Introduction to Scikit-Learn

Group No - 23
Contribution: Report and code both are solely done by Madhav Jivani

October 25, 2024

Teammates

Teammates

Madhav Jivani (202201285)
Tarun Kumar (202412126)
Rupesh Singh (202412124)

GitHub Repository

`https://github.com/madhavJivani/EDA_ASSIGNMENT/tree/main/EDA_ASSIGNMENTS/Assignment-3`

1 Introduction to Scikit-Learn

Scikit-Learn is a powerful and user-friendly open-source library for machine learning in Python. It provides a consistent interface for various algorithms, including classification, regression, clustering, and dimensionality reduction. Its modular nature allows for easy integration with other libraries such as NumPy and Pandas, making it a cornerstone in the data science and machine learning communities.

2 Data Preprocessing Modules

2.1 Encoders

2.1.1 OneHotEncoder

The `OneHotEncoder` converts categorical variables into a format that can be provided to machine learning algorithms to improve predictions. It creates binary columns for each category of a categorical variable, allowing algorithms to understand and utilize categorical information without assigning arbitrary numerical values.

Use Cases:

- Converting nominal data such as colors (red, green, blue) into binary columns.
- Preparing data for tree-based algorithms that do not handle categorical variables inherently.

2.1.2 LabelEncoder

The `LabelEncoder` is used to convert categorical labels into integers. It assigns a unique integer to each category, which is particularly useful for target variables in classification tasks.

Use Cases:

- Converting categorical target variables into a numeric format.
- Simplifying categorical data when the algorithm can handle ordinal relationships.

2.1.3 Additional Encoders

- `OrdinalEncoder`: Converts categorical features to ordinal integers, preserving order.
- `BinaryEncoder`: Combines the properties of one-hot and label encoding to provide a compact representation.

2.2 Scalers

Scaling features is crucial in machine learning as it helps in improving convergence speed and model accuracy.

- **StandardScaler**: Standardizes features by removing the mean and scaling to unit variance. Useful for algorithms sensitive to the scale of input features (e.g., SVM, KNN).
- **MinMaxScaler**: Scales features to a given range, usually between 0 and 1. It is suitable when the distribution of data is not Gaussian.
- **RobustScaler**: Uses the median and interquartile range for scaling, making it robust to outliers. Ideal when dealing with datasets that contain outliers.

3 Model Selection and Data Splitting

The `train_test_split` function is essential for dividing datasets into training and testing sets, allowing the evaluation of model performance on unseen data. Proper splitting ensures that models generalize well, rather than simply memorizing training data.

4 Evaluation Metrics

4.1 Classification Report

The classification report summarizes the precision, recall, f1-score, and support for each class, providing a comprehensive evaluation of model performance.

4.2 Confusion Matrix

The confusion matrix visualizes the performance of a classification model by showing the actual versus predicted classifications. It helps in identifying misclassifications and understanding model behavior.

4.3 Accuracy Score

The accuracy score is the ratio of correctly predicted instances to the total instances. It provides a quick snapshot of overall model performance.

4.4 ROC AUC Score

The ROC AUC score measures the ability of the model to distinguish between classes. A higher score indicates better model performance, making it a valuable metric for binary classification tasks.

5 Feature Selection

Feature selection is critical for improving model performance and reducing overfitting. It involves selecting the most relevant features for the model, thus simplifying the model and enhancing interpretability.

5.1 SelectKBest and f_classif

`SelectKBest` selects features based on univariate statistical tests. It can use various scoring functions, such as `f_classif`, which is the ANOVA F-value between label/target for classification tasks. This method helps to identify the top k features that are most relevant to the target variable.

Purpose: To reduce the dimensionality of the dataset while preserving the most informative features.

5.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms features into a new set of uncorrelated variables called principal components. These components are ordered by the amount of variance they capture from the original dataset.

Purpose: PCA reduces the dimensionality of a dataset while retaining most of the variance, thereby simplifying models without significant loss of information.

Example: Given a dataset with features representing height, weight, and age, PCA could transform these features into two principal components that capture the majority of the variance, allowing for simpler models while still effectively capturing the underlying structure.

Advantages:

- Reduces noise and redundancy in the data, leading to improved model performance.
- Enhances visualization by reducing high-dimensional data to two or three dimensions.

- Helps mitigate overfitting by reducing the number of features.

5.3 Recursive Feature Elimination (RFE)

`RFE` recursively removes features and builds models until the specified number of features is reached. It helps in selecting the most significant features that contribute to the model's performance.

Purpose: To identify the most important features for the model through a systematic elimination process.

5.4 Random Forest and Extra Trees

Both `RandomForest` and `ExtraTrees` provide built-in feature importance measures. They evaluate the significance of each feature based on how much they improve the model's predictions.

Purpose: To automatically assess the importance of features, allowing for the selection of the most relevant ones for improved model performance.

6 Dataset Utilities

Scikit-Learn includes various utilities for accessing popular datasets.

- `load_iris`
- `load_wine`
- `load_digits`

7 Supervised Learning Models

7.1 Logistic Regression

Used for binary classification problems, it models the probability of a class label based on input features.

7.2 Linear Regression

A fundamental regression technique that models the relationship between a dependent variable and one or more independent variables using a linear equation.

7.3 L1 and L2 Regularization

These techniques are used in regression to prevent overfitting. L1 regularization (Lasso) adds a penalty equal to the absolute value of coefficients, while L2 regularization (Ridge) adds a penalty equal to the square of coefficients.

7.4 Support Vector Machine (SVM)

SVM is a powerful classification technique that finds the optimal hyperplane to separate different classes in the feature space.

7.5 Random Forest Classifier

An ensemble learning method that uses multiple decision trees to improve classification accuracy and control overfitting.

8 Dimensionality Reduction

8.1 PCA

PCA is a dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional space, retaining most of the variance in the data. It is often used for preprocessing data before applying machine learning algorithms.

9 Special Features and Unique Aspects of Scikit-Learn

Scikit-Learn stands out due to its model interpretability, user-friendly API, and robust support for cross-validation. It enables users to build complex machine learning workflows easily and efficiently.