Grace Nye
Molecular Solubility in Water Prediction

Problem Statement:

When designing small molecule drugs, it is essential to understand their chemical properties, including their solubility in aqueous solution. This property is also important for chemists working in many other industries, including the agrochemical industry. Therefore, it is worthwhile to have a fast and simple method for estimating the aqueous solubility of a given small molecule. The General Solubility Equation (GSE) is able to provide an accurate description of solubility given the melting point (MP) of the molecule. If that data is not readily available, however, it is much harder to predict solubility. Existing datasets for training ML models for predicting solubility directly from chemical structures are small and highly variable. Here, using a dataset from the Cambridge Crystallographic Data Centre (CCDC) containing 58,000 molecules and their melting points, we train a random forest regressor to predict the melting point of a molecule with a mean average error (MAE) of 30.00, and then use the GSE to calculate the log of solubility with an MAE of 2.80.

$$logS = 0.5 - 0.01 * (MP - 25) - logP$$ **GSE**

Data Wrangling

There were eight datasets used in this study. Two were downloaded from the Cambridge Crystallographic Data Centre (CCDC): the first is a binary file containing RDKit radius 2, 1024 bits Morgan Fingerprints for each molecule; the second is a binary file containing RDKit 2D descriptors. The files were uploaded and formatted into dataframes. Both files contained 58810 rows, each containing information for one molecule, for a total of 58810 molecules in the dataset. The first dataframe had four rows and an index column with the CID of each molecule. The four rows contained the standard melting temperature of the molecule in Kelvin, the morgan fingerprint, and two columns indicating whether the researchers who compiled this dataset used placed it in their training or testing split. The second dataframe had 211 columns, which contain 209 RDKit molecular descriptors, the standard melting temperature, and whether the authors assigned that molecule to training or testing split. For both dataframes, the columns containing information on train/test splits were dropped. Then the dataframes were joined together on the CID of each molecule. There were no missing values in the dataset. After investigating all the columns, three columns were found to only contain zeroes, so these columns were dropped (SMR_VSA8, SlogP_VSA9, and fr_prisulfonamd). The final shape of the dataset was 206 columns, 58810 rows.

The last six datasets were used for external validation of the final model for predicting solubility. Dataset 1 contains 72 small molecule drugs approved during 2016 and 2020. Dataset2 contains 132 compounds compiled from a solubility challenge proposed by Llinas et al.[i] Dataset3 contains 148 compounds compiled by Ran & Yalkowsky[ii]. Dataset4 contains 900 compounds.[iii] Dataset5 contains 8613 compounds curated from AqSolDB.[iv] Finally, Dataset6 contains solubility of 21 proteolysis targeting chimeric (PROTAC) compounds.[v] These datasets

were downloaded from this github[1] and formatted as dataframes. The columns were renamed so that all contained the same naming scheme and the same columns (compound name, SMILES structure, and experimental log of solubility), then all were joined into one dataframe, with a column containing the name of the dataset that row originated from. The final shape of this dataframe is 9886 rows and 6 columns.

Exploratory Data Analysis

No clearly linear relationships with standard melting temperature were discovered at this stage. The feature of interest (standard melting point) has a fairly normal distribution in the dataset.

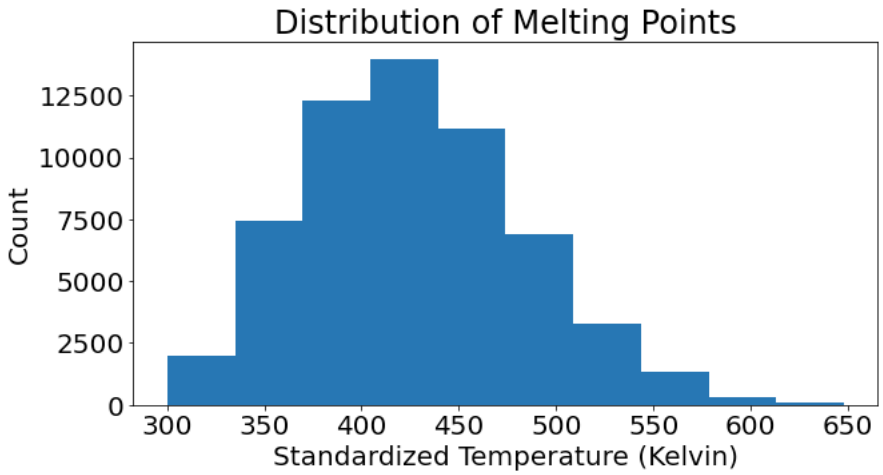
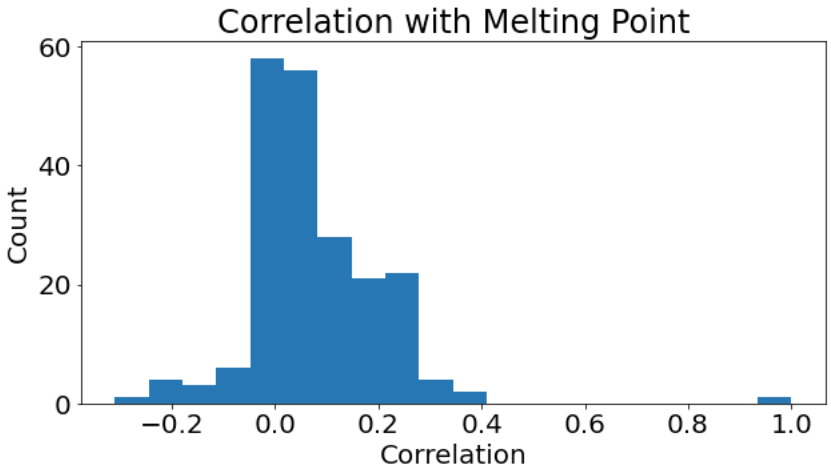


Figure 1: Distribution of melting points in CCDC dataset.

We found that few features had a strong correlation with melting point.



[1] https://github.com/sutropub/OpenSOL/tree/main/Test/clean

Figure 2: Correlation with melting point (calculated for all 206 molecular descriptors). Most features have a correlation between 0-0.2. The most highly correlated feature is the RingCount, which corresponds to the fact that rings tend to make molecules more stable, which leads to an increase in their melting point.

We are primarily interested in small molecules that are of the same size as might be used in drug discovery design. We divided the molecules into categories designed “small”, “medium”, and “large”. “Small” molecules were those having a molecular weight less than 200 amu. “Medium” molecules are those between 200-300 amu. “Large” molecules are those greater than 300 amu. There were 3660 molecules in the “small” set, 17475 in the “medium” set, and 37675 in the “large” set. Molecules typically used in drug discovery and agriculture fall into the “small” and “medium” categories.

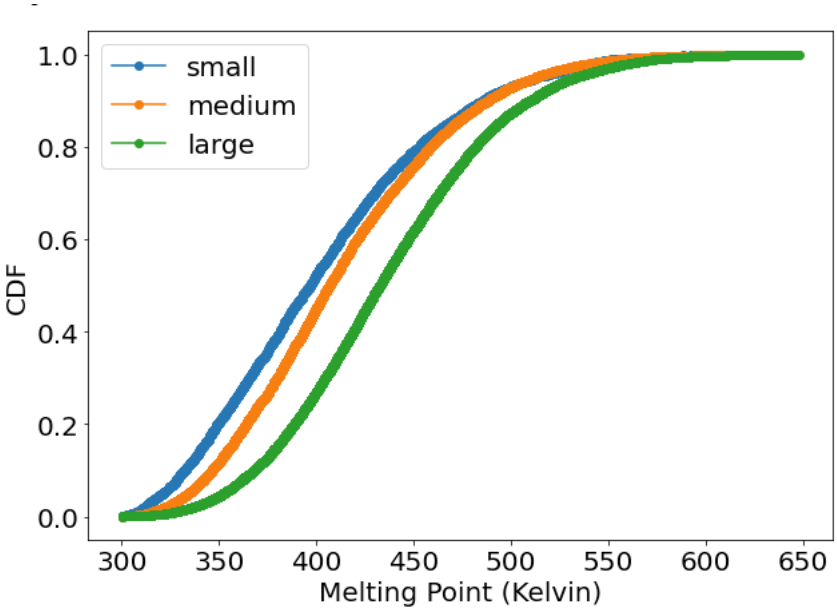


Figure 3: The cumulative distribution functions of the melting points of molecules with the three size groups. Small molecules are less than 200 amu, medium molecules are between 200-300 amu, and large molecules are over 300 amu. We can see that there appears to be a significant difference in the distribution of melting points for molecules with the “large” dataset.

I decided to use all 206 molecular descriptors in my final model, and to use a model that is able to handle features that do not have normal distributions are not linearly related.

For the six external validation datasets to be used to evaluate performance of the final model at predicting solubility, the molecular descriptors were generated using RDKit. By importing the module rdkit.Chem.Descriptors and running Descriptors.CalcMolDescriptors(Chem.MolFromSmiles(<smiles>)) the 2D molecular descriptors of interest were calculated and compiled in a dictionary. Only the same set of features that were available in the CCDC dataset were kept. Rows containing missing values were dropped. Final dataframe shape is 9871 rows and 212 columns.

Preprocessing

For preprocessing, the CCDC dataset was first split into training and test sets, using sklearn.model_selection.train_test_split using the default ratio of 0.25 for test set and 0.75 for train set, and a random split. Then the data was scaled in two different methods to see which performed better: sklearn.preprocessing.StandardScaler (standardizing the data so that it is centered around 0) and sklearn.preprocessing.PowerTransformer (which converts the data to a more Gaussian shape, since many of our features are not normally distributed). Scalers were fit only to the training data for the features, and then used to transform both training and testing datasets.

Modelling

First, in order to establish a baseline for model performance, a linear regression model was built. A linear relationship was not identified in exploratory data analysis, so I did not expect that this model would perform exceptionally well. Using sklearn.linear_model.LinearRegression, I fit three separate models to the unscaled data, standardized data, and log scaled data. After fitting to the training data, the models were then used to predict melting points for the test datasets. The mean absolute error (MAE) of each model was then calculated. Results are shown in **Figure 4**.

Next, a Random Forest Regressor model was tried, using sklearn.ensemble.RandomForestRegressor. Similar to the Linear Regression model, this model was first fit to unscaled data, standardized data, and log scaled data to compare performance. The three models were then used to predict melting points for the test sets, and mean absolute error was calculated. Results are shown below in **Figure 4**.

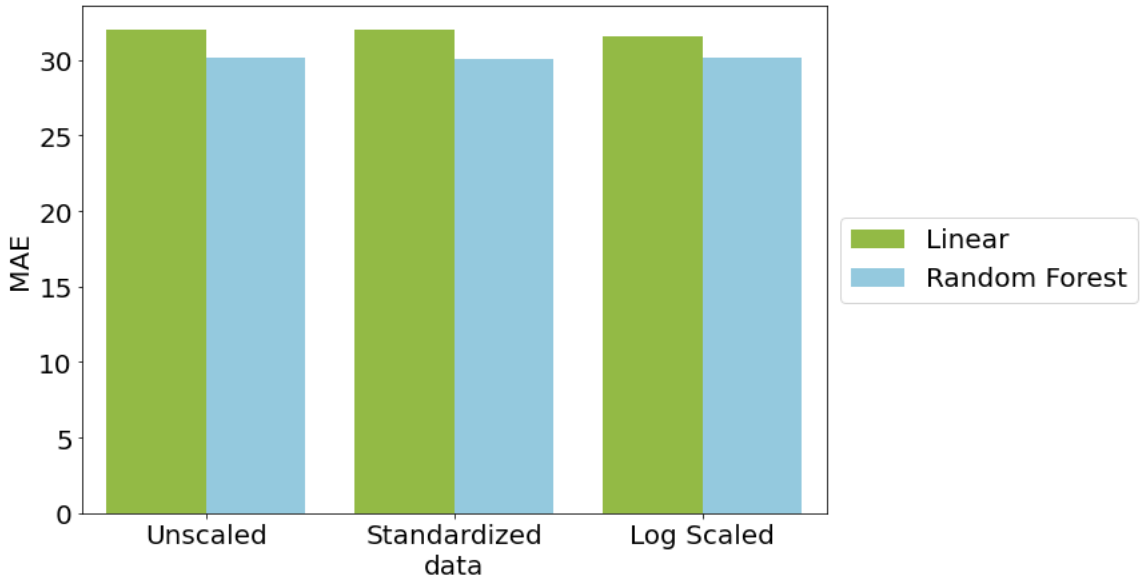


Figure 4: Mean Absolute Error for Linear Regression and Random Forest Regressor models tried on unscaled data, standardized data, and log scaled. Unexpectedly, the Linear Regression model

performs only slightly worse than the Random Forest Regressor. Additionally, it does not appear that scaling the data has any sort of significant impact on model performance.

Unexpectedly, the Random Forest Regressor did not have a significant advantage over the Linear Regression using these metrics. Additionally, it was noted that both these metrics are also comparable to the performance metrics of existing algorithms for predicting melting point data in the literature.

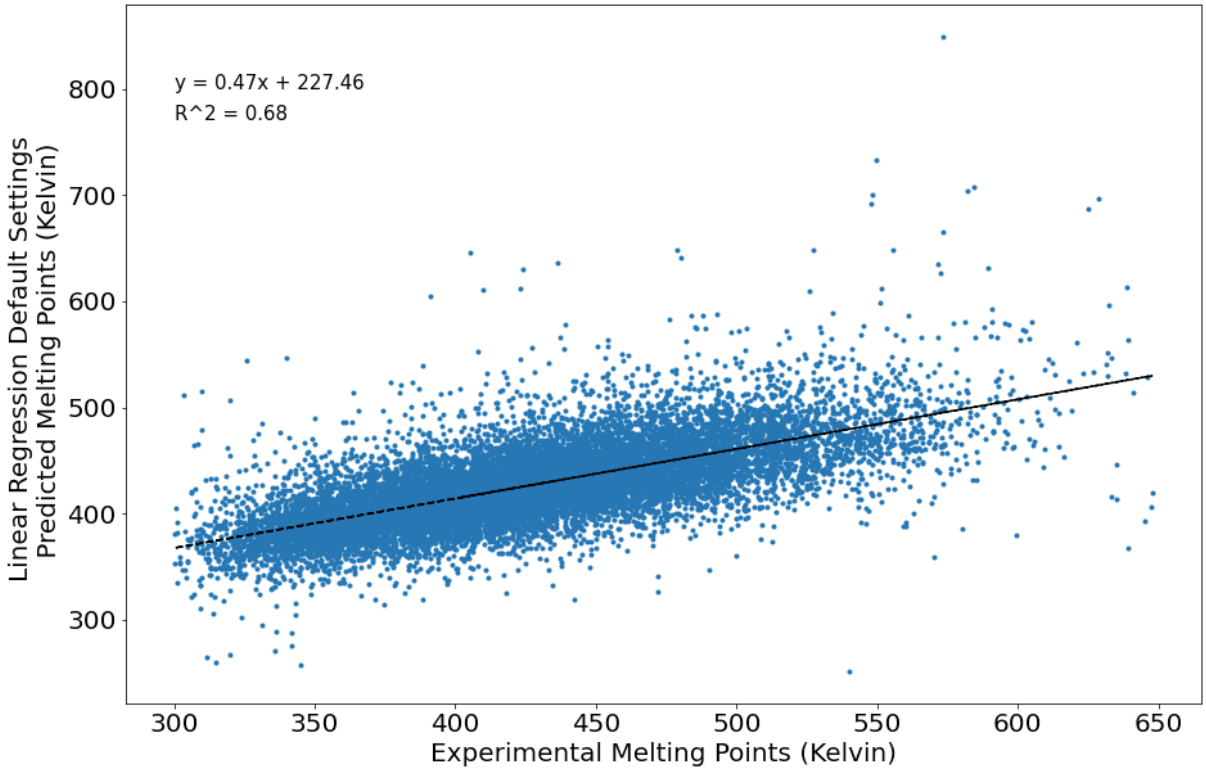


Figure 5: Experimental melting points in test set (Kelvin) vs. predicted melting points (Kelvin) for standardized data with Linear Regression model. The R^2 value = 0.68. RMSE = 41.45, MAE = 31.97.

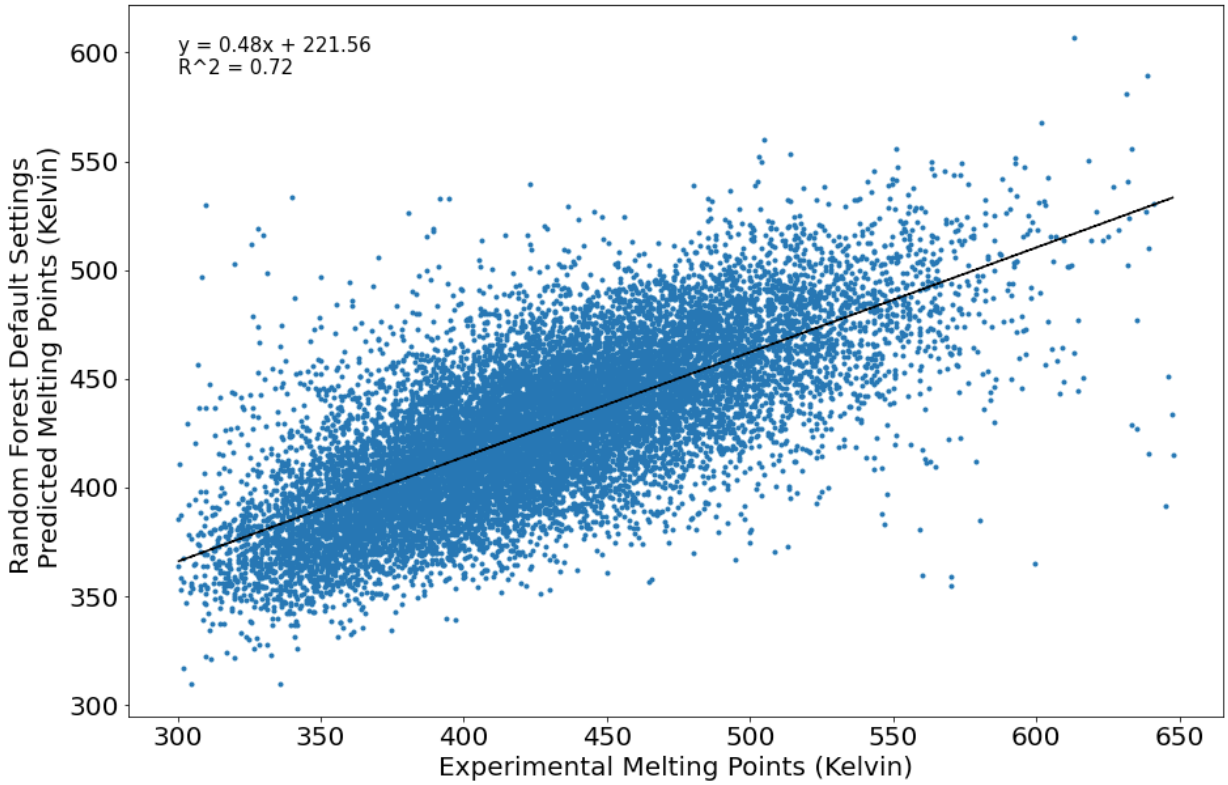


Figure 6. Experimental melting points in test set (Kelvin) vs. predicted melting points (Kelvin) for standardized data with Random Forest Regressor model. The R^2 value = 0.72. RMSE = 39.31, MAE = 30.08.

Although I expected the Linear Regression to be less successful than the Random Forest, I found that they had similar performance with these metrics, with Random Forest having a slight advantage. I decided to move forward with the Random Forest model and tune its parameters to improve performance. Using sklearn.model_selection.RandomizedSearchCV I scanned a variety of parameters and hyperparameters for the best performing model. I investigated the following parameters/hyperparameters:

- n_estimators (number of trees in the forest)
- max_depth (maximum depth of each tree)
- max_features (number of features to consider at every split)
- min_samples_split (minimum number of samples required to split a node)
- min_samples_leaf (minimum number of samples required at each leaf node)

By running RandomizedSearchCV with 100 iterations and a cross-validation strategy of 3 (to help decrease the amount of time the randomized search took to run), and n_jobs=-1 to help speed up the process as well. The randomized search found that the best performing model had the following parameters:

- n_estimators=1600
- min_samples_split=2
- min_samples_leaf=1

- max_features= ‘auto’
- max_depth = 30

The MAE of this model was 30.00, which is only a very minor improvement from the Random Forest model with default settings (30.08). This is a performance improvement of 0.27%.

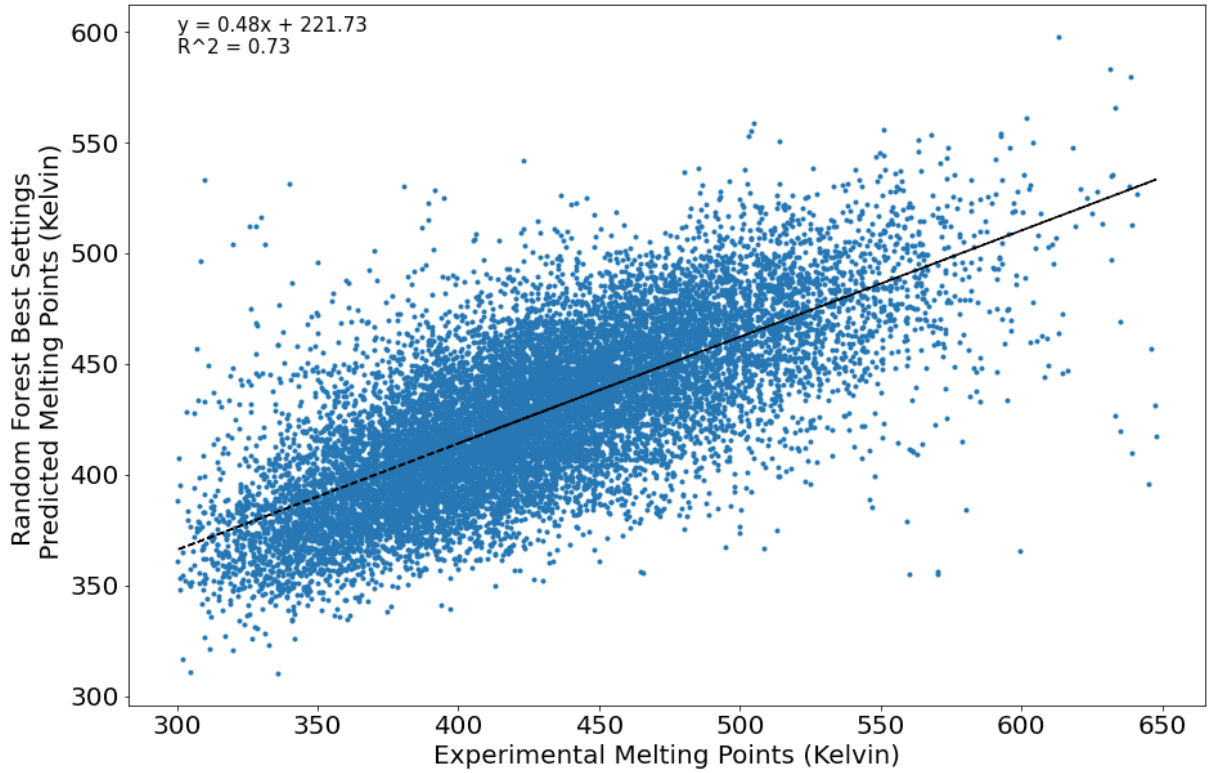


Figure 7: Experimental melting points in test set (Kelvin) vs. predicted melting points (Kelvin) for standardized data with Random Forest Regressor model. The R^2 value = 0.73. RMSE = 39.13, MAE = 30.00.

Using the six external datasets for solubility prediction, I used the best Random Forest Regressor to predict the melting points of each molecule. Then, using the General Solubility Equation (GSE) (Equation 1), I calculated the log solubility for each molecule. The average MAE = 2.79 and the average RMSE = 3.17.

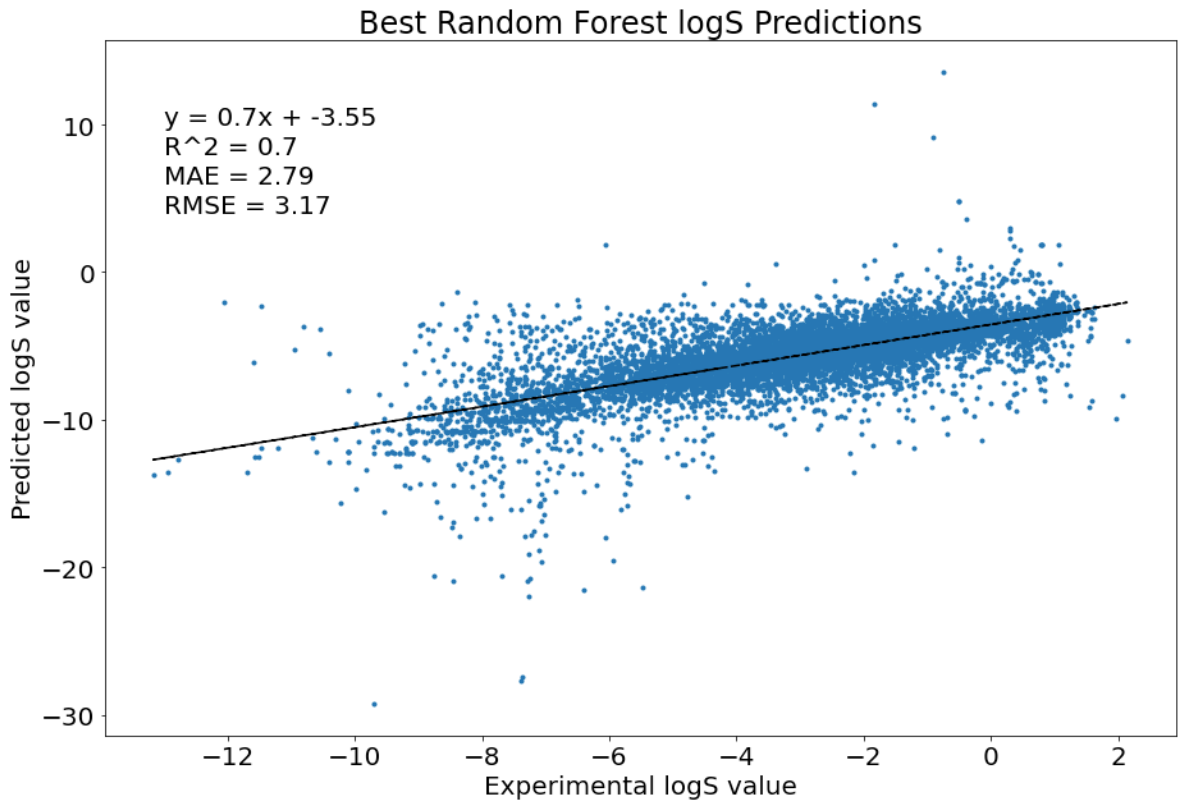


Figure 8: Scatterplots for the six external datasets used to validate the use of melting point predictors to calculate solubility.

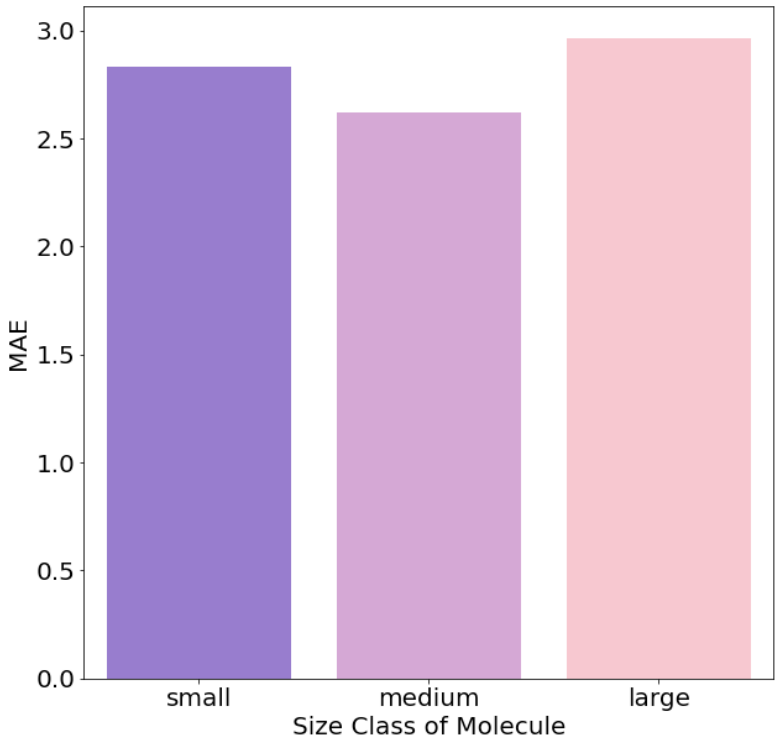


Figure 9: Barplot with MAE of logS for each size class. It appears that performance is slightly worse on larger molecules and is the best on medium sized molecules. This is favorable for our purposes, as we are mostly interested in small and medium sized molecules, for their pharmacokinetic usefulness.

Conclusion & Future Directions

I was able to use a melting point dataset to train a Random Forest Regression algorithm to predict the melting point of a molecule within 30.00 MAE, which is comparable to the performance of existing models in the literature.[vi] This model can then be used to calculate the solubility of molecules using the General Solubility Equation, with an MAE of 2.80.

Future directions include exploring using solubility datasets and molecular descriptors to directly predict solubility. As datasets for this purpose are much smaller and more variable this could be a difficult task. In general, the limiting factor of further work here is the lack of data. Experimentalists will need to compile a larger dataset of solubility before this work can be expanded significantly.

---

[i] Llinas, A.; Oprisiu, I.; Avdeef, A. Findings of the Second Challenge to Predict Aqueous Solubility. *J. Chem. Inf. Model.* **2020**, *60*, 4791– 4803, DOI: 10.1021/acs.jcim.0c00701

[ii] Ran, Y.; Yalkowsky, S. H. Prediction of Drug Solubility by the General Solubility Equation (GSE). *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 354– 357, DOI: 10.1021/ci000338c

[iii] Boobier, S.; Hose, D. R. J.; Blacker, A. J.; Nguyen, B. N. Machine Learning with Physicochemical Relationships: Solubility Prediction in Organic Solvents and Water. *Nat. Commun.* **2020**, *11*, 5753, DOI: 10.1038/s41467-020-19594-z

[iv] Sorkun, M. C.; Khetan, A.; Er, S. AqSolDB, a Curated Reference Set of Aqueous Solubility and 2D Descriptors for a Diverse Set of Compounds. *Sci. Data* **2019**, *6*, 143, DOI: 10.1038/s41597-019-0151-1

[v] García Jiménez, D.; Rossi Sebastiano, M.; Vallaro, M.; Mileo, V.; Pizzirani, D.; Moretti, E.; Ermondi, G.; Caron, G. Designing Soluble PROTACs: Strategies and Preliminary Guidelines. *J. Med. Chem.* **2022**, *65*, 12639, DOI: 10.1021/acs.jmedchem.2c00201

[vi] Zhu, X. *et al.* Building Machine Learning Small Molecule Melting Points and Solubility Models Using CCDC Melting Points Dataset. *J. Chem. Inf. Model.* **2023,** 63, 10, 2948-2959. https://doi.org/10.1021/acs.jcim.3c00308.