Data Analysis Project: Solar Perovskite Prediction

Group 4
Michael Shealy,
Lushuang Zhang,
Bryent Lee

I. Introduction

Project Background

The rise in global warming and the non-renewable energy crisis have made hydrogen fuel an alternative fuel. Methods such as photocatalytic water splitting are being researched to produce hydrogen gas, a clean-burning fuel. Photocatalytic water splitting has the simplicity of using a synthesizable perovskite catalyst; a type of photocatalyst with suitable bandgap and sunlight to produce hydrogen out of water. However, there is currently no technological commercialized way for photocatalytic water splitting. Predicting the stability of the perovskite structure remains a long-standing challenge. Most approaches capable of addressing this challenge require several computationally demanding electronic-structure calculations for each material composition, limiting their use to a small set of materials.[1]

Perovskite structure in general takes the form of ABX_3, where the cations, A and B can span the periodic table and the anion, X, is typically a chalcogen or halogen. Based on past researches, the Goldschmidt tolerance factor, t has been used extensively to predict the stability of the perovskite structure based only on the chemical formula, ABX_3 and the ionic radii, r□ , of each ion (A, B, X). However, its accuracy in recent years is insufficient because of the increase in the number of new perovskite design due to the fact that it has a variety of applications.[1] Using the data from a 2019 research paper that includes a derived feature, new tolerance factor, τ and the oxidation states, n□ of each ion (A,B,X), we can create a new model that takes into account all these 8 features and predicts the synthesizability of any perovskite structure.

Project Overview

The provided dataset that is used for modeling in this project includes partially complete information on synthesizability and bandgap. In order to predict an efficient solar water splitting system, the following objectives will be fulfilled:

1. Use data management strategies to deal with missing values in the provided datasets.
2. Create separate machine learning models to predict synthesizability and band gap of all candidates.
3. Validate the prediction for both synthesizability and band gap using the same set of testing data.

II. Methodology

The modeling for this project mainly dealt with three sections: data management, model training and model validation. For each section, a brief discussion of the methodology used is mentioned below.

Data Management

1. Predict missing values in features “tau” and “t”.
2. Structure data into two separate feature matrices with different observations.
3. Remove outliers that are 3 standard deviations away from the mean of the data.

Model Training

1. For the synthesizability dataset, two classification models (decision tree and random forest) were used
2. For the bandgap data set, Kernel Ridge Regression (KRR) was used as the baseline model. Random forest is used for an improved model due to inadequate performance from the KRR model.

Model Validation

1. For validating each individual model, k-fold cross validation was used combined with GridSearchCV to optimize the hyperparameters of the model.
2. For validating the combined models, the number of data points with both synthesizability and bandgap data were compiled. 50% of these data points were held out from both models and used as the final validation step.

Below is a diagram to further illustrate the process of creating the models.

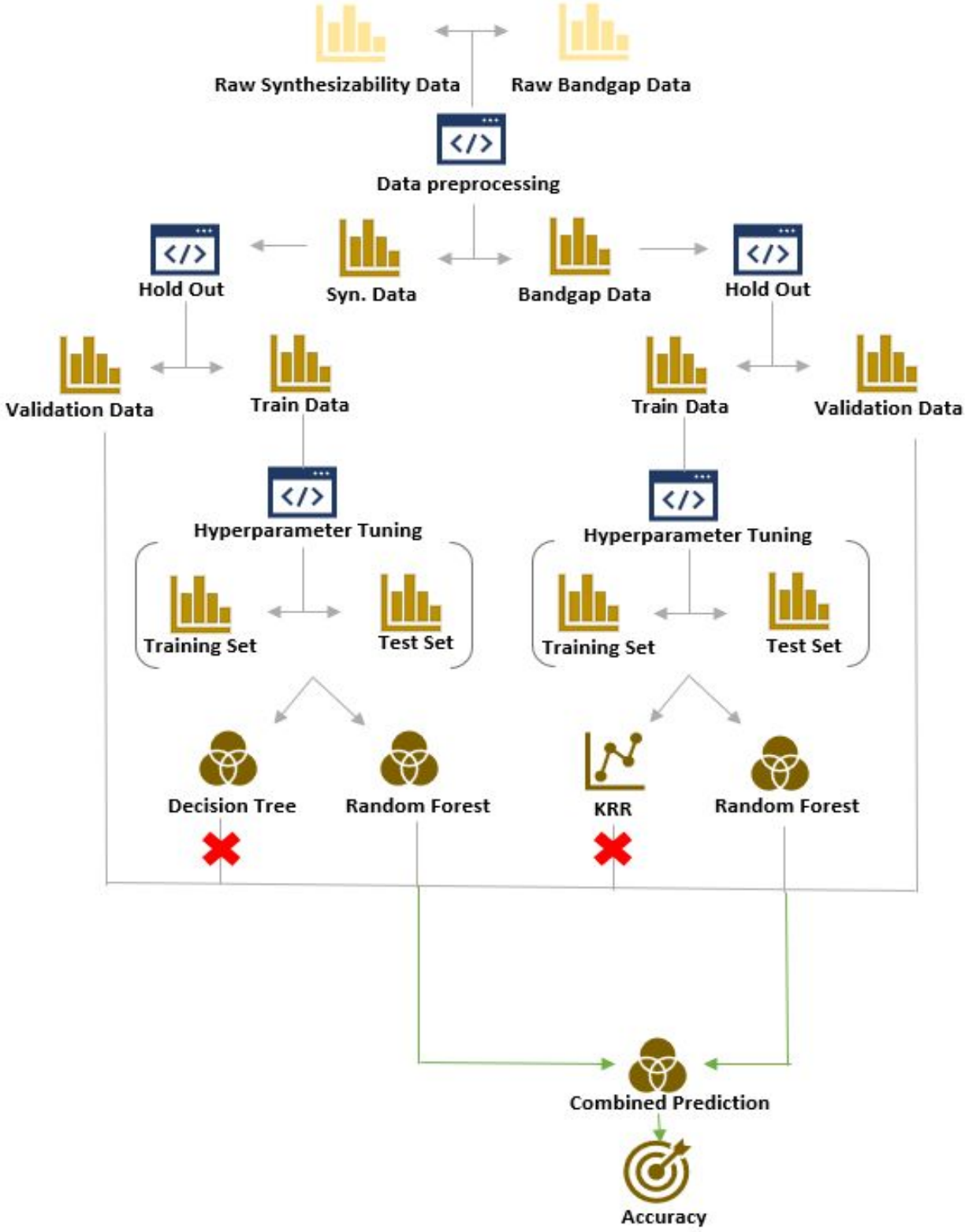

Figure 1: Block Diagram

III. Results and Discussions

Data Management

Data management was first conducted to structure the raw data into useful information for data training and validation. The raw data given from the references [1,2] consist of information on various properties for a total of 685 candidate materials. The first step is to deal with missing values, given that they are all expressed as questions marks in the given dataset. It is also known that the question marks are only present in the columns for t, tau, synthesizability (exp_label) and band gap. The values of t and tau can be predicted using the equations below.[1] Any data points where elements A and B were the same have to be dropped, since this creates arithmetic errors in the 'tau' equation.

$$t = \frac{r_A + r_X}{\sqrt{2}(r_B + r_X)}$$

$$\tau = \frac{r_X}{r_B} - n_A\left(n_A - \frac{r_A/r_B}{\ln(r_A/r_B)}\right)$$

After filling in the missing values for t and tau, we want to reserve some observations where both synthesizability and band gaps data are known for final validation. This hold-out portion will act as testing set for the combined predictions of the two models. Next, we structured the data set into two sets of matrices with the same 8 features (columns) but different observations (rows) based on known values for synthesizability and band gap. This is to prepare both models with adequate data, as summarized in Table I below. The total observations added up by the last three datasets is larger than the total observation in the initial given dataset, which is expected because some observations have both synthesizability and bandgap data, and are used for both. Next, observations outside 3 standard deviation are dropped as outliers for all features, based on the assumption that all observations are normally distributed. It was verified the data removed is no greater than 5% of the original data and thus are dropped for better prediction, also as shown in Table I.

Table I. Summary of the Processed Data

Dataset	#Features	#Observations
Original	**8**	**685**
Tau/t Filled-In	**8**	**679**
Hold-out Data	**8**	**83**
Synthesizability	**8**	**476**
Band gap	**8**	**166**

Model 1: Classification on Synthesizability

In the baseline model, Decision Tree was initially selected for modeling because of its efficiency and relatively simple design compared to other non-linear models. To further validate this method, we used GridSearchCV for hypertuning the max_depth while validating the model's performance since GradSearchCV has a built-in k-fold validation.

For our improved model, Random Forest was used for potential better accuracy. GridSearchCV was also applied, the results are as shown:

Table II: Performance Results for Synthesizability Classification

	Decision Tree	Random Forest
Accuracy on Training Data	**0.997**	**0.994**
Accuracy on Testing Data	**0.866**	**0.908**
False Positive Rate	**10.1 %**	**6.7 %**
False Negative Rate	**3.4 %**	**2.5 %**

For the synthesizability classification, both of the tested classifiers managed to successfully model the data and gave high accuracies on both the training and the testing data sets. While both Decision Tree and Random Forest gave accuracies around 90%, Decision Tree had a higher accuracy for the training set whereas Random Forest had a marginally higher accuracy for the testing set. The reason for this performance essentially comes down to the different logistics behind these two classifiers. While the Decision Tree uses the entire given data to train and build a decision mechanism that classifies the final results, Random Forest randomly selects a subset of the given data, builds a decision tree for each and summarizes the results by letting multiple trees 'vote' collectively. This additional process of Random forest of randomly breaking down the given data helps reduce variance and thus can always give a better accuracy in predicting testing data. As our final goal is to build a model that is able to predict synthesizability for untested compounds, we would prefer Random forest as our final improved model for predicting synthesizability.

Model 2: Classification of Bandgap Data

The baseline approach used to predict the bandgaps of compounds was Kernel Ridge Regression. This was used along with GridSearchCV to validate the model using k-fold cross validation and to optimize the hyperparameter values. The model was first used with an array of magnitudes for the hyperparameter values, and then the optimal hyperparameter values were used in subsequent iterations of GridSearchCV to optimize the values. A 5-fold cross validation (CV=5) approach was used after comparing the results for different amounts of folds.

```
alphas = np.array([1e-7, 1e-6, 1e-5, 1e-4, 1e-3,1e-2, 0.1, 1., 10., 100.])
gammas = np.array([1e7, 1e6, 1e5, 1e4, 1e3, 1e2, 10., 1., 0.1, 0.01, 0.001, 0.0001, 0.00001])
parameter_ranges = {'alpha':alphas, 'gamma':gammas}

KRR = KernelRidge(kernel='rbf')

KRR_search = GridSearchCV(KRR, parameter_ranges, cv=5)
KRR_search.fit(Xgap, ygap)
KRR_search.best_estimator_, KRR_search.best_score_
```

```
C:\Users\CaleShealy\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:813: DeprecationWarning: The default of the
`iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results wh
en test-set sizes are unequal.
  DeprecationWarning)
```

```
(KernelRidge(alpha=1.0, coef0=1, degree=3, gamma=0.1, kernel='rbf',
             kernel_params=None), -0.20210604845017985)
```

```
alphas = np.array([0.2, 0.4, 0.6, 0.8, 1., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.])
gammas = np.array([0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.])

parameter_ranges = {'alpha':alphas, 'gamma':gammas}

KRR_search = GridSearchCV(KRR, parameter_ranges, cv=5)
KRR_search.fit(Xgap, ygap)
KRR_search.best_estimator_, KRR_search.best_score_
```

```
C:\Users\CaleShealy\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:813: DeprecationWarning: The default of the
`iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results wh
en test-set sizes are unequal.
  DeprecationWarning)
```

```
(KernelRidge(alpha=0.2, coef0=1, degree=3, gamma=0.2, kernel='rbf',
             kernel_params=None), -0.16102866052820375)
```

```
alphas = np.array([0.12, 0.14, 0.16, 0.18, 0.2, 0.22, 0.24, 0.26, 0.28, 0.3])
gammas = np.array([0.12, 0.14, 0.16, 0.18, 0.2, 0.22, 0.24, 0.26, 0.28, 0.3])
parameter_ranges = {'alpha':alphas, 'gamma':gammas}
KRR_search = GridSearchCV(KRR, parameter_ranges, cv=5)
KRR_search.fit(Xgap, ygap)
KRR_search.best_estimator_, KRR_search.best_score_
```

```
C:\Users\CaleShealy\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:813: DeprecationWarning: The default of the
`iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results wh
en test-set sizes are unequal.
  DeprecationWarning)
```

```
(KernelRidge(alpha=0.26, coef0=1, degree=3, gamma=0.22, kernel='rbf',
             kernel_params=None), -0.15890706545741903)
```

Even after multiple iterations of GridSearchCV and narrowing down the hyperparameter ranges, the score does not get much better. One reason for this is because of the absence of scaling the data. Scaling could lead to a better regression model, however it would have required dropping the ‘nX’ and ‘rX’ features from the feature matrix. This is because all the data points in these two features were the same value. It was determined to keep the two features instead of dropping them and scaling the data because the good practice of using the same features in both models was deemed more important than scaling the data for one of the models.

Based on the terrible performance of the regression model, the improved approach was to use a Random Forest algorithm to design a classification model of whether a compound would fall within the desired band gap range for solar water splitting. The bandgap data was categorized into whether the values fall within the desired range of 1.23 eV and 3.2 eV to create two separate classes of data. The model was also used with GridSearchCV so that k-fold cross validation could be used and so that the hyperparameter values of the model could be optimized. A 5-fold cross validation approach was used after comparing the results for different amounts of folds. After multiple iterations of the Random Forest model, the model accuracy averaged out to about 90%. This was achieved after making the max depth values range from 2-20 and the n-estimators values range from 2-100. The best max depth and best n-estimators varied from iteration to

iteration, but the accuracy remained the same. The amount of false negatives was anywhere between five and ten times as many as the amount of false positives.

Table III: Performance Results for Band Gap Classification

	KRR	Random Forest
Accuracy on Training Data	**-0.15**	**-**
Accuracy on Testing Data	**-**	**0.897**
False Positive Rate	**-**	**1.8 %**
False Negative Rate	**-**	**8.4 %**

As seen in the results above, the baseline model could not get a positive r^2 value, even after multiple iterations. The improved model is the Random Forest model used to classify whether a compound will have a band gap within the desired range for solar water splitting. Although this model used accuracy as its performance metric instead of r^2, an accuracy of 90% for the algorithm is a much better performance compared to the r^2 values the regression model had. This is because it is a lot easier to predict whether a compound will have a bandgap that will fall within a certain range, compared to predicting the exact band gap value a compound will have.

In context with the objective of this model, an accuracy of 90% means that the model will be correct 90% of the time it predicts whether a compound will or will not fall within the desired band gap range. This means that if the model predicts that a promising perovskite material will have a desired band gap value, there is a 90% chance that the material will actually have a desired band gap value. Likewise, there is a 90% chance that a material will not have a desired band gap value if the model predicts that it won't. The average rate of false negatives was about 8%, which was significantly higher than the average rate of false positives. This means that the model is more likely to predict that a compound will not have a bandgap within the desired range when it actually does, than predicting a compound will have a bandgap within the desired range when it actually does not.

Final Validation Step

Table III: Performance Results for Combined Validation

	Synthesizability	Band Gap
Accuracy on Testing Data	**0.857**	**0.845**
False Positive Rate	**10.7%**	**2.4 %**
False Negative Rate	**3.6%**	**13 %**

For the final validation step, the data that was held out at the beginning was used on both the models together to determine if they make accurate predictions on whether the compounds

are useful for solar water splitting. The synthesizability model had an accuracy of 85.7% on the final validation data. The rates and ratio of false negatives and false positives was also consistent with the rates and ratio from the training and testing data. These are good indicators that the model did not overfit the data and is a useful model to predict the synthesizability of compounds.

Next, the final bandgap validation data was classified into whether it fell into the desired band gap range or not. This classification was used with the final validation feature matrix in the bandgap classification model. The band gap model had an accuracy of 84.5% on the final validation data. The rates and ratio of false negatives and false positives was also consistent with the rates and ratio from the training and testing data. Like from the synthesizability model, these are good indicators that the model did not overfit the data and is a useful model to predict whether or not a compound will have a band gap within the desired range for solar water splitting.

IV. Conclusion and Recommendations

Both classification models have an accuracy score of greater than 90% on their respective testing data, and an accuracy of 85% on the final validation set. The overall ability to predict whether a compound is useful for solar water splitting is based on whether the assumption of bandgap value and perovskite synthesizability are independent is valid or not. If the assumption is valid, then the accuracies of the two models can be multiplied together to have an overall accuracy (if the final validation set accuracies are used, the accuracy would be 72.4%). This means that if the models predict that a compound will form a perovskite and have a bandgap within the desired range, then there is a 72.4% chance that the model is correct. If the independence assumption is not valid, however, then a relationship between perovskite synthesizability and bandgap value must be used to combine the results of the two models. This relationship would likely require the information from the confusion matrices from both models, so that the amount of positive and negative values could be used in said relationship.

To determine if these models are accurate enough for use, an economic analysis of these experiments must be done. This should take into account the cost of resources for these experiments and whether the accuracies are sufficient enough to produce useful perovskites without wasting resources. The higher amount of false negatives compared to false positives means that more compounds that would be useful perovskites would be missed by the model than compounds that would not be useful perovskites, but are predicted to be so by the models. This would mean the model is more conservative with classifying compounds as useful perovskites, which could be economically beneficial. Overall, the models successfully predict whether a compound will form a useful perovskite with a significant amount of accuracy, and further economic analysis should be done to determine if the models should be used.

References:

1. I. E. Castelli, T. Olsen, S. Datta, D. D. Landis, S. Dahl, K. S. Thygesen, and K. W. Jacobsen, “Computational screening of perovskite metal oxides for optimal solar light capture,” Energy Environ. Sci., vol. 5, no. 2, pp. 5814–5819, 2012.

2. C. J. Bartel, C. Sutton, B. R. Goldsmith, R. Ouyang, C. B. Musgrave, L. M. Ghiringhelli, and M. Scheffler, “New tolerance factor to predict the stability of perovskite oxides and halides,” Science Advances, vol. 5, no. 2, p. eaav0693, Feb. 2019.

Individual Contributions

It is agreed that each group members contribute approximately equal amounts of the work in the project. Michael mainly worked on the bandgap model including regression model as baseline model as well as the final validation of the dataset. Lushuang and Bryent mainly focused on the data management and classification of the synthesizability model. The write-up of the report is completed with all three of the group members.