Report: Using decision tree regression with different maximum tree depths to predict the redox potential of molecules

1 Introduction

The application domain of this ML problem is materials science, which is a combination of physics, chemistry, and engineering. It is a research field of properties, designs, and applications of materials. Conventional theory, experimentation, and computation in traditional materials science can be time-consuming and costly [1]. To accelerate the process of materials research, materials scientists are utilizing machine learning, which has led to promising applications in the recent years, for example, the prediction of materials properties [2, 3].
I have been writing my bachelor's thesis at Aalto University, and this project is related to my thesis topic "Machine learning in materials science". Materials are made up by molecules. My thesis includes a short demonstration in which ML models are trained on a dataset of molecular properties to predict the redox potential (i.e., oxidation / reduction potential), which is a molecular property that can affect the properties of materials. The dataset is also used for this course project.

In the report, Section 2 (Problem Formulation) explains the data points, features, and label of this ML problem, and the source of the dataset. Section 3 (Methods) describes and discusses the dataset, feature selection, chosen ML models, loss function, and model validation. Section 4 (Results) discusses the results obtained from the chosen ML models. Section 5 (Conclusion) summarises the report and discuss the limitation of the methods. The code can be found in the Appendix section.

2 Problem Formulation

Each data point is a molecule, which is a group of atoms. Atoms are made up by three main types of particles and one of them is electron. Electrons in atoms have spatial distributions, called orbitals, and each orbital has an orbital energy. Positions and energies of electrons in molecules can be described in molecular orbitals [4].
The type of all data is decimal and data can be either positive or negative. For each molecule, there are 9 numerical features (first 9 columns in the table below) described in the list below.

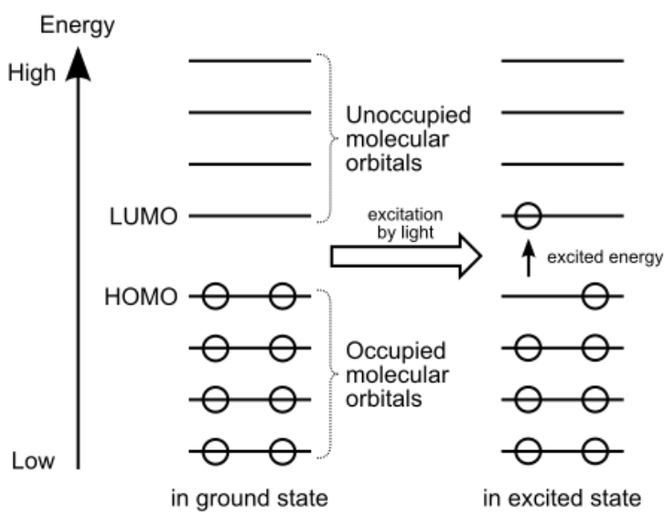


Figure 1: Diagram of HOMO (Highest Occupied Molecular Orbital) and LUMO (Lowest Unoccupied Molecular Orbital) of a molecule [5].

1. HomoA, the Highest Occupied Molecular Orbital (HOMO) of a molecule A. HOMO is the highest-energy molecular orbital that electrons occupy.

2. LumoA, the Lowest Unoccupied Molecular Orbital (LUMO) of a molecule A. LUMO is the lowest-energy molecular orbital that doesn't have any electrons in it.
3. GapA is the energy difference between HOMO and LUMO of molecule A.
4. HomoAH is the HOMO of molecule AH, which is A bounded by one hydrogen atom.
5. LumoAH is the LUMO of molecule AH.
6. GapAH is the energy difference between HomoAH and LumoAH of molecule AH.
7. HomoAH2 is the HOMO of molecule AH2, which is molecule A bounded by two hydrogen atoms.
8. LumoAH2 is the LUMO of molecule AH2.
9. GapAH2 is the energy difference between HomoAH2 and LumoAH2 of molecule AH2.

The goal is to train the models to predict the redox potential of molecules, so the label is the redox potential of a molecule (i.e., the last column in the table below). Redox potential is a numerical measure of how easily a molecule will accept electrons or lose electrons.

	HomoA	LumoA	GapA	HomoAH	LumoAH	GapAH	HomoAH2	LumoAH2	GapAH2	redoxpot1
0	-7.9848	-3.6582	4.3265	-7.5462	-3.5189	4.0272	-6.0675	-0.4076	5.6599	0.9121
1	-6.6740	-3.3543	3.3197	-6.6104	-3.0729	3.5374	-5.8577	-0.2250	5.6327	0.8415
2	-6.3660	-3.3184	3.0476	-6.2117	-3.0280	3.1837	-5.9100	-0.3589	5.5510	0.8053
3	-7.1440	-3.6882	3.4558	-6.8373	-3.5720	3.2653	-5.9666	-0.4971	5.4694	0.9603
4	-7.5521	-3.5521	4.0000	-7.2218	-3.3578	3.8640	-5.9178	-0.1763	5.7415	0.8734
...	...	...	...	...	...	...	...	...	...	...

Figure 2: A part of the input dataset. It is also shown in the beginning of the code in the Appendix.

3 Methods

3.1 Dataset

There are 1989 data points and no missing data. The dataset was generated by quantum chemical computation by the Computational Chemistry research group at Aalto University. Unfortunately, since it is still an internal dataset, access to the dataset needs to be granted by the researcher who owns the dataset. A part of the dataset is shown in the beginning of the code in the Appendix.

3.2 Feature selection

For this ML problem, feature selection is based on domain knowledge. HOMO, LUMO, and Gap (the energy difference between HOMO and LUMO) of a molecule are relevant for its value of redox potential [4], so they are chosen as features. ID number and molecular structures are not relevant for the prediction of redox potential, so they were not chosen as features. Theoretical physics and chemistry are not the focus here because they are not relevant for this ML course.

3.3 Models

In stage 2 of the project, the model was Decision Tree Regressor of Scikit-learn [6] with maximum tree depth of three. In stage 3, the models are Decision Tree Regressor with different maximum tree depths: from 4 to 10. In other words, for the Decision Tree Regressor model, the parameter `max_depth` = 3, 4, ..9, 10. A machine learning method is a combination of model and loss function [7]. Different tree depths results in different models (i.e., hypothesis spaces) and, as a result, different ML methods. The motivations behind this choice are:

- The interaction plots between some features and label seems to be non-linear, so non-linear decision tree was tested.
- Because redox potential (i.e., label/output of this problem) is a numerical value, regression is more reasonable than classification.
- According to some research publications [2, 3], non-linear decision tree-based algorithm is one of the most relevant method in materials science.
- According to the practical user guide on Scikit-learn [8], initially `max_depth` can be set to 3, so that the user can get to know how the tree fits to the data, and then the depth can be

increased. However, if the chosen maximum depth of the tree is too high, it overfits the data [8].

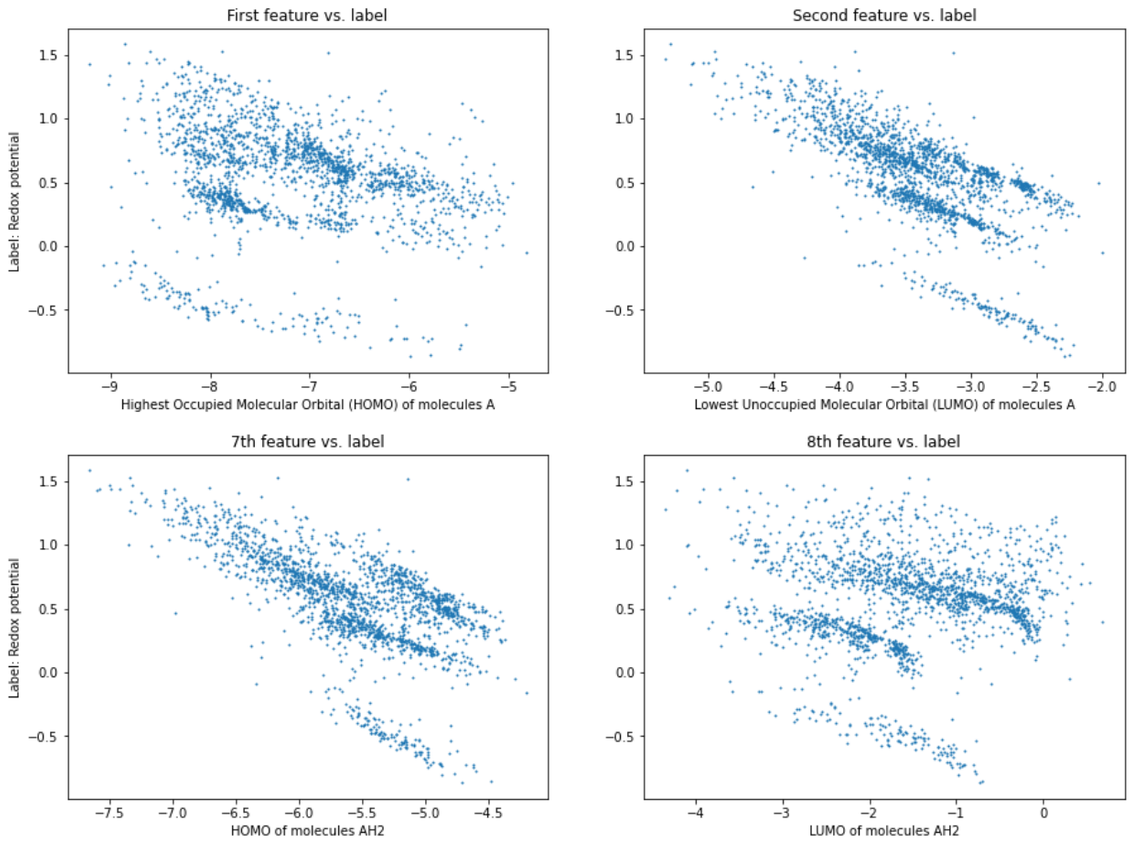


Figure 3: Interaction between some features and label.

3.4 Training, validation, and test set

Although there is no golden ratio for the test set, as a common practice for dataset that has several thousand data points, 20% of the dataset was used as test set. The rest (80%) was used for k–fold cross validation. The purpose of k–fold cross validation is to try validating the models on a different dataset (especially when the dataset is small) and to avoid overfitting [7]. Moreover, the decision tree model is fairly simple and fast, so using k–fold cross validation is not time-consuming. 5 folds (splits) was tested because it is the default number of folds in the library of scikit-learn [9]. The sizes of training set and validation set is default of *KFold* object, which can be found in the document of KFold on Scikit-learn [9].

3.5 Loss

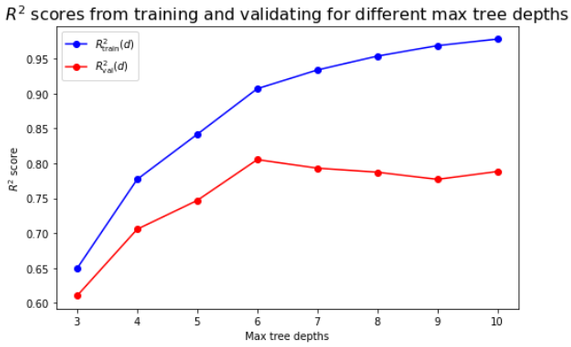
Mean squared errors (MSE) of both training set and validation set was calculated because it is a simple, easy to understand, and common error for decision tree regression. The average validation error of k folds is a better estimation of the expected loss than the validation error obtained from just a single split [7].

In addition, R^2 scores (coefficient of determination) from training and validating each tree depth were obtained because R^2 score tells how accurate the predictions for training set and validation set are. Moreover, R^2 score calculation is convenient because it can be obtained from the method `score` of the Decision Tree Regressor on Scikit-learn library. The better and accurate the prediction is, the closer R^2 is to 1.0 [6].

4 Results

Mean squared errors from training and validating set for different max tree depths are shown in the left figure. R^2 scores from training and validating set for different max tree depths are shown in the right figure. Blue line represents training set and red line represents validation set. As expected, when the tree grows bigger (tree depth from 3 to 6) and predict the data better, the training errors

and validation errors start to decrease, and the R^2 scores increase. However, when the max tree depth becomes 7, the training errors and validation errors start to increase, and the R^2 scores start to decrease. This is when the tree starts to overfit the data. Therefore, the reasonable choice of model is the decision tree regression with maximum tree depth of 6, which gave an average validation error of 0.03.



Explanation for how the test set was constructed is in section 3.4. For the test set, average squared error and R^2 score were obtained for to make it consistent with the training and validation set. From the test set, mean squared error was approximately 0.01 and R^2 score was approximately 0.94. They are also displayed in the code in the Appendix.

5 Conclusion

In conclusion, if the decision tree is too small, the prediction is less accurate, as shown by the lower R^2 score and higher mean squared error of smaller trees. On the other hand, if the tree becomes too complex (i.e., max tree depth increases), the model starts to overfit the data. The obtained results from the final test set are good.

The decision trees have some disadvantages. According to [8], too complex trees do not generalise the data well and result in overfitting. Small variations in the data might lead to a completely different tree. Therefore, the proposed solution is testing decision trees in an ensemble, for example, random forest. Last but not least, biased trees might be generated if some classes dominate. Thus, the dataset should be balanced before training with the decision tree.

References

[1] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, "Data-driven materials science: Status, challenges, and perspectives," *Advanced Science*, vol. 6, no. 21, p. 1900808, Sep 2019. [Online]. Available: http://dx.doi.org/10.1002/advs.201900808

[2] T. Mueller, A. G. Kusne, and R. Ramprasad, "Machine learning in materials science: Recent progress and emerging applications," *Reviews in computational chemistry*, vol. 29, pp. 186–273, 2016.

[3] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, pp. 1–36, 2019.

[4] "Frontier MOs: An Acid-Base Theory," https://chem.libretexts.org/@go/page/53522, Aug 13 2020.

[5] "HOMO and LUMO," https://en.wikipedia.org/wiki/HOMO_and_LUMO.

[6] "Scikit-learn, decision tree regressor," https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html.

[7] A. Jung, *Machine Learning: The Basics*. Springer Nature, 2021.

[8] "Decision trees," https://scikit-learn.org/stable/modules/tree.html#tree.

[9] "Kfold," https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html.

Appendix