Problem Set 05: Model Evaluation

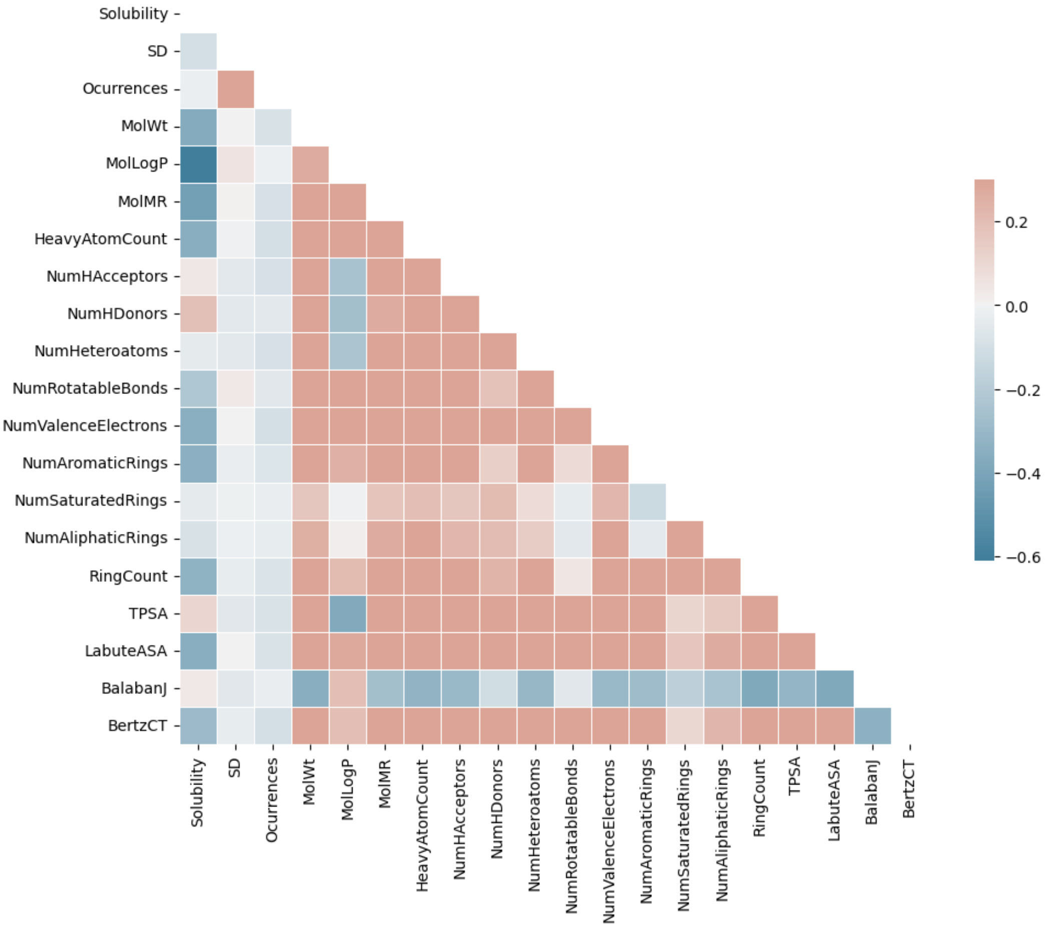
Johannes C. B. Dietschreit, Sascha Mausenberger

Due: 11.11.2024

The problems below are meant to be solved using Python functions and libraries. You can do so by either writing short scripts or using Jupyter notebooks. **Report your results in the Quiz on Moodle** to obtain a grade.

Introduction

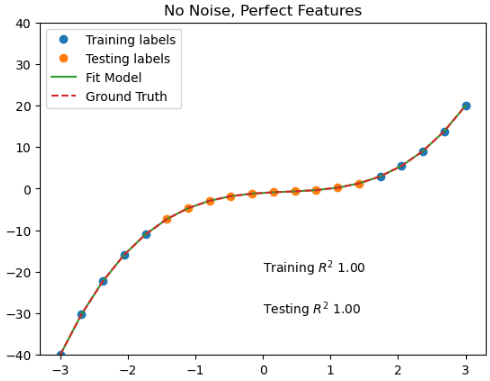
In this PSet you will work on linear regression with and without regularization and model evaluation. You will need a mixture of the packages you have been introduced this far. We will mainly look at the solubility data set again. Here is a correlation plot of all the numerical features. It will help you to understand the relationship between the different numerical, molecular descriptors.

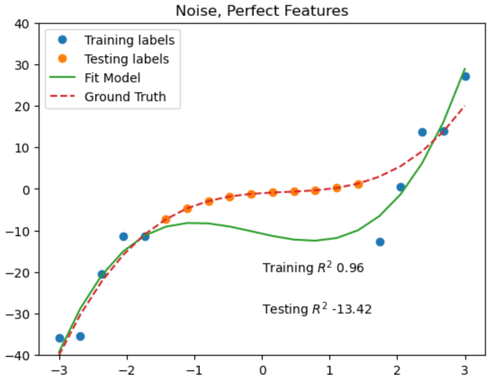


The Problems

1 Noise Linear Regression

In this problem we will deal with synthetic data. Here we have a curve that is created by a higher order polynomial of x. The test points will be those with $|x| < 1.5$. If we fit the outer points without any noise, we perfectly recover the original curve. However, we rarely have access to noiseless data. If we fit the noisy train data with the right order of the polynomial we obtain a curve with a new local minimum, but the fit w.r.t. the test data is significantly worse, the R^2 is negative. But since we know the correct polynomial, this is still one of the best estimate of the original function we can produce. In the real world we usually do not know what polynomial we are trying to fit, so we may have to use high order polynomials to make sure that we included enough orders to fit the data.





1.1 Load the provided train and test arrays. Fit a linear regression model. The fit of the training points should be perfect. Compute the R^2 value (score) on the test set and round the result to 2 digits after the decimal point. (0.5 P)

1.2 Train a ridge regression model with regularization strength of 5 (`alpha=5.0`). Is the score of this model better or worse than the one that was given to you (the one that was fitted on the noisy data with the right polynomial)? (0.5 P)

1.3 Train a Lasso regression model with regularization strength of 5 (`alpha=5.0`). Is the score of this model better or worse than the one that was given to you (the one that was fitted on the noisy data with the right polynomial)? (0.5 P)

1.4 If we do not consider the quality of the fit using R^2, but MAEs and RMSEs, which one produces the better fit for the provided date (LinearRegression, RidgeRegression, LassoRegression)? (1.0 P)

2 Linear Regression

This problem focuses on fitting the solubility column from the `solubility_dataset.csv`.

2.1 Load the data set. Separate all the numerical columns from the ones containing text (no InChI, no SMILES, etc.). Assign the `Solubility` column to a variable y (our target label) and all other columns to our examples $\mathbf{X}$. How many features are in $\mathbf{X}$? (0.5 P)

2.2 Use the scikit-learn function `train_test_split` to split the data set 80:20 in train and test. Use `random_state=42`. Since all features have different physical units (or no units), we need to scale them. Apply the `StandardScaler` from scikit. Make sure to fit the scaler only on your train data before applying it to the test. What is the mean of the normalized `X_test` matrix (features×#number)? Round your answer to 3 digits after the decimal point. (0.5 P)

2.3 Train a LinearRegression model. Plot the predicted (y-axis) vs the true (x-axis) test labels. What does the plot look like? (0.5 P)

What is the feature with the largest absolute coefficient (weight), i.e., which feature is used the most to predict the solubility? (0.5 P)

2.4 To improve generalization, train a RidgeRegression model with a regularization strength of alpha=10.0. Which is the only measure (out of: PearsonR, SpearmanR, R^2, MAE, RMSE, MAPE) that does NOT improve in comparison to the non-regularized model (LinearRegression not RidgeRegression)? (1.0 P)

2.5 If you take a look at the coefficients of the Ridge regression model. What has changed and which parameter has now the largest magnitude compared to the non-regularized linear regression? (0.5 P)

2.6 In order to eliminate nonsensical or linear dependent features, the L^1 regularization of Lasso has to be used. Train a Lasso model with a regularization strength of alpha=1.0. Which is the only non-zero coefficient? (0.5 P)

3 Is it FDA approved?

For studying the effectiveness of a classification model, we will use a data set that contains the information whether a compound is toxic and if it is FDA approved.

3.1 This time, our data does not come with pre-computed descriptors. We only have the SMILES string. We will use rdkit to convert the SMILES string into a molecule, and then we compute a set of descriptors for each molecule.

Some of the molecules will fail to be converted. You will have to remove them. You need to remember which ones were deleted, since you need to remove the failed molecules from the labels in the pandas table.

Use the following code get the featurization of the molecules

```
fpgen = AllChem.GetRDKitFPGenerator()

features = np.array([fpgen.GetFingerprint(x).ToList() for x in valid_mols])
```

What is the size (product of number of features and number of samples) of your feature matrix? (0.5 P)

3.2 Split features and labels in to train and test sets using the familiar scikit-learn function. Use random_state=42 again. Train a LogisticRegression model on the FDA approved label. Use the default settings.

Compute the accuracy of your model (the ratio of (TP+TN)/all). Round your result to 3 digits after the decimal point. (0.5 P)

3.3 What if you used a naive model (instead of training one) that always predicts the same label, namely the one that is more common in the train data set. Would that have a higher accuracy? Round your result to 3 digits after the decimal point. (0.5 P)

3.4 Since we are trying to predict whether a chemical is going to be FDA approved, and therefore harmless, we should not make any mistakes and label something potentially dangerous as likely harmless. Which measures out of |accuracy, precision, recall, F1 score, false positive rate, specificity (true negative rate), balanced accuracy give our logistic regression a better score than the "all approved" model? (0.5 P)

3.5 You cannot compute Matthews correlation coefficient for a model that does not predict any negatives, since the denominator is 0. However, Cohen's Kappa can be computed, and it is surprisingly small for both the trained and the "all approved" model. Report the higher value of the two Cohen's Kappa values. Round your result to 3 digits after the decimal point. (0.5 P)

3.6 Since we have trained a logisitc regression model, it returns internally probabilities for the class membership. The attribute `predict` just picks the class label with the highest probability, which in our case would be the threshold of 0.5. That means, whenever the model is at least certain to 50% that a drug would be approved, the model predicts "approved". However, we might want to be more conservative and want to choose a threshold of 0.6 or higher, to make sure we have as few false positives as possible. The opposite would be the "all approved" model, which basically had a threshold of 0.0.

You can access the predicted probabilities using `predict_proba`. Using only the second column, the one that pertains to the "allowed" label. Let us scan all threshold values in `np.linspace(0.0, 1.0, 2001)`for the predictions done for `X_test`. Compute the tuple (FlasePositiveRate, TruePositiveRate) for each of the models. Sort the results by false positive rate and plot them. Is the AUC (area under the curve) bigger or smaller than 0.5? (1.0 P)