

DESCRIPTION

Problem Statement: Build a tic-tac-toe game classification algorithm using the concept of supervised machine learning.

Requirements:

- Python 3.6
- Scikit-Learn
- Pandas and Numpy

Dataset Used: tic-tac-toe.txt

Attribute Description: Name|Type|Description

top_left_square | string | Value includes x,o or b for blank
top_middle_square | string | Value includes x,o or b for blank
top_right_square | string | Value includes x,o or b for blank
middle_left_square | string | Value includes x,o or b for blank
middle_middle_square | string | Value includes x,o or b for blank
middle_right_square | string | Value includes x,o or b for blank
bottom_left_square | string | Value includes x,o or b for blank
bottom_middle_square | string | Value includes x,o or b for blank
bottom_right_square | string | Value includes x,o or b for blank
class | string | Predictor class: Values can be positive (X won) or negative (X lost or tied)

Dataset Description:

This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row").

Training dataset:

This dataset will be used to test the developer's solution. It will be available at

```
/data/train/tic-tac-toe.data.txt
```

Tasks to be performed:

1. Data Preprocessing:

Use `random_state = 3` while splitting the dataset into train and test set.

```
Label Val | Decoded Val (features)
| 0 | b
| 1 | o
| 2 | x
```

```
Label Val | Decoded Val (class)
```

```
0 | negative
```

```
1 | positive
```

Hint: Use the concept of label encoding i.e. map the parameters manually.

2. Create a Random Forest Model (random_state = 0) using k- Cross-Validation Technique.

3. Apply Ada Boost algorithm to improve the accuracy score (random_state = 0).

Hint: For the above scenario, you can choose the best value of k (from 2 to 10) for Cross-Validation and use n_estimators = 100, n_splits=20 (You need to understand which parameter to use and when).

Print the accuracy score before and after implementing Ada Boost Algorithm.

Output Format:

- Perform the above operations and write your output to a file named **output.csv**, which should be present at the location **/code/output/output.csv**
- **output.csv** should contain the answer to **each question** on consecutive rows.

NOTE: If accuracy before implementing ada boost is 0.713 and after implementing is 0.811 then create a list result = [0.713, 0.811] and convert it to a CSV file(The process of which is mentioned in the stub).

DATASETS

- [Training dataset](#)