Prediction of GS Energies of Small Organic Molecules

Willmor Peña

Final Project

Machine Learning in Molecular Science

Spring 2019

The Problem

There are copious amounts of molecules whose ground state energies would have to be calculated to determine appropriate use in applications (i.e drug discovery). Possibly years of calculations would be needed to obtain the energies of all molecules.

A machine learning model can be trained to predict the ground state of a given molecule.

In this project multiple models are tested to determine which model is able to predict the energies better.

MOLPRO

QUANTUMESPRESS

Q-CHEM™

A QUANTUM LEAP INTO THE FUTURE OF CHEMISTRY

The Data

The dataset was obtained from Kaggle. However the group who made the data available obtained the structure of the molecules from PubChem. The data consists of 16242 molecules each containing atoms C, H, O, N, P, S. and each molecule consists of 50 atoms or less.

The features are flattened entries of the upper triangular part of the Coulomb matrix.

Dimension of entire dataset is 16242 by 1275

Features:

Coulomb matrix entries (1275)

Labels:

Ground State Energy obtained from DFT calculations using Quantum-Espresso (16242)

Energies Distribution in Training Set

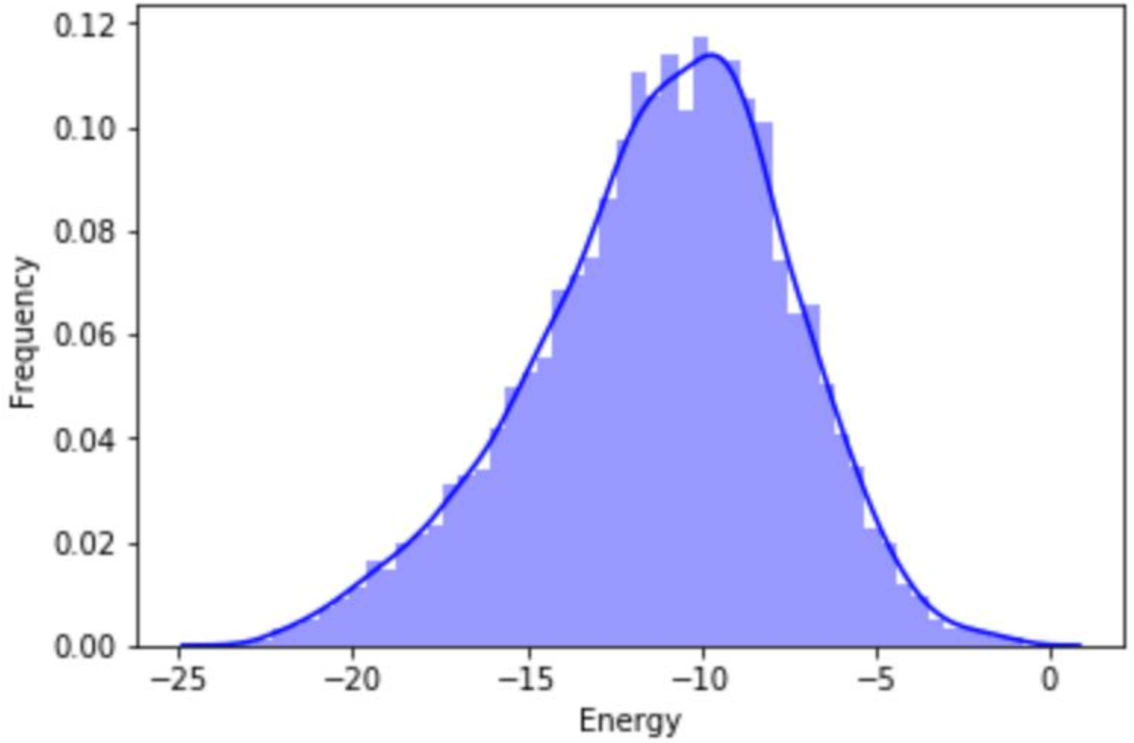


Figure 1: Distribution of labels (Energies), 20 % of data set is set aside for testing.
The remaining dataset is split into 80/20 training /validation when doing cross validation and hyper parameter fine tuning

Energies Distribution in Test Set

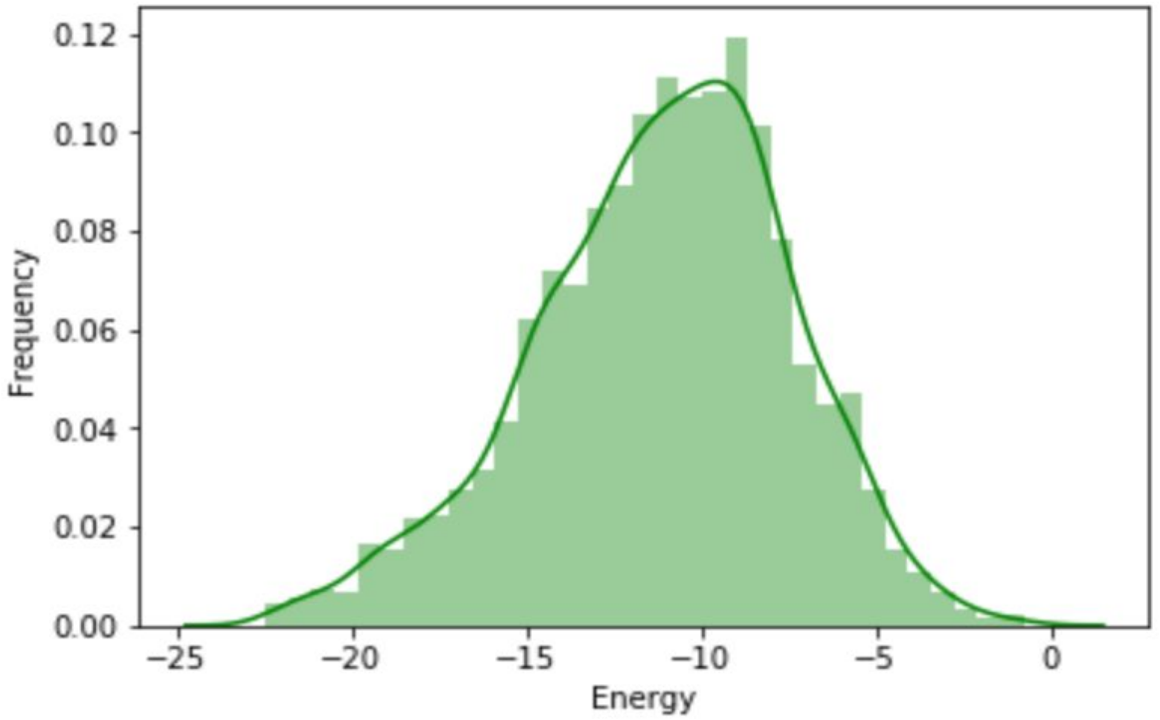


Figure 2: Distribution of labels (Energies), in the test set, distribution is similar to training set.

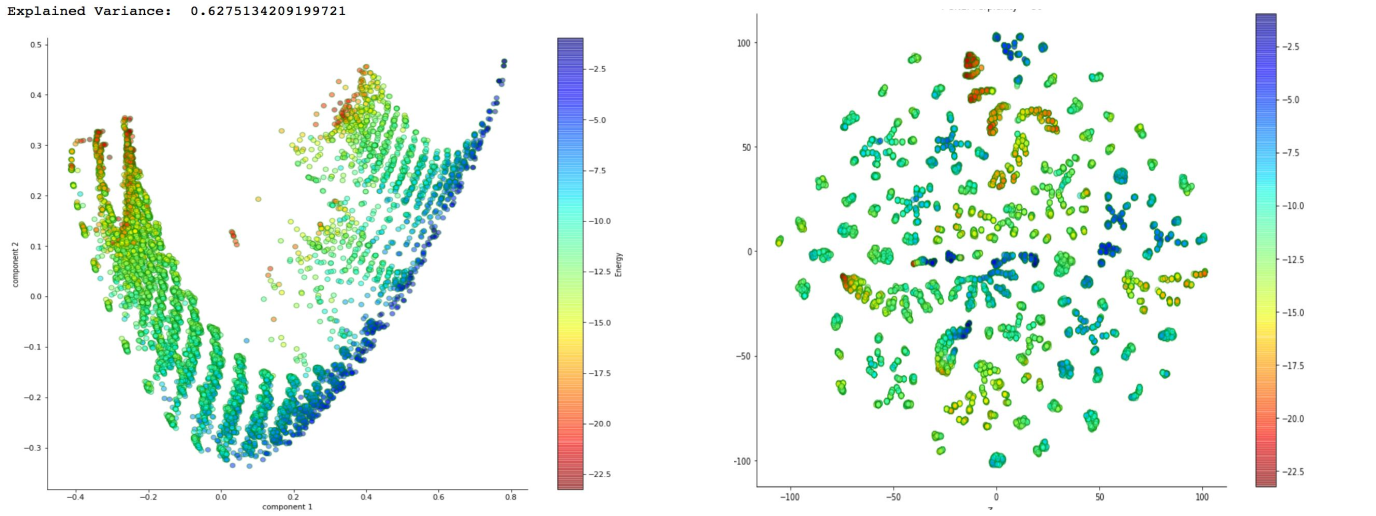
Coulomb Matrix

$$C_{I,J} = \begin{cases} -0.5 Z_I^{2.4} & I = J \\ Z_I Z_J / |R_I - R_J| & I \neq J \end{cases}$$

The Internuclear Coulomb repulsion operator. Where Z's are the nuclear charges, R's are the atomic coordinates, and indices I,J run over the atoms in a given molecule. The off-diagonal terms correspond to Coulomb repulsion between atoms I and J and the diagonal terms encode a polynomial fit of the energies vs. the nuclear charge (Rupp et. al). The features only include the upper triangle part of the matrix. For molecules that consist of less than 50 atoms; their matrices are padded with zeros to maintain a consistent feature dimension.

Preprocessing and Visualization

Dataset features were normalized, and then PCA and T-SNE were performed to visualize features



```
Explained Variance:  0.6275134209199721
```

Figure 3: PCA and T-SNE dimensionality reduction for visualization of the entries of the upper triangle of the Coulomb matrix. The resulting reduced features were not used for training in this project but they could be used if dataset were much larger.

Training models

Linear Models:

1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Elastic Net Regression

Ensemble Models:

1. Random Forest Regression
2. Kneighbors Regression
3. XGBoost Regression

Neural Networks

First, the linear models and ensemble models were trained with the entries of the Coulomb matrix. Having found that the random forest and Xgboost regressor performed the best they were selected for further comparison. A cross validation was performed for both models using both the coulomb matrix and the eigenvalues of the matrix. In both cases the XGBoost regressor performed the best and it was selected for fine tuning. Finally a NN model was trained for comparison.

Eigenspectrum Representation

A problem with the representation of a molecule by its coulomb matrix is that there is not a unique ordering of the atoms, thus by switching the order of the atoms up to d! different but equivalent Coulomb matrices can be obtained (d is the dimension of the square matrix). A solution to this problem proposed by Rupp et. al is to calculate the eigenvalues of the coulomb matrix and ordering them from greatest to lowest. The eigenvalues are invariant to the permutations of the Coulomb matrix. The eigenvalues of the coulomb matrix for each molecule was readily available in the group's github repository.

$$C\mathbf{v} = \lambda\mathbf{v}$$

$$(\lambda_1, \ldots, \lambda_d), \qquad \lambda_i \geq \lambda_{i+1}$$

PCA of Eigenvalues of Coulomb Matrix

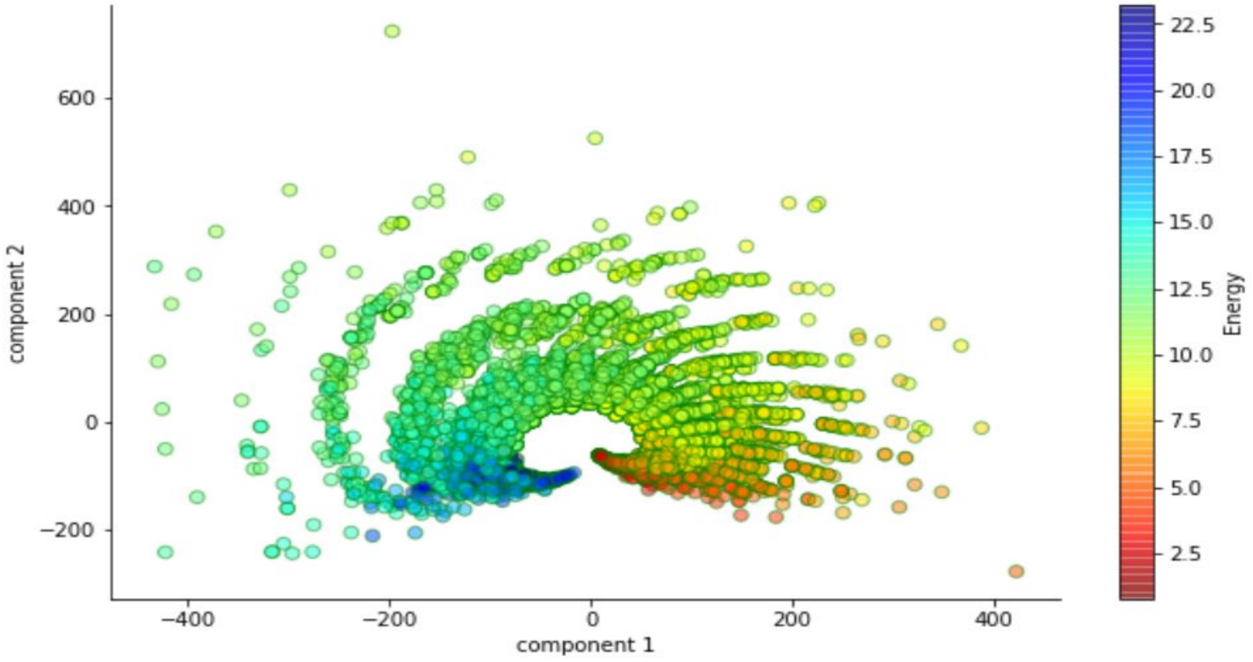


Figure 4: PCA of Eigenvalues of Coulomb matrix. Eigenvalues are used as features in the training of random forest Xgboost regressors, and neural networks

Hyperparameter Fine-tuning

Having read the eigenvalues of the coulomb matrix, they were used as features in the hyperparameter fine-tuning process. This was done as to reduce the training time while finding the best set of hyper-parameters. Furthermore, a randomized search was done as opposed to a grid search. Likewise, this was done to reduce the total number of fits that would have to be done if a full-on gridsearch was performed.

- ❑ *n_estimators (600)*
- ❑ *Min_child_weight (5)*
- ❑ *max_depth (10)*
- ❑ *learning_rate (0.045)*
- ❑ *gamma (0.01)*
- ❑ *colsamply_bytree (0.3)*
- ❑ *alpha (10)*

← Best hyperparameters found by the randomized search

After finding the best hyperparameters, a xgboost regressor was trained with the coulomb matrix entries and a lower prediction error was achieved

Results

Model	**RMSE**	**MAE**
Linear	20.946	2.465
Ridge	0.827	0.553
Lasso	1.577	1.169
ElasticNet	1.694	1.255
K-Neighbors	0.615	0.364
Random Forest	0.188	0.102
XGBooost*	0.148	0.079
NeuralNetwork	0.243	0.108

Table 1: The RMSE and MAE of each model's predictions and true energies of the test set are used as metrics for model performance. Values reported are for models trained with Coulomb matrix entries. *Error reported for XGboost is for the model trained using the hyperparameters found by the randomized grid search.

Eigen Spectrum vs Coulomb Matrix

Model	**RMSE Full features**	**RMSE Eigenvalues**
XGBoost	0.148	**0.151**
Random Forest	0.188	**0.194**
Neural Networks	0.202	**0.317**

Table 2: Comparison of RMSE of test set and predictions from models trained with full features and eigenvalues of Coulomb matrix. The error is expectedly higher for the models trained with the eigenvalues but not significantly higher. Furthermore, training and fine-tuning the model with the eigenvalues takes less time and less computation resources.

Neural Network

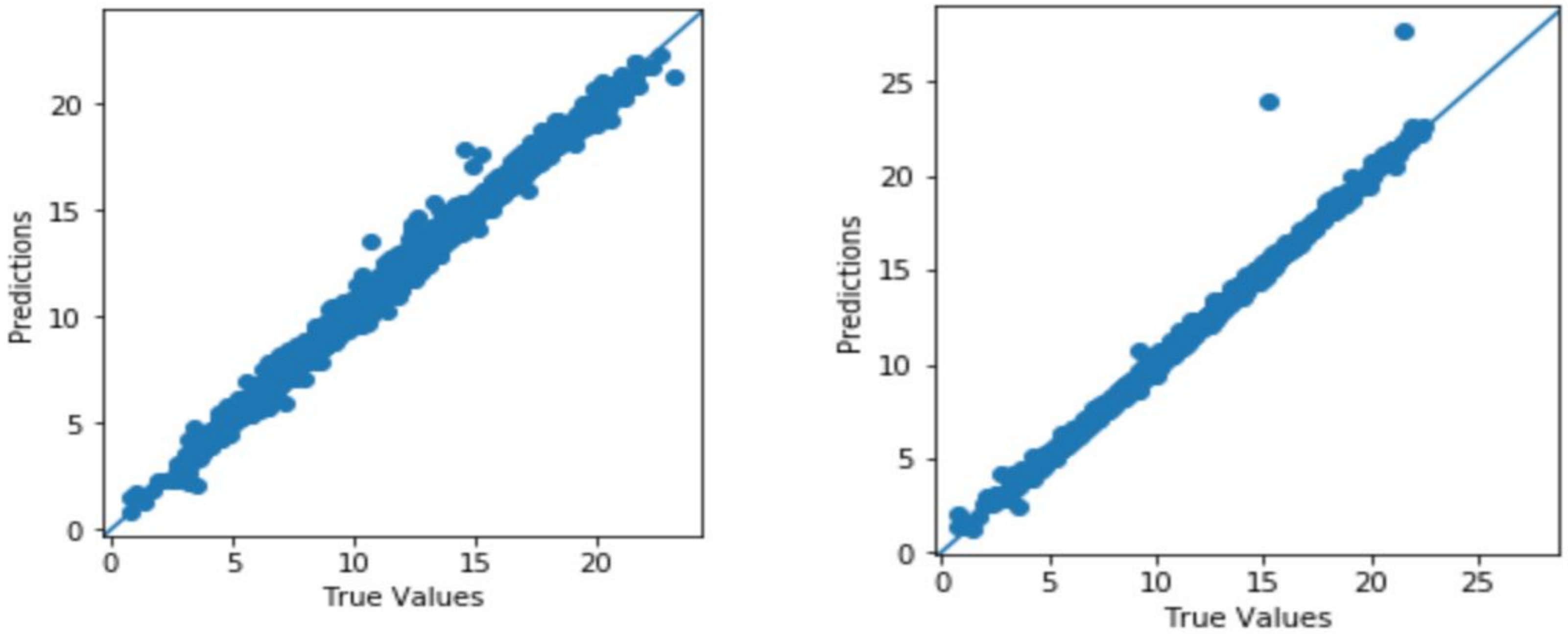
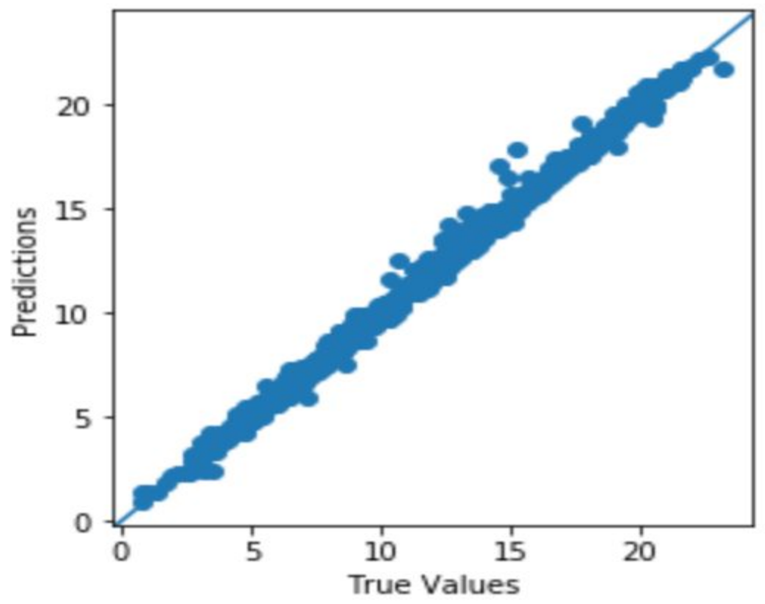


Figure 4: Neural Network model predictions for the test set of the Eigenvalues of the Coulomb matrix (left) and the Coulomb matrix (right).

Random Forest



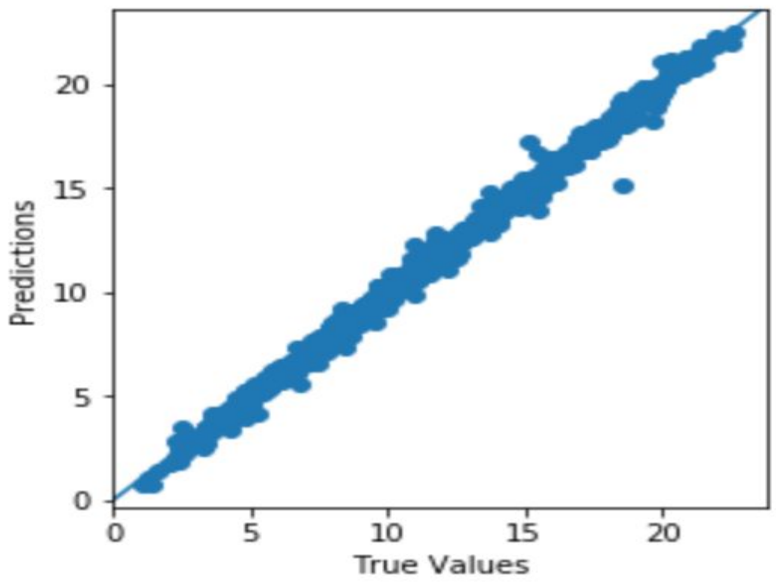
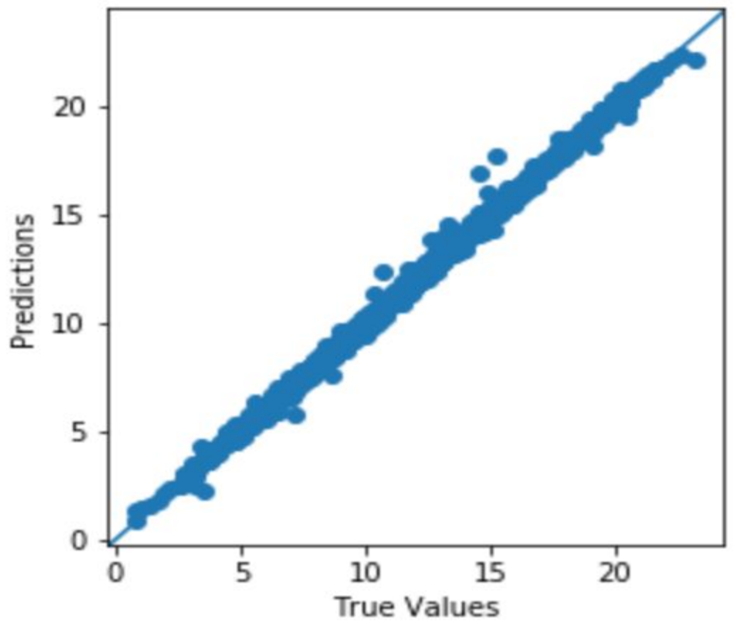


Figure 5: Random Forest model predictions for the test set of the Eigenvalues of the Coulomb matrix (left) and the Coulomb matrix (right).

XGBoost



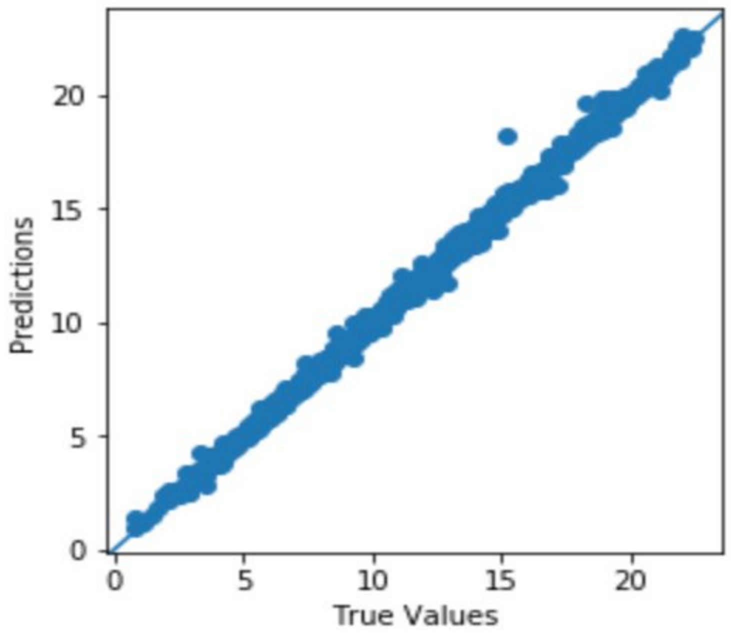


Figure 6: XGBoost model predictions for the test set of the Eigenvalues of the Coulomb matrix (left) and the Coulomb matrix (right).

Conclusion

The ambiguous representation of molecules by the Coulomb matrix was addressed by using the eigenvalues of the matrix as features. The errors achieved are considerably small and predictions are off by about 0.151 Rydberg in average (approximately 0.25 Kcal/mol)

Using this reduced dataset resulted in models prediction errors to increase by 2% but the training time decrease significantly, 90%, when using the eigenvalues.

The XGboost regressor model performed the best. Furthermore, training this model took less time than the random forest and the neural network models.

References

Data Sources:

1. https://www.kaggle.com/burakhmmtgl/energy-molecule/downloads
2. https://github.com/bhimmetoglu/RoboBohr/tree/master/data

Papers:

1. Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Phys. Rev. Lett. 108, 058301
2. Burak, Himmetoglu, J. Chem. Phys. **145**, 134101 (2016)
3. Katja Hansen, Grégoire Montavon, Franziska Biegler, Siamac Fazli, Matthias Rupp, Matthias Scheffler, O. Anatole von Lilienfeld, Alexandre Tkatchenko, and Klaus-Robert Müller. Journal of Chemical Theory and Computation **2013** *9* (8), 3404-3419

Tutorials:

1. https://xgboost.readthedocs.io/en/latest/tutorials/model.html
2. https://www.tensorflow.org/tutorials/keras/basic_regression