rf-classification

April 5, 2024

0.1 Predicting Bill Authenticity Using Random Forest

[2]:

```python
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
data = pd.read_csv(r'C:\Users\ntpc\Desktop\bill_authentication.csv')  # Replace 'path_to_your_dataset.csv' with the actual path to your dataset

# Split the data into features (X) and target variable (y)
X = data.drop('Class', axis=1)
y = data['Class']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize RandomForestClassifier
rf_classifier = RandomForestClassifier()

# Define hyperparameter grid for grid search
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Perform grid search with cross-validation (e.g., 5-fold cross-validation)
grid_search = GridSearchCV(rf_classifier, param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Get the best hyperparameters from grid search
best_params = grid_search.best_params_

# Initialize RandomForestClassifier with the best hyperparameters
```

```python
best_rf_classifier = RandomForestClassifier(**best_params)
best_rf_classifier.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = best_rf_classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f'Best Hyperparameters: {best_params}')
print(f'Accuracy: {accuracy}')
print('Classification Report:')
print(report)
```

```
Best Hyperparameters: {'max_depth': 20, 'min_samples_leaf': 2,
'min_samples_split': 2, 'n_estimators': 100}
Accuracy: 0.9890909090909091
Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       147
           1       1.00      0.98      0.99       128

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275
```