Polymer Properties Predictor

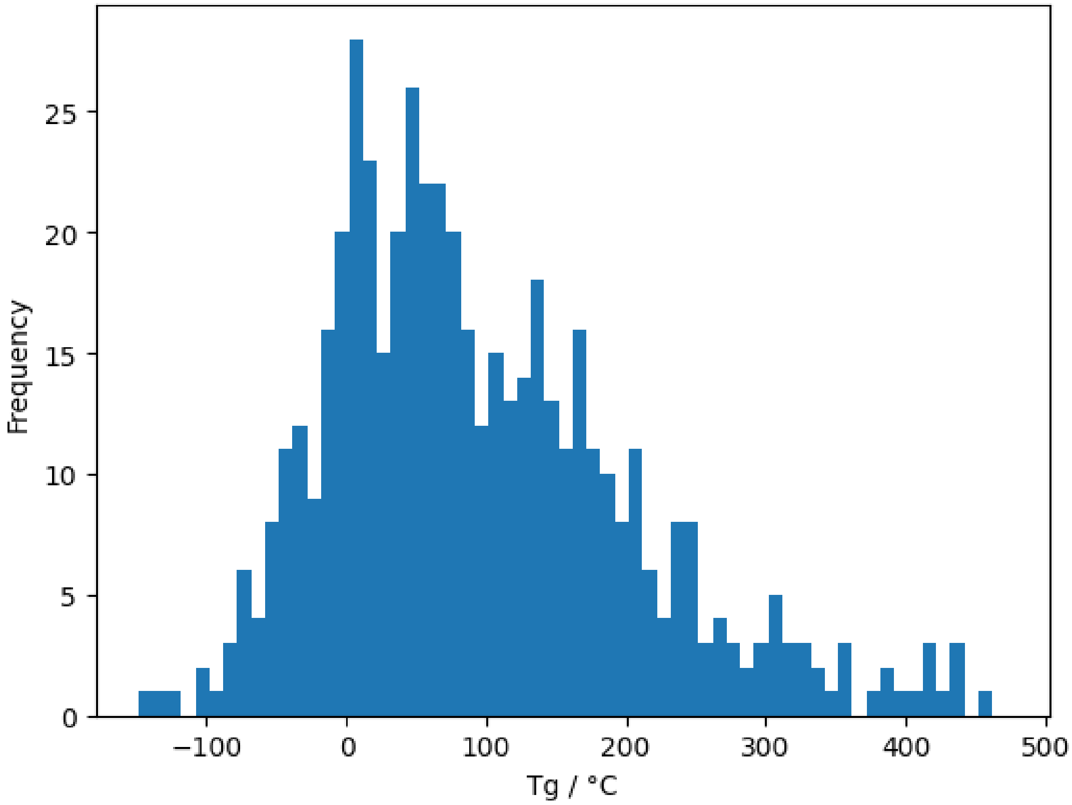
Prediction of 5 different properties of polymers using molecular structure and descriptors.

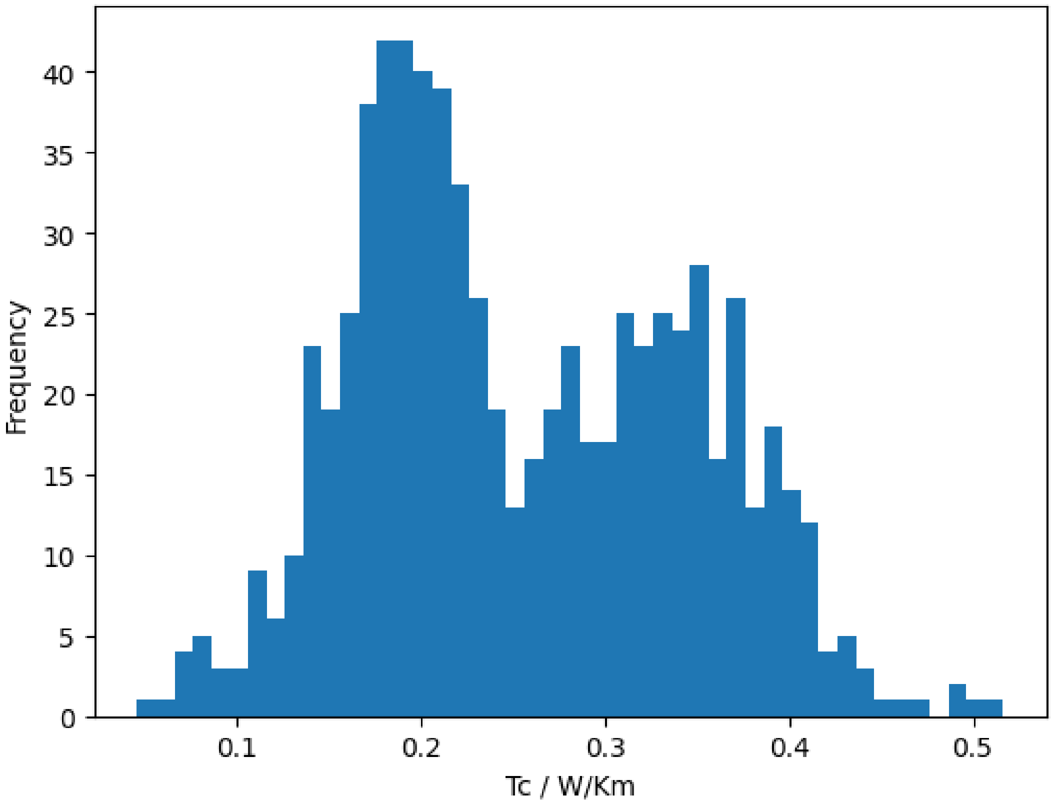
Learning Objectives

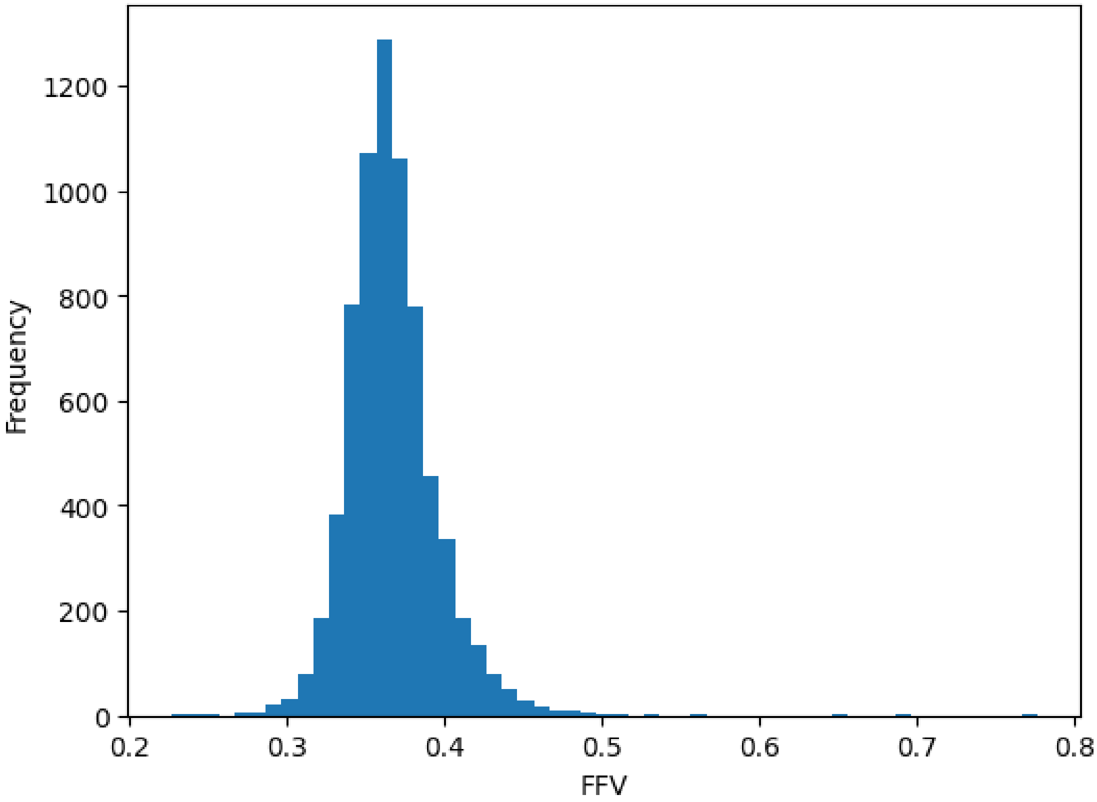
- Multi-Output Regression
- K-Fold Validation
- Ensemble Models

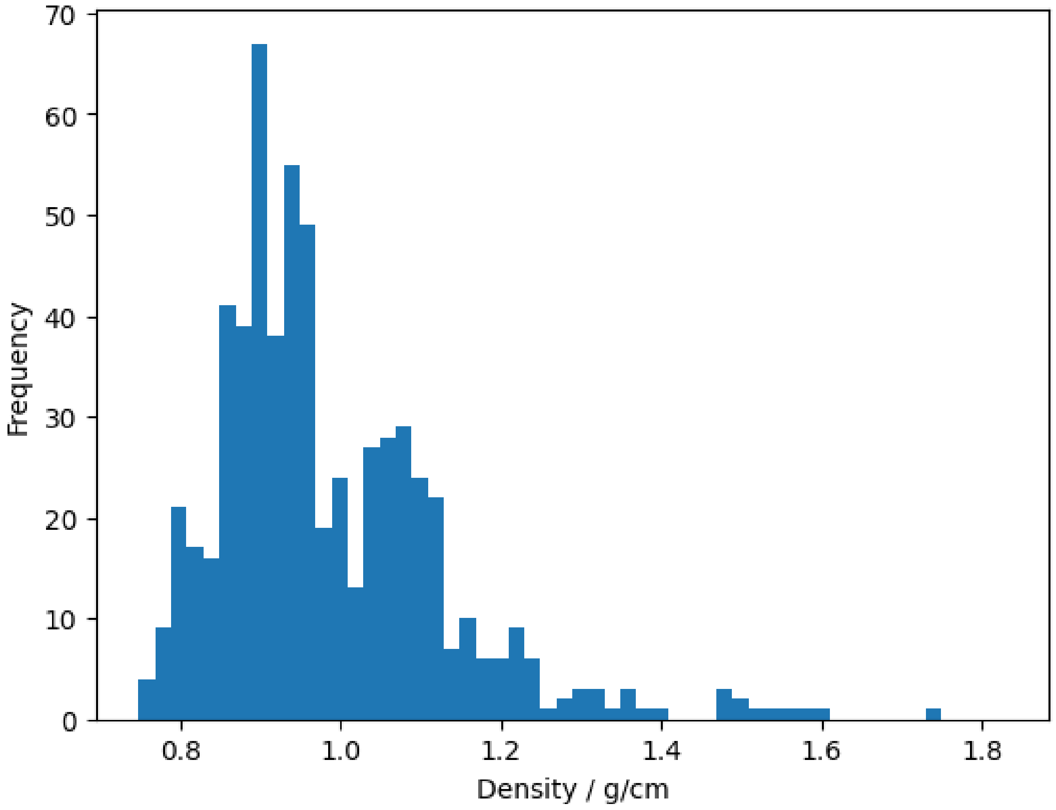
Visualising the Data

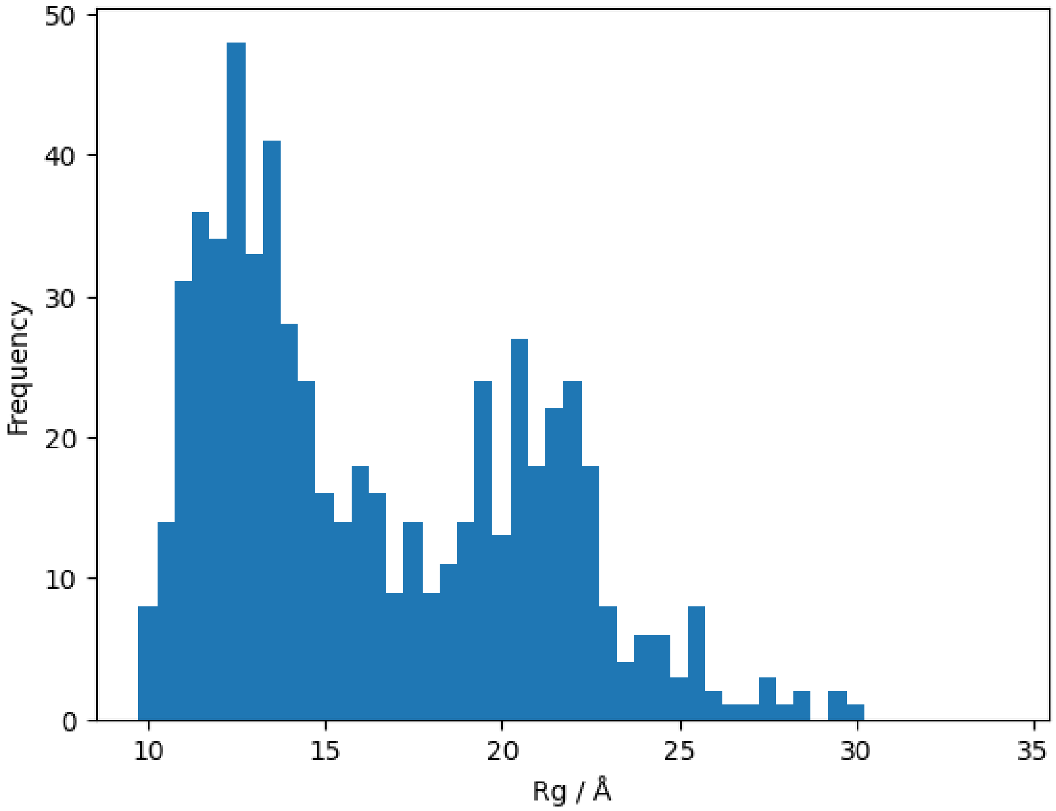
Five different properties and SMILES representations of over 7,500 diverse polymers were collected. These properties were the glass transition temperate (Tg), the fractional free volume (FFV), the thermal conductivity (Tc), the density, and the radius of gyration (Rg). The composition of the dataset is visualised below.

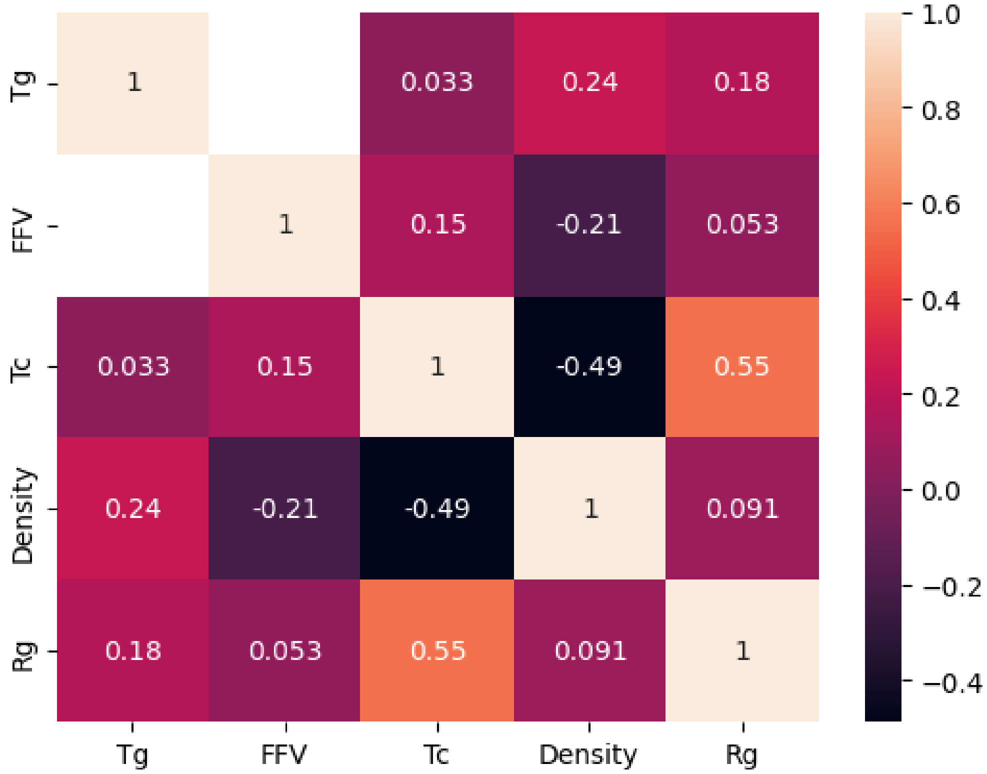



Frequency
0
5
10
15
20
25
30
35
40
0.1
0.2
0.3
0.4
0.5
Tc / W/Km


Frequency
0
200
400
600
800
1000
1200
0.2
0.3
0.4
0.5
0.6
0.7
0.8
FFV


Frequency
0
10
20
30
40
50
60
70
0.8
1.0
1.2
1.4
1.6
1.8
Density / g/cm


Frequency
0
10
20
30
40
50
10
15
20
25
30
35
Rg / Å



There is a strong positive correlation between the radius of gyration and the thermal conductivity of the polymer. Furthermore, there is a strong negative correlation between the density and the thermal conductivity.

There is also a weak positive correlation between the glass transition temperature and the density, as well as a weak negative correlation between the fractional free volume and the density.

These correlations are understandable from a chemical point of view and can be utilised by the multi-output regressors to provide more accurate predictions.

K-Fold Cross Validation

Over 200 numeric descriptors were calculated for each polymer, based on the molecular structure of their monomers.

K-fold cross validation is a statistical test used to ascertain the performance of a model. First, the dataset is divided into K equal subsets (or folds). Next, the model is trained on K – 1 folds and tested on the final fold. This is repeated K times, and the performance of the model is averaged across the K tests.

Using K-fold cross validation can improve the accuracy of the model's prediction. It reduces the variance in its performance by reducing the dependence on each given train / test split and thus reduces overfitting. This provides a reliable estimate of how well the model can generalise. Furthermore, hyperparameter tuning with K-fold cross validation ensures that the hyperparameters are not selected to optimise one specific validation step.

XGBoost regression models were trained using K-fold cross validation. For each iteration, the model was trained on K – 1 folds, using the final fold for validation, and then tested on the test data. At the end, the predictions for each property were averaged across the K runs. The aim of this method was to reduce overfitting.

This method was rerun using different values of K. The results are summarised in the table below.

Testing Data

Property	**Test Score (R2)**			
	1 Fold	**3 Folds**	**5 Folds**	**10 Folds**
FFV	0.5947	0.5774	0.5913	0.5976
Tg	0.5822	0.6254	0.6059	0.6025
Tc	0.6940	0.6852	0.6878	0.6903
Rg	0.5850	0.5967	0.5920	0.5901
Density	0.6866	0.6880	0.6807	0.6732

On the whole, the model's predictions are improved by using K-fold cross validation.

For most of the properties, the best predictions were achieved for K = 3. Large values of K reduce the accuracy of the results as small validation sets lead to noisy error estimates and thus produce a larger variation in the predictions across the iterations.

Bagging

Bagging is an ensemble meta-algorithm designed to increase the accuracy of models by reducing variance and overfitting. It achieves this by generating *N* new datasets by uniformly sampling the original dataset with replacement. This ensures that each dataset is independent. *N* copies of the same model are then trained on each of the new datasets and their predictions are averaged to give an aggregate result.

Bagging regressors were created using different numbers of XGBoost regressors. The results are summarised below.

Testing Data

Property	**Test Score (R2)**			
	1 Estimator	**5 Estimators**	**10 Estimators**	**20 Estimators**
FFV	0.5947	0.7461	0.8200	0.7346
Tg	0.5822	0.6029	0.5842	0.6092
Tc	0.6940	0.8045	0.8537	0.7541
Rg	0.5850	0.6508	0.6067	0.7241
Density	0.6866	0.7935	0.7572	0.7660

In general, increasing the number of estimators increases the accuracy of the ensemble's predictions, until it reaches a plateau. This is because bagging reduces the variance but not the bias.

Voting

Voting is another ensemble meta-algorithm that fits a set of base models and averages their predictions to produce a final result. Unlike bagging, voting permits the use of different base models, and each model is fitted on the whole training dataset.

Three voting base models were used: Linear Regressor, Random Forest Regressor, and K Neighbours Regressor. These were combines into a voting ensemble. The results are summarised in the table below.

Property	**Test Score (R2)**			
	Linear Reg	**RF Reg**	**K Neighbours**	**Voting**
FFV	0.7096	0.7364	0.2432	0.6672
Tg	-0.5200	0.6159	0.2677	0.4558
Tc	0.6200	0.7235	0.4874	0.6935
Rg	0.3157	0.6273	0.1277	0.5225
Density	0.3105	0.6283	0.3483	0.6885

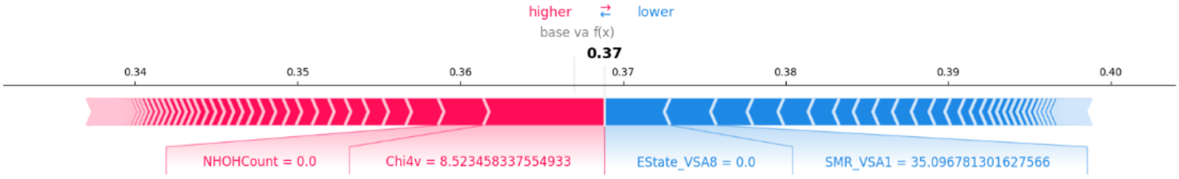
Voting incorporates the benefits and drawbacks of each base model. Thus, poor performing and unsuitable models (such as the linear regressor and K neighbours regressor) can detrimentally affect the ensemble's predictions. Here, the random forest regressor performs better by itself.

Model Insights

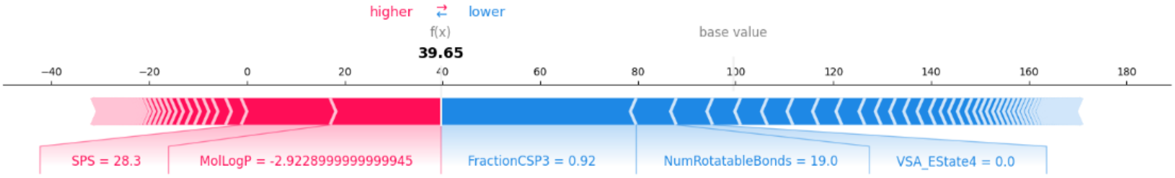
SHAP values were used to uncover insights into the trained XGBoost model.

The SHAP force plots demonstrate how each descriptor impacted the final model prediction, examples of which are given below for each property.

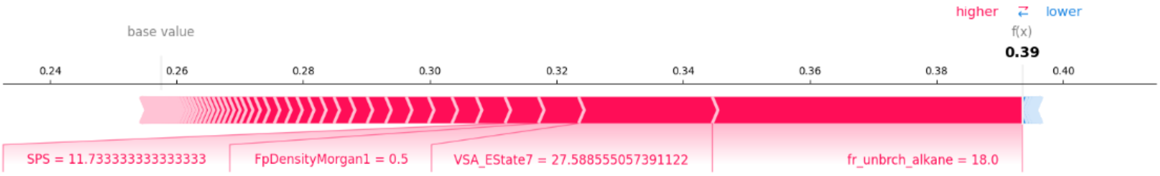
FFV



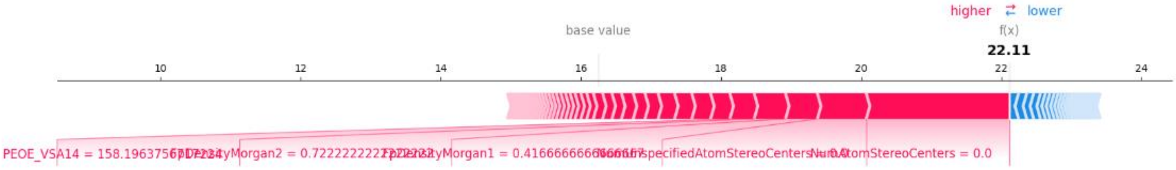
Tg



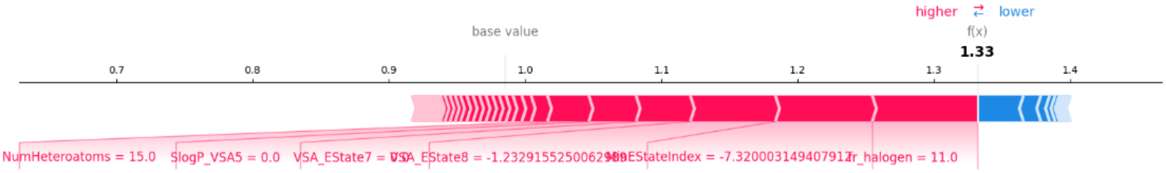
Tc



Rg



Density

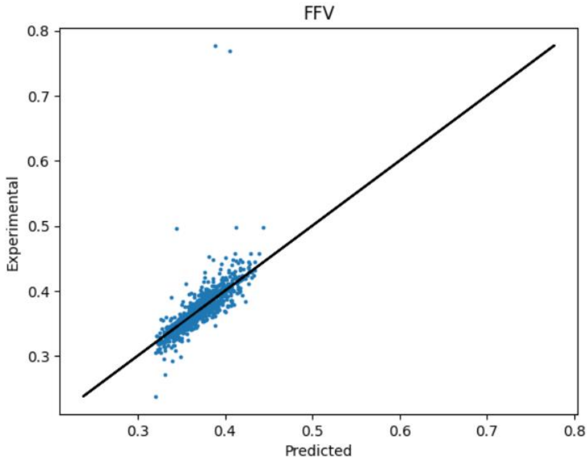


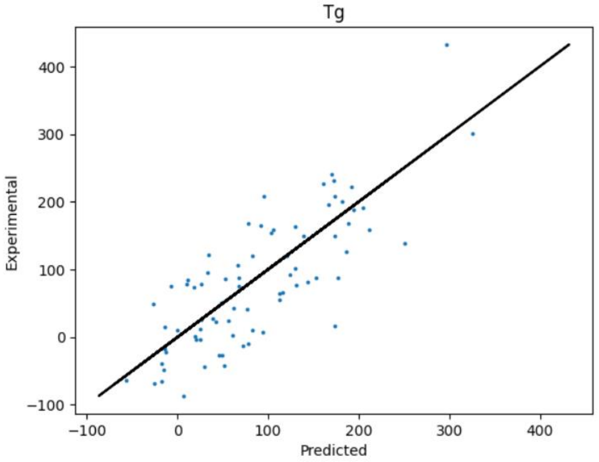
The most important descriptors identified by the SHAP values are difficult to interpret as they correspond to abstract chemical features of the molecules.

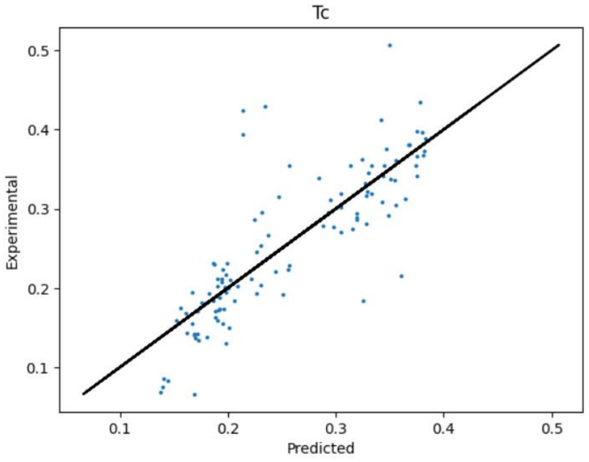
Conclusion

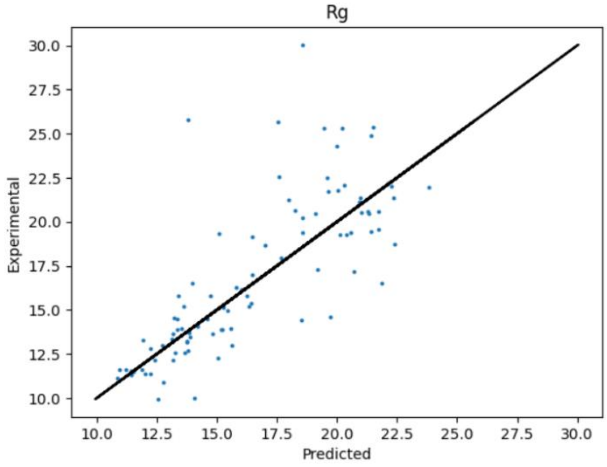
K-fold cross validation and ensemble models can be effective at improving prediction accuracy by reducing variance and overfitting. They achieve this by aggregating the results from multiple different models, averaging their errors.

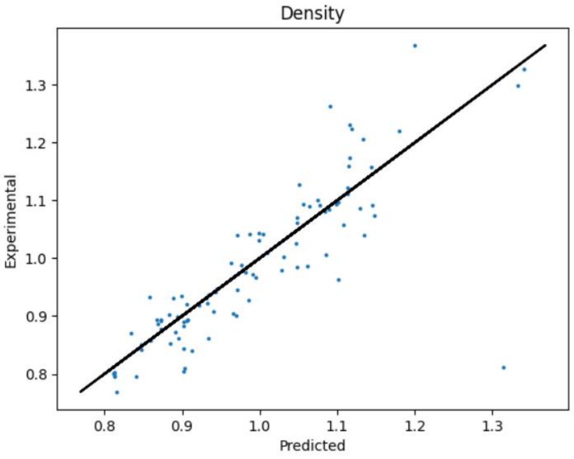
Here, bagging produced the most accurate predictions, whilst the use of unsuitable linear regression models made voting the worst performer.


FFV
Experimental
Predicted


Tg
Experimental
Predicted


Tc
Experimental
Predicted


Rg
Experimental
Predicted


Density
Experimental
Predicted