Autism Classification using Machine Learning

Introduction

Objective: Predict Autism Spectrum Disorder (ASD) classification using machine learning techniques, based on various features (e.g., gender, age, ethnicity).

Dataset: A CSV file (`autism_data.csv`) containing data related to autism diagnosis.

Data Exploration & Preprocessing

- **Initial Data Inspection**:
  - We load the data using `pandas` and inspect its first 5 rows with data.head().
  - We check the size of the dataset using `data.shape`, and identify missing values with `data.isna().sum()`.
  - We also check for duplicate records using `data.duplicated().sum()`.
- **Cleaning Data**:
  - We convert the "age" column to numeric values and replace any erroneous strings.
  - Missing values in the "age" column are filled with the median age value.
  - Categorical columns (e.g., gender, ethnicity) are encoded using `LabelEncoder` to convert them into numeric values.

Splitting Data into Features & Target Variables

- **Features (X)**: All columns except for the target variable (`Class/ASD`).
- **Target (y)**: The `Class/ASD` column, indicating whether an individual has Autism Spectrum Disorder.
- **Train-Test Split**:
- We split the dataset into training (70%) and testing (30%) sets using `train_test_split`.

Data Scaling

- **Standardization**:
  - The data is standardized using `StandardScaler` to ensure that all features have a similar scale, improving model performance.

Model Training

- **Random Forest Classifier**:
  - We train a `RandomForestClassifier`, which is an ensemble learning method using multiple decision trees for prediction.
  - The model is trained on the scaled training data (`x_train`) and evaluated on the test data (`x_test`).
  - The accuracy of the model is calculated using `accuracy_score`.
- **Support Vector Machine (SVM)**:
  - A linear `SVC` (Support Vector Classifier) is also trained with the same dataset to compare performance.
  - Model evaluation includes accuracy score and a detailed classification report using `classification_report`.

Cross-Validation

K-Fold Cross-Validation:

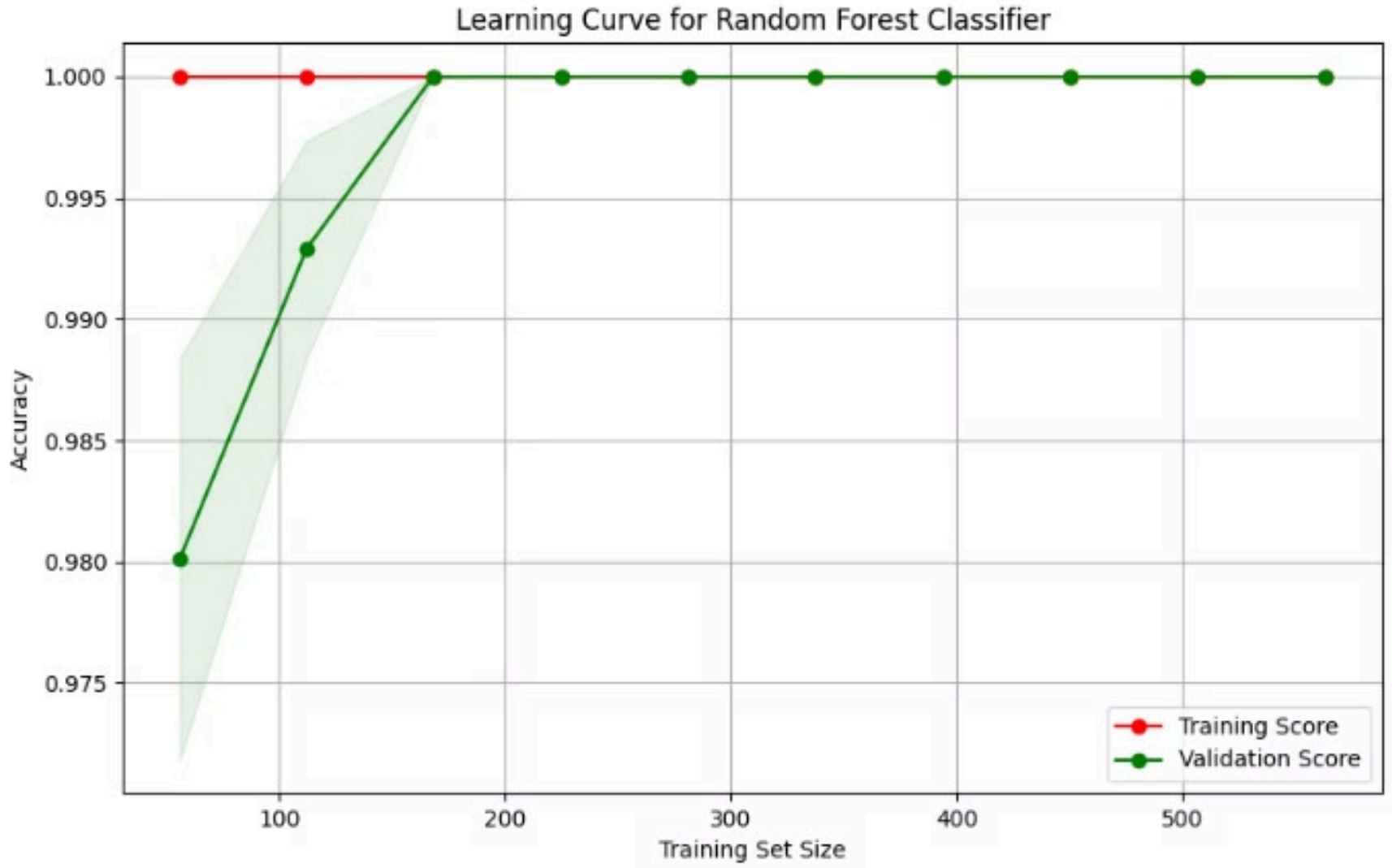
- We use 5-fold cross-validation to assess the model's performance more reliably.
- The mean and standard deviation of cross-validation scores are calculated and printed.

Learning Curve

- **Plotting the Learning Curve**:
  - The learning curve shows the model’s performance as the amount of training data increases.
  - Training and validation scores are plotted for different training set sizes, helping us understand model performance and potential overfitting.
- **Visualization**:
  - The learning curve is generated using `plot_learning_curve`, providing insights into the model's behavior as the training data grows.

Results

- **Random Forest Model**:
  - Accuracy of the Random Forest model: 100%.
- **SVM Model**:
  - Accuracy of the SVM model: 100.
  - Classification report details precision: 1.00, recall: 1.00, and F1-score for each class: 1.00.
- **Cross-Validation**:
  - Mean cross-validation accuracy: 100.00%.
  - Standard deviation of cross-validation scores: 0.0.
- **Learning Curve**:
  - The learning curve shows how the model's performance improves as more training data is used.



Conclusion

Key Insights:

- Random Forest and SVM models are evaluated and compared based on their accuracy, with cross-validation for reliability.
- The learning curve helps us identify whether the model benefits from additional data.