

Lazy Predict

This library offers you the possibility to evaluate many machine learning models at the same time, using `sk-learn` and saving a lot of time and coding.

CLASSIFICATION MODELS

```
# Installation
pip install lazypredict
```

```
# Imports
from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn import datasets
```

```
# Load our toy dataset
df = datasets.load_breast_cancer()
type(df)
[Output]: sklearn.utils.Bunch
```

```
# Transform to Pandas dataframe if you want to explore the data
df2 = pd.DataFrame( df.data , columns=df.feature_names)
df2['target'] = df.target
df2.head()
```

perimeter error	area error	smoothness error	compactness error	concavity error	concave points error	symmetry error	fractal dimension error	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	target
8.59	153.40	0.01	0.05	0.05	0.02	0.03	0.01	25.38	17.33	184.60	2019.00	0.16	0.67	0.71	0.27	0.46	0.12	0
3.40	74.08	0.01	0.01	0.02	0.01	0.01	0.00	24.99	23.41	158.80	1956.00	0.12	0.19	0.24	0.19	0.28	0.09	0
4.58	94.03	0.01	0.04	0.04	0.02	0.02	0.00	23.57	25.53	152.50	1709.00	0.14	0.42	0.45	0.24	0.36	0.09	0
3.44	27.23	0.01	0.07	0.06	0.02	0.06	0.01	14.91	26.50	98.87	567.70	0.21	0.87	0.69	0.26	0.66	0.17	0
5.44	94.44	0.01	0.02	0.06	0.02	0.02	0.01	22.54	16.67	152.20	1575.00	0.14	0.20	0.40	0.16	0.24	0.08	0

```
# X and Y
X, y = df.data, df.target
```

[illegible]

Here is where the magic happens:

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

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
Model					
LinearSVC	0.97	0.97	0.97	0.97	0.02
LogisticRegression	0.97	0.97	0.97	0.97	0.04
Perceptron	0.97	0.97	0.97	0.97	0.03
SGDClassifier	0.96	0.96	0.96	0.96	0.03
PassiveAggressiveClassifier	0.96	0.96	0.96	0.96	0.02
RidgeClassifierCV	0.96	0.95	0.95	0.96	0.04
RidgeClassifier	0.96	0.95	0.95	0.96	0.02
ExtraTreesClassifier	0.96	0.95	0.95	0.96	0.17
KNeighborsClassifier	0.96	0.95	0.95	0.96	0.03
XGBClassifier	0.95	0.94	0.94	0.95	0.09
QuadraticDiscriminantAnalysis	0.95	0.94	0.94	0.95	0.02
SVC	0.95	0.94	0.94	0.95	0.04
LGBMClassifier	0.95	0.94	0.94	0.95	0.16
LinearDiscriminantAnalysis	0.95	0.94	0.94	0.95	0.03
AdaBoostClassifier	0.94	0.93	0.93	0.94	0.19
NuSVC	0.94	0.93	0.93	0.94	0.05
CalibratedClassifierCV	0.94	0.93	0.93	0.94	0.06
BernoulliNB	0.93	0.93	0.93	0.93	0.02
LabelSpreading	0.93	0.92	0.92	0.93	0.04
LabelPropagation	0.93	0.92	0.92	0.93	0.03
RandomForestClassifier	0.93	0.92	0.92	0.93	0.27
NearestCentroid	0.93	0.92	0.92	0.93	0.02
BaggingClassifier	0.92	0.91	0.91	0.92	0.09
GaussianNB	0.92	0.91	0.91	0.92	0.01
ExtraTreeClassifier	0.91	0.91	0.91	0.91	0.02
DecisionTreeClassifier	0.91	0.91	0.91	0.91	0.02
DummyClassifier	0.54	0.52	0.52	0.53	0.02

REGRESSION MODELS

```
from lazypredict.Supervised import LazyRegressor
```

```
# Load toy dataset
boston = datasets.load_boston()
```

```
# X and y split
X, y = boston.data, boston.target
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=.2,
                                                    random_state=12)
```

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

	Adjusted R-Squared	R-Squared	RMSE	Time Taken
Model				
ExtraTreesRegressor	0.88	0.90	2.87	0.26
RandomForestRegressor	0.86	0.88	3.19	0.44
XGBRegressor	0.86	0.87	3.20	0.10
LGBMRegressor	0.85	0.87	3.22	0.08
GradientBoostingRegressor	0.84	0.86	3.34	0.22
BaggingRegressor	0.84	0.86	3.38	0.08
HistGradientBoostingRegressor	0.83	0.85	3.46	0.32
DecisionTreeRegressor	0.80	0.83	3.74	0.03
AdaBoostRegressor	0.79	0.82	3.83	0.17
PoissonRegressor	0.78	0.81	3.92	0.02
ExtraTreeRegressor	0.78	0.81	3.95	0.02
LassoLarsIC	0.71	0.75	4.52	0.04
BayesianRidge	0.71	0.75	4.53	0.02
ElasticNetCV	0.71	0.75	4.53	0.10
LassoCV	0.71	0.75	4.53	0.13
Ridge	0.71	0.75	4.53	0.02
RidgeCV	0.71	0.75	4.53	0.02
SGDRegressor	0.71	0.75	4.53	0.02
HuberRegressor	0.71	0.75	4.53	0.03
LassoLarsCV	0.71	0.75	4.53	0.05
TransformedTargetRegressor	0.71	0.75	4.53	0.01
LinearRegression	0.71	0.75	4.53	0.03
LarsCV	0.70	0.74	4.62	0.08
Lars	0.70	0.74	4.62	0.02