6305	0.6707	0.6419	0.3038
ada	Ada Boost Classifier	0.6386	0.6769	0.6519	0.6401	0.6398	0.2783
xgboost	Extreme Gradient Boosting	0.6323	0.6988	0.6248	0.6371	0.6226	0.2647
gbc	Gradient Boosting Classifier	0.6322	0.7008	0.6043	0.6383	0.6102	0.2643
rf	Random Forest Classifier	0.6321	0.6875	0.6314	0.6321	0.6290	0.2646
lightgbm	Light Gradient Boosting Machine	0.6220	0.6915	0.6110	0.6225	0.6129	0.2442
et	Extra Trees Classifier	0.6153	0.6901	0.6100	0.6145	0.6091	0.2301
lr	Logistic Regression	0.6053	0.6471	0.5776	0.6101	0.5885	0.2109
ridge	Ridge Classifier	0.5986	0.0000	0.5581	0.6024	0.5736	0.1979
qda	Quadratic Discriminant Analysis	0.5890	0.6274	0.4448	0.6556	0.5001	0.1780
knn	K Neighbors Classifier	0.5789	0.6225	0.5514	0.5820	0.5577	0.1579
dt	Decision Tree Classifier	0.5747	0.5746	0.5762	0.5797	0.5742	0.1496
svm	SVM - Linear Kernel	0.5684	0.0000	0.5224	0.5864	0.5288	0.1357
nb	Naive Bayes	0.5456	0.6463	0.1510	0.5109	0.2048	0.0910
dummy	Dummy Classifier	0.5017	0.5000	0.0000	0.0000	0.0000	0.0000

```
<IPython.core.display.HTML object>
```

```
# TUNE MODELS
# tuned_molecular_descriptors_models = tune_model(best_model, optimize="AUC", n_iter
```

1.2 MACCS KEYS

```
# CREATE MODELS WITH PYCARET
maccs_keys_models = setup(
    data=macc_keys_training,
    target="activity",
    test_data=macc_keys_test,
    session_id=123,
)
```

Table 1.3

	Description	Value
0	Session id	123
1	Target	activity
2	Target type	Binary
3	Original data shape	(299, 168)
4	Transformed data shape	(419, 168)
5	Transformed train set shape	(299, 168)
6	Transformed test set shape	(120, 168)
7	Numeric features	167
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	e3dd

```
# COMPARE MODELS
compare_models(verbose=True)
```

```
<IPython.core.display.HTML object>
```

Table 1.4

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	T
lda	Linear Discriminant Analysis	0.6857	0.7090	0.7119	0.6816	0.6910	0.3717	0.3782	0

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
lightgbm	Light Gradient Boosting Machine	0.6622	0.6990	0.6705	0.6674	0.6637	0.3240
gbc	Gradient Boosting Classifier	0.6521	0.7202	0.6438	0.6591	0.6440	0.3039
xgboost	Extreme Gradient Boosting	0.6487	0.6951	0.6710	0.6493	0.6550	0.2976
lr	Logistic Regression	0.6454	0.6798	0.6176	0.6564	0.6293	0.2909
rf	Random Forest Classifier	0.6454	0.6764	0.6843	0.6366	0.6564	0.2909
ridge	Ridge Classifier	0.6421	0.0000	0.6448	0.6424	0.6343	0.2846
dt	Decision Tree Classifier	0.6355	0.6474	0.6514	0.6316	0.6371	0.2713
qda	Quadratic Discriminant Analysis	0.6286	0.6992	0.6243	0.6427	0.6207	0.2575
ada	Ada Boost Classifier	0.6220	0.6836	0.6310	0.6275	0.6234	0.2442
knn	K Neighbors Classifier	0.6187	0.6371	0.6443	0.6166	0.6245	0.2376
et	Extra Trees Classifier	0.6153	0.6840	0.6371	0.6107	0.6202	0.2305
nb	Naive Bayes	0.5924	0.6513	0.3571	0.6584	0.4391	0.1842
svm	SVM - Linear Kernel	0.5586	0.0000	0.5081	0.5835	0.5029	0.1151
dummy	Dummy Classifier	0.5017	0.5000	0.0000	0.0000	0.0000	0.0000

```
<IPython.core.display.HTML object>
```

```
LinearDiscriminantAnalysis(covariance_estimator=None, n_components=None,
                           priors=None, shrinkage=None, solver='svd',
                           store_covariance=False, tol=0.0001)
```

1.3 ECFP4 FINGERPRINTS

```
# CREATE MODELS WITH PYCARET
ecfp4_fingerprints_models = setup(
    data=ecfp4_fingerprints_training,
    target="activity",
    test_data=ecfp4_fingerprints_test,
    session_id=123,
)
```

Table 1.5

	Description	Value
0	Session id	123
1	Target	activity
2	Target type	Binary
3	Original data shape	(299, 1025)
4	Transformed data shape	(419, 1025)
5	Transformed train set shape	(299, 1025)
6	Transformed test set shape	(120, 1025)
7	Numeric features	1024

	Description	Value
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	f45e

```
# COMPARE MODELS
compare_models(verbose=True)
```

```
<IPython.core.display.HTML object>
```

Table 1.6

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	T
ridge	Ridge Classifier	0.7023	0.0000	0.7048	0.7106	0.7044	0.4047	0.4083	0
lr	Logistic Regression	0.6921	0.7363	0.6505	0.7137	0.6761	0.3839	0.3890	0
xgboost	Extreme Gradient Boosting	0.6787	0.7341	0.6643	0.6777	0.6669	0.3576	0.3615	0
rf	Random Forest Classifier	0.6655	0.7230	0.6919	0.6746	0.6727	0.3316	0.3416	0
gbc	Gradient Boosting Classifier	0.6653	0.7298	0.6376	0.6837	0.6561	0.3309	0.3351	0
ada	Ada Boost Classifier	0.6585	0.7308	0.6300	0.6767	0.6460	0.3167	0.3232	0
et	Extra Trees Classifier	0.6522	0.7255	0.7052	0.6411	0.6671	0.3049	0.3100	0
svm	SVM - Linear Kernel	0.6421	0.0000	0.6038	0.6791	0.6222	0.2839	0.2952	0
lightgbm	Light Gradient Boosting Machine	0.6320	0.6829	0.6376	0.6384	0.6315	0.2642	0.2681	0
dt	Decision Tree Classifier	0.6122	0.6126	0.6719	0.6011	0.6326	0.2249	0.2284	0
nb	Naive Bayes	0.6087	0.6095	0.8186	0.5789	0.6748	0.2183	0.2479	0
lda	Linear Discriminant Analysis	0.6086	0.6630	0.6710	0.5933	0.6273	0.2175	0.2201	0
qda	Quadratic Discriminant Analysis	0.5885	0.5883	0.6833	0.5852	0.6240	0.1767	0.1813	0
knn	K Neighbors Classifier	0.5852	0.6566	0.6243	0.5840	0.5986	0.1709	0.1733	0
dummy	Dummy Classifier	0.5017	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0

```
<IPython.core.display.HTML object>
```

```
RidgeClassifier(alpha=1.0, class_weight=None, copy_X=True, fit_intercept=True,
                max_iter=None, positive=False, random_state=123, solver='auto',
                tol=0.0001)
```