BTECH PROJECT PRESENTATION

ON

“Machine Learning with DFT Based Prediction of Band Gap & Phase Stability Analysis of Perovskite Structures”

Under guidance of

Professor Appala Naidu Gandi

and TA : Abhishek Sir

PRESENTED BY

Anamika Choudhary (B21MT004)

Table of Contents

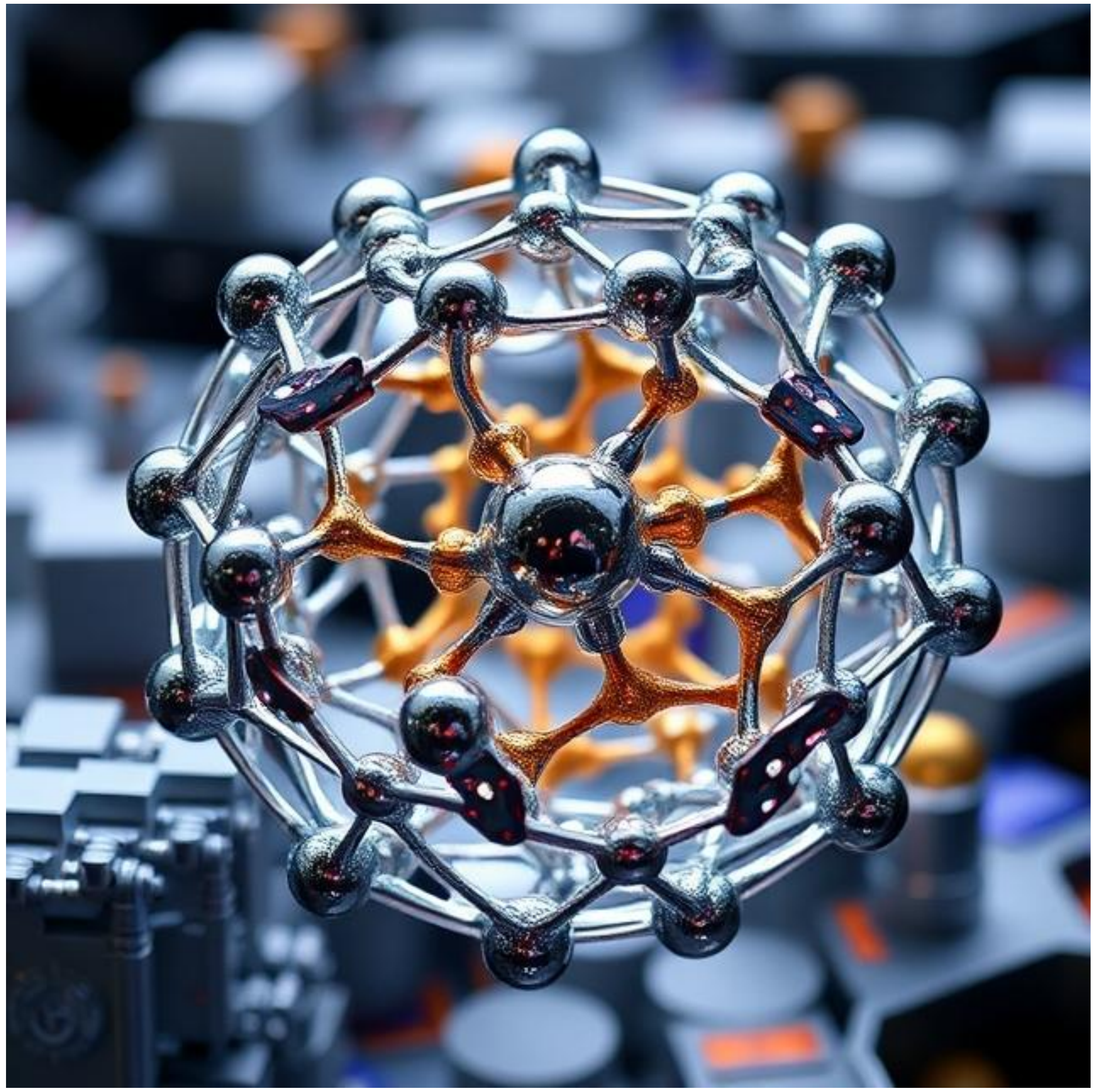
Project Overview

1. Project Overview

- **Objective**: Predict band gaps of materials using machine learning based on compositional and structural properties.
- **Motivation**: Accelerate materials discovery by providing a faster alternative to traditional Density Functional Theory (DFT) calculations.

2. Importance of the Project

- **Speed and Efficiency**:
  - DFT calculations for band gaps are computationally intensive, often taking hours to days per material.
  - Machine learning models predict band gaps in seconds, reducing computation time and enabling faster material screening.
- **Data-Driven Insights**:
  - Large datasets help identify complex relationships between material properties and band gaps, which are challenging to derive from first principles alone.



Perovskite Structures

1. What are Perovskite Structures?

Perovskite structures are a class of crystalline materials with the general formula **ABX3**, where:

- **A** and **B** are cations of different sizes
- **X** is an anion (typically oxygen or a halide)

These structures are named after the mineral perovskite (CaTiO3) and are characterized by a unique crystal lattice. The A cation usually occupies a larger site, while the smaller B cation is in an octahedral coordination with the X anions, creating a versatile framework for various compositions and properties.

Why Perovskite Structures Are Important

1. **Versatile and Tunable Properties:**
   - Perovskites are adaptable materials with a wide range of electronic, optical, and structural properties. By altering the A, B, or X components, perovskites can be customized for various applications, including superconductors, piezoelectrics, and photovoltaic materials.
2. **Breakthroughs in Solar Energy:**
   - Halide perovskites, in particular, have shown remarkable efficiency in solar cells, reaching power conversion efficiencies close to those of silicon-based cells but with simpler and cheaper fabrication processes. This positions perovskites as potential game-changers in the renewable energy sector.
3. **High Potential in Optoelectronics:**
   - Perovskites have promising applications in LEDs, lasers, and photodetectors due to their strong light absorption, high charge-carrier mobility, and ease of tuning their electronic properties.

Why Predicting Band Gaps Is a Valuable Approach

Critical for Device Performance:

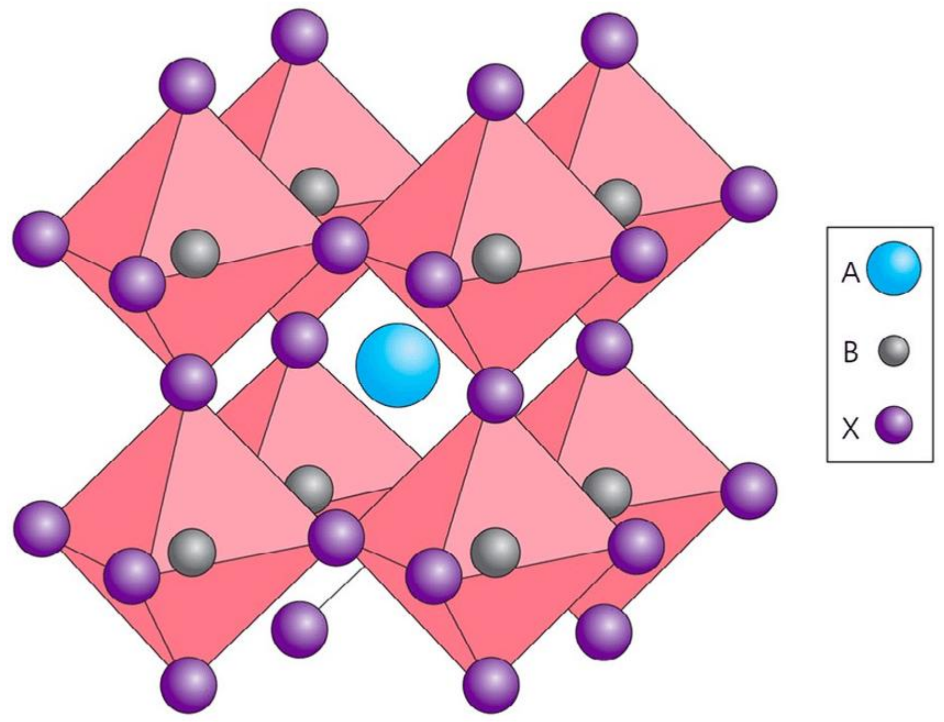
- The band gap determines the material's suitability for electronic and optoelectronic devices. Accurate band gap predictions allow researchers to design perovskite-based materials tailored to specific applications, such as efficient solar cells or LEDs.

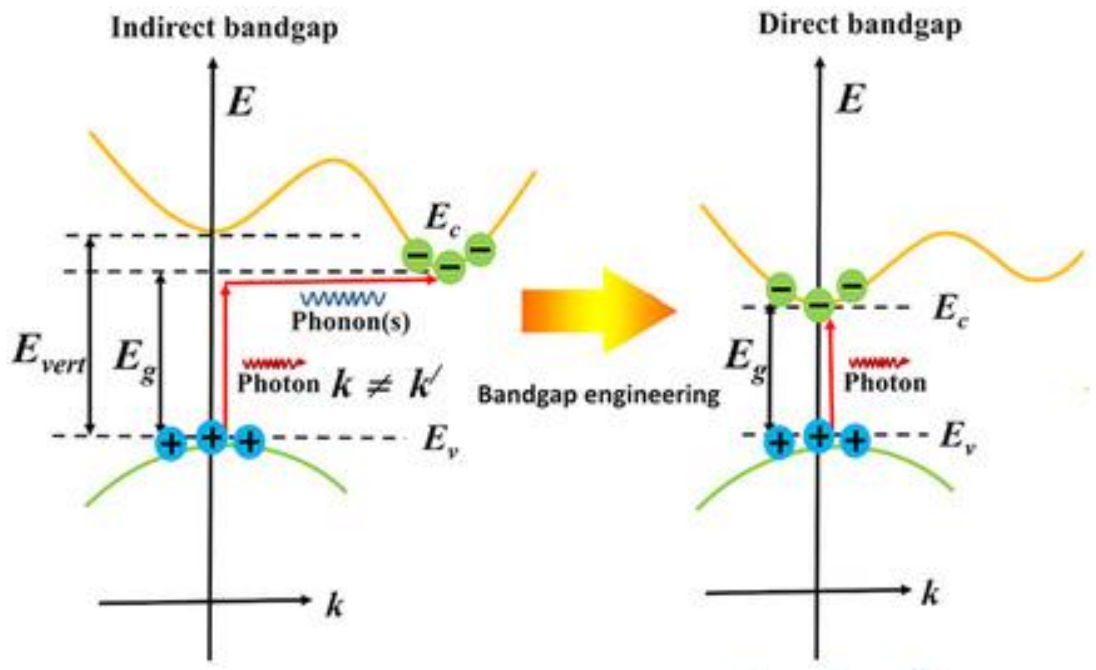
Guides Material Discovery and Optimization:

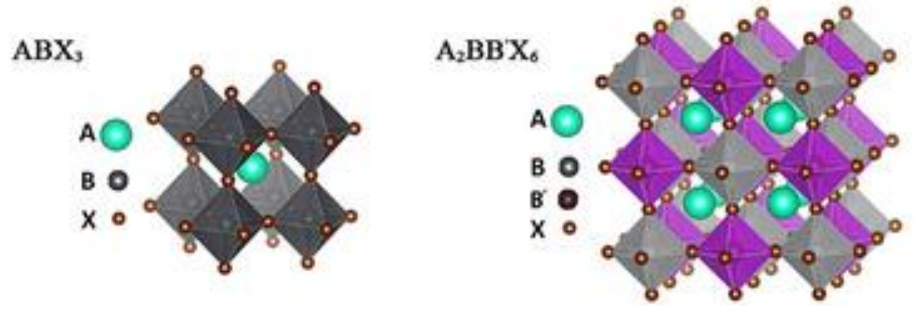
- Band gap prediction helps in screening and optimizing new perovskite compositions without needing extensive experimentation, making the discovery process faster and more cost-effective.

Challenges of Experimental Measurements:

- For complex perovskite compositions or large-scale studies, experimental band gap measurements can be time-consuming and resource-intensive. Computational predictions provide an efficient alternative to obtain reliable estimates of band gaps.







Bird's eye view on machine learning for materials

What is machine learning used for here?
So what to we use machine learning for in material science? The main purpose of machine learning here is the prediction of properties of a given material. This property can either be a class (the classification problem) or a quantity (the regression problem).

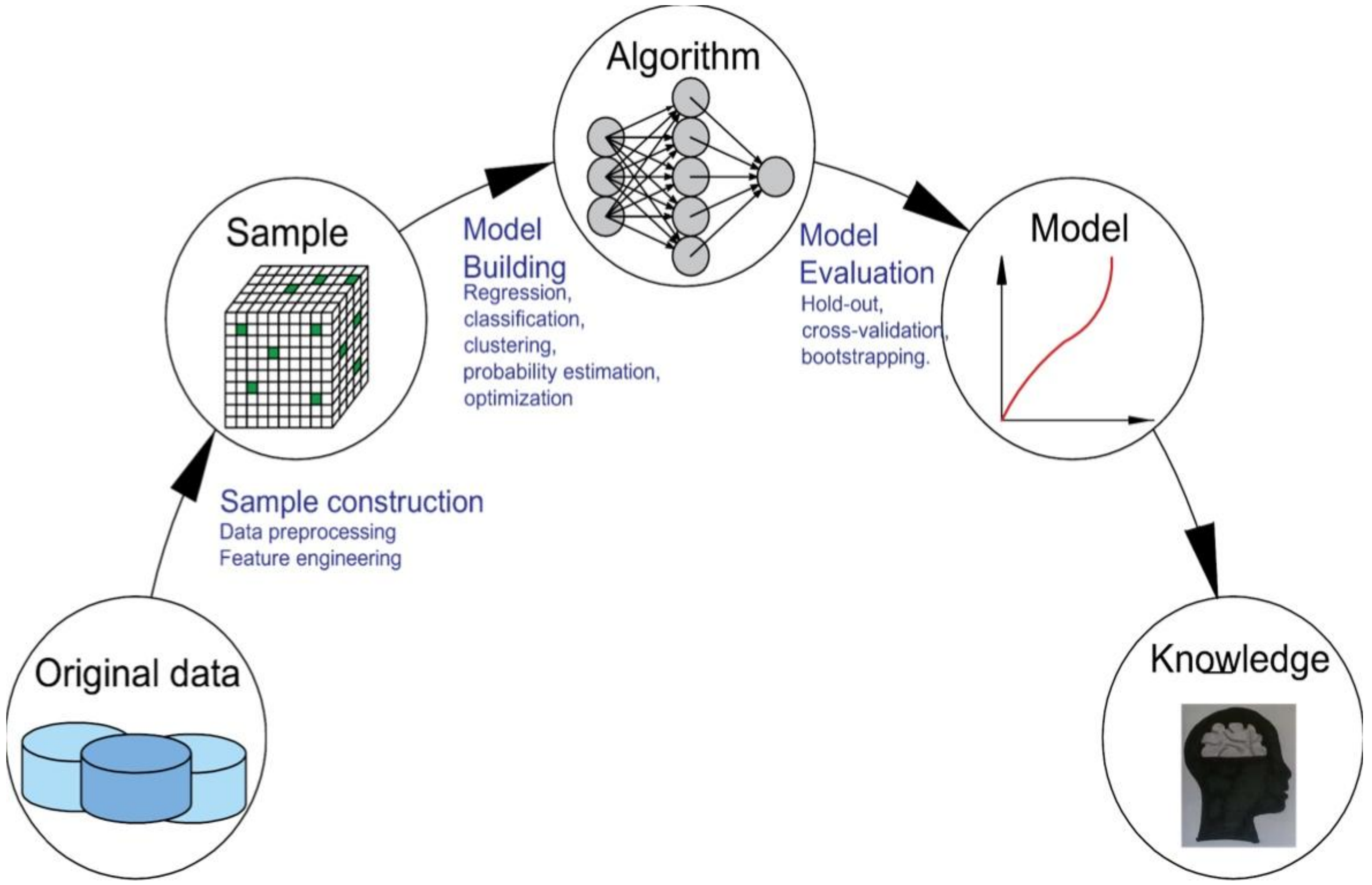
Example problems/papers:

- ❖ Is a given material metallic or semiconducting? An example paper: https://doi.org/10.1038/ncomms15679
- ❖ What is the specific heat capacity of a material? An example paper: https://doi.org/10.1002/adts.201900208

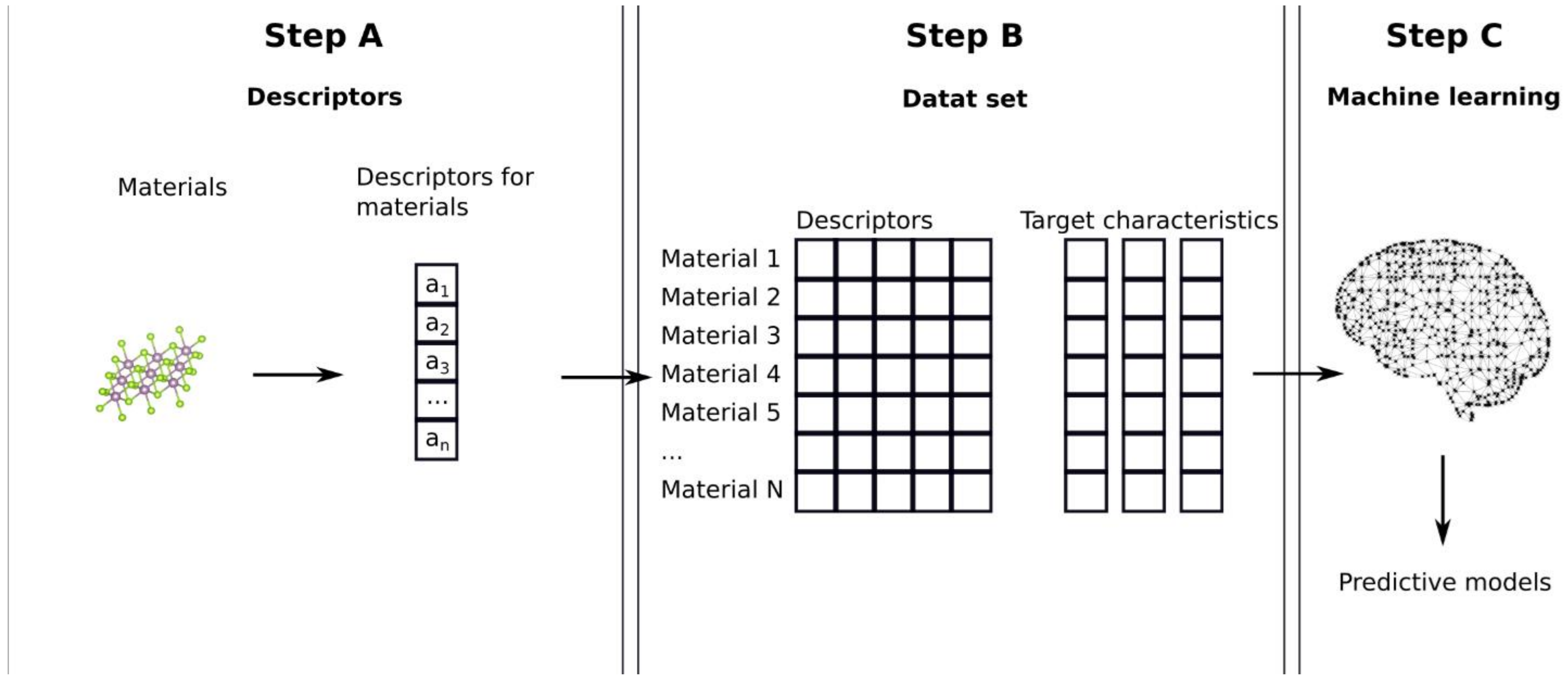
Workflow

So how is this all done?
Generally, we can think of machine learning as a 3-step process:

- ❖ **Step A**: First, find numerical/categorical descriptors that can describe your material. That is: every material in your dataset should be uniquely represented by an array of numbers/categories.
- ❖ **Step B**: Then, apply your procedure to your entire dataset of structures to form a sheet of material descriptors vs. target properties.
- ❖ **Step C**: Use machine learning to predict the target properties based on the descriptors.

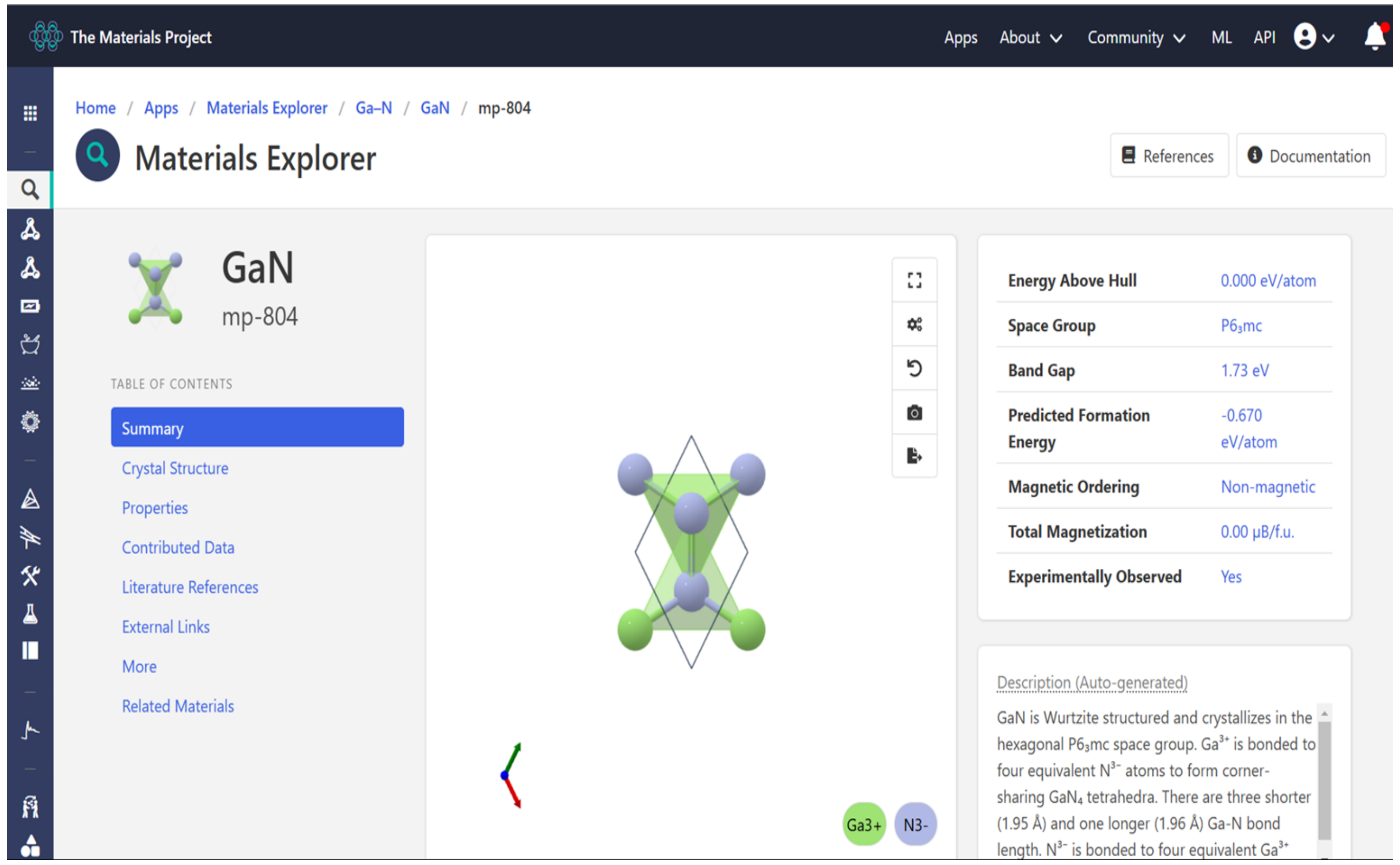


Steps of Model Building



The Materials Project Database

- ❖ The **Materials Project** (MP) database is a massive amount of material science data that was generated using density functional theory (DFT).
- ❖ **Have a look at the database here:** https://materialsproject.org .
- ❖ There are 124,000 inorganic crystals in the database, along with the DFT-calculated properties of these materials. There are also 530,000 nanoporous materials, as well as other interesting properties data.
- ❖ We signed up and got an API key to query data from MP Database.



Energy Above Hull	0.000 eV/atom
Space Group	$P6_3mc$
Band Gap	1.73 eV
Predicted Formation Energy	-0.670 eV/atom
Magnetic Ordering	Non-magnetic
Total Magnetization	0.00 μB/f.u.
Experimentally Observed	Yes

Description (Auto-generated)

GaN is Wurtzite structured and crystallizes in the hexagonal $P6_3mc$ space group. Ga^{3+} is bonded to four equivalent N^{3-} atoms to form corner-sharing GaN_4 tetrahedra. There are three shorter (1.95 Å) and one longer (1.96 Å) Ga-N bond length. N^{3-} is bonded to four equivalent Ga^{3+}

Why is MP great for machine learning?

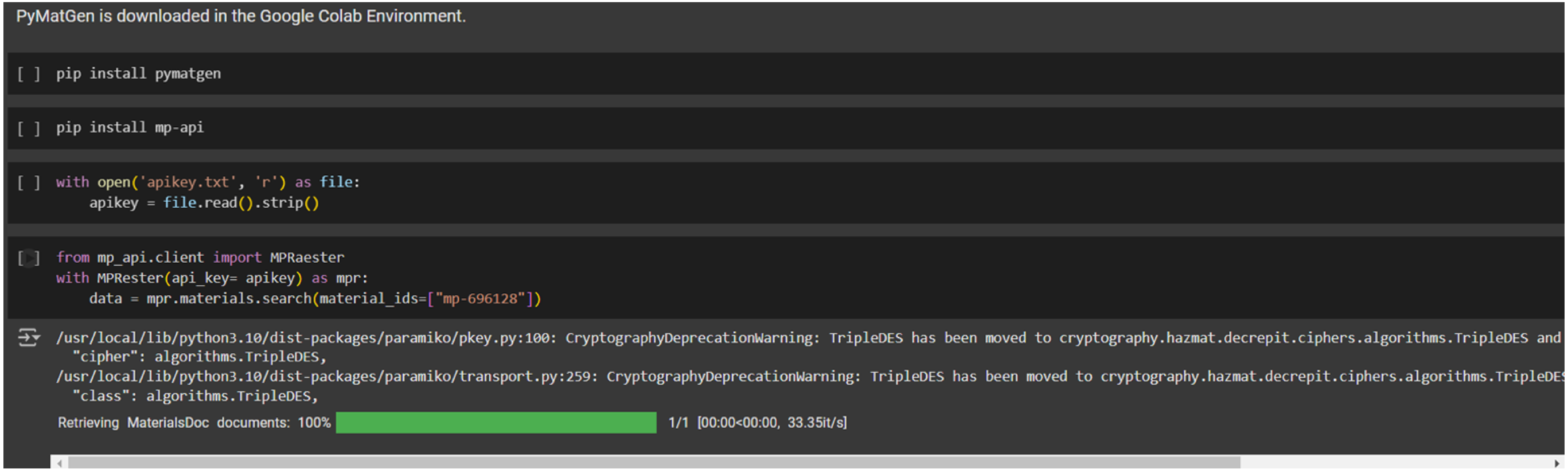
- ❖ Because of the huge amount of materials (dataset) and DFT-calculated properties (target properties).
- ❖ That much data can be utilised using a range of machine learning methods to make predictions with various levels of accuracy.

The DFT properties

- ❖ For a given elemental composition, the lattice parameters and the positions of the atoms within the lattice are all obtained using DFT.
- ❖ For the obtained crystal structure, the Final Magnetic Moment, Formation Energy / Atom, Energy Above Hull / Atom, Band Gap are calculated.
- ❖ The Density is derived from the obtained crystal structure.
- ❖ Further DFT calculations are performed to obtain the band structure as well as other properties.
- ❖ Some of the crystals on MP correspond to crystals that exist in nature, and some are purely hypothetical. The hypothetical crystals have been generated by some algorithm that uses artificial intelligence, or probably by simple elemental substitution.

The PyMatGen python library

- To be able to query the MP database, the MP team provided the community with a python library.
- We loaded it into colab environment and used the API key we obtained earlier to query a test data of mpID-696128 (random material).



PyMatGen is downloaded in the Google Colab Environment.

```
[ ] pip install pymatgen
```

```
[ ] pip install mp-api
```

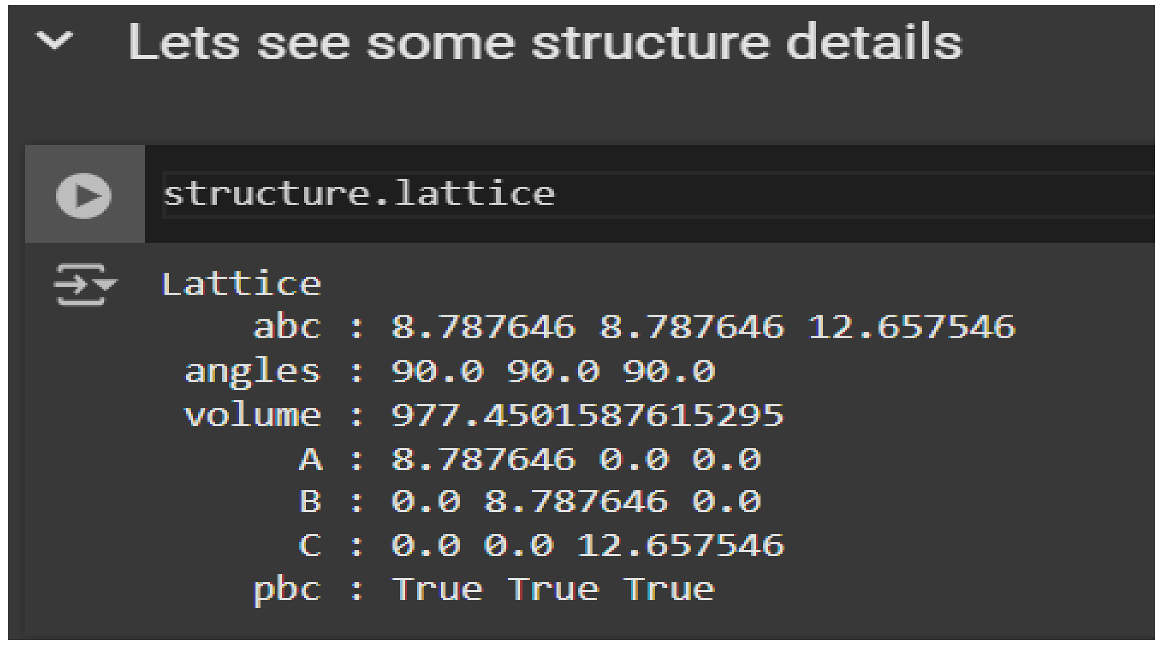
```
[ ] with open('apikey.txt', 'r') as file:
        apikey = file.read().strip()
```

```
[ ] from mp_api.client import MPRaester
    with MPRester(api_key= apikey) as mpr:
        data = mpr.materials.search(material_ids=["mp-696128"])
```

```
/usr/local/lib/python3.10/dist-packages/paramiko/pkey.py:100: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and
  "cipher": algorithms.TripleDES,
/usr/local/lib/python3.10/dist-packages/paramiko/transport.py:259: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES
  "class": algorithms.TripleDES,
Retrieving MaterialsDoc documents: 100% 1/1 [00:00<00:00, 33.35it/s]
```

The PyMatGen python library

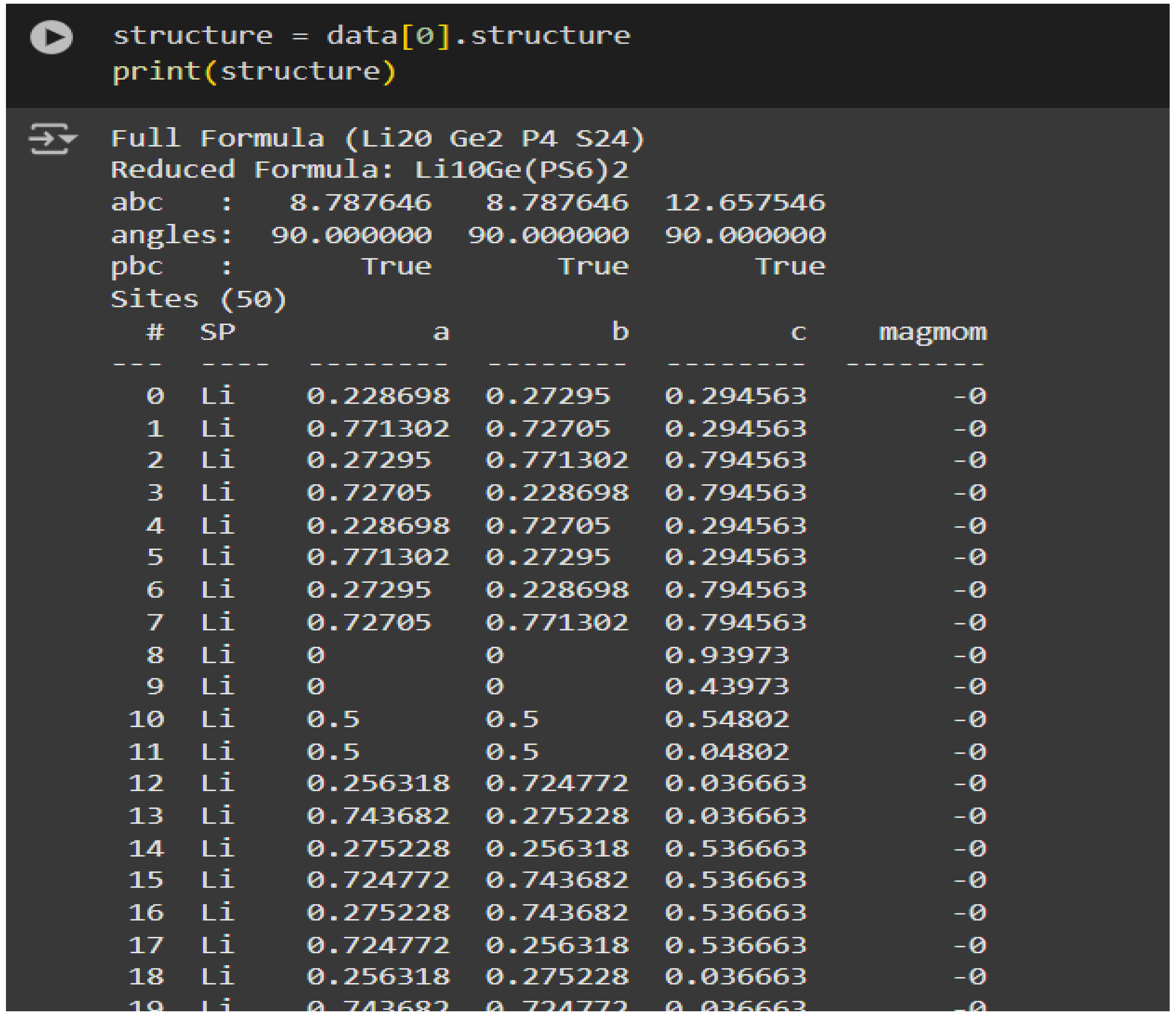
- ❖ We can get the structure details in this following way.
- ❖ So now we have the details of CIF structure in a human readable format which includes the formula, the lattice parameters and the positions of the atoms in the crystals,etc.



Lets see some structure details

```
structure.lattice
```

```
Lattice
    abc : 8.787646 8.787646 12.657546
 angles : 90.0 90.0 90.0
 volume : 977.4501587615295
      A : 8.787646 0.0 0.0
      B : 0.0 8.787646 0.0
      C : 0.0 0.0 12.657546
    pbc : True True True
```



```
structure = data[0].structure
print(structure)
```

```
Full Formula (Li20 Ge2 P4 S24)
Reduced Formula: Li10Ge(PS6)2
abc   :   8.787646   8.787646  12.657546
angles:  90.000000  90.000000  90.000000
pbc   :       True       True       True
Sites (50)
  #  SP           a         b         c    magmom
---  ----  --------  --------  --------  --------
  0  Li    0.228698  0.27295   0.294563        -0
  1  Li    0.771302  0.72705   0.294563        -0
  2  Li    0.27295   0.771302  0.794563        -0
  3  Li    0.72705   0.228698  0.794563        -0
  4  Li    0.228698  0.72705   0.294563        -0
  5  Li    0.771302  0.27295   0.294563        -0
  6  Li    0.27295   0.228698  0.794563        -0
  7  Li    0.72705   0.771302  0.794563        -0
  8  Li    0         0         0.93973         -0
  9  Li    0         0         0.43973         -0
 10  Li    0.5       0.5       0.54802         -0
 11  Li    0.5       0.5       0.04802         -0
 12  Li    0.256318  0.724772  0.036663        -0
 13  Li    0.743682  0.275228  0.036663        -0
 14  Li    0.275228  0.256318  0.536663        -0
 15  Li    0.724772  0.743682  0.536663        -0
 16  Li    0.275228  0.743682  0.536663        -0
 17  Li    0.724772  0.256318  0.536663        -0
 18  Li    0.256318  0.275228  0.036663        -0
```

How do we apply ML to predict crystal properties?

Some Background About Molecules and Crystals:

- ❖ A Crystal is structure that is endowed with periodicity
- ❖ Molecules are fundamentally different from crystals because of the pattern (periodicity)
- ❖ A molecule is just that one single molecule, sitting on its own, in isolation, whereas a crystal is really composed of an infinite number of molecules.

- ❖ The key thing here in the molecular descriptors is that they are based on the atomic positions.
- ❖ In crystals, we can't really use atomic positions like we did with molecules to obtain descriptors. Why?

Because there are many different ways we can represent the unit cell of a crystal. Therefore, we cannot use the atomic coordinates to derive descriptors for crystals, otherwise the derived descriptors, such as the eigenvalues of the Coulomb matrix in the case of molecules, will change dramatically for the same crystal. That is, the descriptor vector is not invariant with respect to translation of the unit cell.

Common Obvious Question Comes:

- **How would the pattern in the crystal make ML for molecules different from ML for crystals?**

- ❖ To predict the properties of the molecules one can derive a set of descriptors for the molecules in the data set that are based on the positions of the atoms within the molecule.
- ❖ One can derive descriptors based on the relative positions, in order to ensure that the descriptors are invariant to transformations: rotation and translation.

SOLUTION - Use of Descriptors

- ❖ We can use some statistics of atomic properties as the descriptor vector. For example:

- ❖ Average of the atomic numbers of all these elements in the crystal. For e.g., in SiC, the average would be 14(Si)+6(C)/2 = 10
- ❖ The average of Ionization potential of the atoms
- ❖ The average of Electron Affinity of the atoms
- ❖ And more Averages
- ❖ Other statistics, such as the standard deviation and the variance. Adding those will triple the number elements in the descriptor vector above.

However there is a Problem:
A lot of materials exist in various phases. That is, for the same atomic composition, let's say SiC, there are several possible structures. Right now, there are 27 possible structures for SiC on MaterialsProject.org.

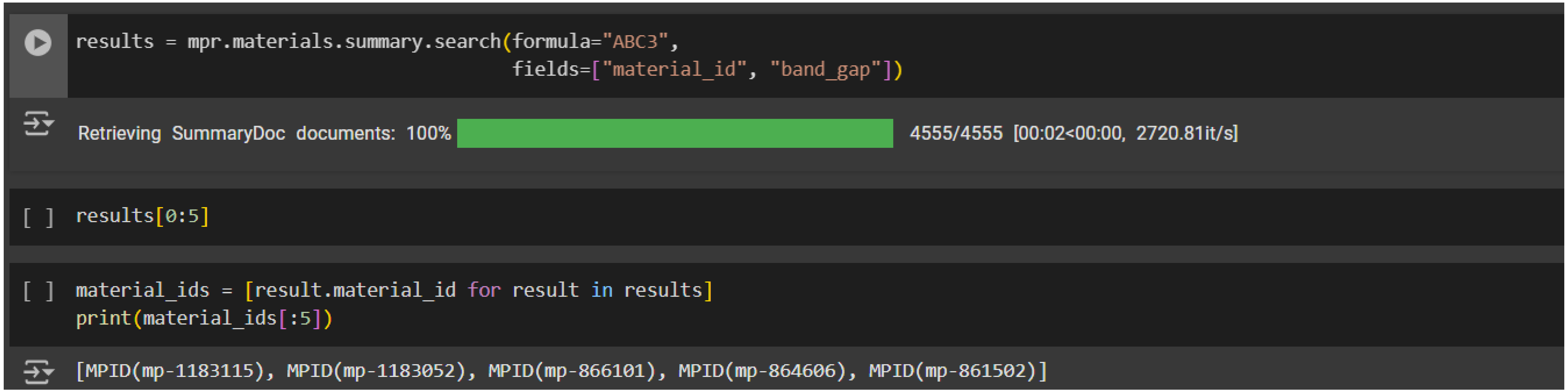
So, the above descriptors won't work. For example, for the case of SiC, all of the 27 SiC phases in MP will have the same values for the statistical values above.

To solve this problem, we have to add descriptors based on the geometrical arrangement of atoms. A simple such descriptor is to average the bond lengths (a bond is formed between two atoms).

Building the Dataset

Step 1: Collecting the structures
We want to predict the bandgaps of structures, so we need to collect the structures (dataset) along with their corresponding bandgaps (target vector).

We will focus on stoichiometric perovskites: these are materials of the form ABC3. The following query will collect the CIFs and bandgaps for these materials from MP.



```
results = mpr.materials.summary.search(formula="ABC3",
                                       fields=["material_id", "band_gap"])
```

```
Retrieving SummaryDoc documents: 100% 4555/4555 [00:02<00:00, 2720.81it/s]
```

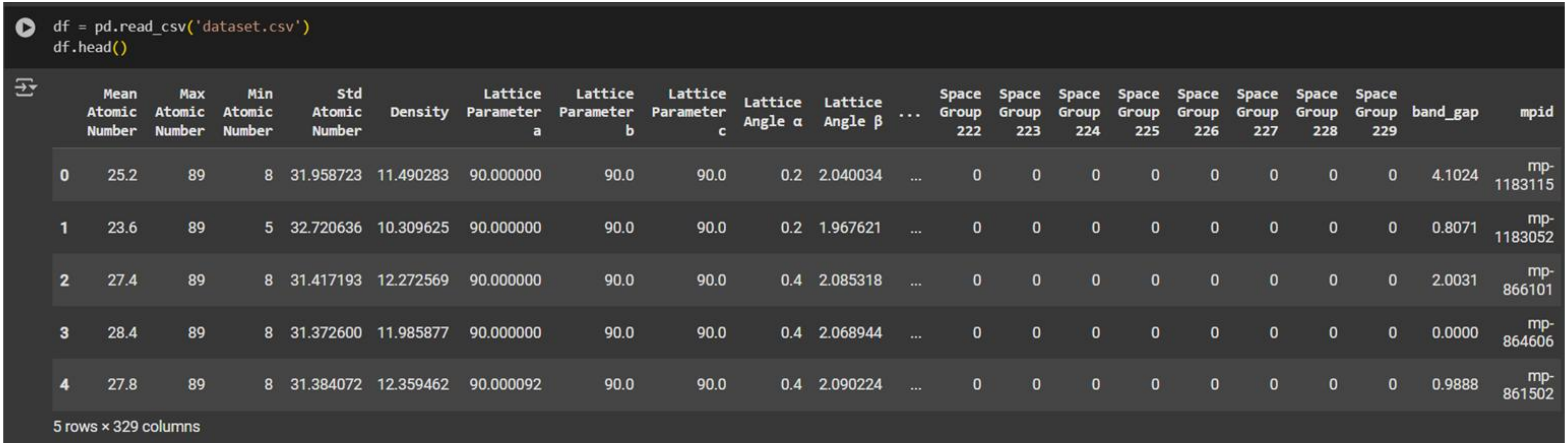
```
results[0:5]
```

```
material_ids = [result.material_id for result in results]
print(material_ids[:5])
```

```
[MPID(mp-1183115), MPID(mp-1183052), MPID(mp-866101), MPID(mp-864606), MPID(mp-861502)]
```

Building the Dataset

Step 2: Pre-Processing
Here we will extract the data we need from the structures, put them in a pandas DataFrame and then apply normalization.
We get 4555 rows of data with their mpIDs.



```
df = pd.read_csv('dataset.csv')
df.head()
```

	Mean Atomic Number	Max Atomic Number	Min Atomic Number	Std Atomic Number	Density	Lattice Parameter a	Lattice Parameter b	Lattice Parameter c	Lattice Angle α	Lattice Angle β	...	Space Group 222	Space Group 223	Space Group 224	Space Group 225	Space Group 226	Space Group 227	Space Group 228	Space Group 229	band_gap	mpid
0	25.2	89	8	31.958723	11.490283	90.000000	90.0	90.0	0.2	2.040034	...	0	0	0	0	0	0	0	0	4.1024	mp-1183115
1	23.6	89	5	32.720636	10.309625	90.000000	90.0	90.0	0.2	1.967621	...	0	0	0	0	0	0	0	0	0.8071	mp-1183052
2	27.4	89	8	31.417193	12.272569	90.000000	90.0	90.0	0.4	2.085318	...	0	0	0	0	0	0	0	0	2.0031	mp-866101
3	28.4	89	8	31.372600	11.985877	90.000000	90.0	90.0	0.4	2.068944	...	0	0	0	0	0	0	0	0	0.0000	mp-864606
4	27.8	89	8	31.384072	12.359462	90.000092	90.0	90.0	0.4	2.090224	...	0	0	0	0	0	0	0	0	0.9888	mp-861502

5 rows × 329 columns

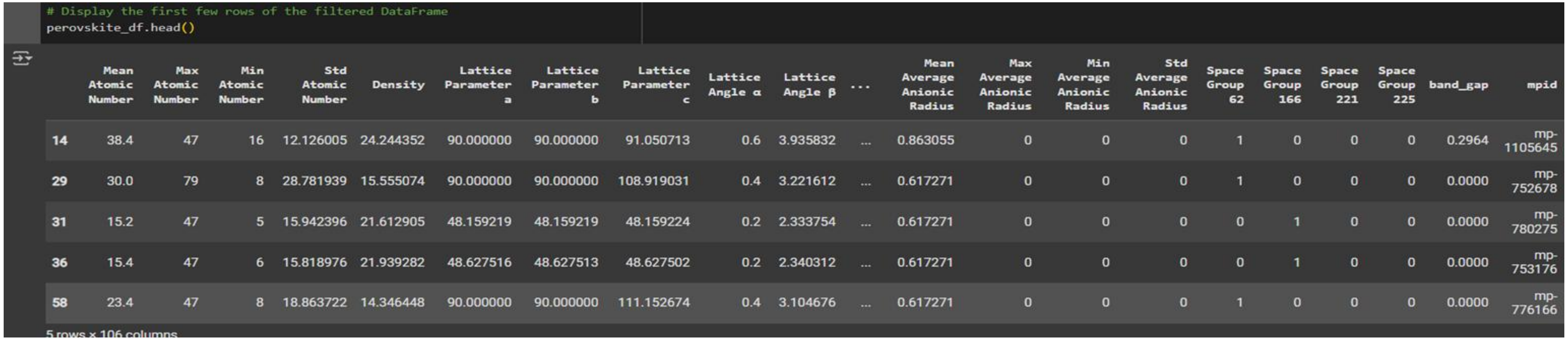
Filter the Dataset to get Perovskite Structures

The space group of perovskite depends on its structure:

- ❖ **Cubic perovskites:** The space group for cubic perovskites is Pm3m (221).
- ❖ **Orthorhombic perovskites:** The space group for orthorhombic perovskites is Pbnm (62). For example, $CaTiO_3$ crystallizes in the orthorhombic Pnma space group

What was done:

1. **Defining Relevant Space Groups:** Common space groups for perovskites include 62 (Pbnm), 221 (Pm3m), 225 (Fm3m), and 166 (R3c).
2. **Applying Lattice Parameter Conditions:**
   a. Tetragonal Structure: Identified by $a = b \neq c$.
   b. Orthorhombic/Triclinic Structure: Characterized by $a \neq b \neq c$.



```
# Display the first few rows of the filtered DataFrame
perovskite_df.head()
```

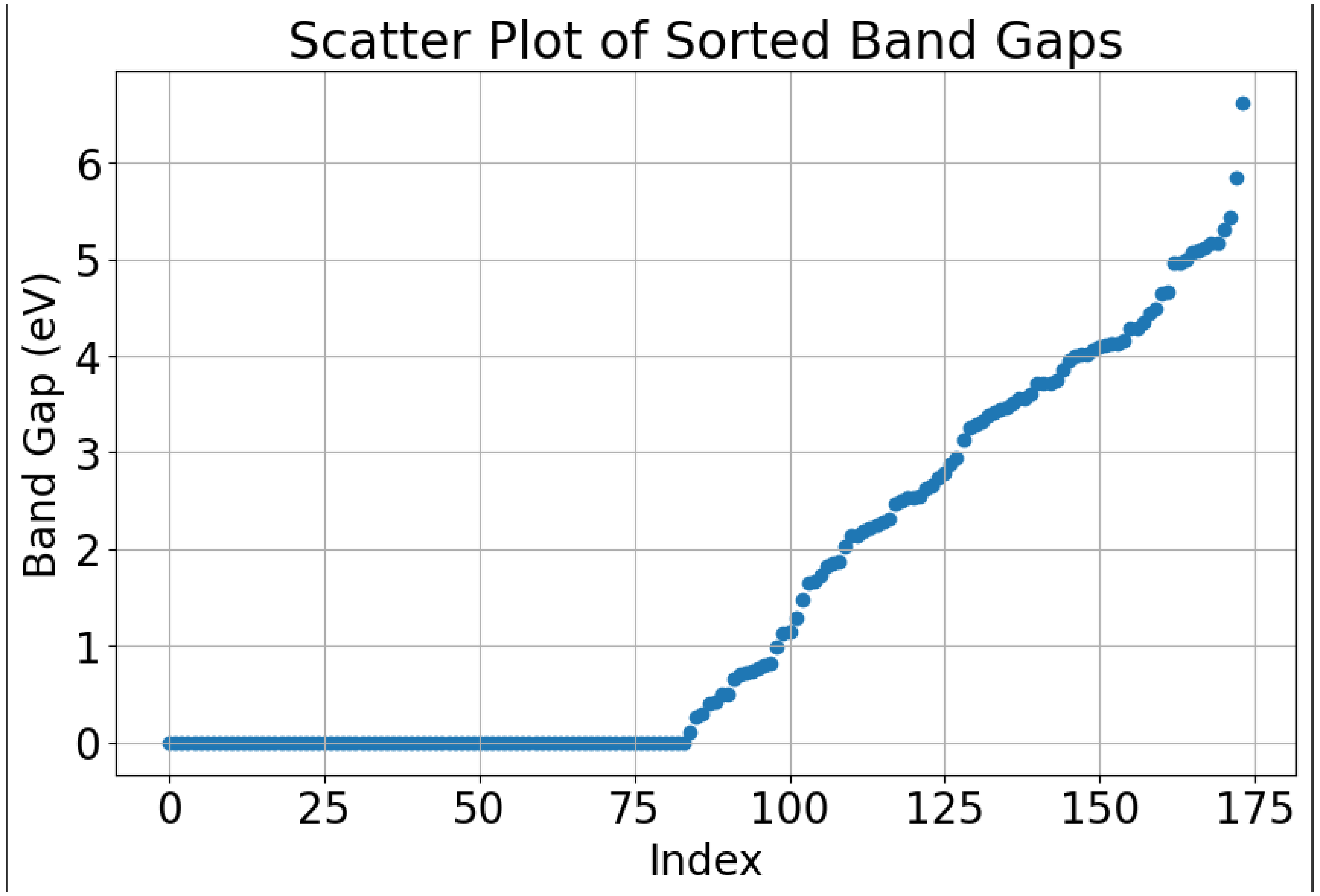
	Mean Atomic Number	Max Atomic Number	Min Atomic Number	Std Atomic Number	Density	Lattice Parameter a	Lattice Parameter b	Lattice Parameter c	Lattice Angle α	Lattice Angle β	...	Mean Average Anionic Radius	Max Average Anionic Radius	Min Average Anionic Radius	Std Average Anionic Radius	Space Group 62	Space Group 166	Space Group 221	Space Group 225	band_gap	mpid
14	38.4	47	16	12.126005	24.244352	90.000000	90.000000	91.050713	0.6	3.935832	...	0.863055	0	0	0	1	0	0	0	0.2964	mp-1105645
29	30.0	79	8	28.781939	15.555074	90.000000	90.000000	108.919031	0.4	3.221612	...	0.617271	0	0	0	1	0	0	0	0.0000	mp-752678
31	15.2	47	5	15.942396	21.612905	48.159219	48.159219	48.159224	0.2	2.333754	...	0.617271	0	0	0	0	1	0	0	0.0000	mp-780275
36	15.4	47	6	15.818976	21.939282	48.627516	48.627513	48.627502	0.2	2.340312	...	0.617271	0	0	0	0	1	0	0	0.0000	mp-753176
58	23.4	47	8	18.863722	14.346448	90.000000	90.000000	111.152674	0.4	3.104676	...	0.617271	0	0	0	1	0	0	0	0.0000	mp-776166

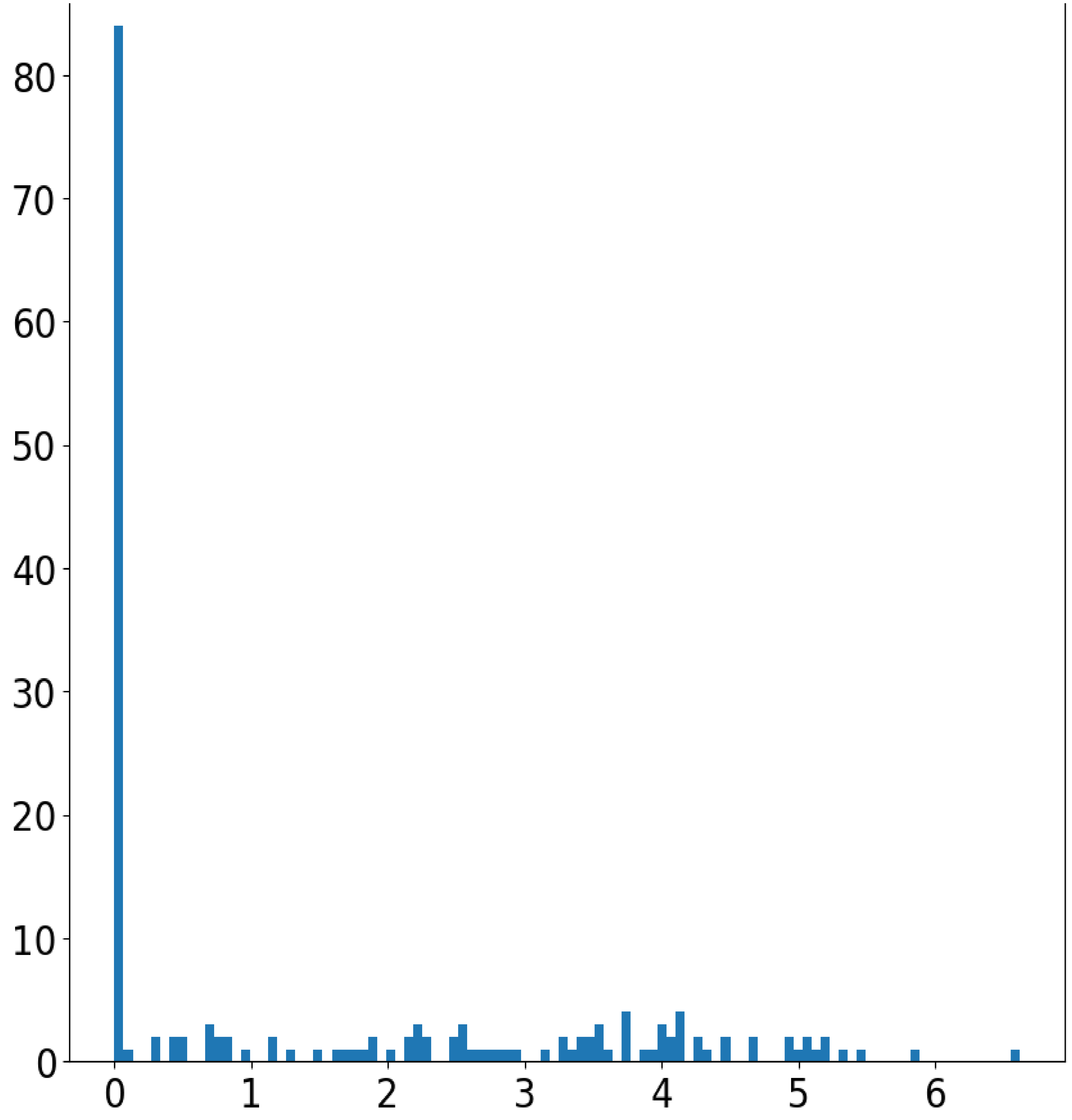
5 rows × 106 columns

Distribution Plots

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

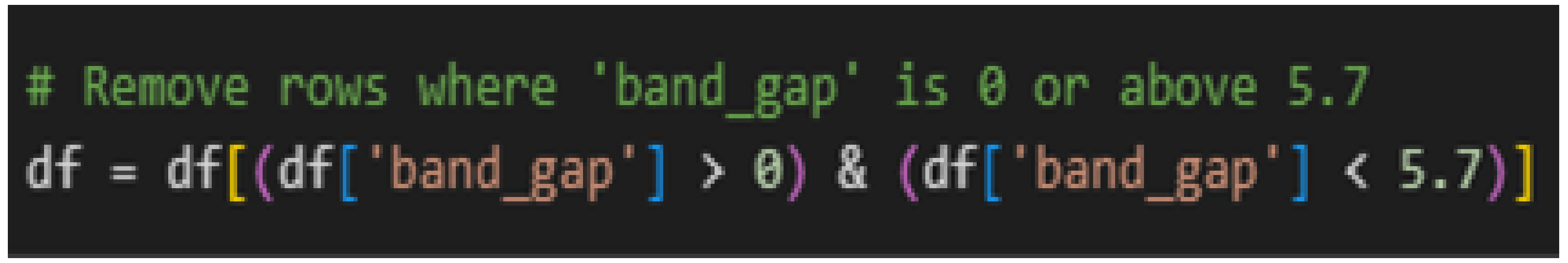
How about a scatter plot?





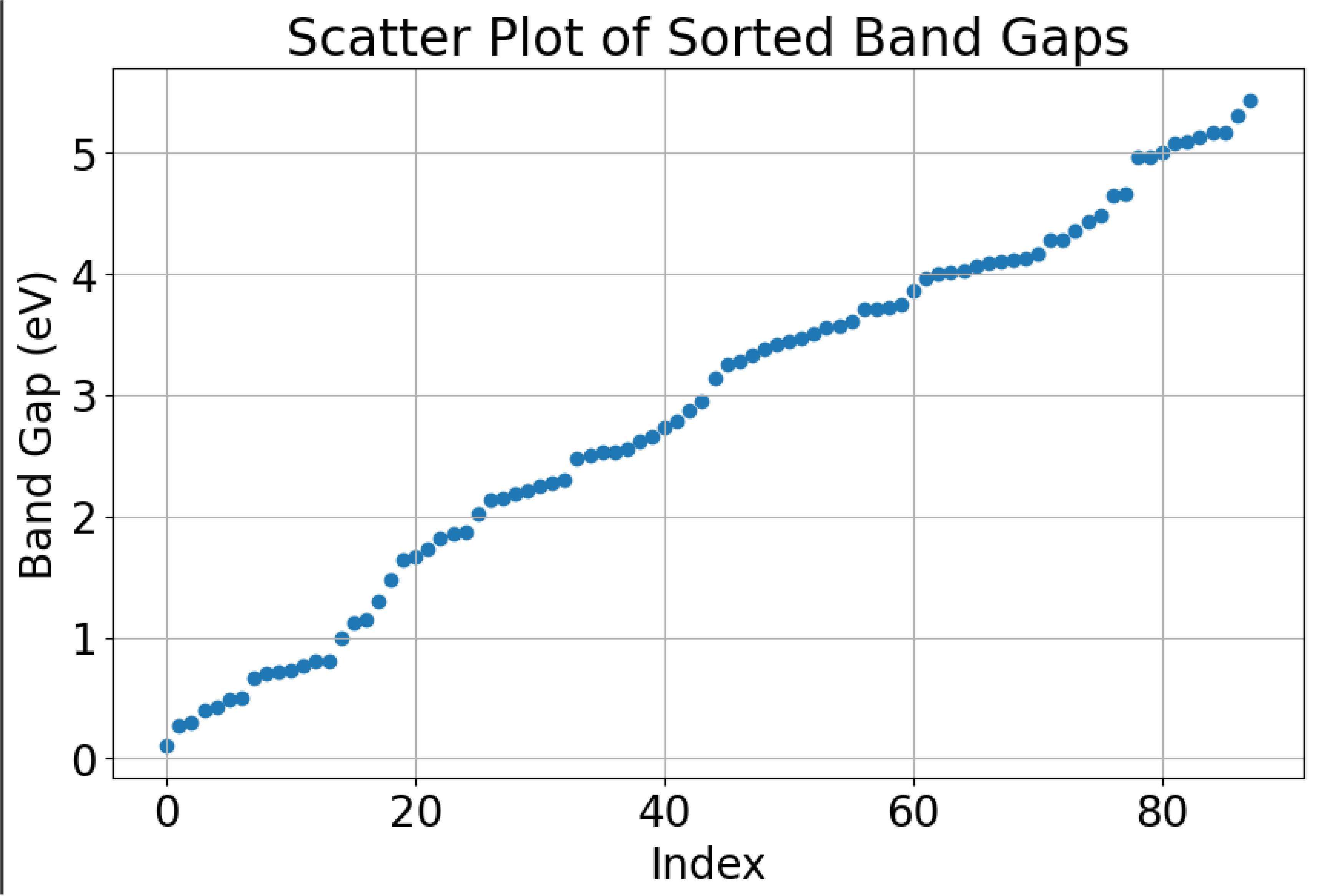
Distribution Plots after Null Removal

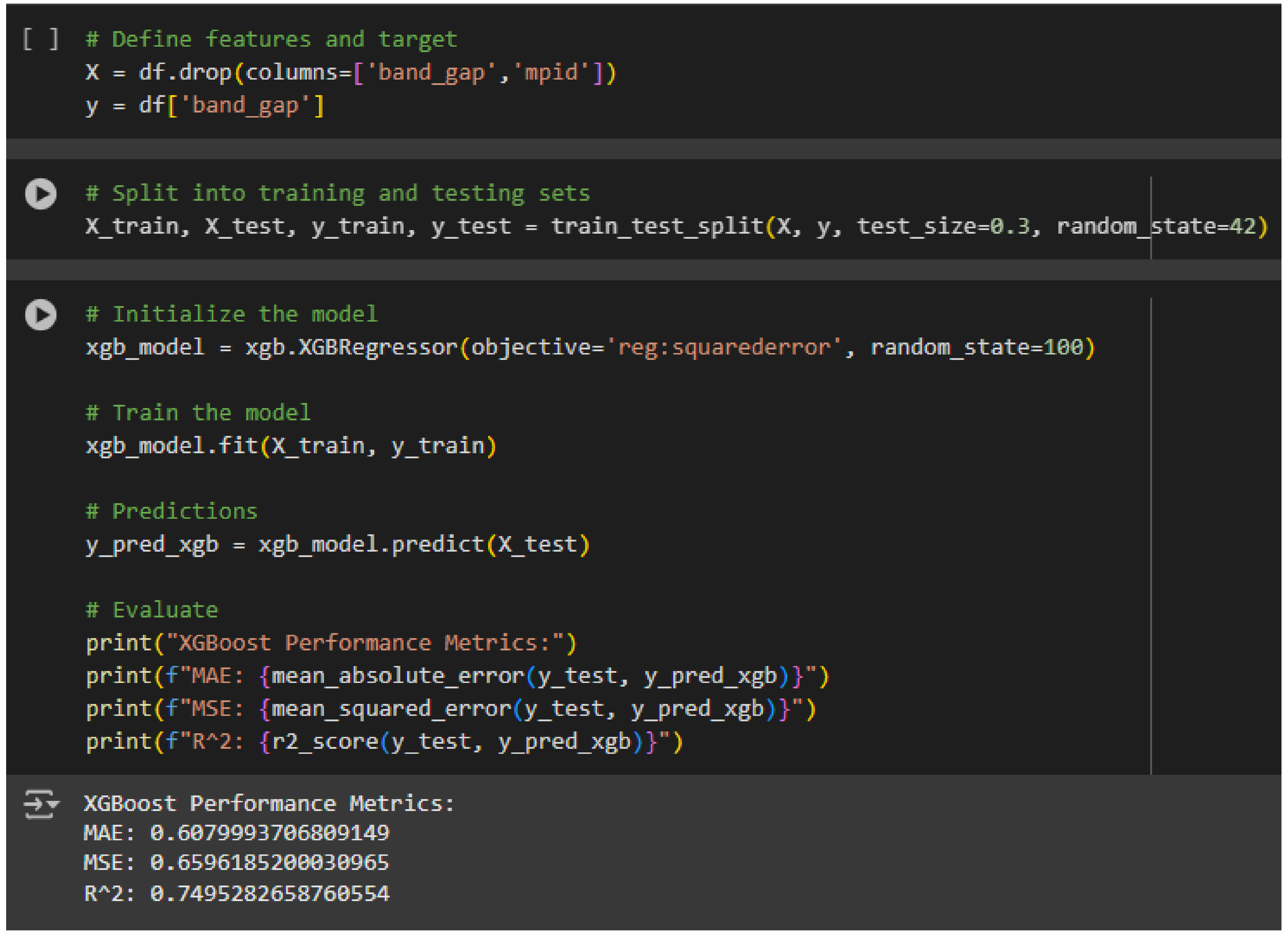
These plots shows that the metals (zero bandgap) were removed.
Then we move onto first model building.



```python
# Remove rows where 'band_gap' is 0 or above 5.7
df = df[(df['band_gap'] > 0) & (df['band_gap'] < 5.7)]
```

Scatter plot after removal:





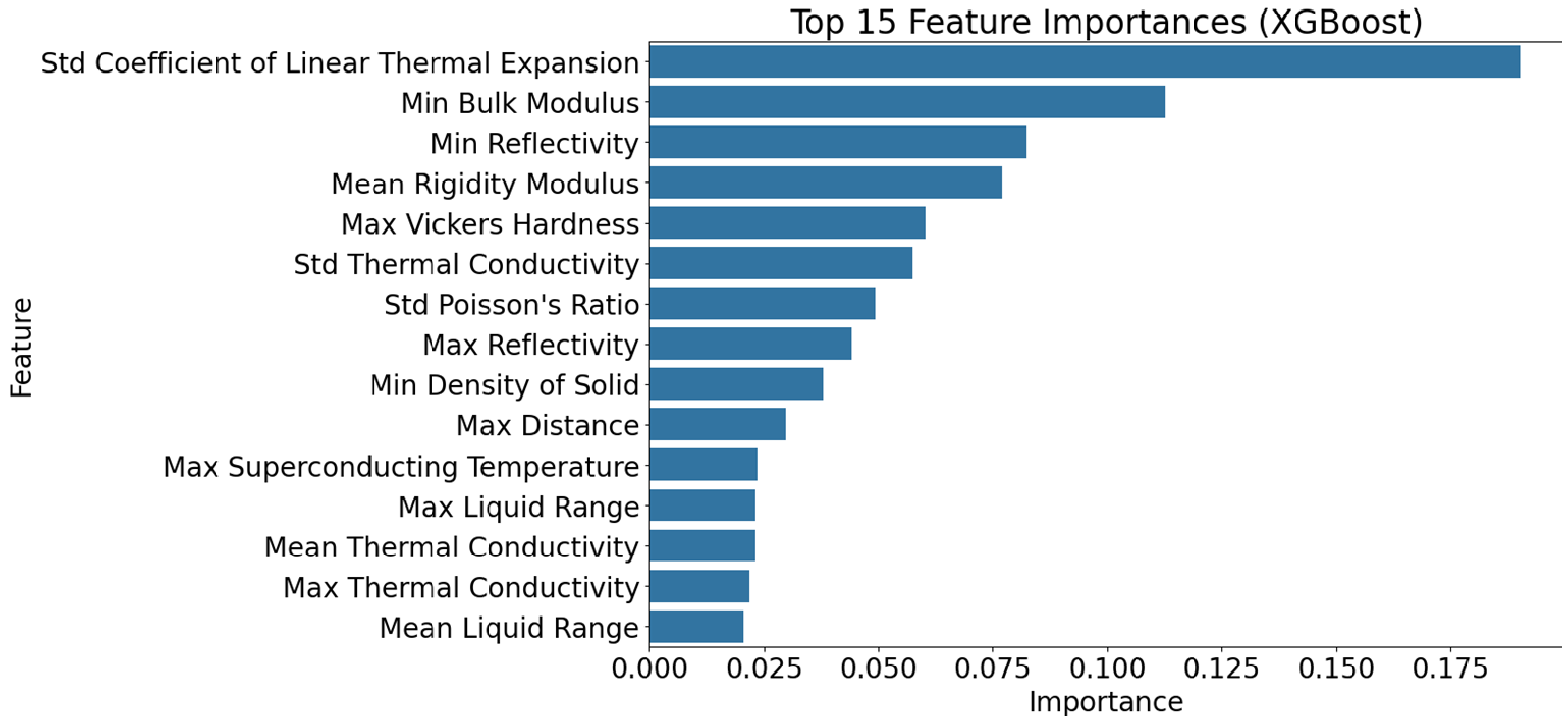
```python
# Define features and target
X = df.drop(columns=['band_gap','mpid'])
y = df['band_gap']
```

```python
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

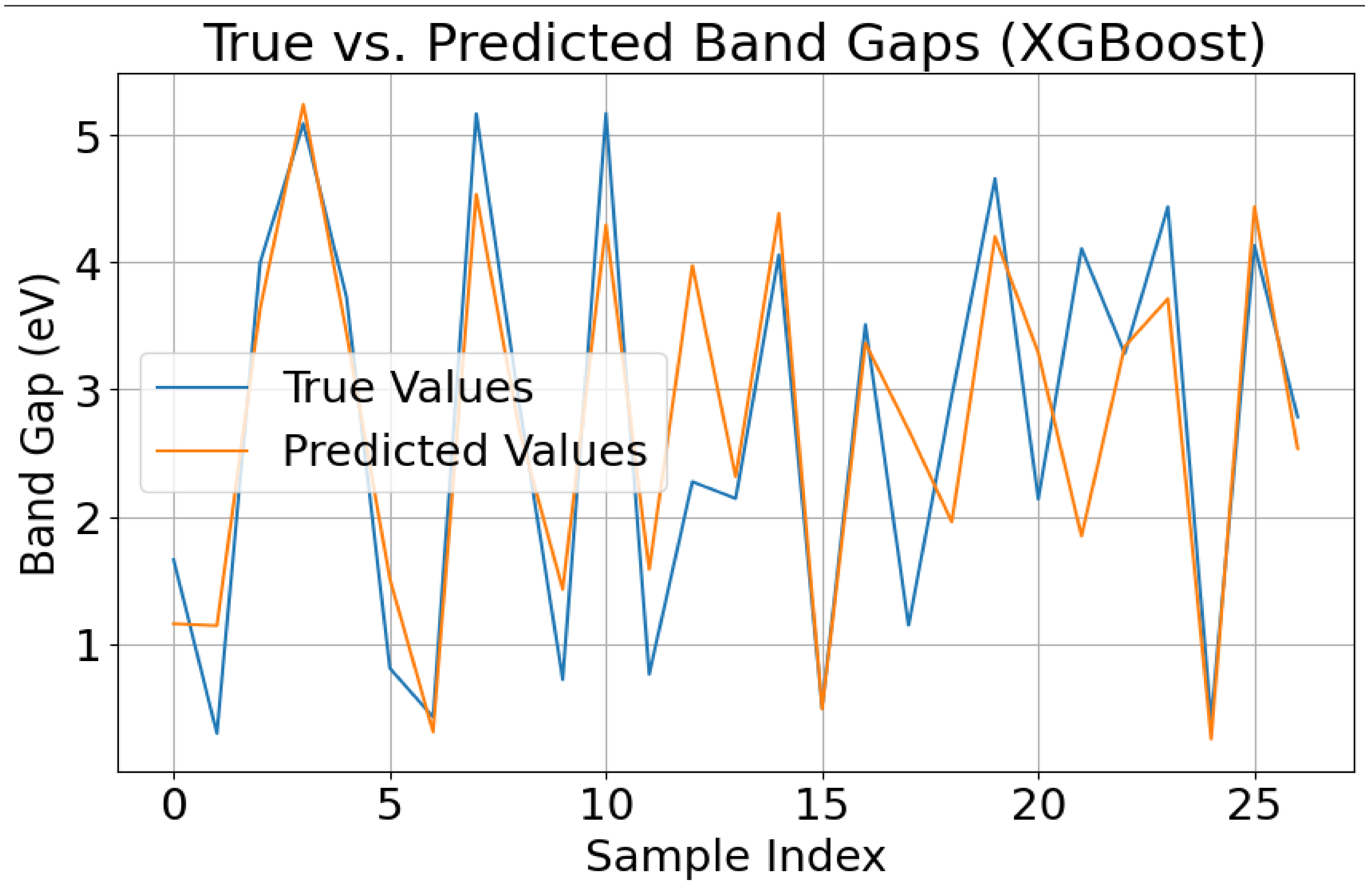
```python
# Initialize the model
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', random_state=100)

# Train the model
xgb_model.fit(X_train, y_train)

# Predictions
y_pred_xgb = xgb_model.predict(X_test)

# Evaluate
print("XGBoost Performance Metrics:")
print(f"MAE: {mean_absolute_error(y_test, y_pred_xgb)}")
print(f"MSE: {mean_squared_error(y_test, y_pred_xgb)}")
print(f"R^2: {r2_score(y_test, y_pred_xgb)}")
```

```
XGBoost Performance Metrics:
MAE: 0.6079993706809149
MSE: 0.6596185200030965
R^2: 0.7495282658760554
```

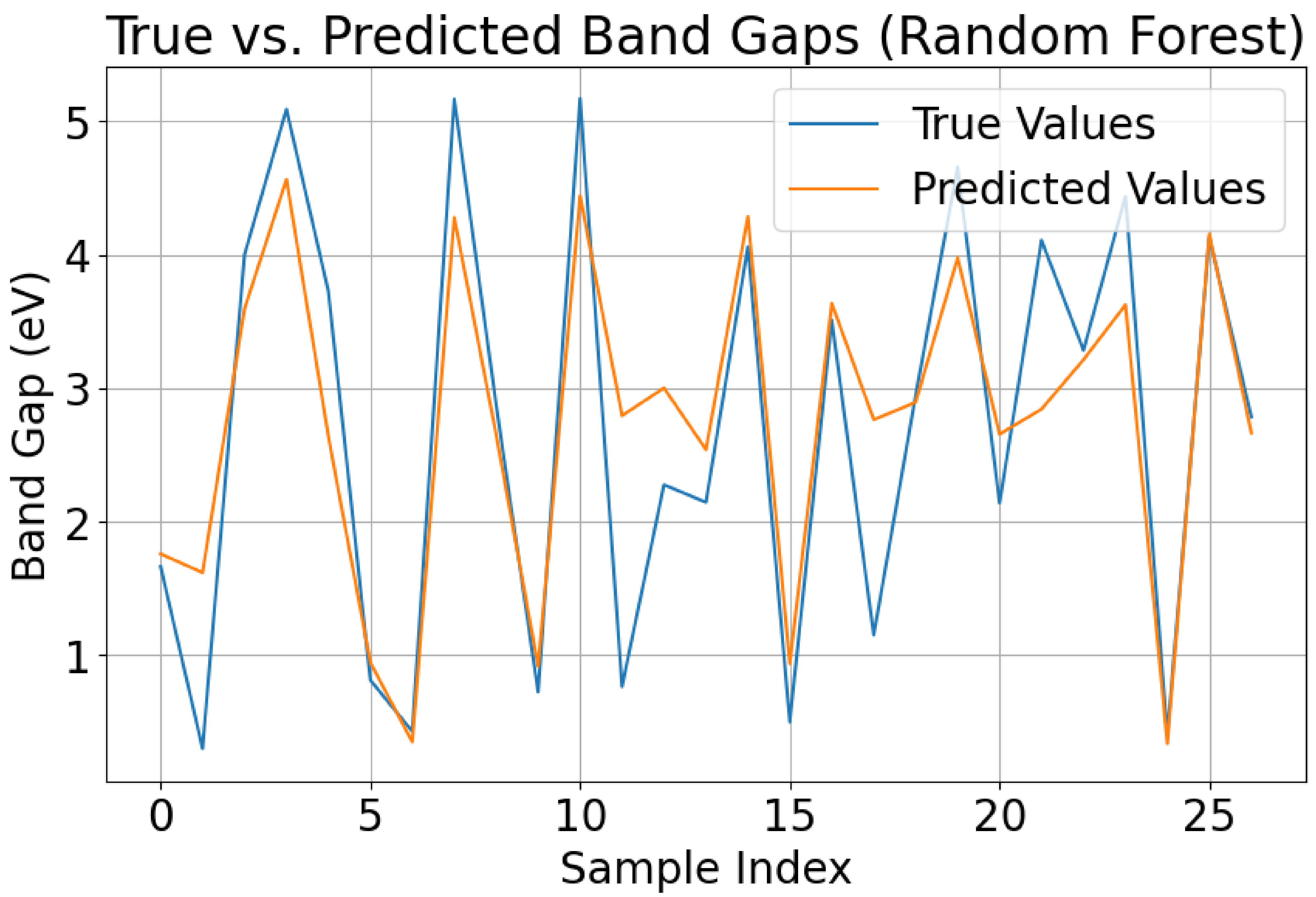
Feature Importances

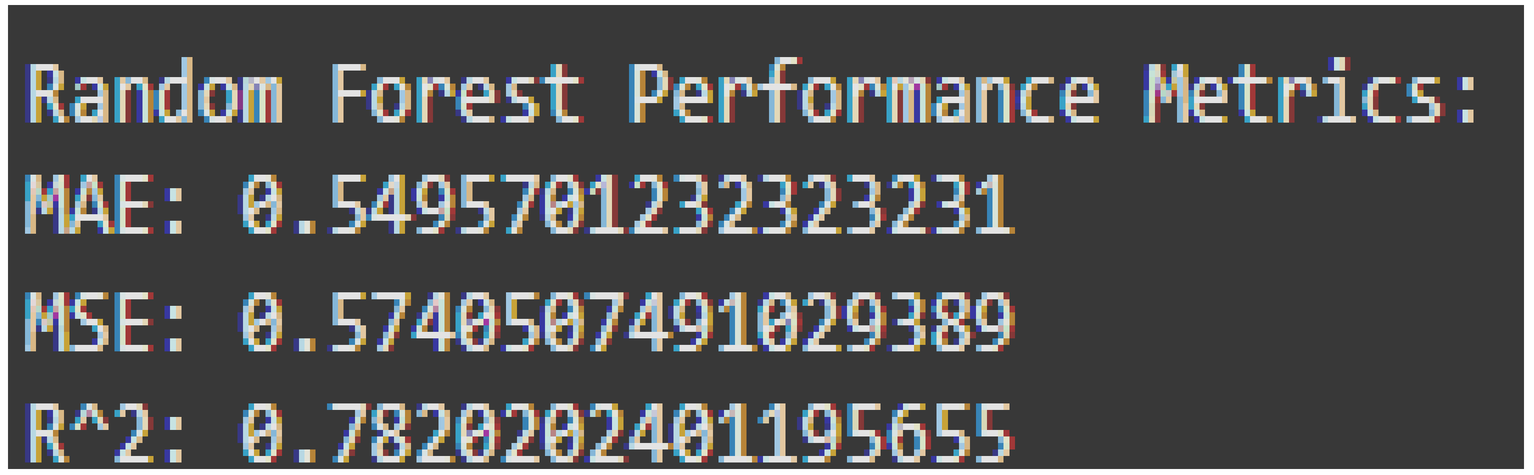


Model’s Goodness of Fit



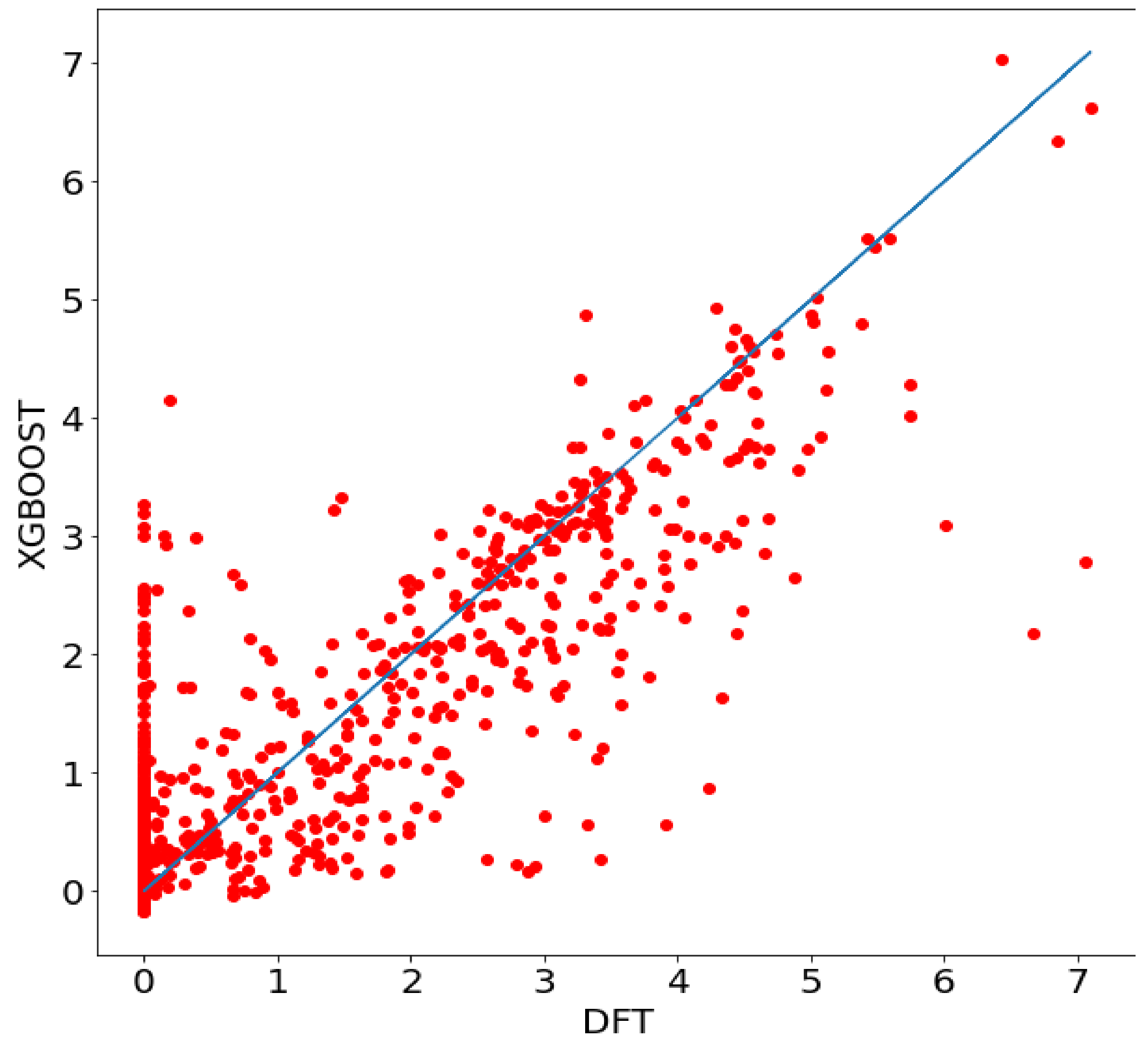
Model's Goodness of Fit



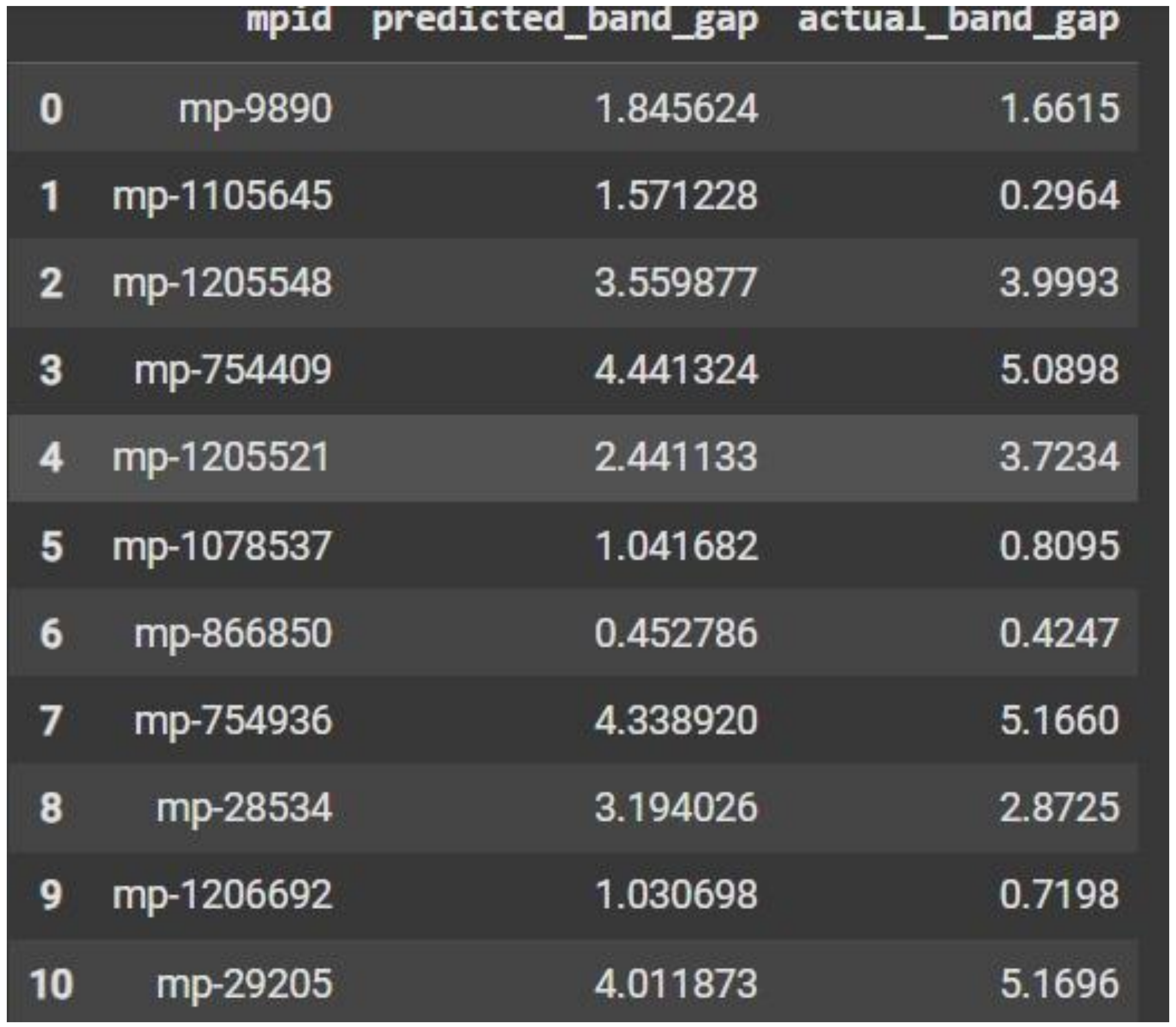


```
Random Forest Performance Metrics:
MAE: 0.549570123232323231
MSE: 0.5740507491029389
R^2: 0.7820202401195655
```

Final Predictions to Test



Complete model



	mpid	predicted_band_gap	actual_band_gap
0	mp-9890	1.845624	1.6615
1	mp-1105645	1.571228	0.2964
2	mp-1205548	3.559877	3.9993
3	mp-754409	4.441324	5.0898
4	mp-1205521	2.441133	3.7234
5	mp-1078537	1.041682	0.8095
6	mp-866850	0.452786	0.4247
7	mp-754936	4.338920	5.1660
8	mp-28534	3.194026	2.8725
9	mp-1206692	1.030698	0.7198
10	mp-29205	4.011873	5.1696

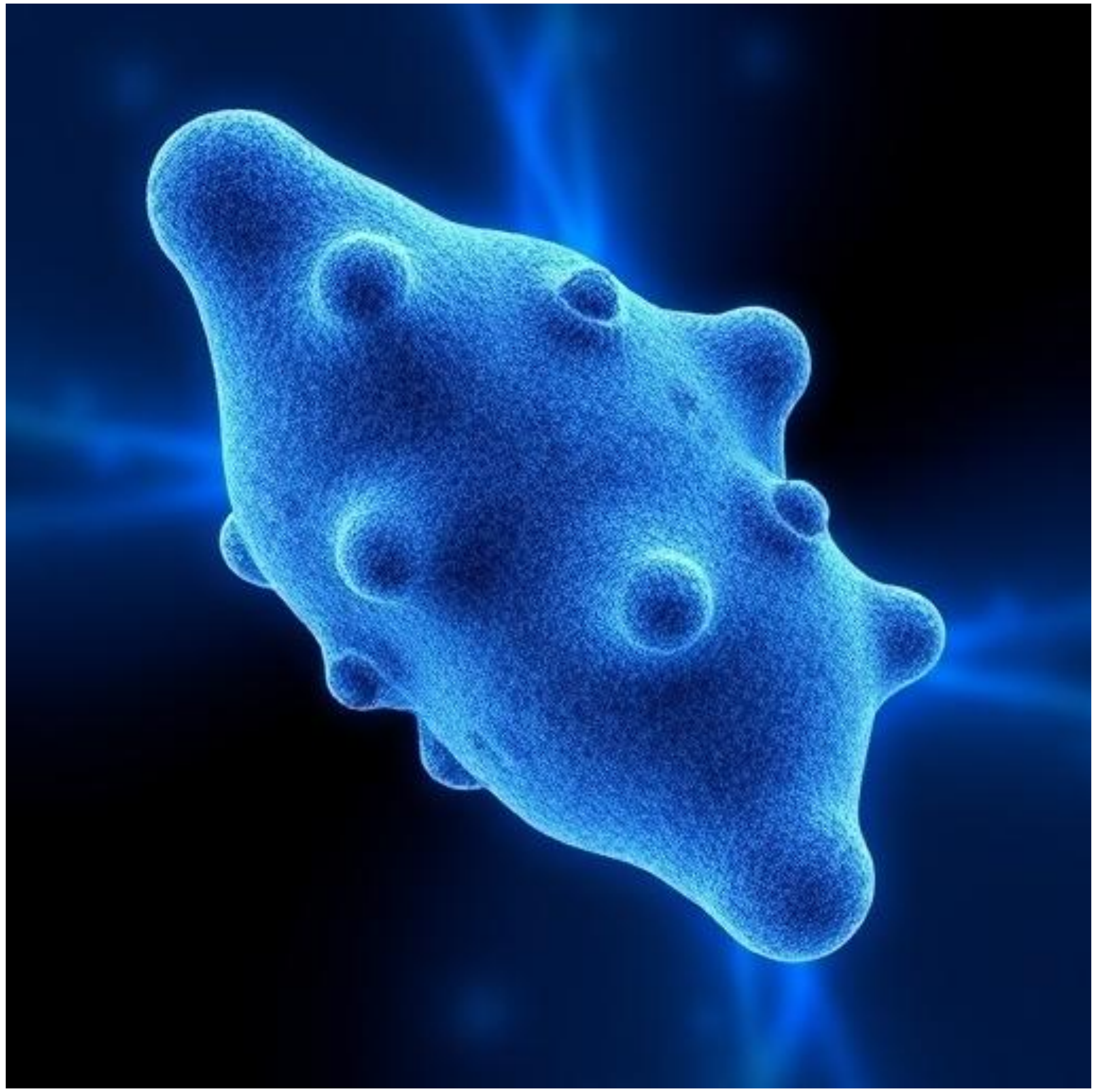
DFT Workflow

1. CIF to POSCAR Conversion
Download CIF files from Materials Project
Convert CIF to POSCAR format using VESTA

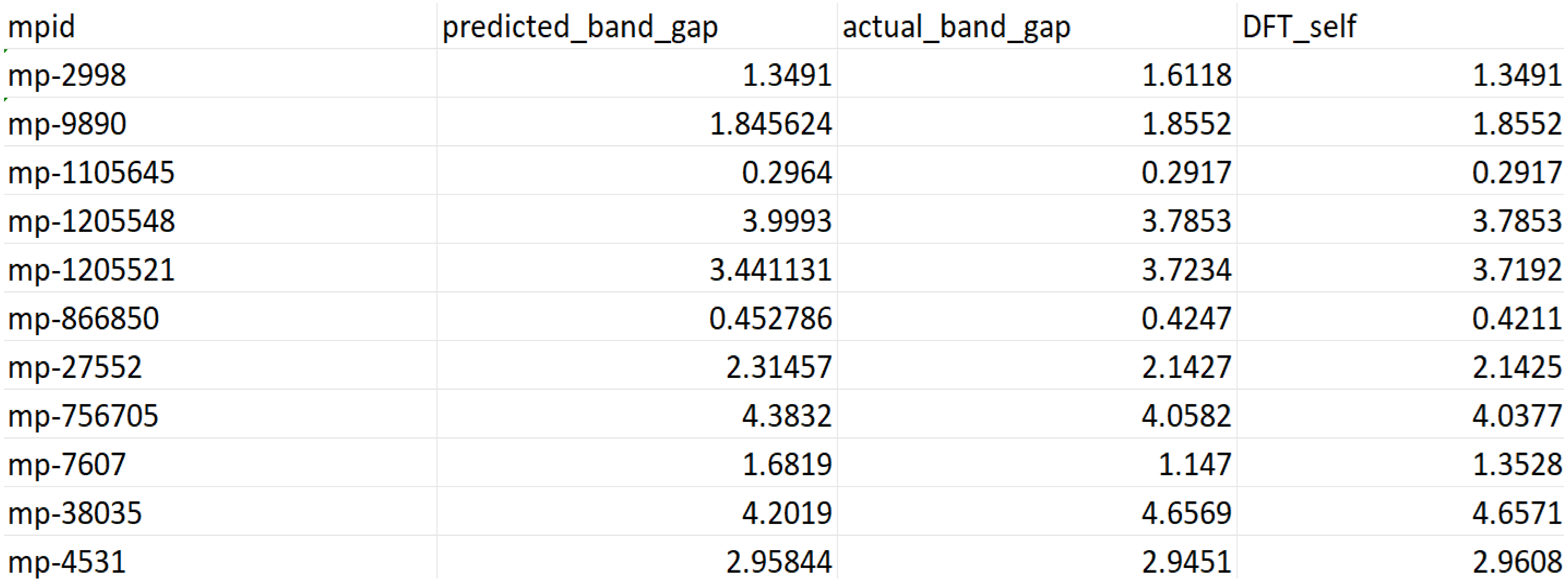
2. Generating Calculation Files
POTCAR: Build using PBE pseudopotentials
INCAR: Set parameters for electronic structure calculation
KPOINTS: Create k-point grid using VASPKIT

3. Self-Consistency Calculation
Run ./JOB file in VASP to achieve self-consistent electron density
Output: CONTCAR (optimized structure)

4. DOS and Band Gap Calculation
Use CONTCAR as new POSCAR for DOS calculation
Run VASPKIT to analyze DOS output and determine Band Gap

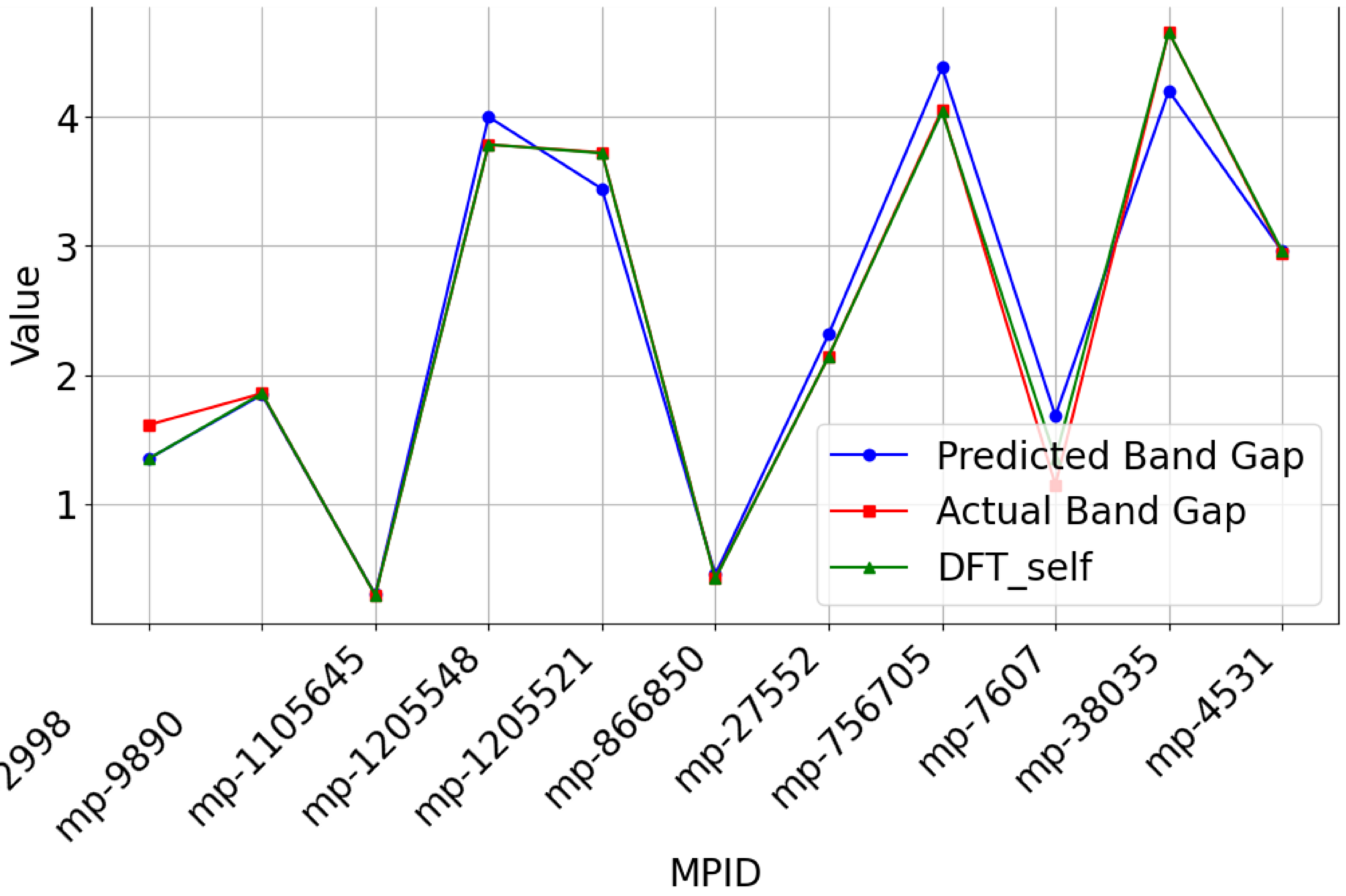


Check with DFT Calculations



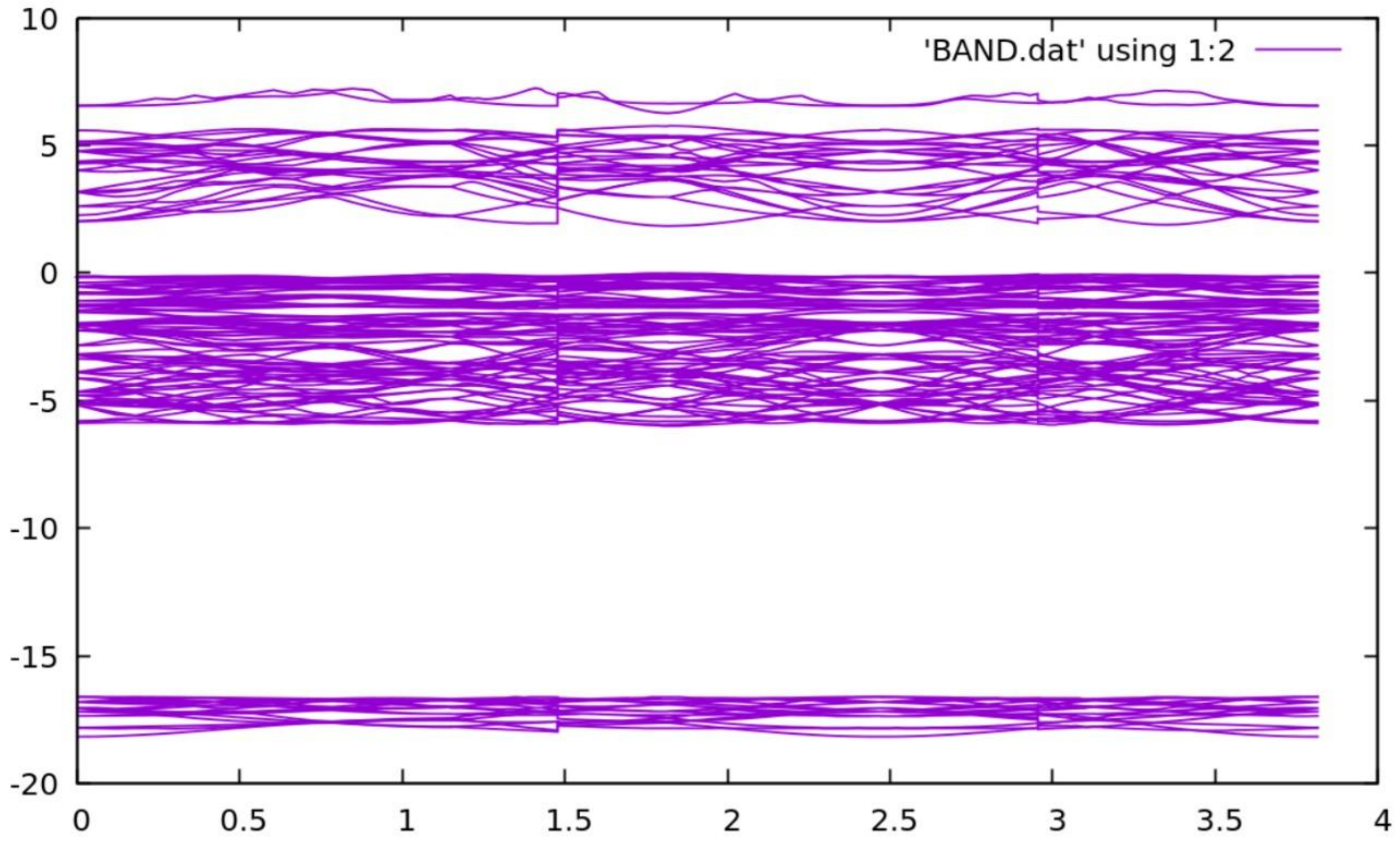
mpid	predicted_band_gap	actual_band_gap	DFT_self
mp-2998	1.3491	1.6118	1.3491
mp-9890	1.845624	1.8552	1.8552
mp-1105645	0.2964	0.2917	0.2917
mp-1205548	3.9993	3.7853	3.7853
mp-1205521	3.441131	3.7234	3.7192
mp-866850	0.452786	0.4247	0.4211
mp-27552	2.31457	2.1427	2.1425
mp-756705	4.3832	4.0582	4.0377
mp-7607	1.6819	1.147	1.3528
mp-38035	4.2019	4.6569	4.6571
mp-4531	2.95844	2.9451	2.9608

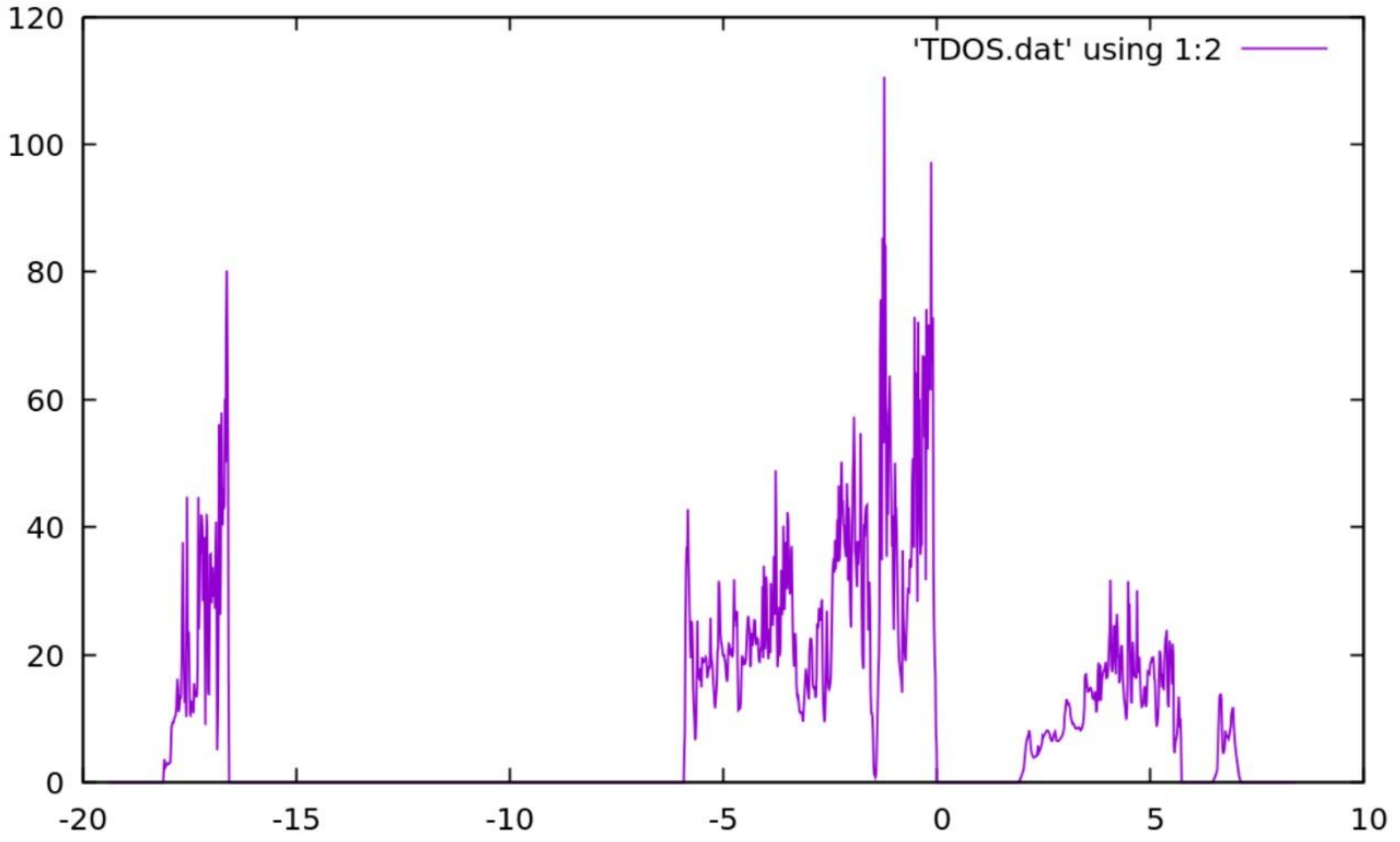
Comparison of Band Gaps (Actual,Predicted and DFT)



Examples of DFT Calculations

DOS and Band Structures of TaAgO3





Other Methods

DFT+U Method:

- Semiempirical approach that introduces an on-site Coulomb interaction (U) to correct for the underestimated band gap in traditional DFT.
- Effective for systems with localized d or f electrons but lacks general theoretical rigor.

Hybrid Density Functionals:

- Incorporates a fraction of nonlocal Fock exchange to improve band gap predictions.
- Reduces band gap error but may vary in accuracy depending on the material and is computationally more demanding than standard DFT.

Modified Semilocal Density Functionals:

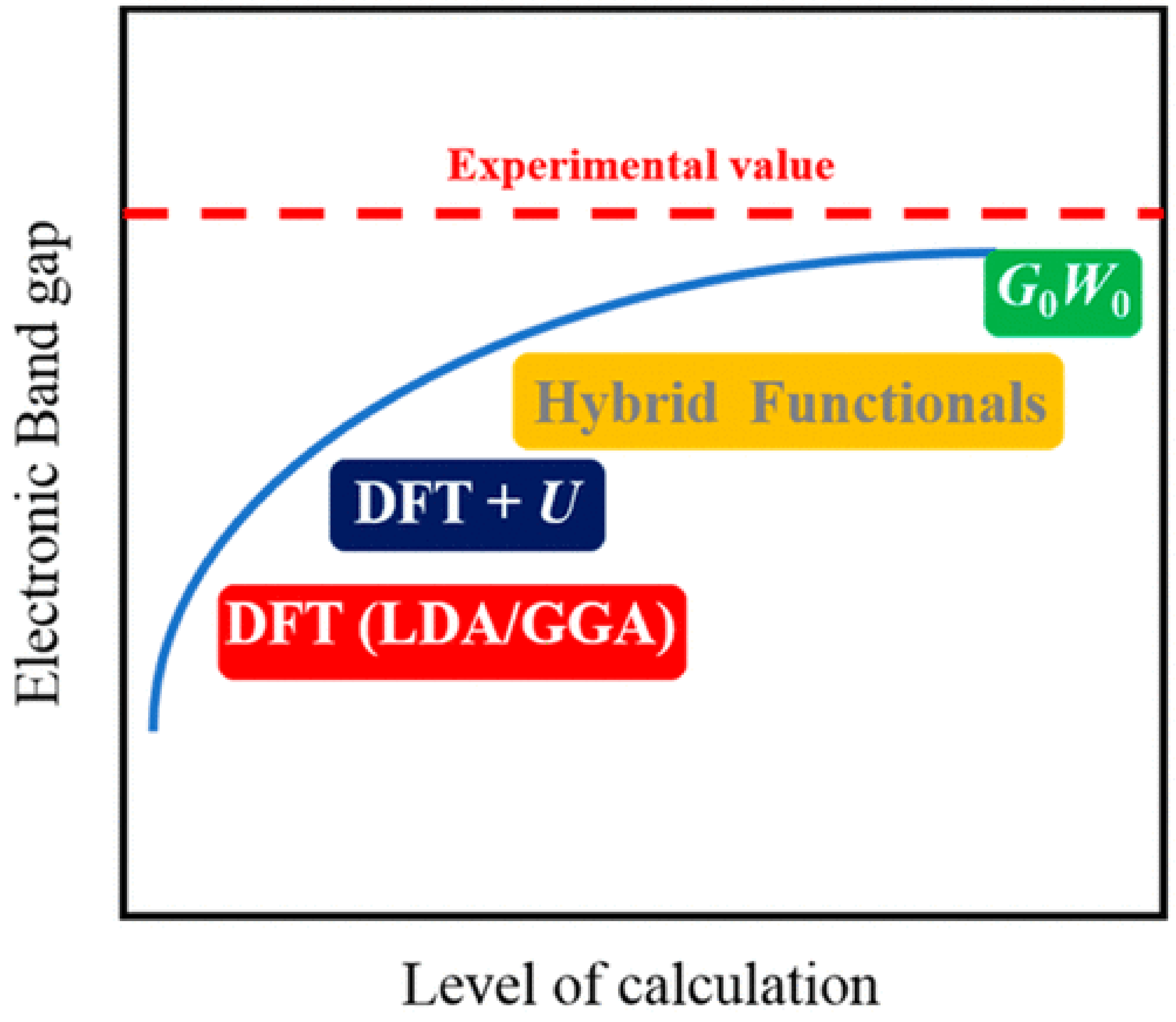
- Adjusts traditional DFT exchange-correlation functionals to better match experimental band gaps.
- Provides more accurate results for specific materials but is still material-dependent.

GW Methods (Many-Body Perturbation Theory):

- Corrects the Kohn–Sham energy levels with a Green's function approach, yielding highly accurate band gaps.
- Computationally intensive and impractical for systems with large unit cells or complex defects.

Linear Correlation Between DFT and GW Band Gaps:

- Observed correlation suggests that standard DFT can approximate GW-quality results by recalibrating band gaps rather than correcting intrinsic DFT limitations.
- This approach can be valuable for complex materials where GW calculations are unfeasible.



Thermodynamic Phase Stability

What is Thermodynamic Phase Stability?

- ❖ Thermodynamic phase stability refers to a material's resistance to decomposition into other phases under equilibrium conditions.
- ❖ A material is considered stable if its energy is at a global minimum when compared to other potential phase combinations it can form.
- ❖ For crystalline materials, stability is evaluated by comparing the formation energy Ef of a compound to the energies of its competing phases.

- **Formation Energy (ΔEf)**: Energy required to form a compound from its constituent elements.

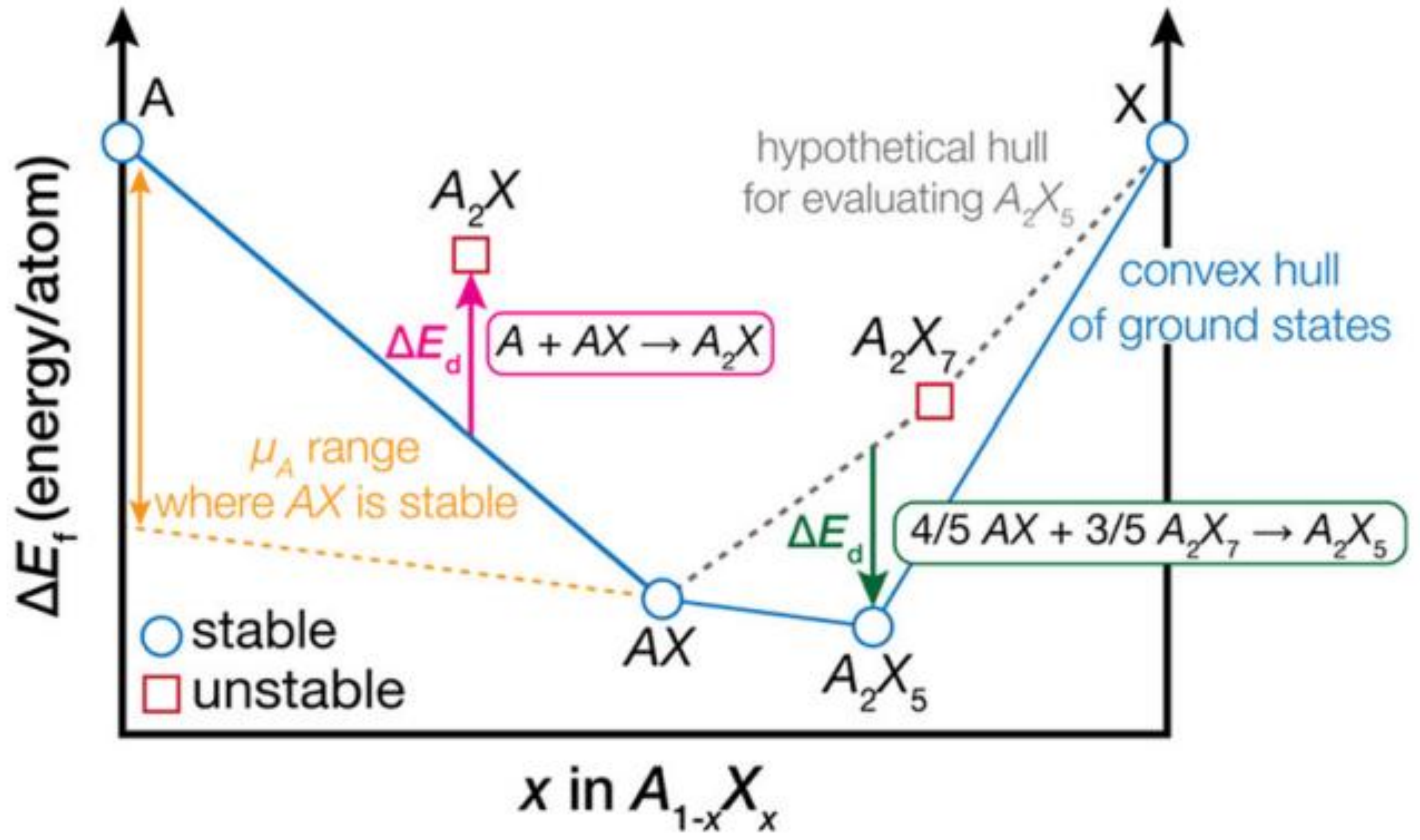
- **Energy Above the Convex Hull (Ehull)**: Measures the energy difference between the compound and the lowest-energy combination of phases (the convex hull) in the phase diagram.

Consider a candidate material with the composition A2X5.

- The formation energy, DEf, of A2X5 is first obtained by referencing to the elemental energies:

$$\Delta E_{f,A_2X_5} = E_{A_2X_5} - 2E_A - 5E_X$$

ΔEf is the reaction energy for forming a compound from its elements (e.g., **2A + 5X -> A2X5**), and **ΔEf>0** indicates the compound is unstable with respect to its elemental phases.



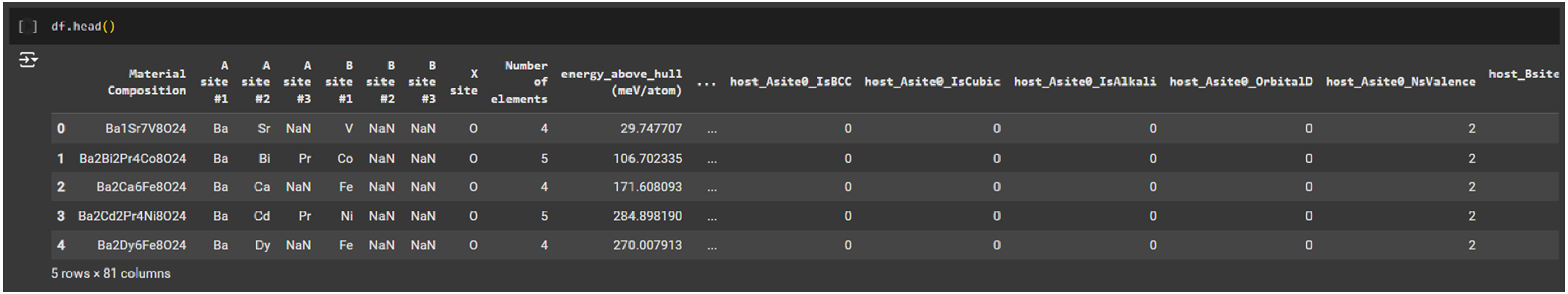
A
X
A_2X
hypothetical hull for evaluating A_2X_5
convex hull of ground states
ΔE_d
$A + AX \rightarrow A_2X$
A_2X_7
μ_A range where AX is stable
ΔE_d
$4/5\ AX + 3/5\ A_2X_7 \rightarrow A_2X_5$
○ stable
□ unstable
AX
A_2X_5
ΔE_f (energy/atom)
x in $A_{1-x}X_x$

Thermodynamic Phase Stability

Phase Stability can be measured by DFT but the significant of computational cost makes such calculation potentially prohibitive when screening large number of possible compounds.

The second work was about the prediction of the formation energies and Stability of family perovskites: AA'BO6

A brief overview of the dataset



```
[ ] df.head()
```

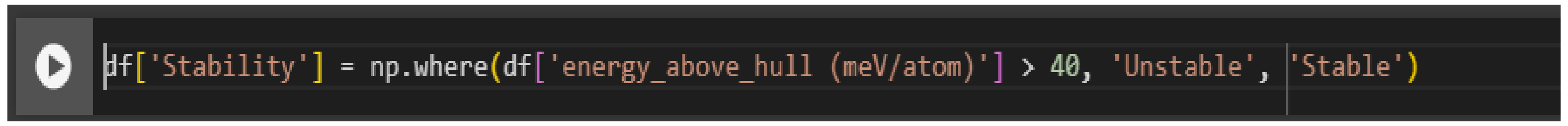
	Material Composition	A site #1	A site #2	A site #3	B site #1	B site #2	B site #3	X site	Number of elements	energy_above_hull (meV/atom)	...	host_Asite0_IsBCC	host_Asite0_IsCubic	host_Asite0_IsAlkali	host_Asite0_OrbitalD	host_Asite0_NsValence	host_Bsite
0	Ba1Sr7V8O24	Ba	Sr	NaN	V	NaN	NaN	O	4	29.747707	...	0	0	0	0	2	
1	Ba2Bi2Pr4Co8O24	Ba	Bi	Pr	Co	NaN	NaN	O	5	106.702335	...	0	0	0	0	2	
2	Ba2Ca6Fe8O24	Ba	Ca	NaN	Fe	NaN	NaN	O	4	171.608093	...	0	0	0	0	2	
3	Ba2Cd2Pr4Ni8O24	Ba	Cd	Pr	Ni	NaN	NaN	O	5	284.898190	...	0	0	0	0	2	
4	Ba2Dy6Fe8O24	Ba	Dy	NaN	Fe	NaN	NaN	O	4	270.007913	...	0	0	0	0	2	

5 rows × 81 columns

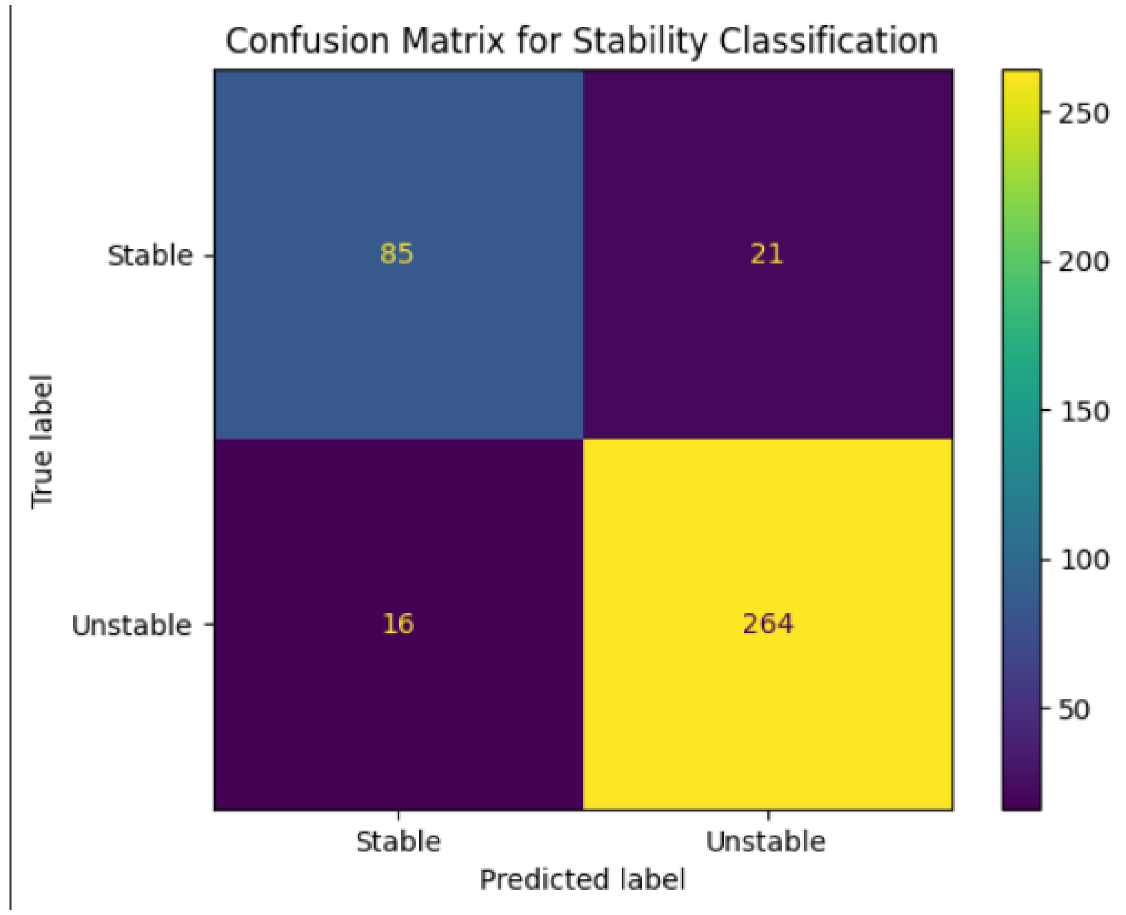
Thermodynamic Phase Stability

Phase Stability was determined using convex hull analysis, with the energy above the convex hull(meV/atom) also called Ehull, providing a direct measure of stability.

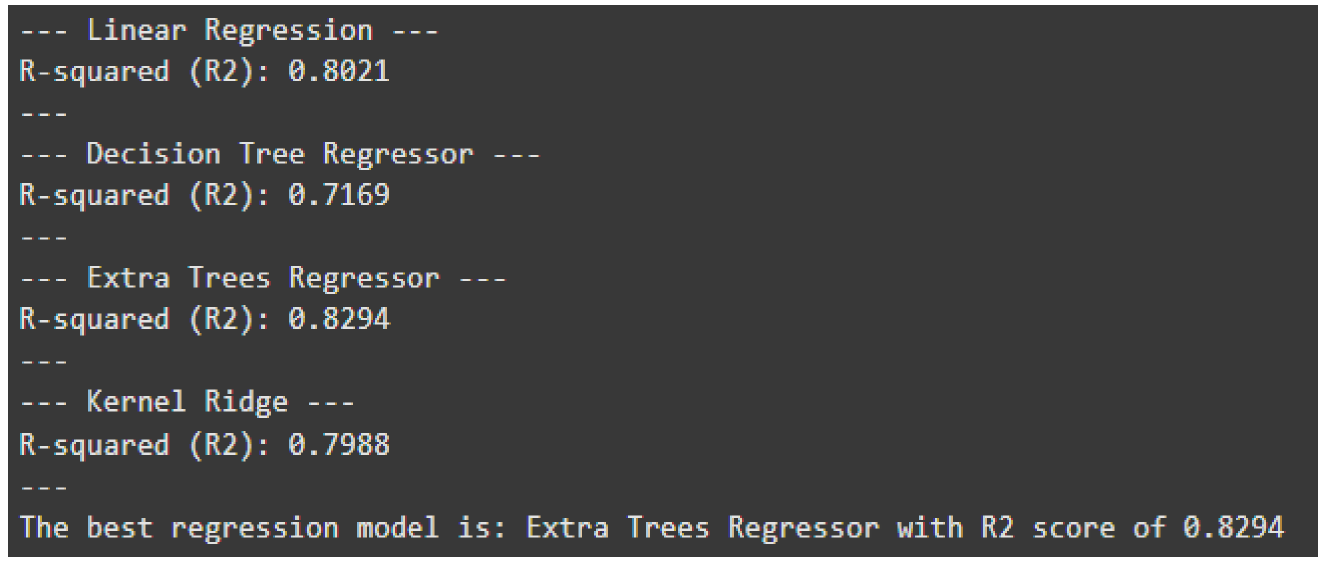
- ❖ **Classification:** For the classification problem, we used a threshold of 40 meV/atom value for Ehull value. Above the threshold value, the material is classified as Unstable.
- ❖ We used Extra Trees Classifier as our best model, with an F1 score of 0.83.
- ❖ The confusion matrix is given for reference.



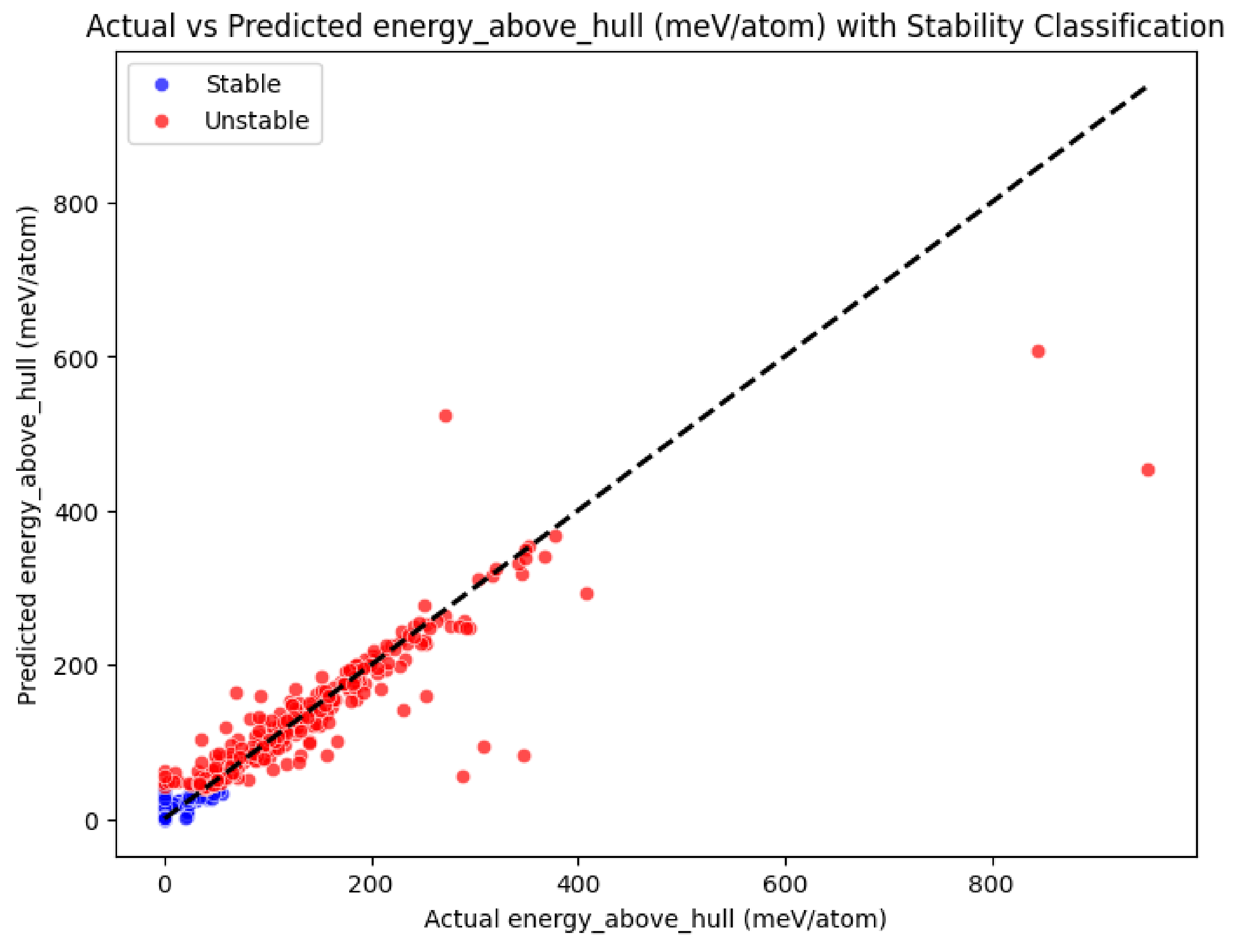
```
df['Stability'] = np.where(df['energy_above_hull (meV/atom)'] > 40, 'Unstable', 'Stable')
```



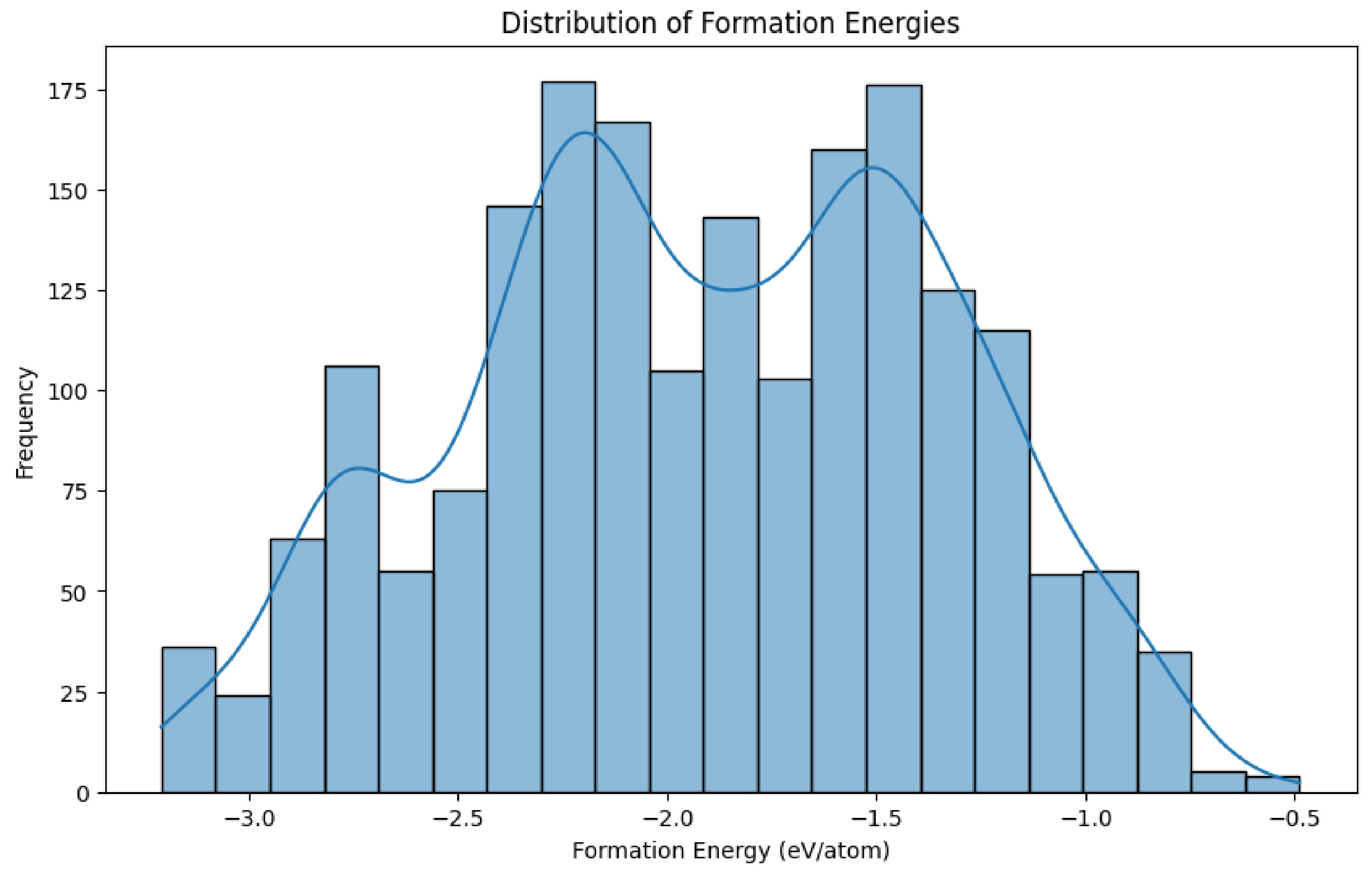
- ❖ **Regression:**
  We used Extra Trees Regressor as our best model,
  giving a R2 value of 0.83 which works fairly well.

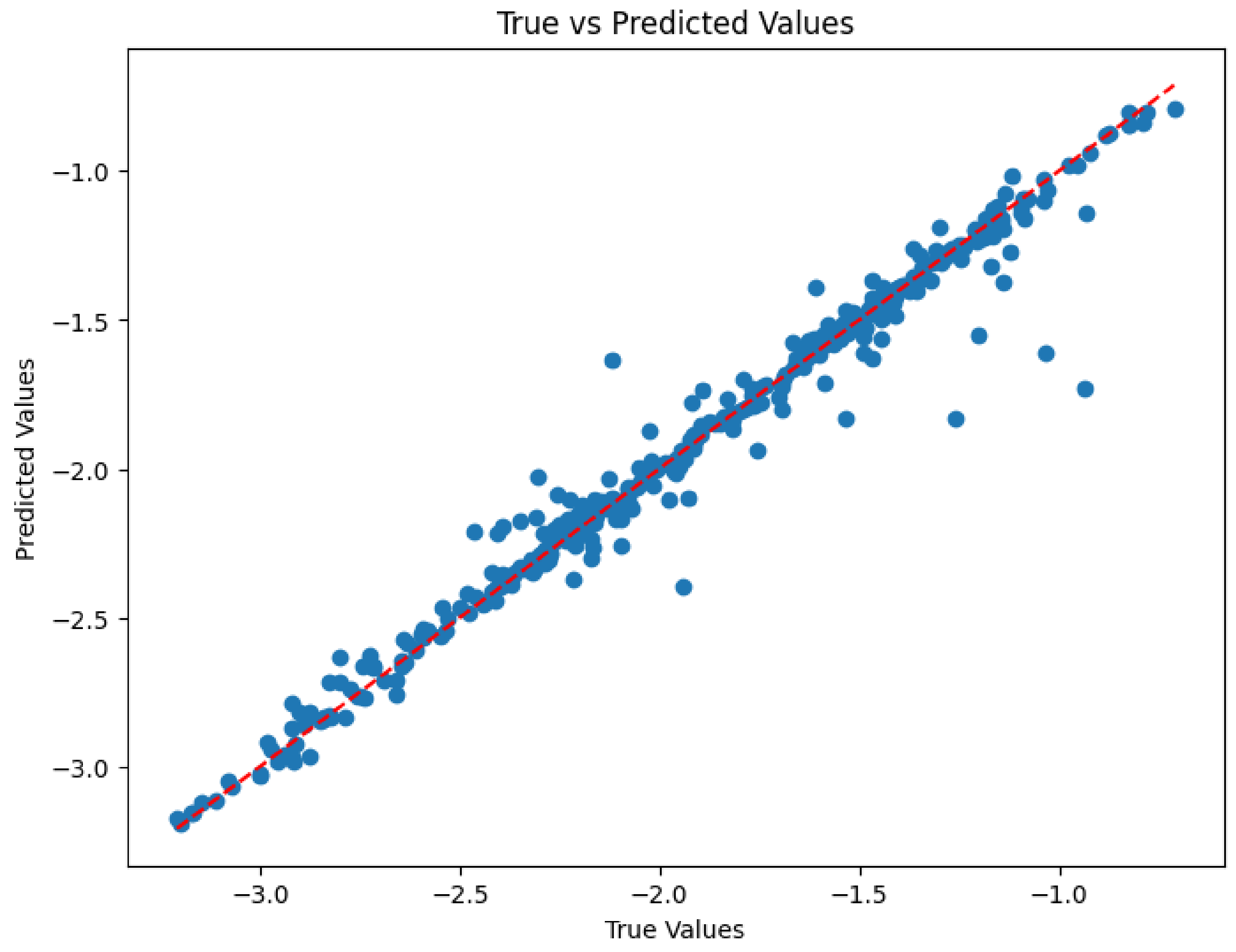


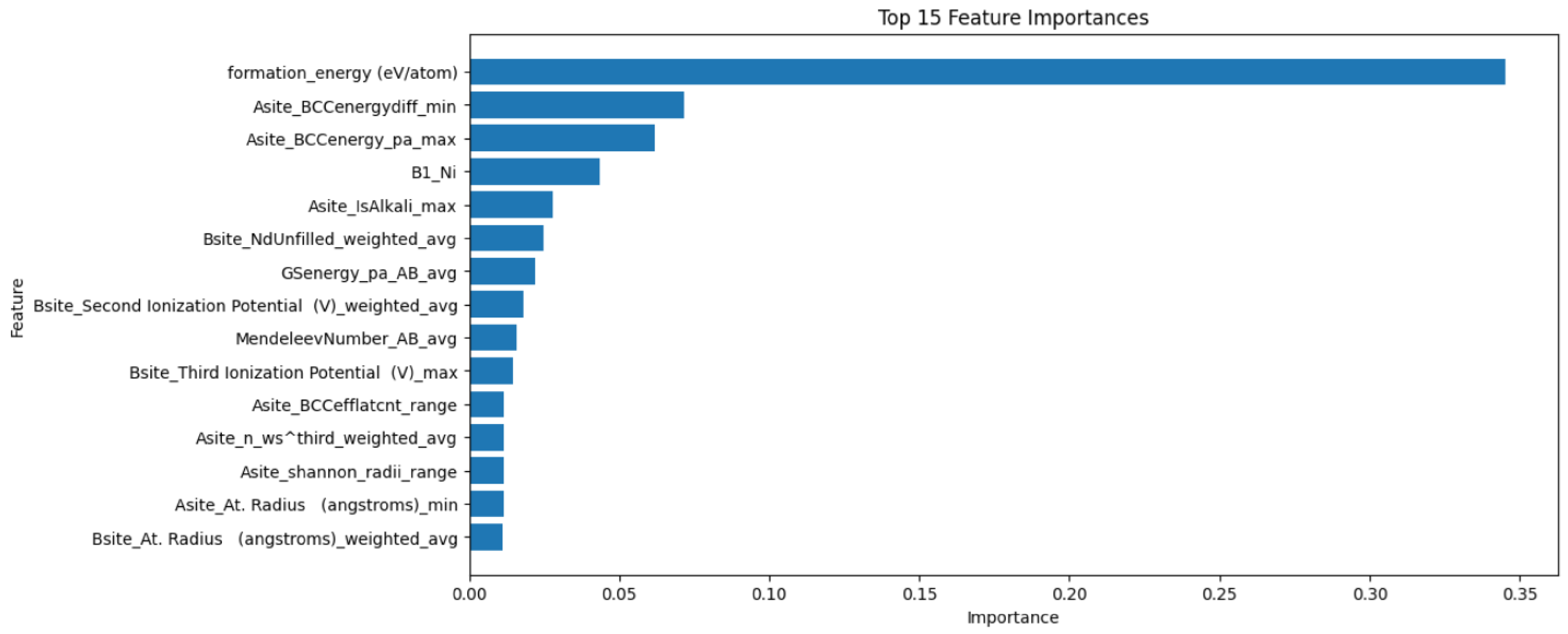
```
--- Linear Regression ---
R-squared (R2): 0.8021
---
--- Decision Tree Regressor ---
R-squared (R2): 0.7169
---
--- Extra Trees Regressor ---
R-squared (R2): 0.8294
---
--- Kernel Ridge ---
R-squared (R2): 0.7988
---
The best regression model is: Extra Trees Regressor with R2 score of 0.8294
```



Some more plots







Thank You!