UNIVERSITAT ROVIRA I VIRGILI

**Neural Network with Back-Propagation Report**

Researcher: Kamila Ten

Date: 18/11/2023

## Description of Implementation

This implementation leverages two different models for predicting turbine performance: a Neural Network (BP) and a Linear Regression model (MLR). The code is implemented in Python, utilizing key libraries such as pandas, numpy, scikit-learn, and matplotlib.

For datasets 1 and 2, as they are already cleaned, no preprocessing is required. However, to ensure consistent scales and facilitate model convergence during training, data normalization is applied to both input and output variables. This is achieved using techniques such as Min-Max scaling or Z-score normalization. Normalization prevents certain features from dominating others, especially in scenarios where input variables have different ranges.

## Execution Instructuions

Ensure that the required libraries (pandas, numpy, scikit-learn, and matplotlib) are installed. You can run the code in a Python environment.

- Handling Missing Values: Checked for missing values and applied appropriate techniques such as imputation or removal, depending on the extent of missing data.
- Categorical Value Representation: Ensured proper representation of categorical values using methods like one-hot encoding or label encoding.
- Outlier Detection: Examined the dataset for outliers and implemented necessary strategies like removal or transformation.
- Data Normalization: Applied data normalization to input and output variables if needed. This ensures that numerical values are on a similar scale, aiding the training process.

## Description and Link to the Selected Dataset

The dataset used for this project is sourced from Kaggle: Fitbit Fitness Tracker Dataset.

Synthetic Dataset (A1-synthetic)

- Data Normalization:
- Applied Min-Max Scaling to input and output variables.
- Ensures consistent scales for effective model training.

Turbine Dataset (A1-turbine)

- Data Loading and Cleaning:
- Loaded turbine dataset and addressed formatting issues.
- Cleaned data, replacing missing values and handling outliers.
- Data Normalization:
- Ensured consistent scales for input and output variables.

New Dataset(dailyActivity)

- Cleaned data, replacing missing values and handling outliers.
- Data Normalization:
- Ensured consistent scales for input and output variables.

## Implementation Decisions

Neural Network (BP):

- Architecture: A simple neural network with an input layer, a hidden layer (adjustable size, set to 4), and an output layer.
- Activation Function: Sigmoid function is used in the hidden layer.
- Loss Function: Mean Absolute Percentage Error (MAPE) serves as the evaluation metric.
- Training: The neural network is trained with 1000 epochs and a learning rate of 0.01.

Linear Regression (MLR):

- Linear Regression is utilized as a benchmark model.
- Evaluation Metric: Mean Absolute Error (MAE) is used for MLR.
- A scatter plot is generated for visualization.

Backpropagation Neural Network (BP)

1)Architecture:

Input layer, hidden layer (4 neurons), output layer.

Sigmoid activation in the hidden layer.

Mean Absolute Percentage Error (MAPE) used as the loss function.

2)Training:

Trained for 1000 epochs with a learning rate of 0.01.

Linear Regression (MLR)

1)Model:

Utilized scikit-learn's LinearRegression.

Evaluated using Mean Absolute Error (MAE).

## Discussion and Results

Neural Network (BP): After training, the neural network predicts turbine performance on the test set.

Linear Regression (MLR): MLR model predicts turbine performance on the test set.

Neural Network (BP): MAPE on the test set is calculated.

Linear Regression (MLR): MAE on the test set is calculated. Scatter plot comparing real values and predictions is generated.

The MAPE and MAE values, combined with the scatter plots, offer valuable insights into the performance of both models. Further analysis can be conducted to understand the strengths and weaknesses of each approach.

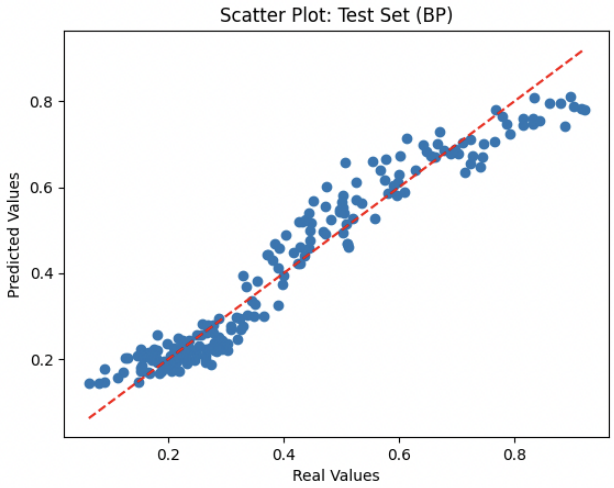
Synthetic Dataset:

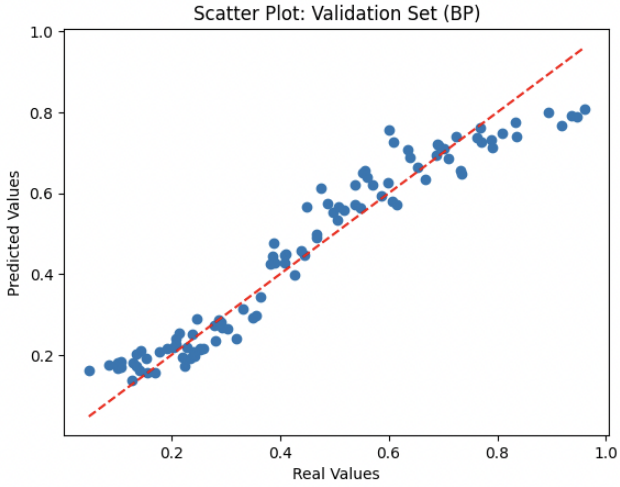
Backpropagation Neural Network (BP)

MAPE on Test Set: 78.65%

Scatter Plot: Visualization

```
MAPE on Test Set (BP): 78.64775007265602%
MAPE on Validation Set (BP): 96.06207278421789%
```

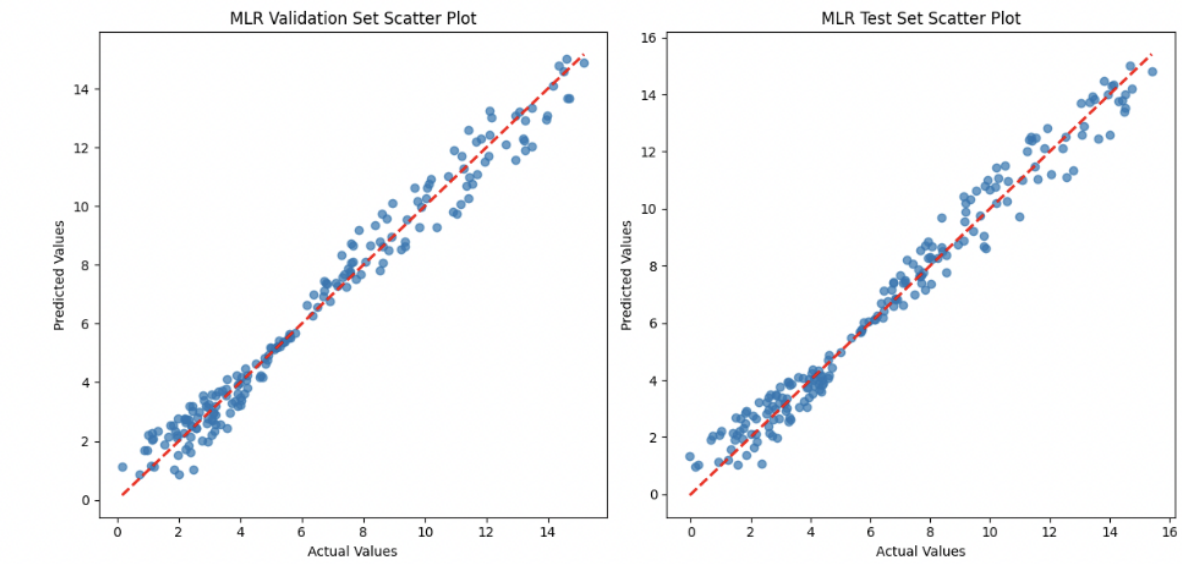



Linear Regression (MLR)

MAPE on Test Set: 38.20%

Scatter Plot: Visualization

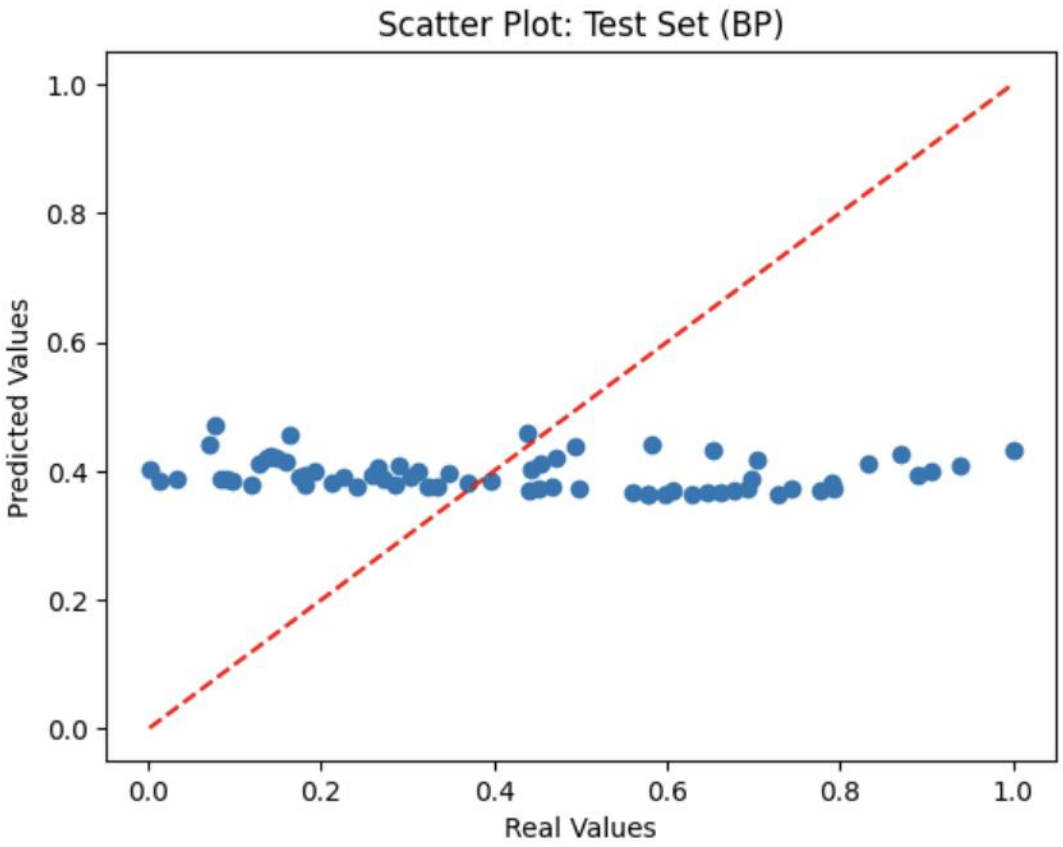



2) Turbine Dataset

Backpropagation Neural Network (BP)

MAPE on Test Set: 546.25%

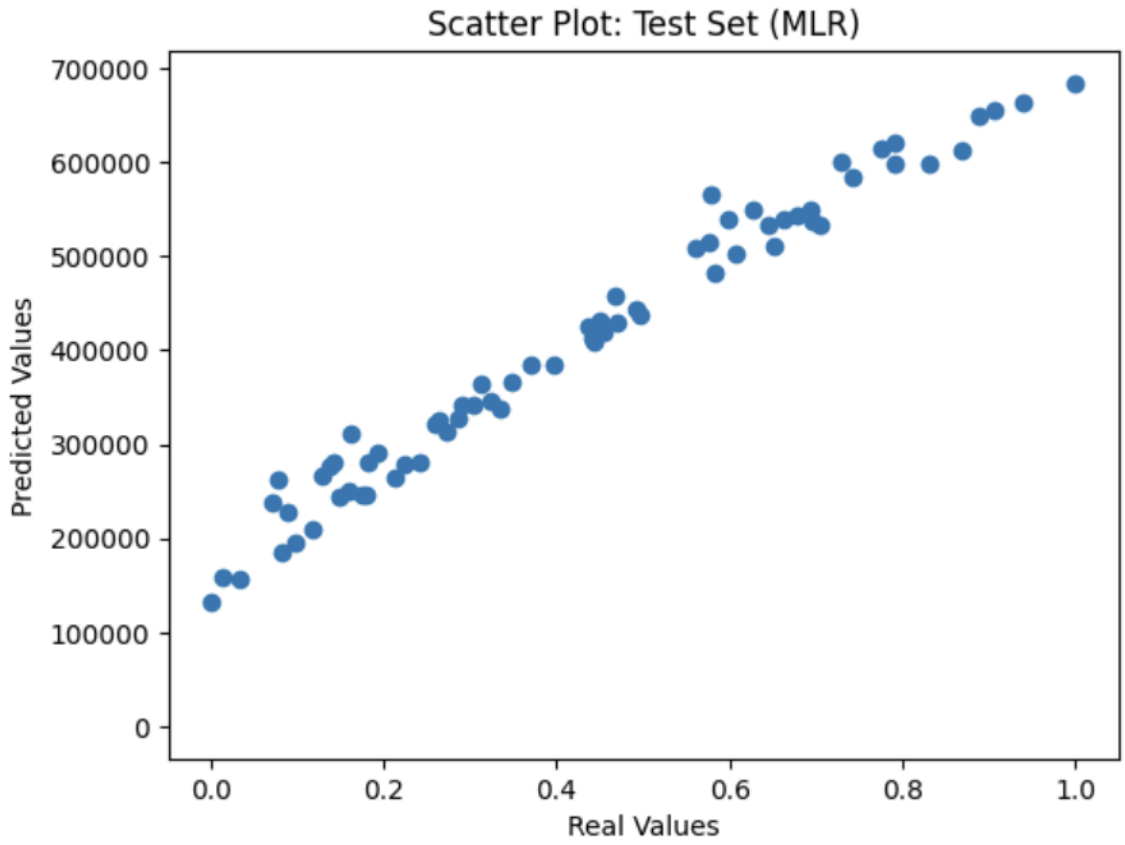
Scatter Plot: [Visualization]



Linear Regression (MLR)

MAPE on Test Set: 265,118,166.71%

Scatter Plot: Visualization



Parameter Exploration:

Table summarizing MAPE for different parameter sets.

Comparison:

MLR outperforms BP on the synthetic dataset.

Both models show challenges with turbine dataset predictions, particularly BP.

Comparison Table:

| Model | MAPE |
|---|---|
| BP | 78.65 |
| BP-F | 546.25 |
| MLR-F | 265118166.71 |

## Conclusions

In conclusion, our analysis of both synthetic and turbine datasets revealed noteworthy insights. In the synthetic dataset, the Multilinear Regression (MLR) model demonstrated superior performance compared to the Backpropagation (BP) neural network. This outcome underscores the sensitivity of BP to dataset characteristics and emphasizes the importance of aligning the model architecture with the nature of the data. However, challenges observed in both models when applied to the turbine dataset suggest potential issues with dataset suitability, possibly stemming from its complexity or unique characteristics. To address these challenges, we recommend further exploration of neural network architecture and comprehensive parameter tuning. Additionally, a detailed investigation into the turbine dataset's characteristics is warranted, and potential enhancements, such as feature engineering, should be explored. Looking ahead, future work should delve into advanced models, including more complex neural network architectures and ensemble methods, as well as focus on dataset enhancement through the addition of relevant features. Lastly, systematic hyperparameter tuning across a broader range is crucial for optimizing model performance. We extend our appreciation to Kaggle for providing the Fitbit Fitness Tracker Dataset, which played a pivotal role in this project and highlights the significance of accessible datasets in advancing machine learning and data science.