

# crop

June 14, 2024

```
[9]: import numpy as np
import pandas as pd

import lazypredict
from lazypredict.Supervised import LazyClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
```

```
[10]: data = pd.read_csv("Crop_Recommendation.csv")
```

```
[11]: data.shape
```

```
[11]: (2200, 8)
```

```
[12]: data.head()
```

```
[12]:
```

	Nitrogen	Phosphorus	Potassium	Temperature	Humidity	pH_Value	Rainfall	\
0	90	42	43	20.88	82.00	6.50	202.94	
1	85	58	41	21.77	80.32	7.04	226.66	
2	60	55	44	23.00	82.32	7.84	263.96	
3	74	35	40	26.49	80.16	6.98	242.86	
4	78	42	42	20.13	81.60	7.63	262.72	

```

Crop
0 Rice
1 Rice
2 Rice
3 Rice
4 Rice
```

```
[13]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Nitrogen        2200 non-null  int64
```

```

1   Phosphorus    2200 non-null    int64
2   Potassium     2200 non-null    int64
3   Temperature   2200 non-null    float64
4   Humidity      2200 non-null    float64
5   pH_Value      2200 non-null    float64
6   Rainfall      2200 non-null    float64
7   Crop          2200 non-null    object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB

```

```
[14]: data.describe().T
```

```

[14]:          count    mean  std   min  25%   50%   75%   max
Nitrogen    2200.00   50.55  36.92   0.00  21.00  37.00   84.25  140.00
Phosphorus  2200.00   53.36  32.99   5.00  28.00  51.00   68.00  145.00
Potassium   2200.00   48.15  50.65   5.00  20.00  32.00   49.00  205.00
Temperature 2200.00   25.62   5.06   8.83  22.77  25.60   28.56   43.68
Humidity    2200.00   71.48  22.26  14.26  60.26  80.47   89.95   99.98
pH_Value    2200.00    6.47   0.77   3.50   5.97   6.43    6.92    9.94
Rainfall    2200.00  103.46  54.96  20.21  64.55  94.87  124.27  298.56

```

```

[15]: X = data.drop('Crop', axis=1)
      y = data['Crop']

```

```

[16]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳ random_state=5)

```

```

[17]: # Defines and builds the lazyclassifier
      clf = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric=None)
      models_train, predictions_train = clf.fit(X_train, X_train, y_train, y_train)
      models_test, predictions_test = clf.fit(X_train, X_test, y_train, y_test)

```

```

87%|          | 27/31 [00:04<00:00, 5.15it/s] File
"/home/user/workspace/.venv/lib/python3.11/site-
packages/joblib/externals/loky/backend/context.py", line 250, in
_count_physical_cores

```

```

    cpu_info = subprocess.run(
        ~~~~~

```

```

File "/usr/lib/python3.11/subprocess.py", line 548, in run
    with Popen(*popenargs, **kwargs) as process:
        ~~~~~

```

```

File "/usr/lib/python3.11/subprocess.py", line 1026, in __init__

```

```

    self._execute_child(args, executable, preexec_fn, close_fds,

```

```

File "/usr/lib/python3.11/subprocess.py", line 1953, in _execute_child
    raise child_exception_type(errno_num, err_msg, err_filename)

```

```

[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000209 seconds.

```

You can set ``force_col_wise=true`` to remove the overhead.

```
[LightGBM] [Info] Total Bins 1336
```

```
[LightGBM] [Info] Number of data points in the train set: 1760, number of used
features: 7
```

```
[LightGBM] [Info] Start training from score -3.142336
```

```
[LightGBM] [Info] Start training from score -3.030418
```

```
[LightGBM] [Info] Start training from score -3.116360
```

```
[LightGBM] [Info] Start training from score -3.078620
```

```
[LightGBM] [Info] Start training from score -3.091042
```

```
[LightGBM] [Info] Start training from score -3.155581
```

```
[LightGBM] [Info] Start training from score -3.007161
```

```
[LightGBM] [Info] Start training from score -3.054228
```

```
[LightGBM] [Info] Start training from score -3.030418
```

```
[LightGBM] [Info] Start training from score -3.042252
```

```
[LightGBM] [Info] Start training from score -3.066350
```

```
[LightGBM] [Info] Start training from score -3.182610
```

```
[LightGBM] [Info] Start training from score -3.066350
```

```
[LightGBM] [Info] Start training from score -3.116360
```

```
[LightGBM] [Info] Start training from score -3.078620
```

```
[LightGBM] [Info] Start training from score -3.142336
```

```
[LightGBM] [Info] Start training from score -3.129264
```

```
[LightGBM] [Info] Start training from score -3.091042
```

```
[LightGBM] [Info] Start training from score -3.142336
```

```
[LightGBM] [Info] Start training from score -3.054228
```

```
[LightGBM] [Info] Start training from score -3.030418
```

```
[LightGBM] [Info] Start training from score -3.182610
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain. best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain. best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain. best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain. best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

```

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

100%|      | 31/31 [00:06<00:00,  4.60it/s]

'tuple' object has no attribute '__name__'
Invalid Classifier(s)

 87%|      | 27/31 [00:03<00:00,  8.59it/s]

[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000183 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 1336
[LightGBM] [Info] Number of data points in the train set: 1760, number of used
features: 7
[LightGBM] [Info] Start training from score -3.142336
[LightGBM] [Info] Start training from score -3.030418
[LightGBM] [Info] Start training from score -3.116360
[LightGBM] [Info] Start training from score -3.078620
[LightGBM] [Info] Start training from score -3.091042
[LightGBM] [Info] Start training from score -3.155581
[LightGBM] [Info] Start training from score -3.007161
[LightGBM] [Info] Start training from score -3.054228
[LightGBM] [Info] Start training from score -3.030418
[LightGBM] [Info] Start training from score -3.042252
[LightGBM] [Info] Start training from score -3.066350
[LightGBM] [Info] Start training from score -3.182610
[LightGBM] [Info] Start training from score -3.066350
[LightGBM] [Info] Start training from score -3.116360
[LightGBM] [Info] Start training from score -3.078620
[LightGBM] [Info] Start training from score -3.142336
[LightGBM] [Info] Start training from score -3.129264
[LightGBM] [Info] Start training from score -3.091042
[LightGBM] [Info] Start training from score -3.142336
[LightGBM] [Info] Start training from score -3.054228
[LightGBM] [Info] Start training from score -3.030418
[LightGBM] [Info] Start training from score -3.182610
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

```

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]





```

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
100%|      | 31/31 [00:05<00:00,  6.15it/s]

```

```
[18]: models_train
```

```

[18]:
Accuracy  Balanced Accuracy ROC AUC  F1 Score  \
Model
LGBMClassifier          1.00          1.00    None    1.00
RandomForestClassifier  1.00          1.00    None    1.00
DecisionTreeClassifier  1.00          1.00    None    1.00
ExtraTreeClassifier     1.00          1.00    None    1.00
ExtraTreesClassifier    1.00          1.00    None    1.00
LabelPropagation        1.00          1.00    None    1.00
LabelSpreading          1.00          1.00    None    1.00
BaggingClassifier       1.00          1.00    None    1.00
QuadraticDiscriminantAnalysis  1.00          1.00    None    1.00
GaussianNB             0.99          0.99    None    0.99
SVC                    0.99          0.99    None    0.99
KNeighborsClassifier    0.98          0.98    None    0.98
LogisticRegression     0.97          0.97    None    0.97
LinearDiscriminantAnalysis  0.97          0.97    None    0.97
LinearSVC              0.97          0.97    None    0.96
CalibratedClassifierCV  0.96          0.97    None    0.96
NuSVC                  0.96          0.96    None    0.96
SGDClassifier           0.94          0.93    None    0.94
PassiveAggressiveClassifier  0.88          0.88    None    0.88
NearestCentroid        0.88          0.88    None    0.87
Perceptron             0.84          0.84    None    0.83
BernoulliNB            0.78          0.77    None    0.77
RidgeClassifier         0.69          0.68    None    0.62

```

RidgeClassifierCV	0.69	0.68	None	0.62
AdaBoostClassifier	0.23	0.23	None	0.18
DummyClassifier	0.05	0.05	None	0.00

Model	Time Taken
LGBMClassifier	1.71
RandomForestClassifier	0.49
DecisionTreeClassifier	0.07
ExtraTreeClassifier	0.04
ExtraTreesClassifier	0.31
LabelPropagation	0.33
LabelSpreading	0.55
BaggingClassifier	0.14
QuadraticDiscriminantAnalysis	0.11
GaussianNB	0.04
SVC	0.24
KNeighborsClassifier	0.13
LogisticRegression	0.18
LinearDiscriminantAnalysis	0.12
LinearSVC	0.14
CalibratedClassifierCV	0.40
NuSVC	0.49
SGDClassifier	0.13
PassiveAggressiveClassifier	0.09
NearestCentroid	0.07
Perceptron	0.11
BernoulliNB	0.06
RidgeClassifier	0.07
RidgeClassifierCV	0.17
AdaBoostClassifier	0.46
DummyClassifier	0.03

```
[19]: models_test
```

```
[19]:
```

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	\
Model					
RandomForestClassifier	1.00	1.00	None	1.00	
QuadraticDiscriminantAnalysis	1.00	1.00	None	1.00	
ExtraTreesClassifier	1.00	1.00	None	1.00	
GaussianNB	1.00	1.00	None	1.00	
LGBMClassifier	1.00	1.00	None	1.00	
BaggingClassifier	0.99	0.99	None	0.99	
SVC	0.98	0.98	None	0.98	
LabelPropagation	0.98	0.98	None	0.98	
LabelSpreading	0.98	0.98	None	0.98	
DecisionTreeClassifier	0.98	0.98	None	0.98	



KNeighborsClassifier	0.97	0.97	None	0.97
LinearDiscriminantAnalysis	0.97	0.97	None	0.97
NuSVC	0.96	0.97	None	0.96
LogisticRegression	0.97	0.97	None	0.97
LinearSVC	0.96	0.96	None	0.96
CalibratedClassifierCV	0.96	0.96	None	0.96
SGDClassifier	0.92	0.93	None	0.92
ExtraTreeClassifier	0.89	0.90	None	0.89
PassiveAggressiveClassifier	0.86	0.86	None	0.86
NearestCentroid	0.85	0.85	None	0.83
Perceptron	0.81	0.82	None	0.80
BernoulliNB	0.76	0.77	None	0.74
RidgeClassifierCV	0.64	0.68	None	0.55
RidgeClassifier	0.63	0.68	None	0.55
AdaBoostClassifier	0.20	0.23	None	0.16
DummyClassifier	0.03	0.05	None	0.00

Model	Time Taken
RandomForestClassifier	0.42
QuadraticDiscriminantAnalysis	0.04
ExtraTreesClassifier	0.23
GaussianNB	0.02
LGBMClassifier	1.55
BaggingClassifier	0.11
SVC	0.09
LabelPropagation	0.20
LabelSpreading	0.43
DecisionTreeClassifier	0.03
KNeighborsClassifier	0.04
LinearDiscriminantAnalysis	0.11
NuSVC	0.28
LogisticRegression	0.15
LinearSVC	0.06
CalibratedClassifierCV	0.37
SGDClassifier	0.12
ExtraTreeClassifier	0.02
PassiveAggressiveClassifier	0.06
NearestCentroid	0.04
Perceptron	0.05
BernoulliNB	0.03
RidgeClassifierCV	0.07
RidgeClassifier	0.04
AdaBoostClassifier	0.42
DummyClassifier	0.02

```
[20]: rf = RandomForestClassifier()

rf.fit(X_train, y_train)

rf_score = rf.score(X_test, y_test)

rf_score
```

```
[20]: 0.9977272727272727
```

```
[21]: input = ((4 , 8.970 , 9.61 , 15.75 ,0.73 , 9.52 , 2.66 ))

input_array = np.asarray(input)

input_array_reshape = input_array.reshape(1, -1)

predictions = rf.predict(input_array_reshape)
print (predictions)

print("predictions are: ", predictions[0])
```

```
['KidneyBeans']
predictions are:  KidneyBeans
```

```
[ ]:
```