

```
1 !pip install lazypredict

1 ### importing lazypredict library
2 import lazypredict
3 ### importing LazyClassifier for classification problem
4 from lazypredict.Supervised import LazyClassifier
5 ### importing LazyClassifier for classification problem because here we are solving Classification use case.
6 from lazypredict.Supervised import LazyClassifier
7 ### importing breast Cancer Dataset from sklearn
8 from sklearn.datasets import load_breast_cancer
9 ### splitting dataset into training and testing part
10 from sklearn.model_selection import train_test_split

1 ### storing dataset in data variable
2 data = load_breast_cancer()

1 ### separating dataset into dependent and independent features
2 X = data.data
3 y = data.target

1 ### splitting dataset into training and testing part(50% training and 50% testing)
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.5, random_state =123)
```

```
1 clf = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric = None)
```

```
1 ### fitting data in LazyClassifier
2 models,predictions = clf.fit(X_train, X_test, y_train, y_test)
3 ### lets check which model did better on Breast Cancer Dataset
4 from pandas import DataFrame
5 import pandas as pd
6 data=pd.DataFrame(models)
7 data
```

100%|████████████████████| 29/29 [00:01<00:00, 21.52it/s]

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
Model					
LinearSVC	0.99	0.99	0.99	0.99	0.02
Perceptron	0.99	0.98	0.98	0.99	0.01
LogisticRegression	0.99	0.98	0.98	0.99	0.04
SVC	0.98	0.98	0.98	0.98	0.03
XGBClassifier	0.98	0.98	0.98	0.98	0.07
LabelPropagation	0.98	0.97	0.97	0.98	0.03
LabelSpreading	0.98	0.97	0.97	0.98	0.03
BaggingClassifier	0.97	0.97	0.97	0.97	0.09
PassiveAggressiveClassifier	0.98	0.97	0.97	0.98	0.02
SGDClassifier	0.98	0.97	0.97	0.98	0.02
RandomForestClassifier	0.97	0.97	0.97	0.97	0.20
CalibratedClassifierCV	0.98	0.97	0.97	0.98	0.06
QuadraticDiscriminantAnalysis	0.96	0.97	0.97	0.97	0.02
ExtraTreesClassifier	0.97	0.96	0.96	0.97	0.14
RidgeClassifierCV	0.97	0.96	0.96	0.97	0.02
LGBMClassifier	0.96	0.96	0.96	0.96	0.09
RidgeClassifier	0.97	0.96	0.96	0.97	0.02
AdaBoostClassifier	0.96	0.96	0.96	0.96	0.19
KNeighborsClassifier	0.96	0.96	0.96	0.96	0.04
BernoulliNB	0.95	0.95	0.95	0.95	0.03
LinearDiscriminantAnalysis	0.96	0.95	0.95	0.96	0.02
GaussianNB	0.95	0.95	0.95	0.95	0.02
NuSVC	0.95	0.94	0.94	0.95	0.03
ExtraTreeClassifier	0.94	0.93	0.93	0.94	0.02
NearestCentroid	0.95	0.93	0.93	0.95	0.02
DecisionTreeClassifier	0.93	0.93	0.93	0.93	0.02
DummyClassifier	0.53	0.50	0.50	0.53	0.01

```
1 ### Importing LazyRegressor
2 from lazypredict.Supervised import LazyRegressor
3 ### Importing dataset available in sklearn
4 from sklearn import datasets
5 from sklearn.utils import shuffle
```

```
import numpy as np
```

```
1  ### storing the Boston dataset in variable
2  boston = datasets.load_boston()
```

```
1  ### loading and shuffling the dataset
2  X, y = shuffle(boston.data, boston.target, random_state=13)
3  offset = int(X.shape[0] * 0.9)
```

```
1  ### splitting dataset into training and testing part.
2  X_train, y_train = X[:offset], y[:offset]
3  X_test, y_test = X[offset:], y[offset:]
```

```
1  ### fitting data in LazyRegressor because here we are solving Regression use case.
2  reg = LazyRegressor(verbose=0, ignore_warnings=False, custom_metric=None)
```

```
1  ### fitting data in LazyClassifier
2  models, predictions = reg.fit(X_train, X_test, y_train, y_test)
3  ### lets check which model did better on Breast Cancer Dataset
```

```
100%|██████████| 42/42 [00:04<00:00, 9.99it/s]
```

```
1  from pandas import DataFrame
2  import pandas as pd
3  data=pd.DataFrame(models)
4  data
```

	Adjusted R-Squared	R-Squared	RMSE	Time Taken
Model				
SVR	0.83	0.88	2.62	0.03
BaggingRegressor	0.83	0.88	2.63	0.07
NuSVR	0.82	0.86	2.76	0.04
RandomForestRegressor	0.81	0.86	2.79	0.42
XGBRegressor	0.81	0.86	2.79	0.09
GradientBoostingRegressor	0.81	0.86	2.84	0.20
ExtraTreesRegressor	0.79	0.84	2.98	0.33
HistGradientBoostingRegressor	0.77	0.83	3.06	0.41
AdaBoostRegressor	0.77	0.83	3.06	0.15
PoissonRegressor	0.77	0.83	3.11	0.02
LGBMRegressor	0.77	0.83	3.11	0.09
KNeighborsRegressor	0.77	0.83	3.12	0.03
DecisionTreeRegressor	0.65	0.74	3.79	0.02
MLPRegressor	0.65	0.74	3.80	1.20
HuberRegressor	0.64	0.74	3.84	0.03
GammaRegressor	0.64	0.73	3.88	0.05
LinearSVR	0.62	0.72	3.96	0.02

✓ 1s completed at 1:13 PM

