CHEM452/CHBE413

Chemical Data Science and Engineering

Homework 4

Due Date: September 25st, 2025

1. **Regression with Cross-Validation** (dataset: “*solubility.csv”*)

In this problem we will integrate two strategies to help reduce overfitting: cross-validation and regularization. The best way to accomplish this is to perform a training/test split on your dataset and then perform K-fold cross-validation within the training set to tune the hyperparameters of your model. Finally, report the model performance with the hyperparameters optimized via cross-validation on the held-out test set.

1. (**2 points)** Check your solubility dataset for any repeated lines, missing values, or obvious outliers. Select only the first 1000 rows of the dataframe for this homework, and then select the following features (columns) for use in your regression models: [*MolLogP,NumHAcceptors,NumHDonors,NumAromaticRings*]. The descriptions of these features may be found here: <https://rdkit.readthedocs.io/en/latest/GettingStartedInPython.html#list-of-available-descriptors>.
2. (**6 points**) Split your 1000 solubility datapoints into 80% training and 20% test using *sklearn.model\_selection.train\_test\_split*. Use the following scikit-learn routines to establish a 5-fold cross validation procedure on the training setusing Lasso regression: *sklearn.cross\_validation.KFold, sklearn.linear\_model.Lasso.* Within each combination of training folds, perform scaling of all feature columns by subtracting the mean, and dividing by the standard deviation, using scikit-learn’s *sklearn.preprocessing.StandardScaler.\**

Make plots of the 5-fold cross-validated average RMSE and R2 as a function of the regularization parameter, alpha (10-3, 10-2, 10-1,100, 101, 102, 103)and provide error bars using the derived standard deviations. Make sure that you have applied the scaler’s inverse transform before reporting these metrics. Report the alpha that results in the best 5-fold cross-validated RMSE and R2. Report the RMSE and R2 of your best model (optimum alpha) on the held-out test set not used during cross-validation. When you report the error on your final held out test set, use the optimal hyperparameters you found using cross-validation to train one model on the entire training set (all K folds) at once – then apply this model to the held-out test set to make the final estimate. How does your cross-validated performance compare to that on your held out test set?

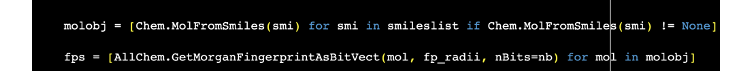
*\*Note – There are (at least) two common errors that occur when someone typically does this. First, the mean and standard deviations used for scaling your columns must only be derived from the data in the K-1 folds used to fit your model. Second, the scaling must also be applied to the features of your held-out fold, prior to applying the regression model. If your RMSE and R2 applied to your held out fold are very poor following training on the K-1 folds, it is likely that you are applying the scalers incorrectly.*

1. (**4 points**) Lasso regression can also be used for feature selection, with the idea that the coefficients corresponding to the least important features will shrink to zero as the regularization parameter, alpha, increases. Based on this interpretation, use the results of (c) to identify the most important molecular feature for predicting solubility. Regression coefficients can be accessed by referencing the .*coef\_* variable in the Python object.
2. (**2 point)** Compute the Pearson correlation matrix for your feature space. What is the most correlated pair of features? What is the least correlated pair of features? Do any pairs exhibit strong anti-correlation? Explain whether these results influence your interpretation of the feature selection results from part (c) in any way?
3. (**2 points**) Describe how you might introduce non-linearity into each of your regression models for this problem (be concrete in your example). What would be the potential benefits and downsides of your proposed strategy?
4. **Molecular Featurization and Tanimoto (aka Jaccard) Similarity (**dataset: ***“****solubility.csv”*)

In problem 1, we used precomputed features for fitting our regression models, but ignored the use of the SMILES representations of each molecule. As we learned in lecture, molecular fingerprints can provide a more general characterization of molecular structure. Here we will use a Morgan fingerprint, which is similar to the Extended Connectivity Fingerprints we learned about in lecture.

1. (**4 points**) A common similarity metric used when comparing molecular structures is the Tanimoto Similarity, defined as

Where xA/B represent the molecular fingerprint bit vector of molecule A/B. Write a function called TanimotoSimilarity() that evaluates the similarity between two molecular fingerprints represented as bit vectors. The following two commands will calculate the fingerprints as bit vectors for you using RDKit:



Use your function to compute the Tanimoto similarity for all pairs of molecules (use only the first 1000 molecules, as in Problem 1) in the dataset and identify the pair of molecules that is most similar and the pair that is least similar and display them in your Jupyter notebook. Also, report the average and standard deviation of all calculated pairwise similarities.