Practical 1: Linear Regression

Joanna Chung, Emily Chen
joannachung@college.harvard.edu, emily-chen@college.harvard.edu
Camelot usernames: jcat1, emchen

February 9, 2018

1 Technical Approach

In constructing and testing our models, we first split the given "training" data set in two, yielding a training set and a validation set. This new training set comprised roughly 80% of the original data set, and the remaining 20% became the validation set, on which we evaluated the different models we constructed.

First Iteration

On the first iteration of our approach, we made use of Python's Scikit-learn package and tested various types of linear regression models as well as other model classes with the same given amount of data:

- Ridge Regression: To construct this model we performed 5-fold cross validation on the training data, iterating through a list of values of λ and selecting the optimal λ that minimizes the Ridge loss function. We iterate through $\lambda = 0.001, 0.005, 1, 5, 10, 50, 100, 500, 1000$ and use $\lambda = 5$ for the L_2 penalty term in our model.
- Lasso Regression: We constructed this model in the same way we performed Ridge regression: perform 5-fold cross validation and select the optimal λ that minimizes the Lasso loss function. In our Lasso regression model, we take $\lambda = 0.001$ for the L_1 penalty term.
- Elastic Net: Combining L_1 and L_2 regularization, we attempted to construct an elastic net model. For this model, we use $\lambda = 1$ with a ratio of penalty terms of 0.5, meaning L_1 and L_2 are weighted equally in this model.
- Neural Network: Here we used a multi-layer perceptron regressor that optimized for the squared-loss using stochastic gradient descent.
- Bayesian Ridge Regression: Here we introduced λ as a random variable to be estimated by the data, and thus tune the regularization constant over uninformed priors. The output y (gap value) was assumed to follow a Gaussian distribution.

The RMSE values of the validation set for each of the aforementioned models are denoted in Table 1 below:

Model	RMSE
Baseline 1: Linear Regression	0.29846
Baseline 2: Random Forest	0.27209
Model 3: Ridge Regression ($\lambda = 5$)	0.29880
Model 4: Lasso Regression ($\lambda = 0.001$)	0.30040
Model 5: Elastic Net	0.40704
Model 6: Neural Network	0.27627
Model 7: Bayesian Ridge Regression	0.29880

Table 1: First iteration of models and their validation set RMSE values

Even after tuning the hyperparameters of our model classes during our first iteration, we found that Random Forest was the best model (as indicated by Table 1). Thus, we turned to methods of feature engineering to further develop our model.

Second Iteration

On the second iteration, we turned to using rd-kit in order to extract additional amounts of data regarding each molecule. We extracted a total of 1024 additional factors, namely the Morgan Fingerprint of each molecule, and added them to the data set.

The RMSE values of the validation set for each of the aforementioned models are denoted in Table 2 below:

Model	RMSE
Model 1: Linear Regression	0.14149
Model 2: Random Forest	0.10931
Model 3: Ridge Regression	0.14144
Model 4: Lasso Regression	0.17319
Model 5: Elastic Net	0.40713
Model 6: Neural Network	0.06878
Model 7: Bayesian Ridge Regression	0.14146

Table 2: Second iteration of models and their validation set RMSE values

Final Iteration

On the final iteration, we used rd-kit to extract the number of atoms in each molecule, and added it to the data set.

The RMSE values of the validation set for each of the aforementioned models are denoted in Table 3 below:

Model	RMSE
Model 1: Linear Regression	0.13843
Model 2: Random Forest	0.10266
Model 3: Ridge Regression	0.13830
Model 4: Lasso Regression	0.16647
Model 5: Elastic Net	0.39826
Model 6: Neural Network	0.06556
Model 7: Bayesian Ridge Regression	0.13831

Table 3: Final iteration of models with additional features and their validation RMSE values

2 Results

The final model we used was random forest trained on the given data in addition to the Morgan fingerprints and number of atoms for each molecule that we extracted. This model yielded the lowest validation RMSE value, which was a good indicator of how our model would perform on the test data. We obtained a validation RMSE value of 0.05950 running on our entire training set (roughly 800,000 molecules).

After training this model and using it to predict y (or gap) values for the given test set, we uploaded our predictions to Camelot. We obtained a test set RMSE value of 0.05958, beating both the baseline linear regression (0.27209) and random forest models (0.29846) by a significant margin. This result from our final iteration - including the column of number of atoms per molecule - was the best score we achieved, although the second iteration of adding Morgan fingerprint features beat both baselines already. The fact that the test RMSE was close to our RMSE value of 0.0590 during model selection from our validation set also confirmed that the our cross-validation method allowed for

3 Discussion

In summary, we began our search for the best model by first using ridge regression and lasso regression to tune the parameters of our model, in addition to the linear regression and random forest models provided. We used 5-fold cross validation to tune the parameters of our ridge and lasso models, including the selection of $= 5$ and $\lambda = 0.001$ as the penalty weights for ridge and lasso regression, respectively. We tested our models on the validation set (which we had separate from the original given training set), and the best model turned out to be random forest. These did not bring us over the baseline RMSEs, which motivated us to try feature manipulation.

Next, we extracted over a thousand additional features per molecule using the RDKit Morgan fingerprint, and ran our models on this extended dataset. This was very effective, and brought us over the baseline linear and random forest RMSEs. Here, the best model was also random forest. We also experimented with the elastic net, neural network, and bayesian ridge regression models, but the random forest model still came out to be the best suited.

Even after having beat the baselines, we knew we could do even better, so to improve our model further, we turned to our knowledge of chemistry to add a column of number of atoms in each molecule, which is closely tied to the energy band gap. This gave us our best results, and our random forest model test set RMSE improved from 0.1097 to 0.05950, setting us at the top 15% of our peers in terms of ranking.

For future experimentation, our implementation of the neural network showed promising results. With fine-tuning of the parameters and modification of the model after gaining deeper knowledge of the algorithm and parameter implications, it may become the model with the best results. The RMSE in our basic implementation was already 0.06556, very close compared to our random forest of 0.05950.