Lazy Predict: fit and evaluate all the models from scikit-learn with a single line of code

Submitted by :- Ambarish Singh

What is LazyPredict in Machine Learning ?

- lazypredict is a convenient wrapper library, that enables us to quickly fit all the models to our dataset and compare their performance. This way, we can see what works well “out of the box”.

Installing lazypredict Library

In [1]:

```
pip install lazypredict
```

```
Collecting lazypredict
  Downloading lazypredict-0.2.12-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages (from lazypredict)
(1.5.1)
Collecting xgboost
  Downloading xgboost-1.7.1-py3-none-manylinux2014_x86_64.whl (193.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 193.6/193.6 MB 14.3 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: joblib in /opt/conda/lib/python3.10/site-packages (from lazypredict)
(1.2.0)
Requirement already satisfied: click in /opt/conda/lib/python3.10/site-packages (from lazypredict)
(8.1.3)
Collecting lightgbm
  Downloading lightgbm-3.3.3-py3-none-manylinux1_x86_64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 76.4 MB/s eta 0:00:00
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.10/site-packages (from lazypre
dict) (1.1.2)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-packages (from lazypredict)
(4.64.1)
Requirement already satisfied: scipy in /opt/conda/lib/python3.10/site-packages (from lightgbm->lazy
predict) (1.9.3)
Requirement already satisfied: numpy in /opt/conda/lib/python3.10/site-packages (from lightgbm->lazy
predict) (1.23.4)
Requirement already satisfied: wheel in /opt/conda/lib/python3.10/site-packages (from lightgbm->lazy
predict) (0.37.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from
scikit-learn->lazypredict) (3.1.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/conda/lib/python3.10/site-packages (fr
om pandas->lazypredict) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas-
>lazypredict) (2022.5)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-date
util>=2.8.1->pandas->lazypredict) (1.16.0)
Installing collected packages: xgboost, lightgbm, lazypredict
Successfully installed lazypredict-0.2.12 lightgbm-3.3.3 xgboost-1.7.1
Note: you may need to restart the kernel to use updated packages.
```

Importing All Necessary Library

In [2]:

```
from lazypredict.Supervised import LazyClassifier, LazyRegressor
from sklearn.model_selection import train_test_split
from sklearn import datasets
```

Classification Model task

In [4]:

```
# load data
data = datasets.load_breast_cancer()
X, y = data.data, data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)

# fit all models
clf = LazyClassifier(predictions=True)
models, predictions = clf.fit(X_train, X_test, y_train, y_test)
```

```
100%|██████████| 29/29 [00:02<00:00, 12.66it/s]
```

Checking All Classification Model Accuracy in details:-

In [5]:

```
models
```

Out[5]:

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
Model					
BernoulliNB	0.98	0.98	0.98	0.98	0.01
PassiveAggressiveClassifier	0.98	0.98	0.98	0.98	0.01
SVC	0.98	0.98	0.98	0.98	0.02
Perceptron	0.97	0.97	0.97	0.97	0.01
AdaBoostClassifier	0.97	0.97	0.97	0.97	0.17
LogisticRegression	0.97	0.97	0.97	0.97	0.03
SGDClassifier	0.96	0.97	0.97	0.97	0.01
ExtraTreeClassifier	0.96	0.97	0.97	0.97	0.01
CalibratedClassifierCV	0.97	0.97	0.97	0.97	0.04
RandomForestClassifier	0.96	0.96	0.96	0.96	0.23
LGBMClassifier	0.96	0.96	0.96	0.96	0.75
GaussianNB	0.96	0.96	0.96	0.96	0.01
ExtraTreesClassifier	0.96	0.96	0.96	0.96	0.14
QuadraticDiscriminantAnalysis	0.96	0.96	0.96	0.96	0.01
LinearSVC	0.96	0.96	0.96	0.96	0.02
BaggingClassifier	0.96	0.95	0.95	0.96	0.07
XGBClassifier	0.96	0.95	0.95	0.96	0.40
LinearDiscriminantAnalysis	0.96	0.95	0.95	0.96	0.02
NearestCentroid	0.96	0.95	0.95	0.96	0.01
NuSVC	0.96	0.95	0.95	0.96	0.03
RidgeClassifier	0.96	0.95	0.95	0.96	0.02
RidgeClassifierCV	0.96	0.95	0.95	0.96	0.03
KNeighborsClassifier	0.95	0.94	0.94	0.95	0.11
DecisionTreeClassifier	0.95	0.94	0.94	0.95	0.02
LabelSpreading	0.94	0.93	0.93	0.94	0.03
LabelPropagation	0.94	0.93	0.93	0.94	0.04
DummyClassifier	0.62	0.50	0.50	0.48	0.01

Checking All Classification Model Prediction in details:-

In [6]:

```
predictions
```

Out[6]:

	AdaBoostClassifier	BaggingClassifier	BernoulliNB	CalibratedClassifierCV	DecisionTreeClassifier	DummyClassifier	Extra
0	1	1	1	1	1	1	
1	0	0	0	0	0	1	
2	0	0	0	0	0	1	
3	1	1	1	1	1	1	
4	1	1	1	1	1	1	
...	...	...	...	...	...	...	
109	1	1	1	1	1	1	
110	0	0	0	0	0	1	
111	1	1	1	1	1	1	
112	1	1	1	1	1	1	
113	0	0	0	0	0	1	

114 rows × 27 columns

Regression Model task

In [8]:

```
# load data
boston = datasets.load_boston()
X, y = boston.data, boston.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)

# fit all models
reg = LazyRegressor(predictions=True)
models, predictions = reg.fit(X_train, X_test, y_train, y_test)
```

```
100%|██████████| 42/42 [00:07<00:00,  5.86it/s]
```

Checking All Regressor Model Accuracy in details:-

In [9]:

```
models
```

```
Out[9]:
```

	Adjusted R-Squared	R-Squared	RMSE	Time Taken
Model				
GradientBoostingRegressor	0.90	0.92	2.49	0.23
XGBRegressor	0.89	0.91	2.60	0.72
RandomForestRegressor	0.88	0.89	2.81	0.34
LGBMRegressor	0.87	0.89	2.83	0.43
ExtraTreesRegressor	0.85	0.87	3.12	0.20
BaggingRegressor	0.84	0.86	3.15	0.04
HistGradientBoostingRegressor	0.84	0.86	3.18	1.05
DecisionTreeRegressor	0.84	0.86	3.23	0.01
AdaBoostRegressor	0.81	0.84	3.45	0.10
PoissonRegressor	0.73	0.76	4.16	0.05
ExtraTreeRegressor	0.71	0.75	4.29	0.01
KNeighborsRegressor	0.68	0.72	4.54	0.01
RANSACRegressor	0.67	0.71	4.58	0.13
Lars	0.62	0.67	4.93	0.02
LassoLarsCV	0.62	0.67	4.93	0.02
TransformedTargetRegressor	0.62	0.67	4.93	0.01
LinearRegression	0.62	0.67	4.93	0.01
Ridge	0.62	0.67	4.93	0.01
LassoCV	0.62	0.67	4.93	0.07
ElasticNetCV	0.62	0.67	4.94	0.07
BayesianRidge	0.62	0.67	4.94	0.01
LassoLarsIC	0.62	0.67	4.95	0.01
RidgeCV	0.62	0.67	4.95	0.01
LarsCV	0.62	0.66	4.96	0.03
SGDRegressor	0.61	0.66	4.99	0.01
SVR	0.60	0.65	5.07	0.02
GammaRegressor	0.59	0.64	5.12	0.01
NuSVR	0.58	0.63	5.17	0.02
MLPRegressor	0.57	0.63	5.21	0.41
Lasso	0.57	0.62	5.25	0.01
HuberRegressor	0.56	0.61	5.32	0.02
ElasticNet	0.56	0.61	5.33	0.01
TweedieRegressor	0.56	0.61	5.33	0.02
OrthogonalMatchingPursuitCV	0.55	0.61	5.37	0.01
LinearSVR	0.53	0.59	5.49	0.01
OrthogonalMatchingPursuit	0.48	0.54	5.79	0.01
GaussianProcessRegressor	0.23	0.33	7.00	0.08
QuantileRegressor	-0.15	-0.00	8.56	2.76
DummyRegressor	-0.17	-0.02	8.66	0.01
LassoLars	-0.17	-0.02	8.66	0.01
PassiveAggressiveRegressor	-0.33	-0.15	9.20	0.01
KernelRidge	-7.68	-6.57	23.56	0.15

Checking All Regressor Model Prediction in details:-

In [12]:

```
predictions
```

Out[12]:

	AdaBoostRegressor	BaggingRegressor	BayesianRidge	DecisionTreeRegressor	DummyRegressor	ElasticNet	ElasticNetC
0	25.59	22.71	28.86	28.10	22.80	26.02	28.
1	30.80	30.38	35.75	33.10	22.80	30.48	35.
2	17.73	18.14	15.38	17.30	22.80	17.85	15.
3	26.26	22.81	24.97	22.00	22.80	24.45	24.
4	17.39	16.12	18.74	23.20	22.80	19.08	18.
...	...	...	...	...	...	...	
97	15.04	13.19	-0.20	16.30	22.80	4.64	-0.
98	15.69	12.70	13.79	11.70	22.80	16.58	13.
99	17.31	13.89	16.07	16.30	22.80	16.57	16.
100	21.87	20.13	22.28	19.00	22.80	22.47	22.
101	24.54	24.12	24.52	26.40	22.80	24.05	24.

102 rows × 42 columns

Thank You