A PRACTICAL IMPLEMENTATION OF THE SEMMA DATA Mining Methodology Using PyCARET

Rutik Darda

September 23, 2023

Abstract

Abstract: In the ever-evolving realm of data analytics, the SEMMA (Sample, Explore, Modify, Model, and Assess) methodology stands out as a systematic and structured approach to data mining. This paper delves deep into each phase of the SEMMA methodology, providing readers with a comprehensive understanding of its intricacies. By leveraging the powerful capabilities of the PyCaret library, we elucidate the practical aspects of implementing SEMMA, from sampling data to assessing model performance. Through this hands-on approach, we aim to bridge the gap between theoretical understanding and real-world application, emphasizing the methodology's significance in contemporary data analysis.

1 Keywords

SEMMA (Sample, Explore, Modify, Model, and Assess) methodology. We'll use the PyCaret library to walk you through the process step by step.

2 SEMMA Data Mining Steps:

1. Sample: Obtain a representative subset of your data.
2. Explore: Investigate the properties and quality of the sampled data.
3. Modify: Prepare the data for modeling.
4. Model: Build predictive models.
5. Assess: Evaluate the models' performance.

Step 1: Sample For this step, we'll first load the data and take a quick look at it to understand its structure.

The dataset consists of the following columns:

1. MemberID: An identifier for the member.
2. Age: Age of the member.
3. Gender: Gender of the member (Male or Female).
4. Height_cm: Height of the member in centimeters.
5. Weight_kg: Weight of the member in kilograms.
6. ActivityLevel: The level of physical activity (e.g., Sedentary, Light, Moderate).
7. CaloriesBurned: The number of calories burned by the member.

3 Step 2: Explore

1. To explore the dataset with PyCaret, you would use the setup() function. This function automatically profiles the dataset and prepares it for modeling. It identifies the data types, checks for missing values, and performs many other preprocessing tasks.

python Copy code

```python
import pandas as pd

# Load the dataset
data = pd.read_csv('/mnt/data/health_fitness_data.csv')

# Display the first few rows of the data
data.head()
```

RESULT

	MemberID	Age	Gender	Height_cm	Weight_kg	ActivityLevel	CaloriesBurn
0	1	37	Female	160	66	Moderate	5
1	2	49	Male	181	83	Moderate	4
2	3	30	Male	154	64	Sedentary	2
3	4	45	Female	186	82	Sedentary	7
4	5	26	Female	150	87	Light	3

python Copy code

```python
from pycaret.classification import setup
exp_clf = setup(data=data, target='ActivityLevel', session_id=123)
```

4 Step 3: Modify

1. PyCaret provides a variety of preprocessing functions such as removing outliers, imputing missing values, and creating transformations. Most of these are automatically handled in the setup() function. If additional modifications are needed, they can be specified as arguments within the setup() function or using other functions like create_model(), tune_model(), etc.

5 Step 4: Model

1. PyCaret offers a simplified workflow for creating and comparing multiple models. The compare_models() function creates multiple models and scores them using cross-validation to help you pick the best one.

python Copy code

```python
from pycaret.classification import compare_models
best_model = compare_models()
```

This will rank various algorithms based on a variety of metrics (like Accuracy, AUC, Recall, Precision, etc.), and the best model (based on a chosen metric) will be assigned to the best_model variable.

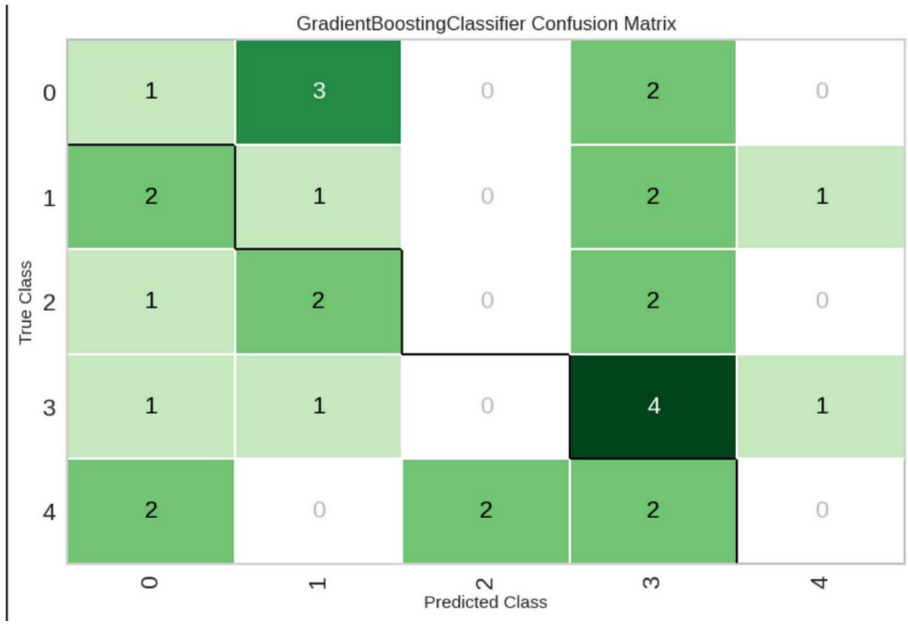
You can also create specific models, tune them, ensemble them, or stack them using functions like create_model(), tune_model(), ensemble_model(), and stack_models().

6 Step 5: Assess

1. Once you've selected a model, you can assess its performance on the validation set using various plots and metrics. For instance:

python Copy code

```python
from pycaret.classification import plot_model
plot_model(best_model, plot='confusion_matrix')
```



This would display the confusion matrix for the model. Other available plots include 'auc', 'feature', 'boundary', and many more.

This is how step by step SEMMA datamining methodology works!