Data Preprocessing Report

Sahil Karande

July 28, 2024

Project Title

Machine Learning Mastery: From Basics to Advanced

1 Concepts Used

1.1 Data Manipulation with Pandas

- `pd.read_csv("Data.csv")`: Reads data from a CSV file into a DataFrame.
- `df.head()`: Displays the first few rows of the DataFrame to preview the data.
- `df.dtypes`: Shows the data types of each column in the DataFrame.

1.2 Handling Missing Values

Concept: Missing values in the dataset must be addressed for accurate analysis and modeling.

- `SimpleImputer`: Replaces missing values with a specified strategy, such as the mean of the column.
  - `fit_transform()`: Fits the imputer on the data and replaces missing values with the mean.
  - **Example**: Missing values are filled with the mean of the respective column.

1.3 Encoding Categorical Data

Concept: Machine learning algorithms require numerical inputs. Categorical data must be converted into numerical format.

- `LabelEncoder`: Converts categorical labels into numerical values.
  - `fit_transform()`: Transforms categorical labels into numerical values.
  - **Example**: Converts 'Yes' to 1 and 'No' to 0.
- `OneHotEncoder`: Converts categorical features into binary vectors.
  - **Example**: Converts 'Mumbai', 'Nagpur', and 'Pune' into binary vectors.
- `ColumnTransformer`: Applies one-hot encoding to specified columns and leaves others unchanged.
  - **Example**: Applies one-hot encoding to the first column of the dataset.

1.4 Splitting the Dataset

Concept: Splitting the dataset into training and testing sets is crucial for evaluating model performance.

- `train_test_split(X, y, test_size=0.2, random_state=0)`: Divides data into training (80%) and testing (20%) sets, with a fixed random seed for reproducibility.

1.5 Feature Scaling

Concept: Standardizing features to have a mean of 0 and a standard deviation of 1 improves the performance of many machine learning algorithms.

- `StandardScaler`: Standardizes feature values by removing the mean and scaling to unit variance.
  - `fit_transform(X_train)`: Computes scaling parameters from the training set and applies the transformation.
  - `transform(X_test)`: Applies the same scaling to the test set.

2 Key Concepts and Tools to Remember

- **Data Manipulation**
  - `pd.read_csv()`
  - `df.head()`
  - `df.dtypes`
- **Handling Missing Values**
  - `SimpleImputer`
  - `fit_transform()`
  - `strategy='mean'`
- **Encoding Categorical Data**
  - `LabelEncoder`
  - `OneHotEncoder`
  - `ColumnTransformer`
- **Splitting the Dataset**
  - `train_test_split()`
  - `test_size`
  - `random_state`
- **Feature Scaling**
  - `StandardScaler`
  - `fit_transform()`
  - `transform()`