Hope Artificial Intelligence

Classification Assignment using GridSearch

Dataset : CKD.csv

1. Identifying Problem statement:

   For the given dataset and the client's requirement the following stages are followed.

   Stage 1 : Domain selection: Machine Learning

   Stage 2 : Selecting the Learning algorithm: Supervised Learning algorithm

   Stage3 : In Supervised learning method this type comes under "Classification Learning method".

2. Information about the given dataset:

   The given dataset has 25 columns. But the column 'sg' split as 3 columns as sg_b, sg_c, sg_d and sg_e. So totally it has 28 columns

   In that Input column's are : 'age', 'bp', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hrmo', 'pcv', 'wc', 'rc', 'sg_b', 'sg_c', 'sg_d', 'sg_e', 'rbc_normal', 'pc_normal', 'pcc_present', 'ba_present', 'htn_yes', 'dm_yes', 'cad_yes', 'appet_yes', 'pe_yes', 'ane_yes', [totally 27]

   Output Column : 'classification_yes' [1 only]

3. Mention the pre-processing method if you're doing any (like converting string to number – nominal data

   There are 12 columns of categorical data. So these categorical data are converted to nominal data and the columns are:
   'sg_b', 'sg_c', 'sg_d', 'sg_e', 'rbc_normal', 'pc_normal', 'pcc_present', 'ba_present', 'htn_yes', 'dm_yes', 'cad_yes', 'appet_yes', 'pe_yes', 'ane_yes', 'classification_yes'

4. Developing the model using the following classification algorithms using Grid Search CV:

   - Logistic Regression method
   - Support Vector Machine Classification method
   - Decision Tree Classification method
   - Random Forest Classification method

5. Research Values of each algorithms:

1. Logistic Regression using GridSearch CV method:

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924946382275899
```

Classification report:

```
The report:
              precision    recall  f1-score   support

       False       0.98      1.00      0.99        51
        True       1.00      0.99      0.99        82

    accuracy                           0.99       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

Best score for this model:

ı_penalty	param_solver	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
l2	newton-cg	{'penalty': 'l2', 'solver': 'newton-cg'}	0.981569	0.981014	0.981217	1.000000	1.000000	0.988760	0.009179	1
l2	lbfgs	{'penalty': 'l2', 'solver': 'lbfgs'}	0.981569	0.981014	0.981217	1.000000	1.000000	0.988760	0.009179	1

2. SVM Classification algorithm using GridSearchCV:

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9924946382275899
```

Classification report:

```
The report:
              precision    recall  f1-score   support

       False       0.98      1.00      0.99        51
        True       1.00      0.99      0.99        82

    accuracy                           0.99       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

Best model for SVM Grid:

n_gamma	param_kernel	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
scale	sigmoid	{'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}	0.981569	1.000000	1.000000	0.981031	1.000000	0.992520	0.009163	1

3. Decision Tree Classification algorithm using GridSearch CV :

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'sqrt', 'splitter': 'random'}: 0.9400566944426594
```

Classification Report:

The report:

	precision	recall	f1-score	support
False	0.91	0.94	0.92	51
True	0.96	0.94	0.95	82
accuracy			0.94	133
macro avg	0.93	0.94	0.94	133
weighted avg	0.94	0.94	0.94	133

Best Model Grid:

ures	param_splitter	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
sqrt	random	{'criterion': 'entropy', 'max_features': 'sqrt...	0.945100	0.981014	1.000000	0.925524	0.981217	0.966571	0.027153	1

4. Random Forest Classification algorithm using GridSearch CV

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 100}: 0.9849624060150376
```

Classification Report:

The report:

	precision	recall	f1-score	support
False	0.98	0.98	0.98	51
True	0.99	0.99	0.99	82
accuracy			0.98	133
macro avg	0.98	0.98	0.98	133
weighted avg	0.98	0.98	0.98	133

Best Model using Grid:

param_n_estimators	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score

100	{'criterion': 'entropy', 'max_features': 'log2...	1.000000	0.971429	0.984615	0.985075	1.000000	0.988224	0.010792	1

6. Final Model for this dataset:

Both Logistic Regression and SVM classification algorithm using Grid are best model because its f1 macro average is 0.9924 and as well as accuracy is 0.99.