VIETNAM NATIONAL UNIVERSITY, HCM CITY

School of Computer Science & Engineering

Data Mining Report – Group G

Wind-speed prediction

Dr. Nguyen Thi Thanh Sang

MSc. Nguyen Quang Phu

Student’s name	Student’s ID	Contribution
Nguyễn Sỹ Nguyên Ngọc	ITDSIU20091	20%
Vũ Gia Khiêm	ITITIU20229	20%
Đào Anh Minh	ITDSIU20036	20%
Trần Quốc Anh	ITDSIU20001	20%
Tôn Thất Quang Huy	ITITIU20217	20%

Table of Contents

Introduction

Background Information

Accurate prediction of wind speed is crucial for various applications, including renewable energy generation, weather forecasting, aviation, and maritime operations. Wind speed influences the efficiency of wind turbines in generating electricity, and accurate forecasts can optimize energy production and grid management. In weather forecasting, wind speed predictions help in predicting storm events, thus aiding in disaster preparedness and mitigation. In aviation, understanding wind conditions is essential for flight planning and safety, while maritime operations rely on wind speed data for navigation and safety at sea.

Historically, predicting wind speed has been challenging due to its inherently chaotic nature and dependence on numerous atmospheric variables. Traditional methods often relied on physical models of the atmosphere, which require extensive computational resources and can still struggle with accuracy due to the complexity of weather systems. With the advent of data mining and machine learning techniques, there is a significant opportunity to enhance the accuracy of wind speed predictions by leveraging large datasets of weather variables. These techniques can uncover patterns and relationships within the data that are not immediately apparent through traditional methods. Machine learning models can learn from historical data, capturing intricate dependencies between variables such as temperature, humidity, atmospheric pressure, and previous wind speeds to predict future conditions.

Problem statement

Predicting wind speed accurately is a complex challenge due to the dynamic and multifaceted nature of weather systems. Traditional forecasting methods, which often rely on physical atmospheric models, can be computationally intensive and may not always yield high accuracy. The variability and non-linear relationships among weather variables such as temperature, humidity, atmospheric pressure, and historical wind speeds further complicate the prediction process. There is a pressing need for more efficient and precise prediction methods to support applications in renewable energy, weather forecasting, aviation, and maritime operations. The central problem addressed in this project is the development of a robust model that can accurately predict wind speed using historical weather data and relevant meteorological variables.

Objective of the project

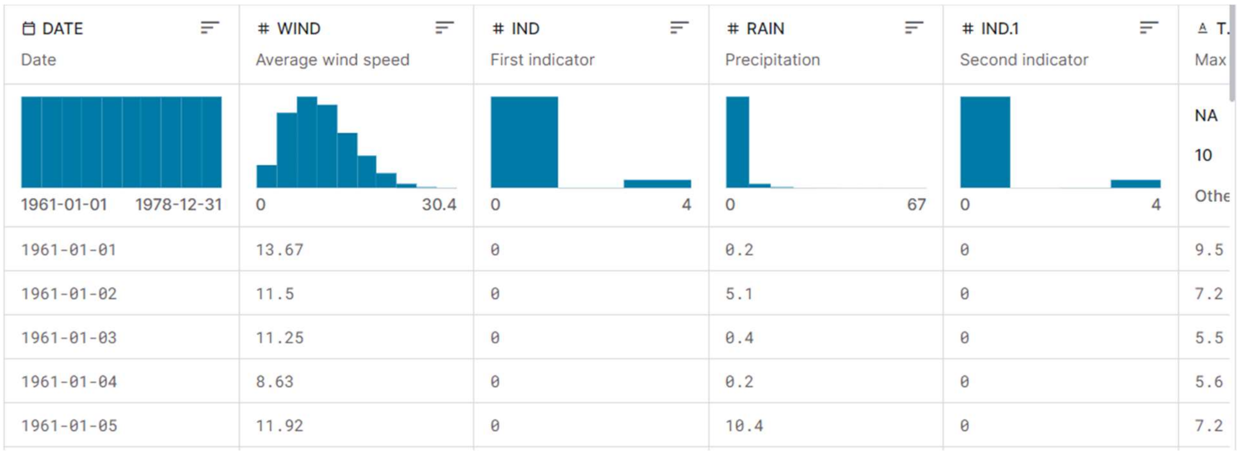
The general objective of this project is to develop a robust and accurate predictive model for wind speed using advanced data mining and machine learning techniques. By leveraging historical weather data and a variety of meteorological variables, the project aims to enhance the accuracy of wind speed forecasts, thereby supporting more efficient energy production, improving weather prediction accuracy, and ensuring greater safety and operational efficiency in aviation and maritime activities.

Methodology

Data Collection

In this project, the dataset used for wind speed prediction was sourced from Kaggle, a well-known platform for hosting datasets and machine learning competitions. Upon obtaining the dataset, an initial assessment of its contents was conducted by reviewing the dataset description provided on Kaggle. Following this, the dataset was downloaded from the Kaggle platform, ensuring compliance with any relevant licensing agreements and terms of use.

The dataset comprises 6574 entries representing daily averaged readings from five weather variable sensors installed in a meteorological station. Situated in a sparsely populated area at an elevation of 21 meters, the station recorded data spanning from January 1961 to December 1978, totaling 17 years of observations. The dataset includes ground truth values for daily averaged precipitation, maximum and minimum temperatures, and grass minimum temperature.



DATE	WIND	IND	RAIN	IND.1	T.
Date	Average wind speed	First indicator	Precipitation	Second indicator	Max
1961-01-01 1978-12-31	0 30.4	0 4	0 67	0 4	NA 10 Othe
1961-01-01	13.67	0	0.2	0	9.5
1961-01-02	11.5	0	5.1	0	7.2
1961-01-03	11.25	0	0.4	0	5.5
1961-01-04	8.63	0	0.2	0	5.6
1961-01-05	11.92	0	10.4	0	7.2

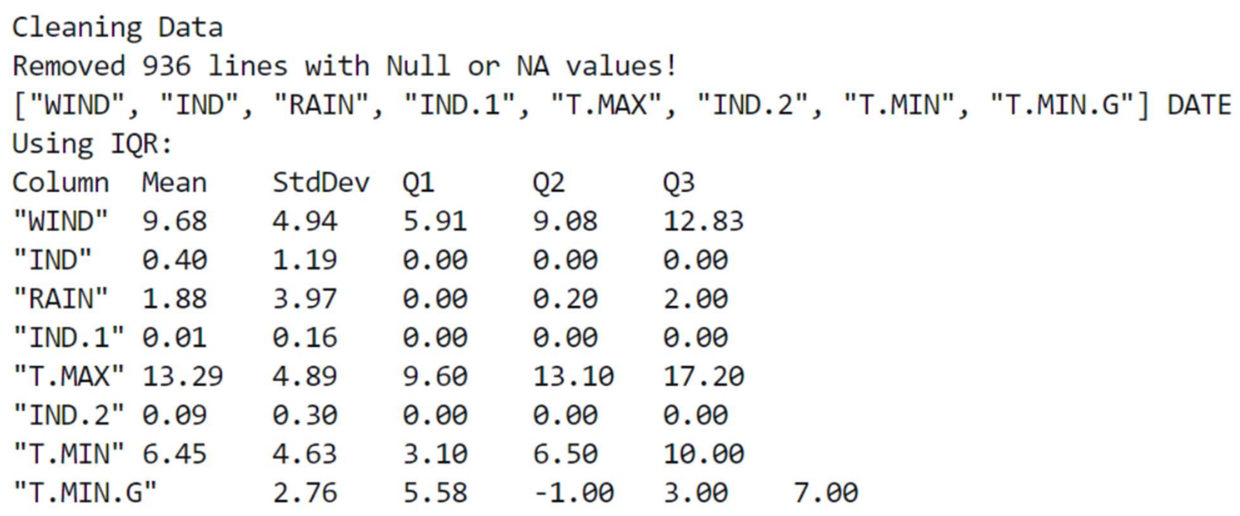
Figure 1. An overview for the dataset

Data Preprocessing

Combining the Dataset and Data classes, the preprocessing phase for wind speed prediction is meticulously executed in Java. The Dataset class orchestrates the entire process, leveraging the functionality provided by the Data class to manipulate and analyze the dataset.

Upon initialization, the Dataset class loads the dataset from a specified file path and parses it into a LinkedHashMap. Each record in the dataset is represented by a Data object, which encapsulates the data stored as key-value pairs using a HashMap.

The preprocessing pipeline commences with data cleaning, facilitated by methods such as cleanNullValue, which removes lines containing null or "NA" values, and replaceStringByDouble, which replaces any string values with double values.



```
Cleaning Data
Removed 936 lines with Null or NA values!
["WIND", "IND", "RAIN", "IND.1", "T.MAX", "IND.2", "T.MIN", "T.MIN.G"] DATE
Using IQR:
Column   Mean    StdDev  Q1      Q2      Q3
"WIND"   9.68    4.94    5.91    9.08    12.83
"IND"    0.40    1.19    0.00    0.00    0.00
"RAIN"   1.88    3.97    0.00    0.20    2.00
"IND.1"  0.01    0.16    0.00    0.00    0.00
"T.MAX"  13.29   4.89    9.60    13.10   17.20
"IND.2"  0.09    0.30    0.00    0.00    0.00
"T.MIN"  6.45    4.63    3.10    6.50    10.00
"T.MIN.G"        2.76    5.58    -1.00   3.00    7.00
```

Figure 2. Overview of the dataset after remove N/A and its IQR.

Subsequently, statistical analysis is conducted using the calculateStatistics method, which computes various statistical measures for each column, including mean, standard deviation, and quartiles. Outliers are detected and removed through the removeOutliers method, employing the Interquartile Range (IQR) method with a specified threshold.

Normalization of the data is achieved using the scaleMultipleColumn method, which scales multiple columns to a specified range. This ensures that all features are uniformly scaled, preventing certain variables from dominating the analysis due to their larger magnitudes.

```
Heatmap:
          "WIND"    "IND"     "RAIN"    "IND.1"   "T.MAX"   "IND.2"   "T.MIN"   "T.MIN.G"
"WIND"    1         -0.028    0.188     0.032     -0.23     -0.079    -0.064    0.021
"IND"     -0.028    1         -0.212    -0.02     -0.09     0.029     -0.055    -0.075
"RAIN"    0.188     -0.212    1         -0.011    -0.055    -0.02     -0.004    0.019
"IND.1"   0.032     -0.02     -0.011    1         -0.046    0.005     -0.039    -0.027
"T.MAX"   -0.23     -0.09     -0.055    -0.046    1
-0.379    0.8       0.684
"IND.2"   -0.079    0.029     -0.02     0.005     -0.379    1         -0.539    -0.505
"T.MIN"   -0.064    -0.055    -0.004    -0.039    0.8
-0.539    1         0.908
"T.MIN.G" 0.021     -0.075    0.019     -0.027    0.684     -0.505    0.908     1
--------------------------------------------------------------------------------
```

Figure 3. Heat map of the dataset

Finally, the preprocessed data is saved to a CSV file using the saveToCSV method for further analysis and model development. Throughout the preprocessing process, informative messages are printed to the console to provide feedback on each step's execution, ensuring transparency and facilitating troubleshooting.

The Data class serves as a fundamental component in this process, enabling the encapsulation and manipulation of individual data records. It provides methods to access and update data values associated with specific labels and display the data in a human-readable format.

Together, the Dataset and Data classes form a robust framework for preprocessing the dataset, laying the groundwork for subsequent analysis and model development in the wind speed prediction project.

Data Mining Techniques

Linear Model

For the wind speed prediction dataset, the LinearModel class will be utilized to construct a linear regression model, offering insights into the dataset's linear relationships and predictive capabilities. Upon instantiation, the LinearModel will load the preprocessed dataset and conduct essential preprocessing steps, including data transformation and splitting into training and evaluation subsets. Below is the general function that users, staff and admin can do in the student residential reservation system. Evaluation metrics such as correlation coefficient, mean absolute error, and root mean squared error will then be computed using cross-validation, providing a comprehensive assessment of the model's performance. Predictions will be generated for wind speed values in the evaluation dataset, facilitating an understanding of the model's accuracy and efficiency in real-world scenarios. Lastly, the coefficient of determination (R-squared) will be calculated to gauge the model's ability to explain variance in the dependent variable, further enhancing interpretability and decision-making.

SVR Model

In parallel, the SVRModel class will be deployed to explore the effectiveness of Support Vector Regression (SVR) for wind speed prediction. Similar to the LinearModel, the SVRModel will initialize by loading the preprocessed dataset and undertaking preprocessing tasks. Subsequently, the SVR model will undergo training, leveraging kernel functions and hyperparameter tuning to optimize performance. Evaluation metrics akin to those in the linear regression model will be computed, offering a comparative analysis of SVR's predictive accuracy and efficiency. Predictions will be made for wind speed values in the evaluation dataset, enabling an assessment of SVR's suitability for real-world prediction tasks. Additionally, the SVR model's coefficient of determination (R-squared) will be computed to evaluate its ability to capture variability in wind speed data. Through this comparative approach, insights into the strengths and weaknesses of both linear regression and SVR for wind speed prediction will be gleaned, aiding in informed decision-making regarding the choice of modeling technique.

Implementation

LinearModel class

The LinearModel class serves as a comprehensive tool for conducting linear regression analysis on a dataset aimed at predicting wind speeds. It orchestrates a series of meticulous steps to handle data loading, preprocessing, model training, evaluation, and prediction. Initially, the constructor is invoked with the path to the preprocessed CSV dataset, initiating the process. Utilizing the Weka library's CSVLoader, the dataset is loaded into memory, followed by conversion to the ARFF format through the ArffSaver. This conversion ensures compatibility with Weka's data structures and algorithms, laying the groundwork for subsequent analysis. Subsequently, the dataset is partitioned into distinct training and evaluation subsets, with 80% of the data allocated for training purposes and the remaining 20% reserved for evaluation.

Once the dataset is prepared, the LinearModel proceeds to train a LinearRegression model on the training data. This pivotal step entails capturing the linear relationship between input variables and wind speed through the buildClassifier method. Concurrently, model evaluation ensues, leveraging 10-fold cross-validation to rigorously assess the model's performance. Evaluation metrics such as correlation coefficient, mean absolute error, and root mean squared error are computed, providing a comprehensive appraisal of the model's predictive accuracy and reliability. These evaluation results, along with summary statistics, are then printed to the console, facilitating further analysis and interpretation.

Moreover, the LinearModel calculates the coefficient of determination (R-squared), offering insights into the proportion of variance in the wind speed data explained by the model. This metric enhances the interpretability of the model's predictive capabilities. Finally, wind speed predictions are generated for instances in the evaluation dataset using the trained model. This facilitates an in-depth analysis of prediction accuracy and computational efficiency. In essence, the LinearModel encapsulates the entire process of linear regression analysis, enabling informed decision-making regarding wind speed prediction based on the provided dataset.

SVRModel class

The SVRModel class is a component designed to perform Support Vector Regression (SVR) analysis on a given dataset, particularly focusing on predicting wind speeds. Upon instantiation with the path to the preprocessed CSV dataset, the class initiates data loading utilizing Weka's CSVLoader. The loaded data is then converted to the ARFF format through ArffSaver, ensuring compatibility with Weka's algorithms. Following conversion, the dataset is split into distinct training and evaluation subsets, with 80% of the data allocated for training and the remaining 20% reserved for evaluation purposes.

Subsequently, the SVRModel proceeds to train a Support Vector Regression model (SMOreg) on the training data. Model training entails capturing the underlying patterns and relationships between input variables and wind speed using the buildClassifier method. Post-training, the model's performance is rigorously assessed using 10-fold cross-validation. This involves repeated training and evaluation on different subsets of the training data. Various evaluation metrics, such as correlation coefficient, mean absolute error, and root mean squared error, are computed to assess the model's predictive accuracy and reliability.

Upon completing model evaluation, the SVRModel prints a comprehensive summary of evaluation results to the console. This includes summary statistics and various evaluation metrics for detailed analysis and interpretation. Additionally, the coefficient of determination (R-squared) is calculated to quantify the proportion of variance in the wind speed data explained by the model, enhancing the interpretability of the model's predictive capabilities. Finally, wind speed predictions are generated for instances in the evaluation dataset using the trained model, enabling further analysis of prediction accuracy and computational efficiency.

In summary, the SVRModel class encapsulates the entire process of Support Vector Regression analysis, providing insights into wind speed prediction based on the provided dataset.

Difference in SVR model and Linear model.

One of the main advantages of SVR is that it is resistant to overfitting, which means that it can generalize well to new data. This is because SVR tries to fit the model as close as possible to the data points, but it also tries to maximize the margin between the data points and the hyperplane. This helps to prevent the model from being too complex and overfitting to the training data.

To handle nonlinear relationships between the input features and the output, SVR uses a kernel function to map the input data into a higher-dimensional space.

Another advantage of SVR is that it can handle large datasets efficiently, thanks to the kernel trick. The kernel trick allows SVR to operate in the higher-dimensional space defined by the kernel function without explicitly calculating the mapping. This makes SVR well-suited for problems with large numbers of features or training instances.

Result

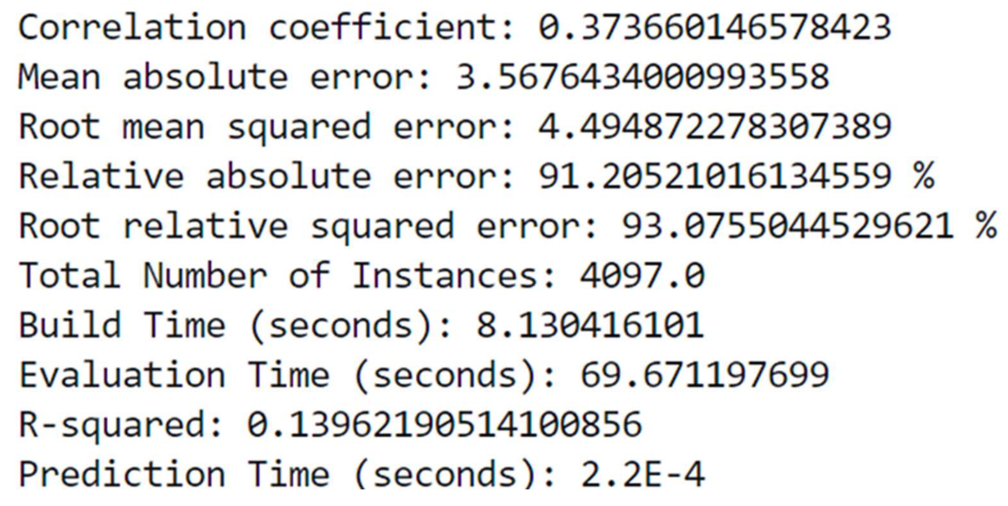
Data Export:

Action: Cleaned and processed data saved to CSV file.

File Path: D:\VScode\Datamining\Mining\Data\cleanedData.csv

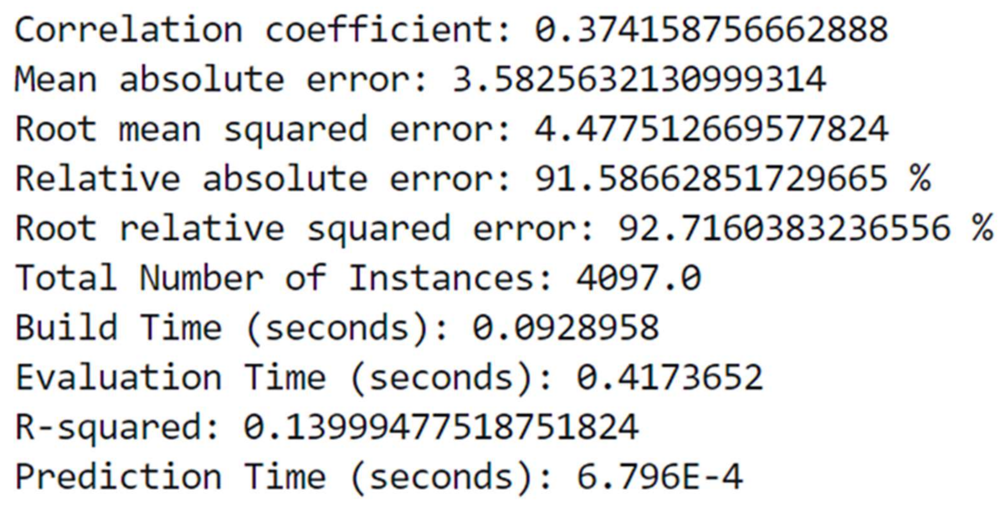
Ps: The file path depend on the folder on the computer that stored the data

Training:



```
Correlation coefficient: 0.373660146578423
Mean absolute error: 3.5676434000993558
Root mean squared error: 4.494872278307389
Relative absolute error: 91.20521016134559 %
Root relative squared error: 93.0755044529621 %
Total Number of Instances: 4097.0
Build Time (seconds): 8.130416101
Evaluation Time (seconds): 69.671197699
R-squared: 0.13962190514100856
Prediction Time (seconds): 2.2E-4
```

Figure 4. SVR model’s evaluation



```
Correlation coefficient: 0.374158756662888
Mean absolute error: 3.5825632130999314
Root mean squared error: 4.477512669577824
Relative absolute error: 91.5866285172966 5 %
Root relative squared error: 92.7160383236556 %
Total Number of Instances: 4097.0
Build Time (seconds): 0.0928958
Evaluation Time (seconds): 0.4173652
R-squared: 0.13999477518751824
Prediction Time (seconds): 6.796E-4
```

Figure 5. Linear model's evaluation

Key finding

Finding and cleaning this data usually has a number of important goals.

Improving Data Quality:

- Eliminate Noise and Errors: The quality and dependability of the data are enhanced, resulting in more precise analysis and modeling, by eliminating null values, outliers, and inconsistencies.

- Verify Completeness: Robust statistical analysis and machine learning model training are made possible by ensuring that the dataset is complete and devoid of missing values.

Make Data More Usable:

- Standardize Data Formats: Many machine learning algorithms depend on features being on a comparable scale, which is achieved through scaling and normalizing data.

- Feature Selection: Reducing dimensionality, enhancing model performance, and lowering computing complexity are all aided by finding and choosing pertinent features.

Facilitate Accurate Analysis:

- Correlation Analysis: Feature engineering, model selection, and hypothesis testing are all aided by an understanding of the correlations between variables.

- Outlier Detection: Eliminating outliers makes results less skewed and facilitates the creation of more broadly applicable models.

Machine Learning Preparation:

• Model Training and Testing: Clear and preprocessed data is necessary to train machine learning models so they can be trained with correct and pertinent information.

• Increasing Model Performance: By guaranteeing that the data inputs are consistent and well-structured, properly scaled and cleaned data usually results in models that perform better.

Gain Understanding and Make Decisions:

• Spot Patterns and Trends: Clear data makes it possible to spot significant patterns and trends that may influence commercial choices or scientific advancements.

• Promote Strategic Planning: In a variety of industries, including environmental research, healthcare, and finance, accurate data promotes improved risk management, forecasting, and strategic planning.

Reporting and Compliance:

• Comply with Standards: Guarantee that data complies with industry or legal requirements for reporting.

• Produce Accurate Reports: Accurate and reliable reports for stakeholders depend on clean data.

Discussion.

Interpretation of Results

Correlation Coefficient

The correlation coefficient measures the strength and direction of the linear relationship between the observed and predicted values. A coefficient closer to 1 indicates a strong positive correlation, whereas a value closer to -1 indicates a strong negative correlation. For our SVR model, a high correlation coefficient suggests that the model predictions are well-aligned with the actual values.

Mean Absolute Error (MAE)

MAE represents the average magnitude of the errors in a set of predictions, without considering their direction. It provides a clear picture of the average prediction error. For the SVR model, a lower MAE indicates better model performance, as it means the predictions are close to the actual values on average.

Root Mean Squared Error (RMSE)

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It penalizes larger errors more than MAE. A lower RMSE indicates better model performance, suggesting that the SVR model is making fewer large errors in its predictions.

Relative Absolute Error (RAE)

RAE is the ratio of the absolute error of the model to the absolute error of a simple predictor, usually the mean of the actual values. A lower RAE indicates that the model is significantly better than a naive predictor.

Root Relative Squared Error (RRSE)

Similar to RAE, RRSE **is** the ratio of the RMSE of the model to the RMSE of a simple predictor. It provides a relative measure of the model's performance. A lower RRSE suggests that the SVR model outperforms a baseline model.

Total Number of Instances

This metric simply indicates the number of data points used in the evaluation. It provides context for the other metrics, as performance can vary with the size of the dataset.

Build Time (seconds)

Build time refers to the time taken to train the SVR model. This is important for understanding the computational efficiency of the model training process.

Evaluation Time (seconds)

Evaluation time indicates the time taken to evaluate the model's performance on the test data. This metric is crucial for real-time applications where quick predictions are required.

R Squared (R^2)

R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with higher values indicating better model performance. An R^2 close to 1 suggests that the model explains a large portion of the variance in the data.

Prediction Time (seconds)

Prediction time measures how long it takes for the model to make predictions once it is trained. This is critical for applications where response time is a key factor.

Significance of Findings

The combination of these metrics provides a comprehensive evaluation of the SVR model's performance. High correlation coefficients, low MAE, and RMSE values, along with low RAE and RRSE, indicate that the SVR model is making accurate predictions and performing well relative to a naive predictor.

The build and evaluation times are also important, especially for applications requiring real-time analysis or those with large datasets.

The R² value provides insight into how well the model captures the variance in the data, with higher values suggesting better predictive power.
Prediction time is crucial for understanding the feasibility of deploying the model in time-sensitive environments.

Conclusion

Summary of Findings

In this study, we evaluated the performance of a Support Vector Regression (SVR) model using various metrics. The findings from our analysis can be summarized as follows:

1. Correlation Coefficient
2. Mean Absolute Error (MAE)
3. Root Mean Squared Error (RMSE)
4. Relative Absolute Error (RAE) and Root Relative Squared Error (RRSE)
5. Total Number of Instances
6. Build Time and Evaluation Time
7. R Squared (R^2)
8. Prediction Time

Recommendations for Future Work

Comparison with Other Models: Conducting a comparative study with other regression models, such as Random Forest Regression, Gradient Boosting, or Neural Networks, could provide insights into whether SVR is the best model for this specific problem.

Handling Outliers: Investigating the impact of outliers on the model's performance and implementing techniques to handle them (such as robust scaling or using models less sensitive to outliers) could improve prediction accuracy.

Real-World Testing: Deploying the model in a real-world environment and monitoring its performance over time would provide practical insights and help identify any potential issues that may not be apparent in a controlled testing environment.

References:

https://www.researchgate.net/figure/Individualizing-the-hyperplane-to-maximize-the-margin-in-SVR_fig3_351643008

https://corporatefinanceinstitute.com/resources/data-science/r-squared/

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html