IEC - 03
Estimating Aqueous Solubility Directly From Molecular Structure

November 4, 2024

Group Members

Sahil Kumar 22114083

Yash Joshi 22114108

Shubham Kr Verma 22114092

Mohammed Haaziq Jamal 22114055

---

1 Introduction

The ability to predict the solubility of compounds is crucial in various fields, including pharmaceuticals and material sciences. In this project, we utilize a dataset containing molecular descriptors and solubility measurements to develop predictive models.

This report outlines the analysis and modeling of solubility using a dataset that includes molecular descriptors. Various regression models were evaluated for their performance in predicting solubility, with parameter tuning conducted to optimize model accuracy.

2 Data Preprocessing

The dataset used in this study is `delaney_solubility_with_descriptors.csv`. The target variable is `logS`, which represents the solubility. We performed the following preprocessing steps:

- Split the dataset into training and testing sets.
- Standardize the features using `StandardScaler`.

3 Exploratory Data Analysis (EDA)

The purpose of this Exploratory Data Analysis (EDA) is to investigate the relationships between molecular features and their solubility as indicated by the logarithm of solubility (logS). Understanding these relationships is crucial for developing predictive models that can accurately estimate solubility based on molecular characteristics.

3.1 Understanding Relationships Between Features and logS

3.1.1 MolLogP vs logS

The analysis reveals a negative correlation between MolLogP and logS (correlation coefficient: -0.828). This suggests that as the MolLogP value increases, solubility (logS) tends to decrease. This linear relationship indicates that linear models, such as Linear Regression, could effectively capture this dependency.

3.1.2 MolWt vs logS

Similar to MolLogP, there is a negative trend between Molecular Weight (MolWt) and logS (correlation coefficient: -0.637). This indicates that larger molecules are likely to be less soluble, further supporting the use of linear modeling techniques while acknowledging the potential for non-linear behaviors in the data distribution.

3.1.3 NumRotatableBonds vs logS

The feature NumRotatableBonds demonstrates a weaker negative correlation with logS (correlation coefficient: -0.239). Molecules with more rotatable bonds tend to have lower solubility, although this relationship is less pronounced, suggesting that it may play a minor role in predicting solubility.

3.1.4 AromaticProportion vs logS

The analysis of AromaticProportion shows that many compounds exhibit values of either 0 or 1, indicating that they are primarily either fully aromatic or non-aromatic. There is no clear linear relationship between AromaticProportion and logS (correlation coefficient: -0.268), suggesting that this feature might not be a strong predictor of solubility.

3.2 Feature Distribution Insights

3.2.1 MolLogP and MolWt

Both MolLogP and MolWt display broad distributions, making them promising candidates for predictive modeling. Notably, MolWt exhibits significant right skewness, indicating that transformations (e.g., log transformation) could be beneficial for modeling.

3.2.2 NumRotatableBonds

This feature primarily consists of discrete values (e.g., 0, 1, 2), suggesting it may behave similarly to a categorical or ordinal feature. Although linear models can handle these values, decision-tree-based models, such as Random Forests, may capture the underlying patterns more effectively.

3.2.3 AromaticProportion

Given its binary nature (mostly values of 0 or 1), AromaticProportion could influence the modeling approach. This feature may work well with models that handle categorical data, such as Logistic Regression or Decision Trees.

4 Model Selection and Evaluation

In this analysis, various machine learning models were evaluated to predict solubility (logS) based on molecular descriptors. The performance of each model was measured using key regression metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination R^2. The table below presents the comparative performance results.

Model	RMSE	MAE	R^2
Neural Network	0.7212	0.5364	0.8926
Random Forest	0.8040	0.5811	0.8665
Gradient Boosting	0.8190	0.6075	0.8615
Support Vector Machine	0.8793	0.6371	0.8403
Linear Regression	1.0103	0.7798	0.7892
Ridge Regression	1.0104	0.7799	0.7891
Lasso Regression	1.0258	0.7989	0.7826

Table 1: Comparative performance of different models for solubility prediction.

4.1 Analysis of Results

1. **Neural Network**: The Neural Network model outperformed other models, achieving the lowest RMSE and MAE, along with the highest R^2 score. This indicates that the Neural Network was able to capture complex, non-linear relationships in the data effectively.

2. **Random Forest**: The Random Forest model also performed well. Its robust handling of non-linear patterns and minimal need for feature scaling make it a reliable choice. However, it falls slightly short of the Neural Network in terms of accuracy.

3. **Gradient Boosting**: Gradient Boosting achieved moderate performance. While close to the Random Forest, additional hyperparameter tuning could further optimize its performance.

4. **Support Vector Machine (SVM)**: The SVM model demonstrated reasonable performance This suggests it captures some complexity in the data but would benefit from further hyperparameter adjustments (e.g., kernel choice, 'C', and 'gamma' parameters).

5. **Linear Models (Linear Regression, Ridge, Lasso)**: Linear Regression and its regularized versions (Ridge and Lasso) had the highest RMSE values and the lowest R^2 scores. These results imply that linear models are less effective for this task, likely due to the data's non-linear nature.

4.2 Feature Scaling

We analyzed the performance of various models with and without feature scaling. The results indicated a significant improvement in model performance when feature scaling was applied. Models trained without scaling showed a noticeable decrease in performance metrics such as R^2.

Figures 1a and 1b illustrate the performance comparison of the models. As shown in the images, feature scaling is crucial for achieving optimal model results.

Model Performance Comparison:

	RMSE	MAE	R^2
Random Forest	0.805900	0.581965	0.865842
Neural Network	0.808483	0.601139	0.864981
Gradient Boosting	0.818983	0.607462	0.861451
Ridge Regression	1.010260	0.779903	0.789176
Linear Regression	1.010295	0.779830	0.789162
Lasso Regression	1.022205	0.798299	0.784161
Support Vector Machine	1.649565	1.226731	0.437927

(a) Performance without Feature Scaling

Model Performance Comparison:

	RMSE	MAE	R^2
Neural Network	0.721189	0.536374	0.892564
Random Forest	0.804029	0.581054	0.866465
Gradient Boosting	0.818983	0.607462	0.861451
Support Vector Machine	0.879279	0.637119	0.840299
Linear Regression	1.010295	0.779830	0.789162
Ridge Regression	1.010388	0.779889	0.789123
Lasso Regression	1.025845	0.798885	0.782621

(b) Performance with Feature Scaling

Figure 1: Comparison of Model Performance with and without Feature Scaling

4.3 Conclusion and Next Steps

Based on these results, the Neural Network model appears to be the best option for solubility prediction, followed closely by the Random Forest model.

5 Hyperparameter Tuning

In the pursuit of optimizing model performance for solubility prediction, we employed both Grid Search and Randomized Search for hyperparameter tuning across two machine learning models: Random Forest and Gradient Boosting. Given the computational intensity of Grid Search, we transitioned to Randomized Search, which offers a more efficient way to explore the hyperparameter space.We also tuned the parameters for neural network by manaully experimenting and adding layers based on insight.

5.1 Results of Randomized Search

After fitting the models to the training data, the best parameters and corresponding Root Mean Squared Error (RMSE) for each model were obtained as follows:

- **Random Forest:**
  - Best Parameters: {'n_estimators': 100, 'min_samples_split': 2, 'max_depth': None}
  - Best RMSE: 0.7416
- **Gradient Boosting:**
  - Best Parameters: {'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.1}
  - Best RMSE: 0.7773
- **Neural Network:**

- Added Dropout Layers,early stopping and regularization.
- Chnaged learning rate ,epoch and batch size.
- Best RMSE: 0.7211

5.2 Final Evaluation Metrics

After hyperparameter tuning, we evaluated the performance of each model using the test dataset, yielding the following results:

- **Random Forest:** RMSE = 0.8040, MAE = 0.5811, $R^2 = 0.8665$
- **Gradient Boosting:** RMSE = 0.8002, MAE = 0.5927, $R^2 = 0.8677$
- **Neural Network:** RMSE = 0.7211, MAE = 0.5363, $R^2 = 0.8925$

5.3 Conclusion

Despite the effort invested in hyperparameter tuning using Randomized Search, the improvements in model metrics were modest. Each model showed comparable performance with RMSE values close to one another. The Neural Network emerged with the best R^2 score among the models, suggesting it captures the variance in the data slightly more effectively.

6 Outlier Detection

In the pursuit of improving model performance, we implemented an outlier detection strategy using the Isolation Forest algorithm from the `sklearn.ensemble` module. The purpose of this approach was to identify and remove anomalous data points that may adversely affect the training of our predictive models.

6.1 Isolation Forest Implementation

We configured the Isolation Forest with a contamination parameter set to 0.05, indicating that we expected approximately 5% of the training data to be outliers.

6.2 Training Data Shape

The shapes of the original and cleaned training datasets were as follows:

- Original training data shape: (915, 4)
- Cleaned training data shape: (869, 4)

This indicates that 46 samples were removed as outliers during the cleaning process.

6.3 Model Performance Comparison

After removing the outliers, we retrained the models and evaluated their performance using key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination R^2. The results are summarized in the table below:

Table 2: Model Performance After Outlier Removal

Model	**RMSE**	**MAE**	R^2
Neural Network	0.7295	0.5418	0.8900
Random Forest	0.8040	0.5811	0.8665
Gradient Boosting	0.8190	0.6075	0.8615
Support Vector Machine	0.8793	0.6371	0.8403
Linear Regression	1.0103	0.7798	0.7892
Ridge Regression	1.0104	0.7799	0.7891
Lasso Regression	1.0258	0.7989	0.7826

6.4 Conclusion

The implementation of the Isolation Forest for outlier detection has led to the removal of 46 anomalous samples, yet resulting in an overall decrement in model performance metrics. The Neural Network model achieved the best performance among the evaluated models, indicated by a lower RMSE and higher R^2 value.Except Neural network other models haven't been much affected by outlier detection.

7 Output Visualization

To analyze the performance of our predictive models, we performed the following visualizations:

- Predicted vs. Actual Solubility:
  - We plotted the predicted solubility against the actual solubility for each model.
  - The Neural Network model showed the best alignment with the $y = x$ line, indicating a strong correspondence between predicted and actual solubility values.
  - This alignment suggests that the Neural Network effectively captures the solubility trends present in the dataset.
- Feature Importance Analysis:
  - We examined feature importance for the Random Forest and Gradient Boosting models, which are suitable for this analysis.
  - The feature importance graphs revealed that one feature clearly dominates the others in its contribution to the predictions.

  - This information is critical for understanding the variables that significantly impact model performance.

- Insights Gained:
  - The visualizations not only demonstrate the efficacy of the Neural Network model but also highlight the critical features driving predictions in the Random Forest and Gradient Boosting models.
  - This dual approach enriches our understanding of the solubility prediction task and aids in further model refinement.

8 Learning and Limitations

We adhered to Sir's motive of the project to showcase what we learned in class .Thus, we worked our ways around insights based on what we learned and trained and selected models accordingly .**Our Group included "Something of Everything" learnt in the class and tutorials**.
Our major setback was the size of the dataset and the fact that we weren't able to get any detailed dataset.Having a smaller dataset restricted the model's capacity to generalize.Unfortunately, generating artificial data is not feasible in this case, as synthetic data may not accurately capture the underlying relationships in molecular properties that affect solubility, leading to unreliable model training and evaluation.

9 Conclusion

In this project,we successfully developed and evaluated multiple machine learning models to predict solubility (logS) based on molecular descriptors. The primary objectives were to identify the best-performing model, optimize its parameters, and understand the underlying factors influencing solubility predictions.

Key findings from the project include:

- Model Performance: The Neural Network model emerged as the top performer, demonstrating superior accuracy with the highest R^2 score, lowest Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The Random Forest and Gradient Boosting models also performed well, indicating the strength of ensemble methods in handling this prediction task.

- Parameter Tuning: Through the use of Randomized Search for hyperparameter tuning, we sought to enhance model performance. Despite significant efforts, the improvements in performance metrics were modest. The results suggest that further exploration of hyperparameter spaces, possibly through Bayesian optimization, may yield better outcomes.

- Outlier Detection: The implementation of Isolation Forest for outlier detection significantly improved the dataset's quality. Post-removal of outliers, we retrained our

models, leading to notable difference in predictive metrics across models, particularly for the Neural Network.Yet despite this, possibly due to lack of sufficient dataset there has been decline in the metrics.

- Feature Importance: Analysis of feature importance for the Random Forest and Gradient Boosting models revealed that certain features significantly influenced predictions. This insight not only informs model interpretation but also guides future feature engineering efforts.

- Visual Analysis: The visualizations of predicted versus actual solubility highlighted the effectiveness of the Neural Network, while feature importance plots underscored the dominance of specific features in driving predictions.

In conclusion, the project has established a robust framework for predicting solubility using machine learning techniques. The findings highlight the potential for further refinements, including deeper parameter tuning, advanced ensemble techniques, and enhanced feature engineering. Moving forward, these insights can inform ongoing research and application in predictive modeling within the field of cheminformatics.

10 Contributions

The contributions to the project are as follows:

10.1 Haaziq

- Contributed to the implementation of outlier detection and removal using Isolation Forest and compared model metrics after removing outliers.
- Conducted exploratory analysis of the data, providing insights into the underlying patterns and distributions, along with visualizations to support findings.

10.2 Sahil

- Contributed to model selection and evaluation for the Neural Network, Random Forest, and Support Vector Machine models, analyzing their performance metrics, such as RMSE, MAE, and R^2.
- Conducted experiments with layers of neural networks to find the best performing model.

10.3 Shubham

- Contributed to model selection and evaluation for the Gradient Boosting and Linear Regression models, analyzing their effectiveness in solubility prediction.
- Participated in hyperparameter tuning for all models, implementing Randomized Search to optimize model performance and analyzing the results of tuned models.

10.4 Yash

- Implemented the Artificial Neural Network model, focusing on its architecture and training process and conducted hyperparameter tuning for the Neural Network to enhance model performance.
- Contributed to the visualization of data, plotting predicted vs. actual solubility values and feature importance graphs for applicable models.

References

[1] `https://keras.io/api/`

[2] `https://scikit-learn.org/stable/user_guide.html`

[3] `https://ieeexplore.ieee.org/abstract/document/9640774`

[4] `https://jcheminf.biomedcentral.com/articles/10.1186/s13321-023-00752-6`

[5] `https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00575-3`