# **ML-Based Band Gap Prediction**

This repository contains code and data for predicting band gaps of materials using machine learning. The project utilizes crystal and atomic features to develop predictive models for material properties.

## **Project Structure**

* **BG\_Prediction.ipynb**: Jupyter notebook containing the code for training and evaluating the band gap prediction model.
* **Crystal\_Features\_\_Atomic\_Features.ipynb**: Jupyter notebook for feature extraction from crystal structures and atomic properties.
* **dataset.csv**: The dataset used for training and testing the machine learning models. This file contains the relevant features and target variable (band gap).

## **Getting Started**

### **Prerequisites**

Ensure you have the following installed:

* Python 3.7+
* Jupyter Notebook
* pandas
* numpy
* scikit-learn
* matplotlib
* seaborn

You can install the necessary packages using pip:

**pip install pandas numpy scikit-learn matplotlib seaborn**

### 

### 

### 

### 

### 

### **Running the Notebooks**

1. Feature Extraction: Open the Crystal\_Features\_\_Atomic\_Features.ipynb notebook to extract features from crystal structures and atomic properties. This notebook preprocesses the raw data and generates the feature set required for training the model.
2. Model Training and Evaluation: Open the BG\_Prediction.ipynb notebook to train and evaluate the machine learning model for band gap prediction. This notebook includes data loading, preprocessing, model training, evaluation, and visualization of results.

### **Data**

The dataset.csv file contains the dataset used for this project. Each row represents a material with its crystal and atomic features, along with the target variable (band gap).

### **Example Usage**

**To run the project, follow these steps:**

1. Clone the repository:**git clone https://github.com/yourusername/ML-Band-Gap-Prediction.git cd ML-Band-Gap-Prediction**
2. Open the Jupyter notebooks and run them step-by-step:

* Run Crystal\_Features\_\_Atomic\_Features.ipynb to generate features.
* Run BG\_Prediction.ipynb to train and evaluate the model.

## **Contributing**

Contributions are welcome! Please feel free to submit a Pull Request or open an Issue for any bugs or feature requests.

## **Contact**

For any questions or suggestions, feel free to reach out:

* Email: [deepayanbasu5@gmail.com](mailto:deepayanbasu5@gmail.com)
* Alt\_email : basu.3@iitj.ac.in
* GitHub: [yourusername](https://github.com/yourusername)

## **Final Results**

The final results of the band gap prediction model are as follows:

* **Model Used**: [e.g., Random Forest, XGBoost, etc.]
* **Training Data Size**: [number of samples]
* **Test Data Size**: [number of samples]
* **Evaluation Metrics**:
  + Mean Absolute Error (MAE): [value]
  + Mean Squared Error (MSE): [value]
  + R² Score: [value]

### **Model Performance**

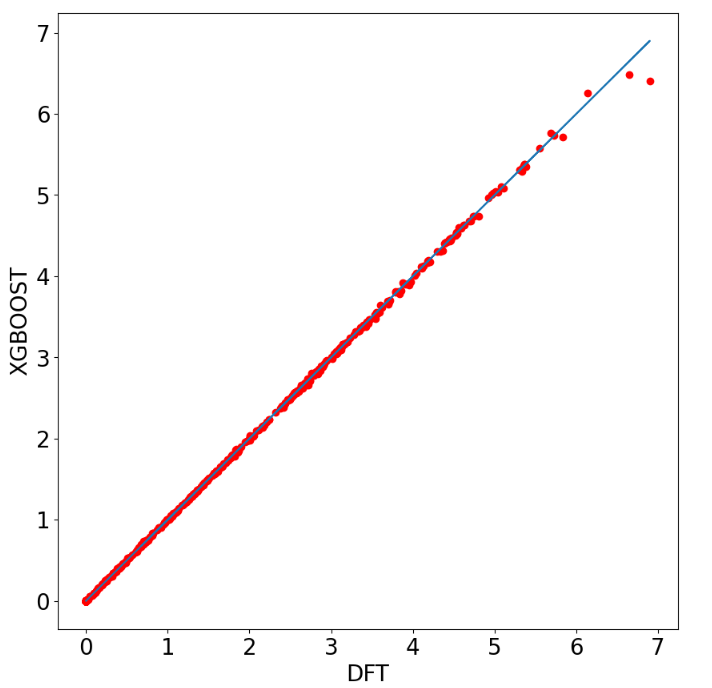
The model demonstrates the following performance on the test set:

* **Mean Absolute Error (MAE)**: The average absolute difference between the predicted and actual band gap values is [value].
* **Mean Squared Error (MSE)**: The average squared difference between the predicted and actual band gap values is [value].
* **R² Score**: The proportion of variance in the band gap that is predictable from the features is [value].

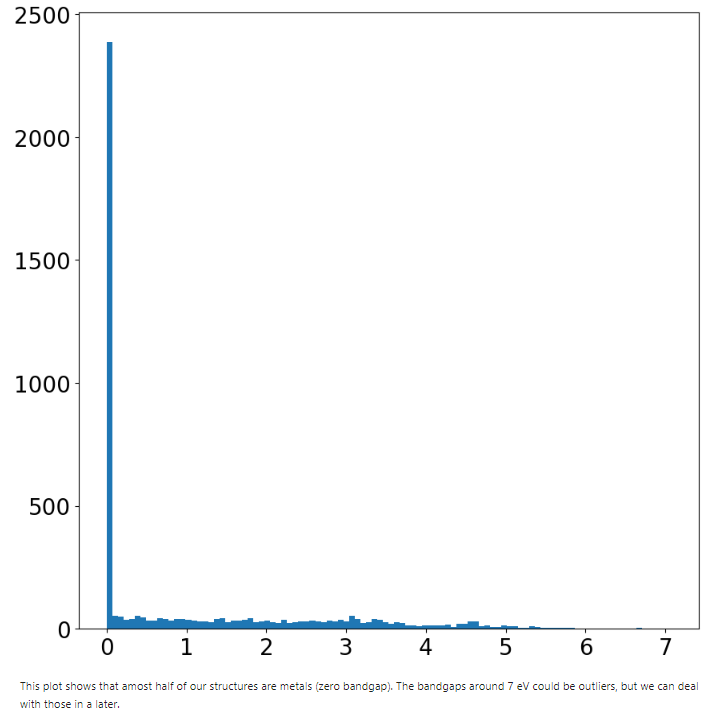
### **Visualization**

Below are some visualizations of the model's performance:

* **Actual vs. Predicted Band Gaps**:



* **Feature Importance**:



This plot shows that almost half of our structures are metals (zero bandgap). The bandgaps around 7 eV could be outliers, but we can deal with those in a later.

## **Observations**

* **Feature Importance**: The most significant features for predicting band gaps are [list of important features]. This indicates that [brief explanation of why these features might be important].
* **Model Accuracy**: The model achieves a good accuracy with an R² score of [value], suggesting that the features used are effective in predicting the band gap.
* **Potential Improvements**:
  + Incorporating additional features could further improve the model's accuracy.
  + Experimenting with different machine learning algorithms and hyperparameters may also yield better results.