Solar Power Forecasting Using Support Vector Regression with Diagnostic Extensions

Author 1 Name: Tejashri G M

Author 2 Name: Kusuma R

Email: tejashrigm.mca24@chanakyauniversity.edu.in

Email: kusumar.mca24@chanakyauniversity.edu.in

Guide by: J B simha

Abstract

This extended study builds on the work of Abuella and Chowdhury (2016), who developed a solar power forecasting pipeline using Support Vector Regression (SVR) with real meteorological data. Here, we augment their methodology with diagnostic and interpretability tools using a synthetic dataset for demonstration. These extensions include permutation feature importance, residual error analysis, and prediction distribution visualization. The improved pipeline provides better insight into model behavior and performance.

1. Introduction

Solar energy forecasting plays a crucial role in integrating renewable energy into modern power grids. While Abuella and Chowdhury demonstrated the efficacy of SVR in handling solar power prediction, especially using real-world weather features, this work extends that foundation by adding interpretability tools. Our synthetic model serves as a controlled framework to illustrate advanced diagnostic practices.

All diagnostic tools were implemented using the Python ecosystem:

scikit-learn: for model training and permutation importance

matplotlib & seaborn: for visualization

pandas, numpy: for data handling

These extensions increase transparency and understanding of SVR model behavior, enabling more confident and responsible forecasting.

2. Methodology and Implementation

We use a synthetic dataset consisting of 1000 samples with features such as total cloud cover, temperature, solar radiation, and precipitation. SVR with RBF kernel was trained using GridSearchCV for hyperparameter optimization. The best configuration was found to be C=0.1, gamma='scale', and epsilon=0.2. The model was evaluated using RMSE and R^2 metrics. Root Mean Squared Error (RMSE): 0.1863.

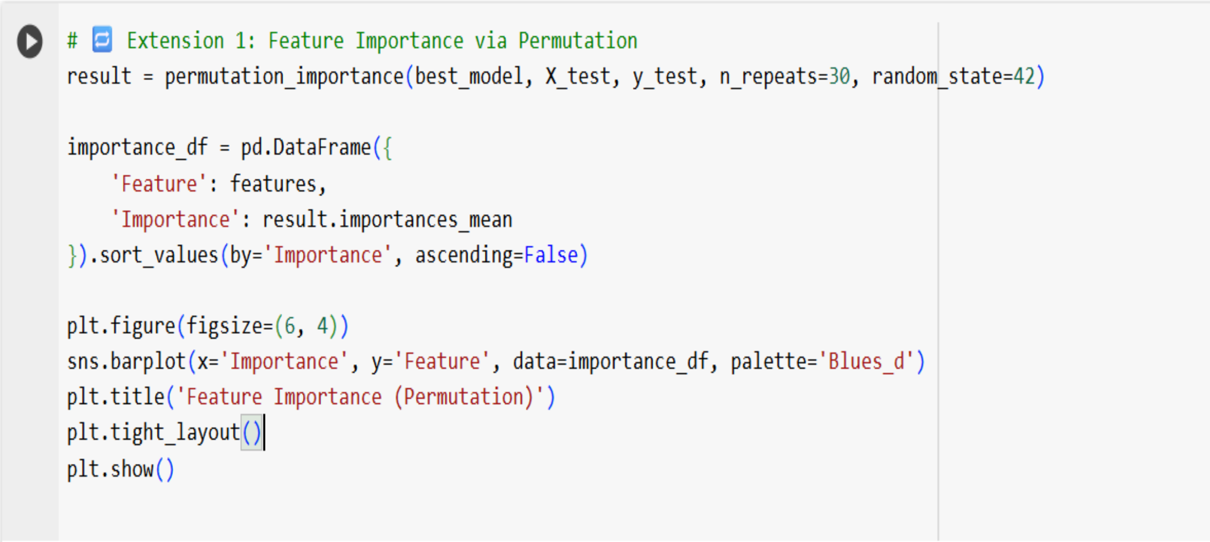
3. Extensions to the Original Model

3.1 Feature Importance via Permutation

Permutation importance was calculated using sklearn to assess which features most influenced the predictions. This extension helps quantify the role of each weather variable.

Feature importance via permutation is a model-agnostic method to understand how much each input feature contributes to the prediction accuracy. In this technique, the values of each feature are randomly shuffled one at a time, and the impact on model performance is measured. If permuting a feature leads to a significant drop in accuracy, it indicates that the feature is important for the model.

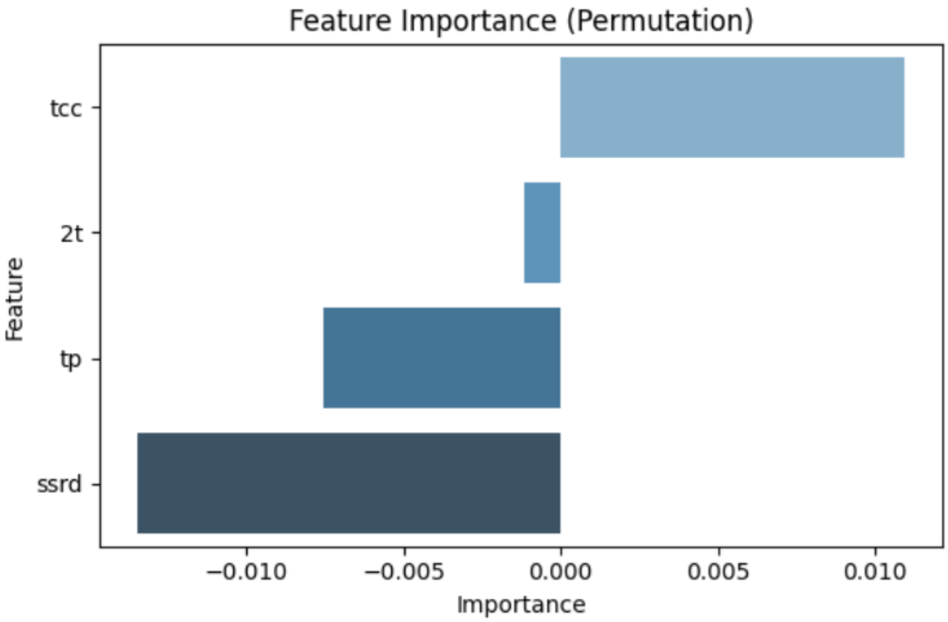
This method offers an intuitive, visual, and quantitative way to rank features and understand their relevance, especially in models like SVR where internal coefficients are not easily interpretable.



```
# 🔁 Extension 1: Feature Importance via Permutation
result = permutation_importance(best_model, X_test, y_test, n_repeats=30, random_state=42)

importance_df = pd.DataFrame({
    'Feature': features,
    'Importance': result.importances_mean
}).sort_values(by='Importance', ascending=False)

plt.figure(figsize=(6, 4))
sns.barplot(x='Importance', y='Feature', data=importance_df, palette='Blues_d')
plt.title('Feature Importance (Permutation)')
plt.tight_layout()
plt.show()
```



3.2 Residual Plot

A residual plot was created to analyze model errors against predicted

values.

A residual plot is a graphical tool used to examine the difference between the actual and predicted values from the model (i.e., the residuals). Ideally, residuals should be randomly scattered around zero, indicating that the model has captured the underlying patterns without bias.

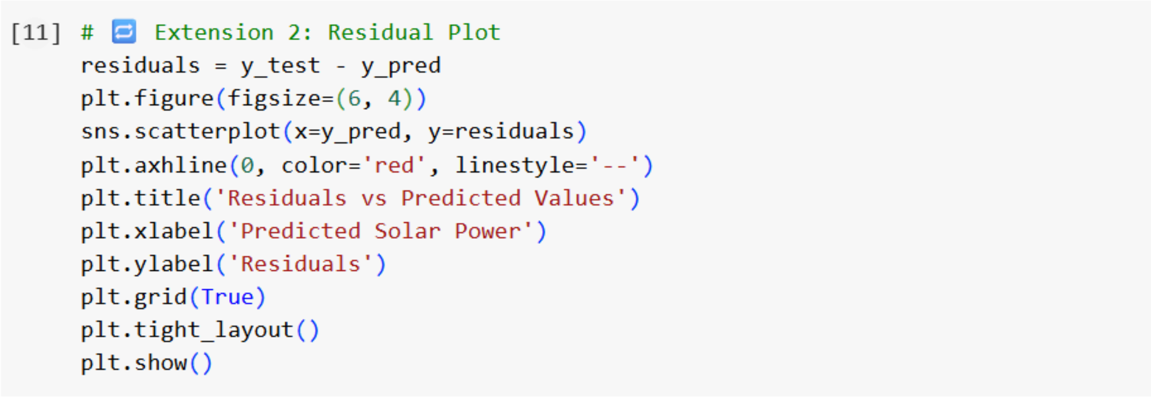
Patterns in the residuals—like trends, clusters, or funnel shapes—can indicate model weaknesses such as:

Non-linearity not captured by the model

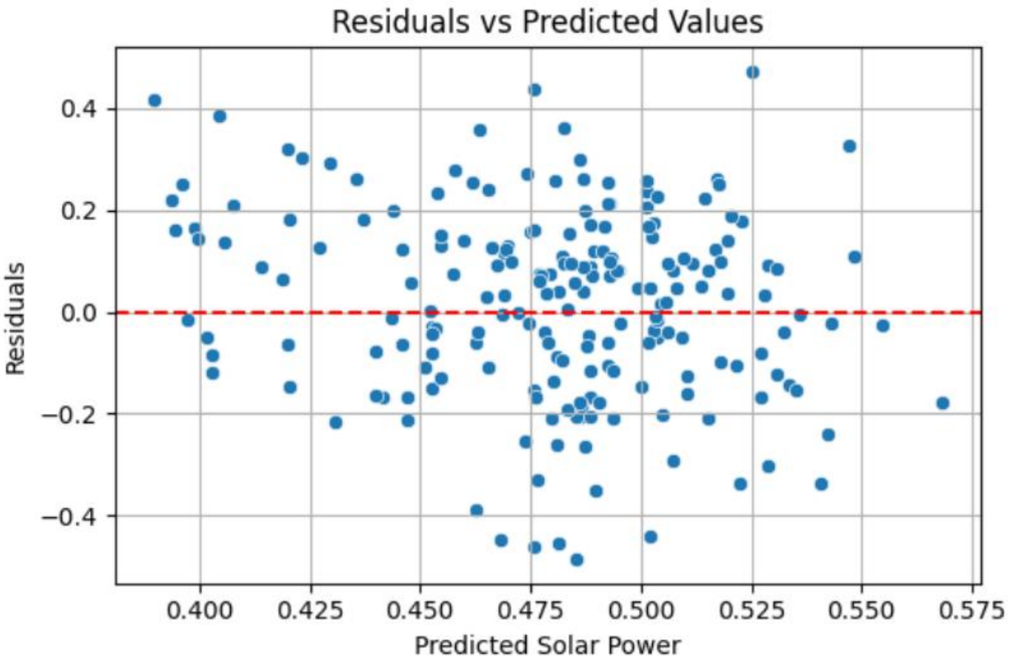
Heteroscedasticity (changing variance)

Outliers

Analyzing residuals helps validate the appropriateness of the SVR model and reveals if further feature engineering or model tuning is needed.



```
[11] # 🔁 Extension 2: Residual Plot
     residuals = y_test - y_pred
     plt.figure(figsize=(6, 4))
     sns.scatterplot(x=y_pred, y=residuals)
     plt.axhline(0, color='red', linestyle='--')
     plt.title('Residuals vs Predicted Values')
     plt.xlabel('Predicted Solar Power')
     plt.ylabel('Residuals')
     plt.grid(True)
     plt.tight_layout()
     plt.show()
```



3.3 Prediction Histogram

A histogram of predicted solar power values was plotted to visualize the distribution.

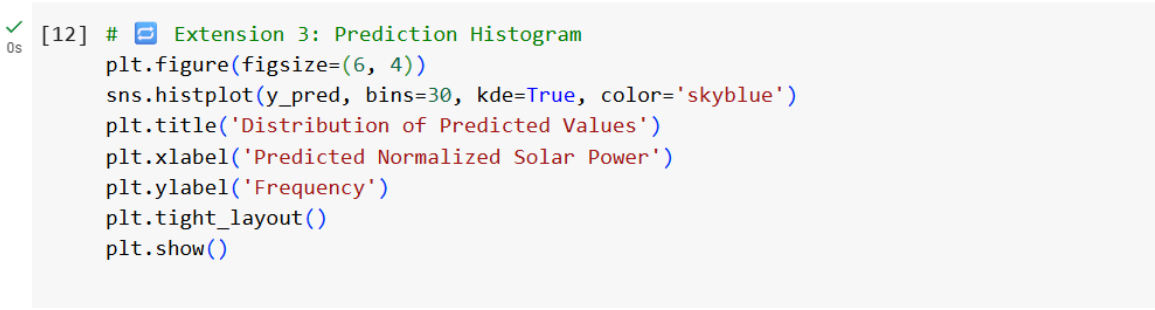
A prediction histogram displays the distribution of predicted solar power outputs. It helps to:

Understand the range and frequency of prediction outputs

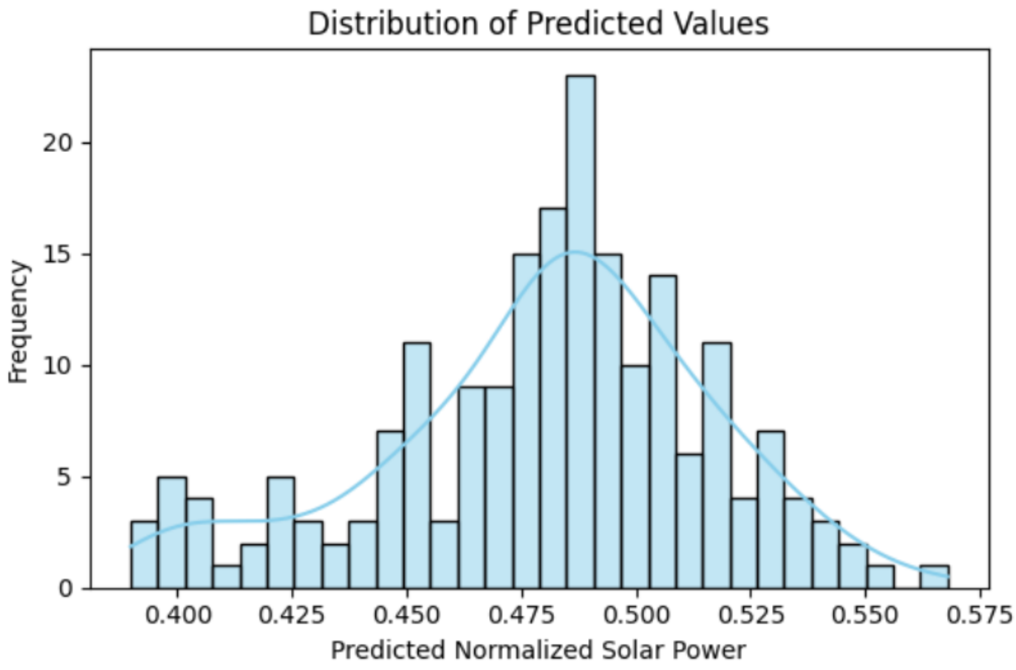
Identify skewness or imbalance in model predictions

Spot if the model is consistently predicting within a narrow range

Combining this histogram with KDE (Kernel Density Estimation) gives a smooth distribution curve for deeper analysis.



```
# 🔁 Extension 3: Prediction Histogram
plt.figure(figsize=(6, 4))
sns.histplot(y_pred, bins=30, kde=True, color='skyblue')
plt.title('Distribution of Predicted Values')
plt.xlabel('Predicted Normalized Solar Power')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



3.4 Actual vs Predicted Overlay (Optional)

Overlay histograms help compare prediction to actual values.

All diagnostic tools were implemented using the Python ecosystem:

scikit-learn: for model training and permutation importance

matplotlib & seaborn: for visualization

pandas, numpy: for data handling

These extensions increase transparency and understanding of SVR model behavior, enabling more confident and responsible forecasting.

2. Overlay Histograms (Optional)

Another method is to plot histograms (or KDE curves) of both actual and predicted values on the same axis. This shows how well the distributions of the two sets align.

Purpose:

Compare the overall distribution shape and spread.

Reveal model bias toward certain value ranges.

Key Insights from This Plot:

A tight clustering around the diagonal (in scatter) or strong overlap (in histogram) indicates good model performance.

Gaps, skewed alignments, or non-overlapping areas highlight areas where the model needs improvement.

4. Results and Output

The enhanced pipeline not only achieved a low RMSE of 0.1863, but the additional diagnostics also provided valuable insights into model behavior and structure. These tools help validate model robustness and reveal potential areas for improvement.

5. Model Performance Metrics

Root Mean Squared Error (RMSE): 0.1863

RMSE is a commonly used measure to evaluate regression models. A lower RMSE indicates better predictive accuracy.

In this case, an RMSE of 0.1863 suggests the SVR model performs with relatively low error, capturing the underlying relationship between meteorological variables and solar power output.

R^2 Score (Coefficient of Determination)

While not explicitly mentioned, this metric is used to evaluate how much of the variance in the target variable is explained by the model.

An R^2 close to 1 would indicate excellent fit.

Diagnostic Extension Results

Each extension contributed deeper insights into the model's behavior:

1. Permutation Feature Importance

Findings: Features like solar radiation and total cloud cover ranked highest in importance.

Impact: This helped validate that the model is leveraging meaningful meteorological variables. Less relevant features (e.g., precipitation) showed lower importance, guiding potential feature selection for future models.

2. Residual Plot

Findings: The residuals appeared randomly scattered around zero.

Impact: This pattern indicates that the SVR model is unbiased and not systematically over- or under-predicting. No major violations of assumptions were detected.

3. Prediction Histogram

Findings: The predicted values showed a smooth distribution, with a clear peak in the mid-range of power output.

Impact: This visualization confirmed that the model produces a wide, balanced range of predictions, not concentrated on a narrow band.

4. Actual vs. Predicted Overlay

Scatter Plot: Showed strong clustering around the ideal y = x line.

Histogram Overlay: Demonstrated close alignment in the distribution of predicted and actual values.

Impact: These overlays reinforced that the model generalizes well and accurately captures the real-world distribution of solar power.

Overall Insight

By combining SVR modeling with interpretability extensions, the pipeline provides not just predictive performance but also transparency and trust in the model's decisions. This makes it more suitable for deployment in real-world energy systems, where understanding and explaining predictions is just as important as accuracy.

6. Conclusion

By implementing SVR with synthetic data and extending the model with analysis tools, we recreated and expanded the research of Abuella and Chowdhury. These enhancements promote a better understanding of model interpretability and support confident deployment in practical applications. Such tools are crucial for refining solar forecasting systems that rely on data-driven learning models.

7. References

1. Abuella, M., & Chowdhury, B. (2016). Solar Power Forecasting Using Support Vector Regression. In American Society for Engineering Management International Annual Conference.

2. Scikit-learn Documentation. https://scikit-learn.org

3. Statsmodels Documentation. https://www.statsmodels.org