The Implementation of Machine Learning to Predict the Solubility of Simple Organic Molecules

Ryan O'Dwyer
Supervisor: Dr. Styliani Consta, Department of Chemistry, Western University
Submitted March 31, 2020

Table of Contents

Abstract

Solubility is an important factor that is used to describe and predict the way that molecules interact with their environment. Solubility plays a crucial role in the environment and in life, such as in the presence of salts and minerals, as well as in the prediction, engineering and discovery of protein and drug structures. Due to the significance of the problem there is much research done to understand the connection between molecular structure and solubility through modern machine learning programs. In this project the effects of various structural features on solubility for small organic molecules are studied. The features considered are the number of carbon, oxygen, and hydrogen atoms, the number of hydroxyl, ether and carbonyl groups, the units of unsaturation, the number of carbon-carbon double bonds, the presence of a ring and the length of the longest chain within the molecule. The interactions between the features is examined through Linear regression, Ridge regression and Lasso. The effects of the presence or absence of features on the accuracy of the solubility prediction are examined. The Lasso models were very useful in eliminating the features negligible to the calculation of solubility, and the most useful model produced was a Linear regression model which employed the number of hydrogen atoms, the number of hydroxyl, carbonyl and ether groups, and the presence of a ring as features.

List of Figures

List of Abbreviations

AI:	Artificial Intelligence
CV:	Cross Validation
GSE:	General Solubility Equation
Lasso:	Least Absolute Shrinkage and Selection Operator
LR:	Linear Regression
ML:	Machine Learning
QSPR:	Quantitative Structure-Property Relationship
R_g:	Radius of Gyration
RMSE:	Root Mean Squared Error
RSS:	Residual Sum of Squares
RR:	Ridge Regression
R^2_a:	Adjusted R^2 value
SE:	Standard Error
#C:	Number of Carbon atoms
#H:	Number of Hydrogen atoms
#O:	Number of Oxygen atoms
UUS:	Units of Unsaturation
#OH:	Number of Hydroxyl groups
#CO:	Number of Carbonyl groups
#CC:	Number of Carbon-Carbon double bonds
#R:	Presence of a Ring
#COC:	Number of Ether groups
LC:	Length of the longest Chain
S:	Solubility parameter

Acknowledgements

Thank you to my supervisor Styliani Consta, as well as Victor Kwan for all the help and guidance during this project. Further thanks are extended to my peers in the Consta group, as well as those organizing the Chem 4491E course, and to Elisa Dong for help with statistics and coding.

1.0 Introduction

Solubility is a key property in many fields such as in the design and study of drugs, proteins and enzymes. Solubility is strongly related to protein expression, and there are a number of diseases associated with poor solubility, such as Alzheimer's disease.[1] In order to predict the bioavailability of a drug, that is by what means and to what extent a drug is absorbed by the body, many researchers have attempted to predict solubility based on intrinsic and experimental properties.[2]

There are a number of equations proposed over the years relating solubility to molecular properties. The General Solubility Equation (GSE), developed by Yalkowsky and Valvani, predicts the solubility of a molecules from its melting temperature and its partition coefficient in octanol.[3] The GSE is a relatively accurate model, although it is not usually applicable in the field of drug development due to its dependence on experimentally produced melting point data, which often doesn't exist for drug candidates in development.[4] In lieu of experimentally based data, solubility is often predicted through analysis of quantitative structure-property relationships (QSPRs) which take into account structural or structurally derived properties.[2] A QSPR developed by Delaney et al. determined that solubility can be predicted with an accuracy comparable to the GSE, taking into account the partition coefficient, molecular weight, number of rotatable bonds and the proportion of heavy atoms in aromatic rings.[3] The solubility of a given molecule in a given solvent depends on a wide array of factors, and it is not always clear which factors will play key roles in affecting solubility for individual molecules. Nevertheless, the prediction of solubility based on independent key groups is useful for most predictive purposes, such as usage as lead optimization in drug development.[5]

The next few paragraphs deal with the basic concepts of Machine Learning (ML). ML is a promising method of analysis for data of the relationship between structure and solubility, and its output can be used in conjunction with more traditional methods such as rational design and directed evolution.[6] ML is a subdiscipline of the field of Artificial Intelligence (AI), first developed in the 1950s, whereby computers learn and adapt from acquired data by incorporating it into statistical models. With ML, the computer infers its own rules of operation with algorithms, which offers a simpler and faster alternative for AI learning in comparison to earlier learning systems, which employed human-coded logic-based rules for all possible outcomes. There have been three major periods of growth for ML in the past. ML was first conceived in the 1950s along with the perceptron, which is the simplest binary unit within a neural network. In the 1980s, the development of back propagation and slow learning allowed ML to be more efficient, and to resemble the function of actual neurons. In the 2010s, the enormous amount of data available online, termed "Big Data", has allowed ML to be applied to many fields by both corporations and researchers. There are many modern uses of ML, which range from automation, to speech and image recognition.[7]

In the context of this paper, ML is best defined as the process of acquiring, organizing and refining complex knowledge, and using algorithms to make predictions based on the provided examples, referred to as training sets. Specifically, inductive ML algorithms can detect the trends of data within a system in order to produce a desired output.[8] Computational methods are used to map the provided input variables to the desired output variables. As this map is based entirely on the provided training sets, it is important that these sets are both appropriate and relevant to the object of study. The maps produce from ML algorithms will reflect the inherent biases of the data in the training sets, so these biases must be taken into account for any ML system in use.[9] In order to extrapolate to a prediction outside of the domain designated by the training set, a process of active learning must be employed, as described in Podryabinkin et al.[10] Thus, while ML has the potential to greatly decrease the computational cost of calculating many variables and is a highly generalizable process, its accuracy depends heavily on the quality of the data provided and the efficacy of the chosen algorithm.[6]

There are many different types of learning employed by ML algorithms, which are employed for a range of diverse situations and data sets. The programs in this work employ a supervised learning program, which labels the training set data with the desired responses and uses them to predict a response for future inputs. This is opposed to an unsupervised learning program, for which there is no output associated with the training sets, and which will instead summarize and find relationships between the input variables. The programs used here are considered passive learners and not active learners, because the programs observe the data without asking the user for feedback or clarifications to assist its learning. The process used also reflects online learning as opposed to batch learning, as predictions are made throughout the learning process of developing an appropriate algorithm to be used, instead of making predictions exclusively at the end after all the training has been completed.[11]

The purpose of this work is to build a model through machine learning that will explore the relationship between various structural features and the solubility parameter (S) of a molecule. The program will employ a Linear regression (LR),[12] Ridge regression (RR)[12] or Lasso[12] model, and will take an array of structural features for each molecule as the input in order to output a value of S. To better understand the connections between these features and S, the scope of this work will be limited to simple organic molecules, not containing more than six carbon atoms and containing no atoms other than carbon, oxygen and hydrogen. The program will not employ active learning and is intended to be used only for molecules whose structure matches the same limitations as those imposed on the data set.

In addition to the work on solubility, I also examined the use of ML to predict the crystal structures of molecules. A crystal structure program would have taken similar inputs as the solubility program and would have outputted the structure and associated Steinhardt order parameters[13] of a given molecular crystal. The focus of this work was shifted to solubility in order to make better use of the limited timeframe to produce a more robust set of ML models.

2.0 Methodology

2.1 Theory of Statistical Learning

The process of predicting a response or output from input variables can be modelled with the following equation

$$Y = f(\boldsymbol{X}) + \epsilon \qquad (1)$$

where Y is the output and **X** is the input, and **X** = (x_1, x_2, … x_n). f(**X**) is a fixed but unknown function of **X**, and ε is a random error function, independent of **X**.[14] Since neither Y nor f(**X**) is known, both are approximated with functions estimating their values, being $\hat{Y}$ and $\hat{f}(\boldsymbol{X})$,[14] as shown in the following equation

$$\hat{Y} = \hat{f}(\boldsymbol{X}). \qquad (2)$$

There are two types of error associated with $\hat{Y}$: The reducible error and the irreducible error. The reducible error refers to the error in $\hat{f}(\boldsymbol{X})$, while the irreducible error refers to the error in ε. Statistical modelling can improve the accuracy of $\hat{f}(\boldsymbol{X})$ to more closely align it to $f(\boldsymbol{X})$, but ε represents the unaccounted-for variables and unmeasurable variation, and it can only be improved through improving the data set used. The connection between input and output variables can be used for prediction and inference. Prediction refers to calculating the output Y from a given input **X**, while inference refers to understanding the relationship between **X** and Y, and understanding which predictors are most important with respect to the response.[14] Both applications will be explored in this work.

2.1.1 Linear Regression

The function $\hat{f}(\boldsymbol{X})$ can be predicted either parametrically or non-parametrically.[14] For more details, see Appendix 1. $\hat{f}(\boldsymbol{X})$ is expressed parametrically as

$$Y \approx \hat{f}(\boldsymbol{X}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{3}$$

where x_i refers to a series of variables and β_i refers to their associated coefficients for a Linear regression (LR) model of a system with n input variables.[14] β_0 can be interpreted as the intercept, while the β_i coefficients are the slope of the line of best fit between each variable and $\hat{f}(\boldsymbol{X})$.[14] Eq. 3 can be simplified to

$$\hat{f}(\boldsymbol{X}) = \beta_0 + \textstyle\sum_{i=1}^{n} \boldsymbol{X}_i \hat{\beta}_i. \tag{4}$$

A simple example of $\hat{f}(\boldsymbol{X})$ for a model only incorporating two features, being the number of carbon and hydrogen atoms, denoted #C and #H respectively, is shown in the following equation along with arbitrary coefficient values

$$\hat{f}(\boldsymbol{X}) = 1.5 + 2.0(\#C) + 1.0(\#H). \tag{5}$$

Using Eq. 5, the value of $\hat{f}(\boldsymbol{X})$ can be calculated for any molecule given the values of #C and #H of that molecule. The goal of this work is to produce an equation similar to Eq. 5, but for a wider range of structural features related to S.

LR can predict a model by minimizing the Residual Sum of Squares (RSS), expressed as

$$RSS(\beta_i) = \textstyle\sum_{j=1}^{N} (y_j - \hat{f}(\boldsymbol{X}_j))^2. \tag{6}$$

Eq. 6 is minimized by selecting the optimal value of the coefficient β_i for each feature x_i for a total of N data points.[14] By combining Eq. 4 and Eq. 6 the following equation is produced

$$RSS(\beta_i) = \textstyle\sum_{j=1}^{N} (y_j - \beta_0 - \sum_{i=1}^{n} x_{ij}\beta_i)^2. \tag{7}$$

Eq. 7 signifies the exact RSS equation used in this study.[14] Thus, the LR model of this study estimates values based on ordinary least squares, which uses the RSS in Eq. 7.

Linear models of regression are the most widely applicable model for problems like the one explored in this work, but there are a number of potential problems with it, discussed in detail in Appendix 1. For a dataset with many parameters as the one used in this work, LR has a high risk of error due to collinearity. Collinearity describes a situation in which different features are correlated and related to each other, making it difficult to understand their individual effects on the response.[14] Within this study, the level of collinearity between the features was examined by comparing the correlation values between each feature.

2.1.2 Ridge Regression and Lasso

RR is a shrinkage ML method, used to reduce the value of the feature coefficients in order to decrease the variance of a model, which is often high in multiple linear regression models like the one examined in this study. Collinearity in a linear model can sometimes result in disproportionally large coefficient values for LR models, and RR was developed to increase the interpretability and improve inference for regression models. Instead of minimizing the RSS as is done with simple linear regression, RR minimizes a penalized RSS, where the parameter λ is included to determine the amount of shrinkage intended.[14] The penalized RRS of RR is expressed by the following equation

$$RSS_{Ridge}(\beta_i) = \sum_{j=1}^{N}(y_j - \beta_0 - \sum_{i=1}^{n} x_{ij}\beta_i)^2 + \lambda \sum_{i=1}^{n} {\beta_i}^2. \tag{8}$$

The penalized RSS in Eq. 8 includes λ in the penalty factor, with the constraint of $\lambda \geq 0$.[14] As λ increases, the calculated coefficients will decrease, thus alleviating problems associated with collinearity. If two features are highly correlated, then a LR model could produce a disproportionally large positive coefficient on one variable in pair with an equally disproportionally large negative coefficient on the variable highly correlated with the former, which greatly decreases the interpretability of the model. The penalty on coefficient size imposed by RR can be used to solve this potential problem in normal LR, although the nature of the RR penalty factor means that while the value of the coefficients will approach 0, they will never reach 0 and thus no features are eliminated from the model by RR. Furthermore, the coefficient β_0, representing the intercept of the equation, is not penalized, as it is not at risk of the same problems associated with collinearity as the other coefficients are. However, as RR is unable to completely eliminate redundant or negligible features, a model such as Lasso can be better suited for a dataset such as the one used in this work.[12]

Lasso, an acronym for least absolute shrinkage operator, is a shrinkage method similar to RR, but with a differently formulated error term.[14] The penalized RRS of Lasso is expressed by the following equation

$$RSS_{Lasso}(\beta_i) = \frac{1}{2}\sum_{j=1}^{N}(y_j - \beta_0 - \sum_{i=1}^{n} x_{ij}\beta_i)^2 + \lambda \sum_{i=1}^{n}|\beta_i|. \quad (9)$$

The penalized RSS in Eq. 9 includes λ as the penalty factor in the same way as for RR.[14] The difference between the RR penalty factor and the Lasso penalty factor is that the latter has the important consequence of allowing the value of feature coefficients in Lasso to be minimized to 0. Thus, a Lasso model is capable of eliminating irrelevant or redundant features from the model by reducing their coefficients to 0, which RR cannot do. Similar to RR, Lasso is used to account for collinearity in a model and is used to decrease the variance of a model.[12]

2.1.3 Model Accuracies

In order to determine the accuracy of the models, the data set was randomly split into a training set and a test set. 80% of the data was used as the training set, on which regression was performed and a model was produced, while the remaining 20% of the data was used as the test set, and the values of S for this set were compared against the predicted values of S to determine the accuracy of each model. The split between training and test sets was randomly chosen, and this split was identical for each LR, RR and Lasso program for each feature set in order to more easily compare the models. While there are alternative methods of testing model accuracy, such as *k*-fold cross validation (discussed in Appendix 1), the data was split 80/20 due to time constraints. As the training/test split was only performed once in this study, there is a relatively high level of bias associated with the produced models, but a relatively low level of variance.

To more easily compare the relative accuracies of the models used, the programs include in their output the R^2, the adjusted R^2 values as well as the root mean squared error (RMSE), which takes the same units as the output. The adjusted R^2 value is an important metric by which the programs are compared, as it accounts for the effectiveness of the different features used. The R^2 value often increases as the number of features increases, regardless of their correlation with the output, while the adjusted R^2 value will increase only as the number of non-negligible features increases and will decrease as the number of negligible features increases.[14] As the adjusted R^2 value more closely represents the effectiveness of the features used, it was deemed a more appropriate measure of accuracy than the R^2 value when comparing the different models. The formula for the adjusted R^2 value, designated as $R^2{}_a$, is shown in the following equation

$$R^2{}_a = 1 - (1 - R^2)(N - 1)/(N - p - 1). \quad (10)$$

The number of features and data points are designated as p and N respectively.[15]

To predict S within a 95% confidence interval, the following equation is used to determine the error associated with the final calculated value of S

$$SE = {}^{s}\!/_{\sqrt{N}}. \tag{11}$$

Eq. 11 represents the standard error (SE)[16] of an estimate, and 95% of predictions fall within ± 2 SE values.[17] *s* represents the standard deviation of the S test set data, being 3.6, and N represents the number of data points in the test set, being 25. For the test set used in this work, SE = 0.72, and 2*SE = 1.4, with SE taking the same unit as S.

2.2 Structural Features

The structural features studied in this work are the number of carbon atoms (#C), the number of hydrogen atoms (#H), the number of oxygen atoms (#O), the units of unsaturation (UUS), the number of hydroxyl groups (#OH), the number of carbonyl groups (#CO), the number of carbon-carbon double bonds (#CC), the presence of a ring (#R), the number of ether groups (#COC) and the length of the longest chain (LC). Two molecules from the dataset are shown in Figures 1 and 2, with their feature values indicated.

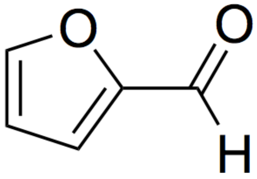

Figure 1: *2-Furaldehyde, a molecule for which #C=5, #H=4, #O=2, UUS=4, #OH=0, #CO=1, #CC=2, #R=1, #COC=1, LC=5.*

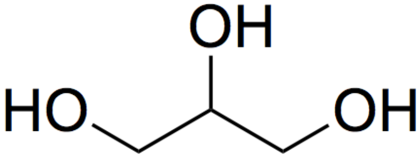

Figure 2: *Glycerol, a molecule for which #C=3, #H=8, #O=3, UUS=0, #OH=3, #CO=0, #CC=0, #R=0, #COC=0, LC=5.*

The features chosen for a QSPR equation such as the one produced in this work should be directly calculable from the molecular structure. Some typical descriptors used in similar works are the molecular weight; the solvent-accessible surface area; the number of potential hydrogen bond acceptors and donors; the number of specific functional groups; the number of rotatable bonds; and electrostatic potential data.[2]

The molecular weight is represented by the features #C, #H and #O. The molecular weight can be directly calculated from the numbers of each atom in its molecular structure, and the only atoms present in the dataset were carbon, hydrogen and oxygen.

The solvent-accessible surface area and number of rotatable bonds is represented most strongly by the LC feature, which counts the number of carbon and oxygen atoms in the longest chain, or the number of atoms in a ring for those molecules with rings. LC can also be used to differentiate between structural isomers by roughly corresponding to the surface area of a molecule. There are other features that could be used to differentiate between structural isomers such as the radius of gyration (R_g), however R_g depends on the orientation of the molecule and thus one can measure many different values for larger molecules, making it difficult to use in this simple model. Furthermore, it is also much less simple to calculate the R_g of a molecule than to count its LC, making LC a more widely accessible and easily understood feature.

The only potential hydrogen bond donors in the dataset are hydroxyl groups, represented by #OH, while the hydrogen bond acceptors are accounted for by #OH, #CO and #COC. Oxygen is the only source of potential hydrogen bonds in the dataset, and it is only present in these three functional groups.

Sources of unsaturation, although not a typical feature analyzed in solubility QSPRs,[2] is examined in this work through the features UUS, #CC, #CO and #R. Each carbon-carbon double bond, each carbonyl group and each ring represents one unit or degree of unsaturation, and UUS can also be calculated directly from the molecular formula through the following equation

$$UUS = (2 \times \#C - \#H + 2)/2. \qquad (12)$$

All the features examined are quantitative, however due to the nature of the dataset #R functions as a classification predictor. No molecule has more than one ring in the dataset, so #R represents a binary value, with a value of 1 indicating the presence of a ring and a value of 0 representing its absence. This is the appropriate handling of qualitative features in regression problems.[14]

2.3 System Details, Programs and Datasets

The full dataset, containing a total of 123 molecules, is included in Appendix 3, the data for which was taken from *Organic Solvents: Physical Properties and Methods of Purification.*[18] The molecules in this dataset were all racemic stereoisomer mixtures, thus the models are unable to distinguish between different stereoisomers. The measurements of S were assumed to be taken at 25°C, however not every molecule had a listed temperature for the measurement of its data. For a complete list of which molecules were definitively measured at 25°C, see Appendix 3. For the cross-validation of the data, 80% (i.e., 98) of the molecules were used to train the model, and the remaining 20% (i.e., 25) of the molecules were used to test the model.

The output variable for the models was the solubility parameter (S), with units of $\sqrt{J/mL}$. This parameter was developed by J.H. Hildebrand,[19] and it represents the isothermal energy of vaporization of the liquid into its ideal gaseous state divided by the volume of the liquid. S may also be understood as the energy required to overcome all the molecular forces holding the liquid together,[18] and this relationship is represented in the following equation

$$S = \frac{\Delta E_v}{V}. \tag{13}$$

Eq. 13 demonstrates the connection between the energy of vaporisation (ΔE_v), volume (V) and S. The nature of S, as it was initially defined by Hildebrand, makes it a more qualitative than quantitative parameter. Nevertheless, S can be used in the comparison and selection of appropriate solvents, which can then be further tested by more quantitative means if accuracy is desired.[19]

The algorithms were written in the Python coding language, while the specific programs used are included and described in Appendix 2. LR, RR and Lasso were run on eighteen different datasets, which each have the same molecules, but with different numbers of features used in each dataset. Each dataset contains associated programs for LR, RR and Lasso models, as well as csv files for the **X** (i.e., the features) and y (i.e., S) values associated with the dataset. Each LR, RR and Lasso program for the various datasets differs only in which csv file it calls up to perform the regression upon.

3.0 Results and Discussion

The correlation chart (see Appendix 4) indicated that the features with the highest degree of correlation, and thus the highest risk of collinearity, were #C and #H. They had a correlation value of 0.83, while the second-most correlated pair of features were UUS and #CO with a value of 0.70. No other pairs of features had values higher than 0.70, indicating that the only features with a strong possibility of collinearity were the four mentioned above. The correlation chart also indicated that the feature most correlated with S was #OH. #C, #H and #O were moderately correlated with S; #R, #COC and LC were weakly correlated with S; While UUS, #CO and #CC were very weakly correlated with S. Thus, given how both UUS and #CO were so weakly correlated with S, their high correlation with each other was not an issue as neither was a critical feature required to calculate S. However, as both #C and #H are roughly equally correlated with S, the final linear model was expected to only include one or the other feature. #C is slightly more correlated with S than #H, so judging only on the correlation chart it is likely that #C would be included in the most accurate model and not #H. A correlation table, the output of the csvscatterplot.py file, shows the correlations between each feature (as well S) with a matrix of scatterplots in Figure 3.

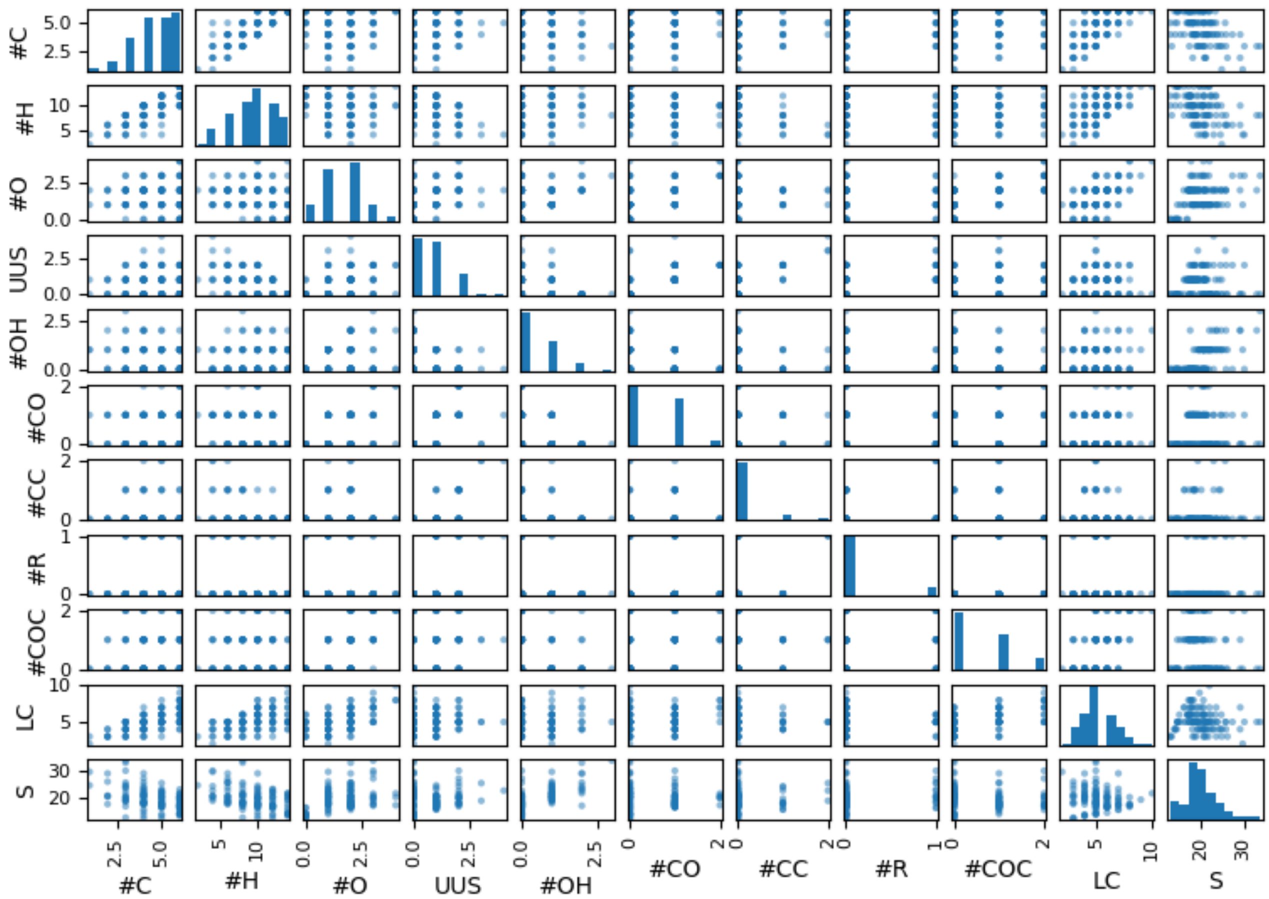

Figure 3: *A correlation table between all features and S. Scatterplots can be used to compare between the features, and histograms can be used to examine their distributions.*

3.1 Full Feature Set

From the R^2_a values, the data set including all features was most accurately represented by the Lasso model, while the LR and RR models were slightly less accurate than the Lasso model and were identical to each other. There is no difference in accuracy when comparing the R^2 and

RMSE values. The λ value for the Lasso model was optimized to three decimal places, and the same value was used for the RR model to simplify comparison between the models. The statistical values associated with each model, as well as the coefficients associated with each feature, are shown in Table 1. Figure 4 represents a scatterplot between the actual values of S for the test set, as well as the values of S calculated from the test set **X** values with the produced equation. The straight diagonal line represents the ideal fit, and if the model had a R^2 value of 1.0 then all points would be on that line. The vertical distance between each point and the line thus represents its accuracy as being the deviation from the true values of S.

Table 1: Statistical values and coefficients for the models of the full dataset.

	LR	RR	Lasso
R^2	0.78	0.78	0.78
Adjusted R^2	0.76	0.76	0.77
RMSE [(J/mL)^0.5]	1.7	1.7	1.7
Optimal λ value	N/A	0.002	0.002
Intercept / β_0	22	22	22
#C Coefficient	0.13	0.13	0
#H Coefficient	-0.57	-0.57	-0.51
#O Coefficient	1.5	1.5	1.3
UUS Coefficient	0.41	0.41	0
#OH Coefficient	3.2	3.2	3.4
#CO Coefficient	-0.78	-0.78	0
#CC Coefficient	-1.3	-1.3	-0.79
#R Coefficient	2.5	2.5	3.0
#COC Coefficient	-0.88	-0.88	-0.62
LC Coefficient	-0.13	-0.13	-0.12

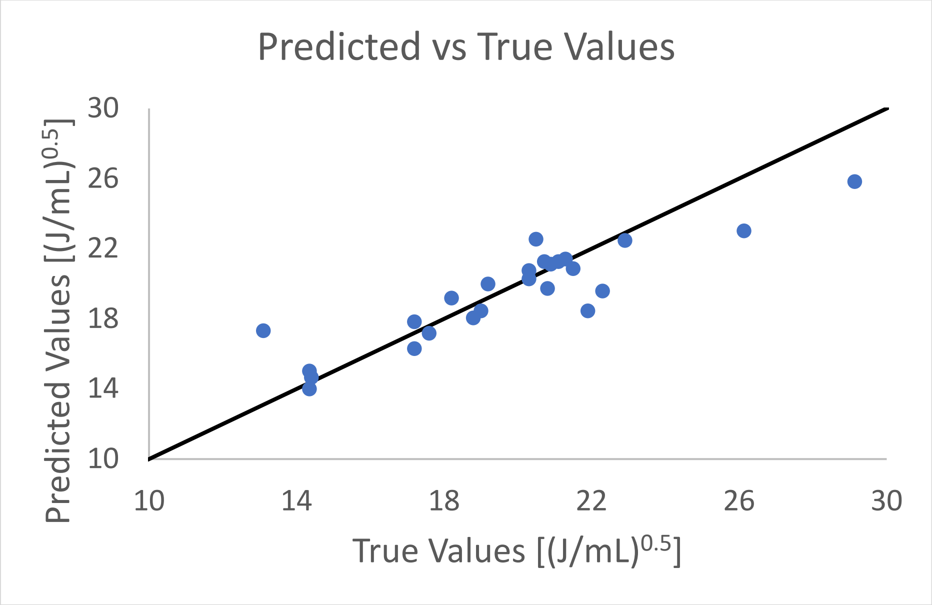

Figure 4: *A scatterplot graph between the actual values of S on the x-axis and the values of S predicted from the test set using the equation of the LR model for the full feature set.*

The value of λ was optimized for the Lasso model, and the same λ value was used for the RR model to simplify comparisons between the models. The accuracy of the RR model increased as the value of λ decreased, making it more similar to the LR model. The RR and LR models are so similar due to the relatively small value of λ, and the two models diverge (with RR becoming less accurate) as λ increases in value. For the Lasso model, the value of λ=0.002 resulted in the coefficients of three features being reduced to 0, namely #C, UUS and #CO. #C was likely eliminated due to its high correlation with #H, while UUS and #CO were likely eliminated due to their low correlation with S as well as their high correlation with each other.

Both #C and #H are negatively correlated with S, but in the LR and RR models, only #H had a negative coefficient while #C was assigned a positive coefficient. This is a clear example of the decrease in interpretability and inference due to the effect of collinearity. Instead of assigning both features relatively small negative coefficients reflective of their correlations with S, the model assigned them relatively higher coefficients with opposite signs, so that the influence of one feature would cancel out the influence of the other feature. Similarly, although UUS and #CO had correlation values with S of less than 0.10, each feature was given moderately large coefficients of opposing sign in order to cancel out each other's influence in the equation for S. It is thus clear that, when using a feature set including all ten features, the coefficients do not accurately represent the actual effects of each feature on S, and should not be taken as an accurate measure of the relationship of each feature to S.

3.2 9-Feature Sets

Two 9-Feature sets were tested, with #C and #H being the features left out in the two sets (Table 2). As #C and #H are so closely correlated with each other, they were deemed to be the most immediately apparent negligible features and were removed in order to reduce the collinearity of the models. From the R^2_a values, the optimized Lasso model for the feature set without #C was the most accurate, while the other models were slightly less accurate and equal in accuracy to each other. There is no difference in accuracy when comparing the R^2 and RMSE values.

Table 2: Statistical values and coefficients for the ‘No #H’ and ‘No #C’ Feature sets.

	No #H LR	No #H RR	No #H Lasso	No #C LR	No #C RR	No #C Lasso
R^2	0.78	0.78	0.78	0.78	0.78	0.78
Adjusted R^2	0.76	0.76	0.76	0.76	0.76	0.77
RMSE [(J/mL)^0.5]	1.7	1.7	1.7	1.7	1.7	1.7
Optimal λ value	N/A	0.012	0.012	N/A	0.002	0.002
Intercept / β_0	20	21	21	22	22	22
#C Coefficient	-1.0	-1.0	-1.0	0	0	0
#H Coefficient	0	0	0	-0.51	-0.51	-0.51
#O Coefficient	1.6	1.6	1.9	1.5	1.5	1.3
UUS Coefficient	1.3	1.2	0.31	0.50	0.50	0
#OH Coefficient	3.1	3.1	2.7	3.2	3.2	3.4
#CO Coefficient	-0.55	-0.55	0	-0.75	-0.75	0
#CC Coefficient	-1.0	-1.0	0	-1.3	-1.3	-0.79
#R Coefficient	2.8	2.8	3.6	2.6	2.6	3.0
#COC Coefficient	-0.96	-0.96	-1.2	-0.89	-0.89	-0.62
LC Coefficient	-0.13	-0.13	-0.13	-0.13	-0.13	-0.12

As with the full feature set, the values of λ were optimized for the Lasso models and the optimized λ values were then applied to the RR models to simplify comparison. The accuracy of RR for the 9-Feature sets decreased as the value of λ increased, thus decreasing their similarities to the LR model. The similarity between the LR and RR models is due to the relatively small λ values.

The values of the coefficients for #C and #H in the 9-Feature sets are both negative, because now that the issue of collinearity between the two features has been removed the model can more accurately represent the actual relationship between each feature and S. The removal of the #CO and #CC features from the ‘No #H’ Lasso model resulted in a decreased UUS coefficient, as the effects of the three features were likely cancelling each other out. Thus, without the negative coefficients of #CO and #CC, UUS was assigned a less positive coefficient. Similarly, the removal of #CO and UUS resulted in the coefficients for #CC and #R to become more positive to account for the missing features. A likely reason why the ‘No #C’ model was more accurate than the ‘No #H’ model is that #H is more strongly correlated with the other features than #C, and thus #H can better reflect the removed features. Although the removed features were only slightly correlated with S, they nevertheless exert some small effect on S, and due to #H being more correlated with the removed features it is more suitable to account for these small effects in the absence of the relatively negligible features.

3.3 7-Feature Sets

Two 7-Feature sets were tested, with the first containing all features except for #H, #O and UUS, and the second containing all features except for UUS, #CO and #CC (Table 3). The features selected for removal from the first set were due to redundancies, while the second excluded features were based on a lack of correlation with S. Oxygen is only present in the training data as part of a hydroxyl, carbonyl or ether functional group, and the only units of unsaturation in the training data come from carbonyl groups, carbon-carbon double bonds and rings, making both #O and UUS redundant if each of their corresponding features are included. #H was excluded due to its high correlation with #C. From the RMSE values, the optimized Lasso model for the 1st set was the most accurate, while the other models were slightly less accurate and equal in accuracy to each other. From the R^2_a values, the least accurate model was the Lasso model for the 2nd set. There is no difference in accuracy when comparing the R^2 values.

Table 3: Statistical values and coefficients for the two 7-Feature sets.

	1st Set LR	1st Set RR	1st Set Lasso	2nd Set LR	2nd Set RR	2nd Set Lasso
R^2	0.78	0.78	0.78	0.78	0.78	0.78
Adjusted R^2	0.77	0.77	0.77	0.77	0.77	0.76
RMSE [(J/mL)^0.5]	1.7	1.7	1.6	1.7	1.7	1.7
Optimal λ value	N/A	0.023	0.023	N/A	0.010	0.010
Intercept / β_0	21	21	21	21	21	21
#C Coefficient	-1.0	-1.0	-1.1	-0.80	-0.80	-0.63
#H Coefficient	0	0	0	-0.11	-0.11	-0.19
#O Coefficient	0	0	0	2.1	2.1	1.8
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	4.7	4.7	4.5	2.6	2.6	2.8
#CO Coefficient	2.3	2.3	2.1	0	0	0
#CC Coefficient	0.22	0.22	0.098	0	0	0
#R Coefficient	4.1	4.1	3.9	3.9	3.9	3.5
#COC Coefficient	0.64	0.64	0.44	-1.4	-1.4	-1.2
LC Coefficient	-0.13	-0.12	0	-0.13	-0.13	-0.13

As with the previous feature sets, the value of λ were optimized for the 1st set Lasso model and the optimized λ value was then applied to the corresponding RR model to simplify comparison. However, for the 2nd set the accuracy of both RR and Lasso decreased as λ increased, making LR the most appropriate model for the 2nd set. This is because the 2nd set Lasso model did not eliminate any feature without a correspondingly large decrease in R^2_a, and thus no features were indicated to be redundant for the 2nd set. Thus, Lasso is usually only more accurate than LR for sets in which a feature is eliminated, such as the eliminated LC feature in the 1st set. LC and #COC are very similarly weakly correlated with S, but #COC is more strongly correlated with the missing features, which is likely why LC and not #COC was eliminated.

For the 1st set, the removal of #O greatly increased the coefficient values of #OH, #CO and #COC as these features now had to account fully for the presence of oxygen instead of combining their total effect with #O in previous feature sets. The greatest change from previous models is the positive value of the #COC coefficient, in light of its weakly negative correlation with S. The removal of LC also caused the coefficients of the two features to which it was most highly correlated, being #C and #COC, to both become more negative to account for the loss of the slightly negative LC value.

For the 2nd set, the values of #C and #H are much more accurately represented than for previous sets, as both features have negative coefficient values. Nevertheless, to eliminate the collinearity in the model they should not be both included for an ideal model of S. For the Lasso model, #O was slightly more negative, and to account for this the coefficient values of the other oxygen-containing features were slightly more positive. #R is the only feature in this set corresponding to units of unsaturation, and its coefficient value is unusually high given its low correlation with S, which may indicate that the high value is compensating for the sum of the weak effects of the other unsaturation-related features.

3.4 6-Feature Sets

Two 6-Feature sets were tested, with the first set containing #H, #O, #OH, #R, #COC and LC, while the second set contains #C, #OH, #CO, #R, #COC and LC (Table 4). UUS and #CC were removed from both sets due to their weak correlations with S, while #H and #C were included in different sets due to their high collinearity. #O was included in the 1st set, while #CO was included in the 2nd set as the inclusion of all oxygen-containing functional groups negates the necessity of the inclusion of #O. From the R^2 and R^2_a values, the 2nd set was overall more accurate at predicting S than the 1st set. From the RMSE values, the most accurate model was the optimized Lasso model for the 2nd set.

Table 4: Statistical values and coefficients for the two 6-Feature sets.

	1st Set LR	1st Set RR	1st Set Lasso	2nd Set LR	2nd Set RR	2nd Set Lasso
R^2	0.77	0.77	0.77	0.78	0.78	0.78
Adjusted R^2	0.76	0.76	0.76	0.77	0.77	0.77
RMSE [(J/mL)^0.5]	1.7	1.7	1.7	1.7	1.7	1.6
Optimal λ value	N/A	0.006	0.006	N/A	0.023	0.023
Intercept / β_0	21	21	21	21	21	21
#C Coefficient	0	0	0	-1.0	-1.0	-1.1
#H Coefficient	-0.42	-0.42	-0.43	0	0	0
#O Coefficient	1.5	1.5	1.5	0	0	0
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.3	3.3	3.4	4.7	4.7	4.4
#CO Coefficient	0	0	0	2.3	2.3	2.1
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	3.1	3.1	3.0	4.1	4.1	3.9

#COC Coefficient	-0.71	-0.71	-0.64	0.64	0.63	0.44
LC Coefficient	-0.28	-0.28	-0.26	-0.13	-0.12	0

As with the previous feature sets, the value of λ were optimized for the Lasso models of each set and the optimized λ value was then applied to the corresponding RR model to simplify comparison. It is notable that although the 1st set Lasso model did not reduce any coefficients to 0, its accuracy peaked at λ=0.006 and did not continue to increase as the value of λ decreased.

For the 1st set, the LC coefficient is relatively high to compensate for the lack of #C, to which LC is highly correlated. The higher value of the LC coefficient is the probable reason that it was not eliminated in the Lasso model as was done in the 2nd set. The only features whose coefficients did not decrease in absolute value between the LR and Lasso model are #H, #O and #OH, indicating that these are the most relevant features within this set with which to calculate S.

For the 2nd set, the values of #OH, #CO and #COC are much more positive to account for the lack of #O. The elimination of LC resulted a more negative value of all of the remaining coefficients, and not just the ones most closely correlated with LC (being #C and #COC). Thus, LC appears to be a relatively negligible feature, as its impact on S can be factored into all the other features with a corresponding decrease in RMSE.

Thus far, the results of the full feature set, the 9-Feature sets, the 7-Feature sets and the 6-Feature sets indicate that the most relevant features to calculate S accurately are #C, #H, #O, #OH, #CO, #R, #COC and LC. Further data sets have indicated optimal combinations of these features, however the data indicates that UUS and #CC are relatively uncorrelated with S and do not tend to add accuracy to models which include them.

3.5 5-Feature Sets

Six 5-Feature sets were tested. The 1st set contains #C, #OH, #R, #COC and LC; the 2nd set contains #C, #OH, #CO, #R and #COC; the 3rd set contains #H, #OH, #CO, #R and #COC; the 4th set contains #C, #O, #OH, #R and #COC; the 5th set contains #H, #O, #OH, #R and #COC; and the 6th set contains #H, #O, #OH, #R, LC. The previous feature sets seem to indicate that #OH and #R are both key features, but the different 5-Feature sets were selected to compare different combinations of #C, #H, #O, #CO, #COC and LC. From all measures of accuracy, the 1st set was the least accurate, the 6th set was slightly more accurate, and sets 2 through 5 were equal in accuracy. There were no differences in accuracy between the models within each set, except for the 1st set in which the Lasso model was slightly less accurate than the LR and RR models in terms of its R^2 value. None of the Lasso models resulted in any features being eliminated, making it likely that the minimum number of features required in a set is five. A model with more features is unnecessary, and a model with fewer features is insufficient to accurately calculate S, as shown by section 3.6 of the Results. The accuracies of each model are shown in Table 5, while the statistical values and coefficients associated with the 1st and 3rd set are shown in Table 6. The statistical values of coefficients of all 5-Feature sets can be found in Appendix 5.

Table 5: Statistical values and accuracies of each 5-Feature Model.

	1st Set	2nd Set	3rd Set	4th Set	5th Set	6th Set
Features	#C, #OH, #R, #COC, LC	#C, #OH, #CO, #R, #COC	#H, #OH, #CO, #R, #COC	#C, #O, #OH, #R, #COC	#H, #O, #OH, #R, #COC	#H, #O, #OH, #R, #LC
LR R^2	0.71	0.79	0.79	0.79	0.79	0.76
LR R^2_a	0.69	0.77	0.77	0.77	0.77	0.75
LR RMSE [(J/mL)^0.5]	1.9	1.6	1.6	1.6	1.6	1.7
RR R^2	0.71	0.79	0.79	0.79	0.79	0.76
RR R^2_a	0.69	0.77	0.77	0.77	0.77	0.75
RR RMSE [(J/mL)^0.5]	1.9	1.6	1.6	1.6	1.6	1.7
Lasso R^2	0.70	0.79	0.79	0.79	0.79	0.76
Lasso R^2_a	0.69	0.77	0.77	0.77	0.77	0.75
Lasso RMSE [(J/mL)^0.5]	1.9	1.6	1.6	1.6	1.6	1.7

Table 6: Statistical values and coefficients for the 1st and 3rd Feature sets.

	1st Set LR	1st Set RR	1st Set Lasso	3rd Set LR	3rd Set RR	3rd Set Lasso
R^2	0.71	0.71	0.70	0.79	0.79	0.79
Adjusted R^2	0.69	0.69	0.69	0.77	0.77	0.77
RMSE [(J/mL)^0.5]	1.9	1.9	1.9	1.6	1.6	1.6
Optimal λ value	N/A	0.010	0.010	N/A	0.010	0.010
Intercept / β_0	22	22	22	21	21	21
#C Coefficient	-1.4	-1.4	-1.4	0	0	0
#H Coefficient	0	0	0	-0.51	-0.51	-0.52
#O Coefficient	0	0	0	0	0	1.1

UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.5	3.5	3.5	4.7	4.7	4.6
#CO Coefficient	0	0	0	1.2	1.2	1.2
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	3.3	3.3	3.2	3.0	3.0	2.8
#COC Coefficient	0.23	0.23	0.21	0.47	0.47	0.46
LC Coefficient	0.40	0.40	0.39	0	0	0

The value of λ was only optimized at 0.010 for the 6th set Lasso model. None of the other Lasso or RR models had an optimal λ value and their accuracies increased as λ decreased. Nevertheless, the λ value of each RR and Lasso model was set at 0.010 to simplify comparison. It is notable that the 6th set Lasso model had a non-zero peak value, despite the model not reducing any of the feature coefficients to zero.

The low accuracy of the 1st set is likely due to its lack of both #O and #CO. The #O feature is redundant if #OH, #CO and #COC are all included, but the impact of oxygen on S isn't fully accounted for if neither #O nor all three of the corresponding functional groups aren't included. Thus, for a truly accurate model of S, the presence of oxygen atoms must be fully accounted for.

The low accuracy of the 6th set is likely due to the inclusion of LC and the exclusion of #COC. The 6th set is nearly identical to the 5th set, except for these two features, which indicates that #COC is more relevant to S than LC. LC and #COC are moderately correlated with each other, but the presence of oxygen in #COC likely makes it more relevant than LC, resulting in the 5th set being more accurate than the 6th set. It is notable that #H was selected over #C in previous feature sets, and that #COC was selected over LC in this set as in previous feature sets, given how #C is slightly more correlated with S than #H and how LC is slightly more correlated with S than #COC. This indicates that the correlation between a feature and S is not the only indication of a feature's importance when calculating S.

Sets 2 through 5 have equal accuracies in terms of R^2, R^2_a and RMSE. Each of these sets include #OH, #R and #COC and have either #H or #C, and either #O or #CO. However, the model with the coefficients most representative of the relationships between each feature and S is likely to be the 3rd set. With regard to #H and #C, previous model sets have indicated that models with #H are more able to represent the effects of missing features than models with #C. #O is a negligible feature if the #OH, #COC and #CO features are all included, and without #O each of those features can better reflect the individual effects of each functional group on S. Thus, the 5-

Feature set most likely to accurately represent the relationship between each features and S is the 3rd set. There is no difference between the accuracies of LR, RR or Lasso for this set, but since LR is the simplest model it can be said to be best suited to represent this feature set.

The equation for S produced by the LR model of the 3rd set is shown in the following equation, using the intercept and coefficient values taken from Table 6

$$S = 21 - 0.51\#H + 4.7\#OH + 1.2\#CO + 3.0\#R + 0.47\#COC. \qquad (14)$$

Eq. 14 calculates S within ± 1.4 $\sqrt{J/mL}$ for a 95% confidence range.

The results of the 5-Feature set models indicate that the most important features to calculate S accurately are #C/#H, #O/#CO, #OH, #R and #COC. #C, #H, #O and #CO are each relevant to S, but #C is relatively interchangeable with #H, while #O is relatively interchangeable with #CO. The set which most likely represents an accurate relationship between 5 features and S is the 3rd set, which includes #H, #OH, #CO, #R and #COC as features.

3.6 4-Feature and 3-Feature Sets

Three 4-Feature sets and two 3-Feature sets were tested. The 1st set contains #H, #O, #OH and #R; the 2nd set contains #H, #O, #OH and #COC; the 3rd set contains #H, #O, #OH and LC; the 4th set contains #H, #O and #OH; the 5th set contains #C, #O and #OH. The 4-Feature sets are used to determine the relative importance of #R, #COC and LC, while the 3-Feature sets are used to further analyze the importance of #H as opposed to #C. Each model includes #O and not #CO in order to fully account for all oxygen atoms given the limited number of features in each set. From all measures of accuracy, the 1st set was the most accurate and the 5th set was the least accurate. The 2nd and 4th sets were equal in accuracy to each other. The LR and RR models of the 3rd set were slightly less accurate than the corresponding Lasso model, which was equal in accuracy to the other models of the 2nd and 4th sets. The accuracies of each model are shown in Table 7, while the coefficients of each set can be found in Appendix 5.

Table 7: Statistical values and accuracies of each 3 and 4-Feature Model.

	1st Set	2nd Set	3rd Set	4th Set	5th Set
Features	#H, #O, #OH, #R	#H, #O, #OH, #COC	#H, #O, #OH, LC	#H, #O, #OH	#C, #O, #OH
Features used in Lasso model	#H, #O, #OH, #R	#H, #O, #OH	#H, #O, #OH	#H, #O, #OH	#C, #O, #OH
Optimized λ value	0.010	0.025	0.033	0.010	0.010
LR R^2	0.78	0.69	0.68	0.69	0.55
LR R^2_a	0.77	0.68	0.67	0.68	0.54
LR RMSE [(J/mL)^0.5]	1.7	2.0	2.0	2.0	2.4
RR R^2	0.78	0.69	0.68	0.69	0.55
RR R^2_a	0.77	0.68	0.67	0.68	0.54
RR RMSE [(J/mL)^0.5]	1.7	2.0	2.0	2.0	2.4
Lasso R^2	0.78	0.69	0.69	0.69	0.55
Lasso R^2_a	0.77	0.68	0.68	0.68	0.54
Lasso RMSE [(J/mL)^0.5]	1.7	2.0	2.0	2.0	2.4

The λ values were only optimized for the Lasso models of sets 2 and 3, the other Lasso and RR models had no optimal value, and their accuracies continued to decrease as λ increased. Nevertheless, the RR λ value of sets 2 and 3 were matched to their corresponding Lasso models, and the values of the other models was set at 0.010 to simplify comparison.

The difference in accuracy between sets 1-3 indicates that between #R, #COC and LC, the most relevant feature to the calculation of S is #R, followed by #COC and LC. The Lasso models of sets 2 and 3 eliminated the #COC and LC features, indicating that the connection of these two features to S is partially dependent on the presence of #R.

The great decrease in accuracy between the 4th and 5th set indicates that #H is much more important than #C when using very few features. The effects of the missing features are better represented by #H than #C due to the higher correlations between #H and the missing features. #H is moderately correlated with many features, while #C is only significantly correlated with #H and LC. This is likely why the difference in accuracy between these sets is much greater than that between the two 9-Feature sets.

From the results of the 3 and 4-Feature sets, the four most important features with which to calculate S are #H, #O, #OH and #R.

4.0 Conclusions

In this thesis, the most accurate method of predicting the solubility parameter (S) as well as determining the relationship between S and certain structural features was studied through the use of ML. This was done by formulating a number of different feature sets with different combinations of the ten features, being #C, #H, #O, UUS, #OH, #CO, #R, #COC and LC, and analyzing these feature sets through Linear regression, Ridge regression and Lasso models.

For the LR models, within each set the accuracies of the LR and RR models were equal, with very little differences in coefficient values between the two models. LR models tended to be less accurate than their corresponding Lasso models if features were eliminated from the Lasso models, but they tended to be equal in accuracy to the Lasso models if all features were retained. The most accurate LR models used the 2nd, 3rd, 4th and 5th 5-Feature sets, which included #C/#H, #O/#CO, #OH, #R and #COC as features. These models were equal in accuracy to their associated RR and Lasso models with regard to R^2, R^2_a and RMSE, but the 3rd set, containing #H, #OH, #CO, #R and #COC, is the model best suited for accurately representing the relationship between these features and S.

For the RR models, they were equal in accuracy to their corresponding LR models for the tested sets, and no RR model had an optimal λ value. This indicates that the coefficients produced from LR were not unreasonably large with the tested features, and that collinearity was not a large issue with the selected features, aside from the correlation of #C and #H. RR would likely be more useful in a dataset with a larger number of features with larger associated coefficient values. The most accurate RR models used the same sets as the most accurate LR models.

For the Lasso models, they were always the most accurate model when they eliminated a feature by reducing its coefficient to 0, in which cases there was an optimal λ value. The only cases where there was an optimal λ value without the elimination of a feature was for the 1st 6-Feature set and the 6th 5-Feature set, which contained #H, #O, #OH, #R, #COC, LC and #H, #O, #OH, #R, LC respectively. Lasso models were very useful for eliminating negligible features from the dataset, but once those features were eliminated the models were as accurate as LR. Thus, for smaller feature sets without any negligible features, LR should be used instead of Lasso due to the simplicity of LR.

Overall, the most accurate models in terms of R^2, R^2_a and RMSE of the test set were the models of the 2nd, 3rd, 4th and 5th 5-Feature sets. These models each had a R^2 value of 0.79, a R^2_a

value of 0.77, and a RMSE value of 1.6 $\sqrt{J/mL}$. Thus, for molecules with no more than 6 molecules and with only carbon, hydrogen and oxygen atoms, 95% of the values of S produced by these models will be within ± 1.4 $\sqrt{J/mL}$ of the actual value of S. The features not included in these models, being UUS, #CC and LC, are essentially negligible to the calculation of S, as their minor influences on solubility can be accounted for with the other features.

The optimized features are mainly correlated with the molecular weight and number of hydrogen bond acceptors and donors for each molecule. Thus, the size of a molecule and its ability to form hydrogen bonds are factors which are highly correlated with that molecule's solubility. The surface area and number of rotatable bonds were roughly represented by LC, and the removal of this feature indicates these factors to be less important towards calculating S for this dataset. The features representing sources of unsaturation were mostly eliminated, except for #R, which indicates that unsaturation is relatively uncorrelated with S.

In the future, feature sets using all possible combinations of the ten structural features should be performed, to be able to definitively produce models with the most accurate combination of features. Furthermore, different combinations of higher order parameters should be tested, such as the square or square root of features, or the product of two different features. A wider variety of structural features should be examined as well, such as a feature directly measuring molecular weight and features for specific counts of hydrogen bond donors and acceptors. A larger and more diverse dataset should also be considered, in order to examine the possible effects of aromaticity and other elements (e.g., N, F, Cl, etc.) on S. Furthermore, linear models should be tested on the logarithm of S, to better elucidate the relationship between S and various molecular features. The training set and test sets should be split up with k-fold CV to reduce the chance of error affecting the data. Different ML models such as neural networks or tree-based models should be performed to determine whether S truly has a linear relationship with the selected features.

The use of ML models to predict a value such as solubility based on molecular structure can also be applied to other fields, such as crystal structure prediction. The results of this work could be applied to similar ML models to predict Steinhardt order parameters for molecules based on similar structural features. However, future work should examine linear as well as non-linear ML models for this field, in order to fully understand the relationship between molecular structure and crystal structure in different environments.

The prediction of solubility from structurally derived features has important applications for the field of drug design. Models of solubility, such as the one produced in this work, have the capacity to compare the relative solubilities of potential drug candidates and to predict the solubilities of proteins within the body. Due to the potential usage of these solubility models in medicinal and organic chemistry, it is important that such models are both easily interpretable by medicinal and organic chemists and widely applicable, through the use of features that are computationally relatively straightforward to calculate and relate to the physical structure of the molecule. The model produced in this work is simple and specialized for a relatively small set of molecules, however the basic principles on which it is founded can be easily expanded to more accurately predict S for a wide range of molecules relevant to the study of drugs, proteins and enzymes.

References

[1] J. D. Harper, P. T. Lansbury, *Annu. Rev. Biochem.* **1997**, *66*, 385–407.

[2] W. L. Jorgensen, E. M. Duffy, *Adv. Drug Deliv. Rev.* **2002**, *54*, 355–366.

[3] J. S. Delaney, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000–1005.

[4] J. Ali, P. Camilleri, M. B. Brown, A. J. Hutt, S. B. Kirton, in *J. Chem. Inf. Model.*, American Chemical Society, **2012**, pp. 420–428.

[5] C. A. Lipinski, F. Lombardo, B. W. Dominy, P. J. Feeney, *Adv. Drug Deliv. Rev.* **1997**, *23*, 3–25.

[6] S. Mazurenko, Z. Prokop, J. Damborsky, *ACS Catal.* **2020**, *10*, 1210–1223.

[7] M. I. Jordan, T. M. Mitchell, *Science (80-.).* **2015**, *349*, 255–260.

[8] D. M. Dutton, G. V. Conroy, *Knowl. Eng. Rev.* **1997**, *12*, 341–367.

[9] T. J. Hughes, S. Cardamone, P. L. A. Popelier, *J. Comput. Chem.* **2015**, *36*, 1844–1857.

[10] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, A. R. Oganov, *Phys. Rev. B* **2019**, *99*, DOI 10.1103/PhysRevB.99.064114.

[11] S. Shalev-Shwartz, S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, New York, NY, **2013**.

[12] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York, New York, NY, **2009**.

[13] P. J. Steinhardt, D. R. Nelson, M. Ronchetti, *Phys. Rev. B* **1983**, *28*, 784–805.

[14] G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer New York, New York, NY, **2013**.

[15] M. K. Chung, *Encyclopedia of Research Design*, SAGE Publications, Inc., **2012**.

[16] J. H. Rosenthal, M. J. Evans, **2009**.

[17] D. Freedman, R. Pisani, R. Purves, *Statistics*, W. W. Norton & Company, New York ; London, **2007**.

[18] J. A. Riddick, W. B. Bunger, T. K. Sakano, *Organic Solvents: Physical Properties and Methods of Purification. Fourth Edition*, Wiley, New York ; Toronto, **1986**.

[19] J. H. Hildebrand, *Chem. Rev.* **1949**, *44*, 37–45.

[20] D. Kriesel, *A Brief Introduction to Neural Networks*, **2007**.

[21] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, **2015**.

5.0 Appendices

Appendix 1: Additional Information

1.1 Neural Networks

Computational programs tend to be static and difficult to apply to different situations, being unable to learn and change over time as the brain does. Thus, the theory of a neural network was developed, to study ways in which a program can 'learn' from data on its own, to be able to generalize and associate the data it was given to apply it to different situations.[20] Neural networks can be visualized as graphs of nodes, organized in layers where the outputs of nodes are weighted to provide the inputs of the nodes on the subsequent layer. An example of a simple feed-forward (i.e., acyclic) neural network is shown in Figure 5.

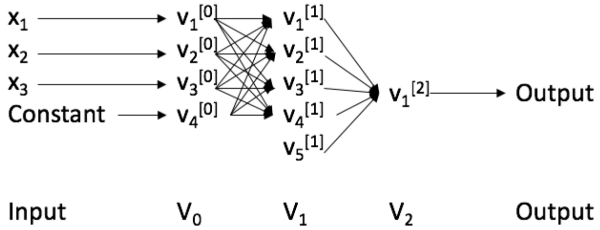

Figure 5: *A diagram of a feedforward neural network, with an Input Layer (V_0), a Hidden Layer (V_1), an Output Layer (V_2) and 10 nodes. This network thus has a depth of 2, a size of 10 and a width of 5.*

Neural networks are organized in layers, the first being the Input Layer (V_0). $v_j^{[l]}$ represents the *j*th node in the *l*th layer, while $o_j^{[l]}(x)$ represents the output of $v_j^{[l]}$ when the network is fed with the input vector x_j.[21] Layers V_1, V_2, … V_{L-1} are referred to as Hidden Layers, and V_L is referred to as the Output Layer, which will be composed of only a single neuron if a simple neural network is used. The depth of a network refers to the number of layers (i.e., the L of V_L), the size of the network refers to the total number of nodes/neurons (the terms can be used interchangeably), and the width of the network refers to the size of its largest layer.[11]

1.2 Characteristics of Regression models

LR, RR and Lasso are all parametric models, as they fit coefficient values to different parameters (or features) in order to predict Y. Parametric models are easier to fit to f(**X**) and are more appropriate for inference, but have a greater risk of error if the chosen model doesn't match

Y. Non-parametric models avoid making explicit assumptions about f($\mathbf{X}$), but require a great number of data points to be accurate, and are better for prediction.

For any model there is always a trade-off between the flexibility of the model and its interpretability, as the flexibility increases the interpretability will decrease and vice versa. Low flexibility and high interpretability are ideal for inference problems, while the reverse is true for prediction problems. There are two types of problems and variables analyzed, being regression problems, with quantitative data, and classification, with qualitative or categorical values. Regression models can include certain categorical values, such as binary features (e.g., #R in this work), but classification models cannot easily include quantitative data.[14]

A common problem with regression models is overfitting, when the model is so closely fit to the training set that it detects 'trends' in the data that are just due to random chance. Overfitting is closely associated with a high amount of variance, or the degree to which $\hat{f}(\boldsymbol{X})$ would change given a different training set, and with a low amount of bias, or the error resulting from reducing complex problems to simpler models. Similar to flexibility and interpretability, variance and bias each increase as the other decreases, and in an ideal model both types of errors are minimized but never eliminated fully. In order to reduce the impact of overfitting, the dataset can be split into training and test sets in a process called cross validation (CV). The training set is used to fit the model while the test set is used to examine the accuracy of the model, because the applicability of the model to non-training data is much more important than how well the model can estimate the $\hat{f}(\boldsymbol{X})$ of data that is already known. In other words, it is not important to understand how well the model has predicted the data that it has been given, it is only important to understand how well the model can predict future data that will be given to it outside of its training data.[14]

For data sets that are as relatively small as the one used in this study, the data set can also be split up through *k*-fold CV. When using *k*-fold CV, the data set is divided into *k* roughly equal sections and then *k* CV runs are implemented using each of the *k* sections once as the test set and leaving the other *k*-1 sections as the training sets. The average error of these *k* runs is calculated and reported. With a relatively computationally inexpensive program such as linear regression, it is feasible to test and average N different *k*-fold CV runs, thus the individual datapoints are used once each as the full test set for N different runs. When using N = *k* for the *k*-fold CV, it is often referred to as leave-one-out CV. If implementing N different *k*-fold CV runs is unfeasible, standard values for *k* are 5 or 10.[12] However, there is a bias-variance trade-off when one chooses to use N instead of 5 or 10 as the k value. Running a leave-one-out CV will have low bias but high variance, as each training set is composed of N-1 data points and will thus be very similar and highly correlated with each other. On the other hand, running a 5 or 10 k-fold CV will have lower variance but higher bias as the outputted error depends on the randomness of how the data set was split in k different ways.[14] As the training/test split was only performed once in this study, there is a relatively high level of bias associated with the produced models, but a relatively low level of variance.

1.3 Errors and Assumptions of Linear Regression models

Linear models of regression, such as the LR, RR and Lasso models examined in this work, make a number of assumptions about the data under analysis, which can be a source of reducible error.

The models assume that the relationship between an individual variable and the output is additive; that it is independent of any other input variable; that these relationships are linear; and that the change in a variable is independent of the variable's value. To account for additive factors, interaction terms can be introduced into the regression equation,[14] as shown in the following equation for a simple model with only 2 input variables

$$\hat{f}(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 \tag{15}$$

where x_i refers to the variables and β_i refers to their associated coefficients. To account for quadratic factors, higher order variables can be introduced into the model, which is still solved the same way as normal regression models,[14] with the new variable treated the same as any other, as shown in the following equation

$$\hat{f}(X) = \beta_0 + \beta_1 x_1 + \beta_2 {x_1}^2. \tag{16}$$

Another assumption made by regression models is that the error terms of the variables are unrelated, which can result in a higher reported accuracy than the true accuracy if the error terms are related. However, this is an issue mainly associated with time series data and is thus not a major source of error for this study. It is also assumed that the variance of the error terms is constant.[14]

The accuracy of regression models can be decreased by the presence of outliers and high leverage points. Outliers are points with normal input variables but abnormal outputs, while high leverage points are points with abnormal input variables or combinations thereof. Outliers often indicate a deficiency in the model, like an unaccounted-for predictor, and significantly affect the confidence and accuracy of a model, while high leverage points can greatly affect the shape of the linear model. A model with many predictors is at risk of error due to collinearity, which is defined in section 2.1.1 of the methodology. Collinearity can be accounted for by eliminating the redundant variables, which will usually increase the accuracy of the model as well. LR is at more of a risk from collinearity than RR or Lasso, due to the penalty factor reducing coefficient size in RR and Lasso.[14]

Appendix 2: Codes Used

The algorithms were written in the Python coding language, using programs downloaded through the Anaconda package of data and software, which is available for free online. The specific programs used are described in Table 8.

Table 8: Anaconda Programs used in Coding.

Anaconda program used	Specific software and functions used
matplotlib	pyplot
pandas	plotting, scatter_matrix, read_csv
numpy	sum, mean
sklearn	linear_model, preprocessing, model_selection, train-test-split, metrics, mean_squared_error, LinearRegression, Ridge, Lasso
scipy	stats
math	sqrt

2.1 Code for csvscatterplot.py

This program outputs a matrix of scatterplots between each feature and between the features and S. It also outputs a table of variables, an individual scatterplot for any two desired features, and a line chart representing the values of each feature associated with the individual data points. Comments within the code are preceded with a '#' symbol.

```
# Scatter Plot Matrix
import matplotlib.pyplot as plt
import pandas
from pandas.plotting import scatter_matrix
import numpy
```

```
colnames = ['#C', '#H', '#O', 'UUS', '#OH', '#CO', '#CC', '#R', '#COC', 'LC', 'S']
data = pandas.read_csv('solubilitieswordless.csv', names=colnames)
print(data.head())     # <This would print a table of the variables for the first five molecules.
scatter_matrix(data)
plt.show()
data.plot.scatter(x='#C', y='#H', title = 'Number of Carbon and Hydrogen Atoms')
plt.show()
        # ^Lines 16 & 17 print a scatterplot between any two variables;
        # Here I've chosen #C and #H.
data.drop(['Solubility'], axis=1).plot.line(title='Solubility')
plt.show()
        # ^Lines 21 & 22 print a line chart showing the variables associated with each molecule.
```

2.2 Code for statsfordata.py

This program outputs a correlation, covariance and description table. The description table includes the number of data points, the mean value, the standard deviation, minimum value, 25% value, 50% value, 75% value and max value for each feature and for S. These tables are saved as csv files. Comments within the code are preceded with a ‘#’ symbol.

```
import pandas
colnames = ['#C', '#H', '#O', 'UUS', '#OH', '#CO', '#CC', '#R', '#COC', 'LC', 'S']
data = pandas.read_csv('solubilities1.csv',index_col='Molecule',parse_dates=True)
file = pandas.read_csv('solubilities1.csv')
      #Printing Description, Correlation and Covariance Tables:
print(data.describe())
print(data.corr())
print(data.cov())
      #Creating .csv files for the Description and Correlation Tables:
data.corr().to_csv('Correlation of variables.csv')
data.describe().to_csv('Description of variables.csv')
```

2.3 Code for solublinreg.py (For Full Dataset)

This program runs a LR model on the full dataset, including all features. It uses csv files of the features and S values as an input and separates the dataset into a training and test set. It outputs the R^2, R^2_a and RMSE value, as well as the intercept and coefficients for each feature in the LR model. It also outputs a scatterplot between the values of S from the test set and the predicted values of S based on the test set feature values and the produced model. Comments within the code are preceded with a '#' symbol.

```
import pandas as pd
from sklearn import linear_model, preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import sklearn
import statsmodels.api as sm
from scipy import stats
from matplotlib import pyplot as plt
from sklearn.metrics import mean_squared_error
from math import sqrt
plt.ioff()
x = pd.read_csv('solubxonly.csv') # load the csv file of the independent variables
y = pd.read_csv('solubyonly.csv') # load the csv file of the dependent variable
# Create Training and Testing Variables
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=31) #20% of data will be test
print ('Training (#data points, #variables):', X_train.shape, y_train.shape)
print ('Test (#data point, #variables):', X_test.shape, y_test.shape)
# Fit a linear model
lm = linear_model.LinearRegression()
model = lm.fit(X_train, y_train)
predictions = lm.predict(X_test)
#print('Predictions:', predictions[0:])
```

```
#print('True Values:', y_test[0:])
#^ This would print the predicted values and true values
rms = sqrt(mean_squared_error(y_test, predictions))
trainR2 = model.score(X_train, y_train)
R2 = model.score(X_test, y_test)
adjR2 = 1 - ((1-R2)*(98-1)/(98-10-1))
# Output the % Accuracy and a plot
print ('Score(R^2):', R2)
print ('Adj. R^2:', adjR2)
print ('Coefficients [#C, #H, #O, UUS, #OH, #CO, #C=C, #Rings, #COC, LC]:', lm.coef_)
print ('Intercept:', lm.intercept_)
print ('RMSE:', rms)
print ('trainR2', trainR2)
plt.scatter(y_test, predictions)
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.title('True Values and Predictions')
plt.show()
```

2.4 Code for ridgereg.py (For Full Dataset)

This program runs a RR model on the full dataset including all features. It uses csv files of the features and S values as an input and separates the dataset into a training and test set. It outputs the R^2, R^2_a and RMSE value, as well as the intercept and coefficients for each feature for the RR model. The same values are also outputted for the equivalent LR model, for comparison. The function Ridge_score(a) is defined to run a RR model on any desired penalty value, designated as ‘a’ and described as ‘alpha’ in the code. Comments within the code are preceded with a ‘#’ symbol.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
from math import sqrt
x = pd.read_csv('solubxonly.csv')
y = pd.read_csv('solubyonly.csv')
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=31)
lr = LinearRegression()
lr.fit(X_train, y_train)
predictions = lr.predict(X_test)
rms = sqrt(mean_squared_error(y_test, predictions))
# higher the alpha value, more restriction on the coefficients;
# low alpha > more generalization, coefficients are barely restricted
# alpha is the lambda penalty factor
test_score=lr.score(X_test, y_test)
adjR2 = 1 - ((1-test_score)*(98-1)/(98-10-1))
def Ridge_score(a):
    rra = Ridge(alpha=a)
    rra.fit(X_train, y_train)
    Ridge_score = rra.score(X_test, y_test)
    predictionsrr = rra.predict(X_test)
    rmsrr = sqrt(mean_squared_error(y_test, predictionsrr))
    adjr2 = 1 - ((1-Ridge_score)*(98-1)/(98-10-1))
    print('RR R^2 (alpha =', a, '):', Ridge_score)
    print('RR Adj. R^2 (a =', a, '):', adjr2)
    print('RR Coefficients:', rra.coef_)
    print('RR Intercept:', rra.intercept_)
    print('RR RMSE:', rmsrr)
print("LR R^2:", test_score)
print('LR Adj. R^2:', adjR2)
print('LR Coefficients [#C, #H, #O, UUS, #OH, #CO, #C=C, #Rings, #COC, LC]:', lr.coef_)
print('LR Intercept:', lr.intercept_)
print('LR RMSE:', rms)
Ridge_score(0.001)
Ridge_score(0.002)
Ridge_score(0.01)
Ridge_score(0.1)
Ridge_score(1)
Ridge_score(10)
Ridge_score(100)
```

2.5 Code for lasso.py (For Full Dataset)

This program runs a Lasso model on the full dataset including all features. It uses csv files of the features and S values as an input and separates the dataset into a training and test set. It outputs the R^2, R^2_a and RMSE value, as well as the intercept and coefficients for each feature for the Lasso model. The same values are also outputted for the equivalent LR model, for comparison. The function lassoalpha(a) is defined to run a Lasso model on any desired penalty value, designated as ‘a’ and described as ‘alpha’ in the code. Comments within the code are preceded with a ‘#’ symbol.

```
import math
import pandas as pd
import numpy as np
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
# difference of lasso and ridge regression is that some of the coefficients can be zero
# i.e., some of the features are completely neglected
# alpha is the lambda penalty coefficient
x = pd.read_csv('solubxonly.csv')
y = pd.read_csv('solubyonly.csv')
```

```
X_train,X_test,y_train,y_test=train_test_split(x ,y, test_size=0.2, random_state=31)
lr = LinearRegression()
lr.fit(X_train,y_train)
lr_test_score=lr.score(X_test,y_test)
predictions = lr.predict(X_test)
rms = sqrt(mean_squared_error(y_test, predictions))
adjR2 = 1 - ((1-lr_test_score)*(98-1)/(98-10-1))
print ("LR R^2: ", lr_test_score)
print ('LR adj R^2:', adjR2)
print ("Number of features used: 10")
print ('LR RMSE:', rms)
def lassoalpha(a):
   lassoa = Lasso(alpha=a, max_iter=10e5)
   fitted_lasso = lassoa.fit(X_train, y_train)
   test_scorea = lassoa.score(X_test,y_test)
   coeff_useda = np.sum(lassoa.coef_!=0)
   adjr2 = 1 - ((1-test_scorea)*(98-1)/(98-coeff_useda-1))
   preda = lassoa.predict(X_test)
   rmsa = sqrt(mean_squared_error(y_test, preda))
   print("R^2 for (alpha =", a, "):", test_scorea)
   print('Adj. R^2 for (alpha =', a, '):', adjr2)
   print("Number of features used:", coeff_useda)
   print('Coefficients:', lassoa.coef_)
   print('Intercept:', lassoa.intercept_)
   print('RMSE:', rmsa)
lassoalpha(1)
lassoalpha(0.1)
lassoalpha(0.01)
lassoalpha(0.002)
lassoalpha(0.001)
lassoalpha(0.0001)
```

lassoalpha(0.00001)

lassoalpha(0.000001)

Appendix 3: Full Dataset

The following table lists the feature and S values for each molecule in the dataset. The units of S are $\sqrt{\mathrm{J/mL}}$, and the final column, titled 'T (°C)', indicates which molecule had its S value measured at 25°C. The molecules labelled '25' were measured at 25°C, the molecules labelled '/' did not have a listed temperature.

Index	Molecule	C#	H#	O#	UUS	#OH	#CO	#CC	#R	#COC	LC	S	T (°C)
0	Propane	3	8	0	0	0	0	0	0	0	3	13.1	25
1	Butane	4	10	0	0	0	0	0	0	0	4	13.9	25
2	2-Methylpropane	4	10	0	0	0	0	0	0	0	3	12.78	25
3	Pentane	5	12	0	0	0	0	0	0	0	5	14.36	25
4	2-Methylbutane	5	12	0	0	0	0	0	0	0	4	13.81	25
5	2,2-Dimethylpropane	5	12	0	0	0	0	0	0	0	3	12.7	25
6	Cyclopentane	5	10	0	1	0	0	0	1	0	5	16.57	25
7	Methanol	1	4	1	0	1	0	0	0	0	2	29.7	25
8	Ethanol	2	6	1	0	1	0	0	0	0	3	26.14	/
9	1-Propanol	3	8	1	0	1	0	0	0	0	4	24.91	/
10	2-Propanol	3	8	1	0	1	0	0	0	0	3	23.5	25
11	1-Butanol	4	10	1	0	1	0	0	0	0	5	23.73	/
12	2-Butanol	4	10	1	0	1	0	0	0	0	4	22.1	/
13	2-Methyl-1-propanol	4	10	1	0	1	0	0	0	0	4	21.5	/
14	2-Methyl-2-propanol	4	10	1	0	1	0	0	0	0	3	21.7	/
15	1-Pentanol	5	12	1	0	1	0	0	0	0	6	22.3	25
16	2-Pentanol	5	12	1	0	1	0	0	0	0	5	22	25
17	3-Pentanol	5	12	1	0	1	0	0	0	0	5	20.8	25
18	3-Methyl-1-butanol	5	12	1	0	1	0	0	0	0	5	22.5	25
19	2-Methyl-2-butanol	5	12	1	0	1	0	0	0	0	4	20.5	/
20	3-Methyl-2-butanol	5	12	1	0	1	0	0	0	0	4	20.5	25
21	2-Propen-1-ol	3	6	1	1	1	0	1	0	0	4	24.1	/
22	1,2-Ethanediol	2	6	2	0	2	0	0	0	0	4	29.1	/
23	1,2-Propanediol	3	8	2	0	2	0	0	0	0	4	25.8	/
24	1,3-Propanediol	3	8	2	0	2	0	0	0	0	5	33	25

25	1,3-Butanediol	4	10	2	0	2	0	0	0	0	5	23.7	/
26	1,4-Butanediol	4	10	2	0	2	0	0	0	0	6	24.7	25
27	2,3-Butanediol	4	10	2	0	2	0	0	0	0	4	22.7	25
28	1,5-Pentanediol	5	12	2	0	2	0	0	0	0	7	23.5	25
29	Glycerol	3	8	3	0	3	0	0	0	0	5	33.8	/
30	Methyl ether	2	6	1	0	0	0	0	0	1	3	18	/
31	Ethyl vinyl ether	4	8	1	1	0	0	1	0	1	5	16	25
32	Ethyl ether	4	10	1	0	0	0	0	0	1	5	15.1	/
33	1,2-Dimethoxyethane	4	10	2	0	0	0	0	0	2	6	17.6	25
34	Propylene oxide	3	6	1	1	0	0	0	1	1	3	18.8	/
35	1,3-Dioxolane	3	6	2	1	0	0	0	1	2	5	20.9	/
36	1,2-Epoxybutane	4	8	1	1	0	0	0	1	1	3	18.4	25
37	Furan	4	4	1	3	0	0	2	1	1	5	19.2	/
38	Tetrahydrofuran	4	8	1	1	0	0	0	1	1	5	20.3	/
39	p-Dioxane	4	8	2	1	0	0	0	1	2	6	20.72	/
40	Dimethoxymethane	3	8	2	0	0	0	0	0	2	5	17.2	/
41	Acetaldehyde	2	4	1	1	0	1	0	0	0	3	21.1	/
42	Propionaldehyde	3	6	1	1	0	1	0	0	0	4	19.3	25
43	Butyraldehyde	4	8	1	1	0	1	0	0	0	5	18.4	/
44	Isobutyraldehyde	4	8	1	1	0	1	0	0	0	4	17.9	25
45	2-Propenal	3	4	1	2	0	1	1	0	0	4	20	25
46	2-Propanone	3	6	1	1	0	1	0	0	0	3	20.5	/
47	2-Butanone	4	8	1	1	0	1	0	0	0	4	19	/
48	Cyclopentanone	5	8	1	2	0	1	0	1	0	5	21.3	/
49	2-Pentanone	5	10	1	1	0	1	0	0	0	5	18.2	/
50	3-Pentanone	5	10	1	1	0	1	0	0	0	5	18	/
51	Methanoic acid	1	2	2	1	1	1	0	0	0	3	24.8	/
52	Ethanoic acid	2	4	2	1	1	1	0	0	0	3	20.7	/
53	Propionic acid	3	6	2	1	1	1	0	0	0	4	20.5	25
54	Butyric acid	4	8	2	1	1	1	0	0	0	5	21.5	/
55	2-Methylpropanoic acid	4	8	2	1	1	1	0	0	0	4	21.1	25
56	Pentanoic acid	5	10	2	1	1	1	0	0	0	6	19.4	/
57	3-Methylbutanoic acid	5	10	2	1	1	1	0	0	0	5	19.6	25
58	2-Propenoic acid	3	4	2	2	1	1	1	0	0	4	24.5	25
59	Methylacrylic acid	4	6	2	2	1	1	1	0	0	4	22.9	25
60	Acetic acid anhydride	4	6	3	2	0	2	0	0	1	5	21.1	/
61	Methyl methanoate	2	4	2	1	0	1	0	0	1	4	20.9	/
62	Ethyl formate	3	6	2	1	0	1	0	0	1	5	19.2	/
63	Propyl formate	4	8	2	1	0	1	0	0	1	6	18.8	/

64	Butyl formate	5	10	2	1	0	1	0	0	1	7	17.8	/
65	Isobutyl formate	5	10	2	1	0	1	0	0	1	6	16.8	25
66	Methyl acetate	3	6	2	1	0	1	0	0	1	4	19.6	/
67	Ethenyl acetate	4	6	2	2	0	1	1	0	1	5	18.4	25
68	Ethyl acetate	4	8	2	1	0	1	0	0	1	5	18.6	/
69	2-Propenyl acetate	5	8	2	2	0	1	1	0	1	6	18.8	25
70	Propyl acetate	5	10	2	1	0	1	0	0	1	6	18	/
71	1-Methylethyl acetate	5	10	2	1	0	1	0	0	1	5	17.2	/
72	Ethyl propionate	5	10	2	1	0	1	0	0	1	6	17.2	25
73	Methyl 2-propenoate	4	6	2	2	0	1	1	0	1	5	18.21	/
74	Ethyl 2-propenoate	5	8	2	2	0	1	1	0	1	6	17.19	/
75	Methyl methacrylate	5	8	2	2	0	1	1	0	1	5	18	25
76	2(3H)-Dihydrofuranone	4	6	2	2	0	1	0	1	1	5	25.8	/
77	1,3-Dioxolan-2-one	3	4	3	2	0	1	0	1	2	5	30.1	/
78	Propylene carbonate	4	6	3	2	0	1	0	1	2	5	27.3	25
79	Ethyl carbonate	5	10	3	1	0	1	0	0	2	7	18	/
80	2-Methoxyethanol	3	8	2	0	1	0	0	0	1	5	22.1	/
81	2-Ethoxyethanol	4	10	2	0	1	0	0	0	1	6	20.3	/
82	Furfuryl alcohol	5	6	2	3	1	0	2	1	1	5	25.6	/
83	2,2'-Oxybisethanol	4	10	3	0	2	0	0	0	1	7	29.13	/
84	2-(2-Methoxyethoxy)ethanol	5	12	3	0	2	0	0	0	1	8	17.4	/
85	2-Furaldehyde	5	4	2	4	0	1	2	1	1	5	22.9	/
86	Ethyl lactate	5	10	3	1	1	1	0	0	1	6	20.5	/
87	2-Methoxyethyl acetate	5	10	3	1	0	1	0	0	2	7	18.8	/
88	Methyl 3-oxobutanoate	5	8	3	2	0	2	0	0	1	6	21.8	25
89	Cyclohexane	6	12	0	1	0	0	0	1	0	6	16.78	25
90	Hexane	6	14	0	0	0	0	0	0	0	6	14.87	25
91	2-Methylpentane	6	14	0	0	0	0	0	0	0	5	14.36	25
92	3-Methylpentane	6	14	0	0	0	0	0	0	0	5	14.58	25
93	2,2-Dimethylbutane	6	14	0	0	0	0	0	0	0	4	13.73	25
94	2.3-Dimethylbutane	6	14	0	0	0	0	0	0	0	4	14.26	25
95	Cyclohexanol	6	12	1	1	1	0	0	1	0	6	23.3	/
96	1-Hexanol	6	14	1	0	1	0	0	0	0	7	21.9	25
97	2-Methyl-1-pentanol	6	14	1	0	1	0	0	0	0	6	20.8	25
98	2-Methyl-2-pentanol	6	14	1	0	1	0	0	0	0	5	19.6	25
99	4-Methyl-2-pentanol	6	14	1	0	1	0	0	0	0	5	20.5	25
100	2-Ethyl-1-butanol	6	14	1	0	1	0	0	0	0	5	21.5	25
101	Isopropyl ether	6	14	1	0	0	0	0	0	1	5	14.4	/

102	Butyl vinyl ether	6	12	1	1	0	0	1	0	1	7	16.1	25
103	1,2-Diethoxyethane	6	14	2	0	0	0	0	0	2	8	17	25
104	1,1-Diethoxyethane	6	14	2	0	0	0	0	0	2	7	15.9	25
105	Cyclohexanone	6	10	1	2	0	1	0	1	0	6	20.3	/
106	2-Hexanone	6	12	1	1	0	1	0	0	0	6	17.6	/
107	4-Methyl-2-pentanone	6	12	1	1	0	1	0	0	0	5	17.2	/
108	4-Methyl-3-penten-2-one	6	10	1	2	0	1	1	0	0	5	18.4	25
109	Hexanoic acid	6	12	2	1	1	1	0	0	0	7	19.4	25
110	Propionic anhydride	6	10	3	2	0	2	0	0	1	7	20.5	/
111	Ethylene glycol diacetate	6	10	4	2	0	2	0	0	2	8	20.5	/
112	Butyl acetate	6	12	2	1	0	1	0	0	1	7	17.4	/
113	Isobutyl acetate	6	12	2	1	0	1	0	0	1	6	17	/
114	1-methylpropyl acetate	6	12	2	1	0	1	0	0	1	6	16.8	/
115	Ethyl butyrate	6	12	2	1	0	1	0	0	1	7	17.4	/
116	Ethyl oxalate	6	10	4	2	0	2	0	0	2	8	17.6	25
117	2-Butoxyethanol	6	14	2	0	1	0	0	0	1	8	18.2	/
118	Triethylene glycol	6	14	4	0	2	0	0	0	2	10	21.9	/
119	2-(2-Ethoxyethoxy)ethanol	6	14	3	0	1	0	0	0	2	9	19.6	/
120	4-hydroxy-4-methyl-2-pentanone	6	12	2	1	1	1	0	0	0	5	18.8	/
121	2-Ethoxyethyl acetate	6	12	3	1	0	1	0	0	2	8	17.8	/
122	Ethyl 3-oxobutanoate	6	10	3	2	0	2	0	0	1	7	20.1	25

Appendix 4: .csv Files

4.1 Correlation Table, from 'Correlation of Variables.csv'

	#C	#H	#O	UUS	#OH	#CO	#CC	#R	#COC	LC	S
#C	1	0.83	0.01	0.04	-0.19	0.08	-0.02	-0.02	0.12	0.66	-0.46
#H	0.83	1	-0.19	-0.52	0.08	-0.32	-0.37	-0.27	-0.05	0.52	-0.43
#O	0.009	-0.19	1	0.34	0.221	0.52	0.01	-0.07	0.65	0.56	0.38
UUS	0.044	-0.52	0.34	1	-0.43	0.7	0.63	0.45	0.27	0.09	0.06
#OH	-0.19	0.08	0.22	-0.43	1	-0.4	-0.11	-0.21	-0.35	0.02	0.66
#CO	0.084	-0.32	0.52	0.7	-0.4	1	0.05	-0.1	0.22	0.18	-0.05
#CC	-0.02	-0.37	0.01	0.63	-0.11	0.05	1	0.21	0.07	-0.05	0.04
#R	-0.02	-0.27	-0.07	0.45	-0.21	-0.1	0.21	1	0.2	-0.05	0.17
#COC	0.122	-0.05	0.65	0.27	-0.35	0.22	0.07	0.2	1	0.55	-0.1

LC	0.662	0.52	0.56	0.09	0.022	0.18	-0.05	-0.05	0.55	1	-0.12
S	-0.46	-0.43	0.38	0.06	0.658	-0.05	0.04	0.17	-0.1	-0.12	1

4.2 Description Table, from 'Description of Variables.csv'

	#C	#H	#O	UUS	#OH	#CO	#CC	#R	#COC	LC	S
count	123	123	123	123	123	123	123	123	123	123	123
mean	4.5	9.4	1.6	0.80	0.50	0.52	0.15	0.13	0.58	5.2	20
std	1.3	3.0	0.91	0.83	0.68	0.59	0.42	0.34	0.70	1.4	3.9
min	1	2	0	0	0	0	0	0	0	2	12.7
25%	4	8	1	0	0	0	0	0	0	4	17.6
50%	5	10	2	1	0	0	0	0	0	5	19.6
75%	6	12	2	1	1	1	0	0	1	6	21.9
max	6	14	4	4	3	2	2	1	2	10	33.8

Appendix 5: Feature Set Values

5.1 All 5-Feature Set Values

	1st Set LR	1st Set RR	1st Set Lasso	2nd Set LR	2nd Set RR	2nd Set Lasso
R^2	0.71	0.71	0.70	0.79	0.79	0.79
Adjusted R^2	0.69	0.69	0.69	0.77	0.77	0.77
RMSE [(J/mL)^0.5]	1.9	1.9	1.9	1.6	1.6	1.6
Optimal λ value	N/A	0.010	0.010	N/A	0.010	0.010
Intercept / β$_0$	22	22	22	21	21	21
#C Coefficient	-1.4	-1.4	-1.4	-1.1	-1.1	-1.1
#H Coefficient	0	0	0	0	0	0
#O Coefficient	0	0	0	0	0	0
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.5	3.5	3.5	4.6	4.6	4.5
#CO Coefficient	0	0	0	2.2	2.2	2.2
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	3.3	3.3	3.2	4.1	4.2	4.0
#COC Coefficient	0.23	0.23	0.21	0.48	0.48	0.46
LC Coefficient	0.40	0.40	0.39	0	0	0

	3rd Set LR	3rd Set RR	3rd Set Lasso	4th Set LR	4th Set RR	4th Set Lasso
R^2	0.79	0.79	0.79	0.79	0.79	0.79
Adjusted R^2	0.77	0.77	0.77	0.77	0.77	0.77

RMSE [(J/mL)^0.5]	1.6	1.6	1.6	1.6	1.6	1.6
Optimal λ value	N/A	0.010	0.010	N/A	0.010	0.010
Intercept / β_0	21	21	21	21	21	21
#C Coefficient	0	0	0	-1.1	-1.1	-1.1
#H Coefficient	-0.51	-0.51	-0.52	0	0	0
#O Coefficient	0	0	0	2.2	2.2	2.2
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	4.7	4.7	4.6	2.3	2.3	2.4
#CO Coefficient	1.2	1.2	1.2	0	0	0
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	3.0	3.0	2.8	4.2	4.2	4.0
#COC Coefficient	0.47	0.47	0.46	-1.8	-1.8	-1.7
LC Coefficient	0	0	0	0	0	0

	5th Set LR	5th Set RR	5th Set Lasso	6th Set LR	6th Set RR	6th Set Lasso
R^2	0.79	0.79	0.79	0.76	0.76	0.76
Adjusted R^2	0.77	0.77	0.77	0.75	0.75	0.75
RMSE [(J/mL)^0.5]	1.6	1.6	1.6	1.7	1.7	1.7
Optimal λ value	N/A	0.010	0.010	N/A	0.010	0.010
Intercept / β_0	21	21	21	22	22	22
#C Coefficient	0	0	0	0	0	0
#H Coefficient	-0.51	-0.51	-0.52	-0.44	-0.44	-0.45
#O Coefficient	1.2	1.2	1.1	1.1	1.1	1.1
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.4	3.4	3.5	3.7	3.7	3.7
#CO Coefficient	0	0	0	0	0	0
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	3.0	3.0	2.8	2.7	2.7	2.6
#COC Coefficient	-0.77	-0.77	-0.64	0	0	0
LC Coefficient	0	0	0	-0.32	-0.32	-0.28

5.2 All 3 and 4-Feature Set Values

	1st Set LR	1st Set RR	1st Set Lasso	2nd Set LR	2nd Set RR	2nd Set Lasso
R^2	0.78	0.78	0.78	0.69	0.69	0.69
Adjusted R^2	0.77	0.77	0.78	0.68	0.68	0.68
RMSE [(J/mL)^0.5]	1.7	1.7	1.7	2.0	2.0	2.0
Optimal λ value	N/A	0.010	0.010	N/A	0.025	0.025
Intercept / β_0	22	22	22	23	23	23

#C Coefficient	0	0	0	0	0	0
#H Coefficient	-0.55	-0.55	-0.55	-0.62	-0.62	-0.62
#O Coefficient	0.76	0.76	0.75	0.60	0.60	0.68
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.8	3.8	3.8	3.7	3.7	3.6
#CO Coefficient	0	0	0	0	0	0
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	2.6	2.6	2.5	0	0	0
#COC Coefficient	0	0	0	0.16	0.16	0
LC Coefficient	0	0	0	0	0	0

	3rd Set LR	3rd Set RR	3rd Set Lasso	4th Set LR	4th Set RR	4th Set Lasso
R^2	0.68	0.68	0.69	0.69	0.69	0.69
Adjusted R^2	0.67	0.67	0.68	0.68	0.68	0.68
RMSE [(J/mL)^0.5]	2.0	2.0	2.0	2.0	2.0	2.0
Optimal λ value	N/A	0.033	0.033	N/A	0.010	0.010
Intercept / β_0	23	23	23	23	23	23
#C Coefficient	0	0	0	0	0	0
#H Coefficient	-0.59	-0.59	-0.62	-0.62	-0.62	-0.62
#O Coefficient	0.80	0.80	0.67	0.70	0.70	1.2
UUS Coefficient	0	0	0	0	0	0
#OH Coefficient	3.6	3.6	3.6	3.6	3.6	2.9
#CO Coefficient	0	0	0	0	0	0
#CC Coefficient	0	0	0	0	0	0
#R Coefficient	0	0	0	0	0	0
#COC Coefficient	0	0	0	0	0	0
LC Coefficient	-0.088	-0.088	0	0	0	0

	5th Set LR	5th Set RR	5th Set Lasso
R^2	0.55	0.55	0.55
Adjusted R^2	0.54	0.54	0.54
RMSE [(J/mL)^0.5]	2.4	2.4	2.4
Optimal λ value	N/A	0.010	0.010
Intercept / β_0	22	22	22
#C Coefficient	-1.3	-1.3	-1.3
#H Coefficient	0	0	0
#O Coefficient	1.2	1.2	1.2
UUS Coefficient	0	0	0
#OH Coefficient	2.9	2.9	2.9
#CO Coefficient	0	0	0
#CC Coefficient	0	0	0

#R Coefficient	0	0	0
#COC Coefficient	0	0	0
LC Coefficient	0	0	0