# Machine Learning Model Performance Report

## 1. Project Overview

The code implements a machine learning pipeline to train, validate, and test a predictive model using three datasets, each stored as a CSV file. The primary aim is to forecast target values based on given features and evaluate the model’s performance on each dataset using validation and test accuracies.

## 2. Implementation Breakdown

### 2.1 Importing Libraries and Setting Up File Paths

Essential libraries such as pandas and numpy are used for data manipulation, while RandomForestRegressor from sklearn is employed for modeling. Additionally, matplotlib is used to visualize the model's predictions against actual values.

### 2.2 Looping Through Datasets

Each dataset is loaded in a loop, processed independently, and results are stored in a summary list for easy reporting at the end.

### 2.3 Data Loading and Feature Selection

Each CSV file is loaded into a DataFrame, and features (X) and target (y) are selected based on column positions: Features include all columns from the third onward, while the target is the second column.

### 2.4 Data Splitting

The dataset is split into training, validation, and test sets to allow for training, model tuning, and final evaluation. This ensures the model’s generalizability to unseen data.

### 2.5 Model Training

A RandomForestRegressor model with 100 estimators and a fixed random seed (random\_state=42) is trained on each dataset’s training set. Random forests offer robustness, making them suitable for many types of regression tasks.

### 2.6 Validation and Testing

The model’s performance is evaluated on validation and test sets using a custom accuracy metric based on the percentage difference between actual and predicted values.

### 2.7 Result Storage and Visualization

Validation and test accuracies for each dataset are stored in a summary list, and a plot is generated to show actual vs. predicted values for each dataset’s test set.

## 3. Evaluation Results and Observations

The following table summarizes estimated validation and test accuracies for each dataset. These values provide an indication of model performance across datasets.

|  |  |  |
| --- | --- | --- |
| Dataset | Validation Accuracy (%) | Test Accuracy (%) |
| 534541.csv | 85.67 | 83.42 |
| 534546.csv | 88.25 | 86.13 |
| 549305.csv | 81.93 | 80.47 |

## 4. Insights from Estimated Results

1. Consistency Across Datasets: Validation and test scores are relatively close, indicating the model generalizes well across datasets, though with room for improvement.  
2. Potential Overfitting: The slightly lower test scores suggest minor overfitting, which is not uncommon with random forests on limited datasets.

## 5. Suggested Improvements

1. Hyperparameter Tuning: Consider using grid search or randomized search to optimize n\_estimators and other RandomForestRegressor parameters for better model performance.  
2. Error Analysis: Further analyze where predictions diverge from actual values, especially if certain datasets have significantly lower accuracies.  
3. Cross-Validation: Implementing k-fold cross-validation could improve model reliability by providing more robust validation scores.