Project Title: Prediction of Band Gap in Perovskite Materials using ML

- **Subtitle:** A Data-Driven Study Using the `castelli_perovskites` Dataset
- **Author:** Eisuke Otsuka, Ren Sudo, Mamoru Sakaibara
- **Date:** May 2, 2025

Update 2

Update 1

- Accomplished
    - Loaded and explored the `castelli_perovskites` dataset (18,928 entries, 66 columns).
    - Extracted atomic fraction features from chemical formulas using `pymatgen`.
    - Conducted data cleaning:
        - Dropped `structure` due to data type incompatibility.
        - Removed `cbm` and `vbm` because they directly define `gap gllbsc`.
    - Performed EDA:
        - Checked for missing data (none).
        - Visualized correlations and distributions via heatmaps and pairplots.
    - Built and evaluated baseline ML models:
        - Linear Regression and Random Forest with 5-fold CV.
        - Achieved R^2 = 0.539 ± 0.059 and MAE = 0.068 ± 0.006 with Random Forest.
    - Interpreted results with Permutation Importance (Fig.5).
- Did Not Work / Postponed
    - MEGNet-based modeling with structure data is not yet implemented (planned for Update 2).
    - SHAP analysis and uncertainty estimation are still pending.
- Planned Next Steps
    - Implement structure-based graph construction and modeling via MEGNet.
    - Explore Gradient Boosting and model comparison.
    - Perform SHAP interpretation for better feature understanding.
    - Consider adding space group, density, and crystal features.
- Questions / Reflections
    - How can we best evaluate feature redundancy in atomic fractions?
    - What is the minimum structural information needed to significantly improve performance?
- Referenced Figures:
    - Fig.1: Pairplot

    - Fig.2: Correlation Heatmap
    - Fig.3–5: Model predictions and Permutation Importance

Excuetive Summary

- Dataset
    - Used `castelli_perovskites` dataset from `matminer`.
    - 18,928 perovskites with features such as band gap (`gap gllbsc`), formation energy (`e_form`), magnetic moment (`mu_b`), and chemical formula.
- Data Cleaning
    - Dropped non-numeric `structure` column.
    - Removed redundant `cbm` and `vbm` due to deterministic relationship with band gap.
- Exploratory Data Analysis (EDA)
    - No missing data; feature types = 63 float, 2 object, 1 boolean.
    - Heatmaps and pairplots identified top compositional features correlated with band gap (e.g., `O`, `N`, `mu_b`, `e_form`).
- Modeling
    - Trained Linear Regression and Random Forest using 5-fold cross-validation.
    - Random Forest outperformed Linear Regression:
        - R^2 = 0.539 ± 0.059, MAE = 0.068 ± 0.006
    - Feature importance (via permutation) highlighted `e_form`, `mu_b`, and atomic fractions (Si, B, Al, Sc).
- Key Insights
    - Band gap prediction from composition alone is feasible but limited (R^2 ~0.54).
    - More complex models or structural data are likely needed for further improvement.
- Next Steps
    - Implement graph-based learning using MEGNet.
    - Introduce SHAP for interpretability and structure-aware features.

Abstract

- This project explores the prediction of electronic band gaps in hypothetical perovskite materials using the `castelli_perovskites` dataset provided by the `matminer` library. We applied classical machine learning methods to investigate the relationship between physicochemical descriptors and the computed band gap (`gap gllbsc`). Our approach included exploratory data analysis, element-wise composition vectorization, and supervised learning (linear regression and random forest). While linear models performed poorly, random forest achieved a moderate predictive accuracy, highlighting the

non-linear nature of band gap formation. Future work may incorporate structural features or deep learning approaches.

Introduction

- In materials science, predicting key physical properties from chemical composition and crystal structure is a critical task in accelerating the discovery of novel materials. One particularly important property is the electronic band gap, which governs a material's electrical behavior—determining whether it acts as a metal, semiconductor, or insulator. This makes band gap prediction a central problem in materials informatics.
- Perovskite materials (especially ABX_3-type compounds) are of growing interest due to their versatile physical and chemical properties, which make them promising candidates for applications such as photovoltaics, optoelectronics, and photocatalysis. In this project, we aim to predict the electronic band gaps of hypothetical perovskites using data-driven methods.
- We use the `castelli_perovskites` dataset from the `matminer` library, which includes 18,928 computationally generated perovskite structures along with properties such as:
  - `gap gllbsc` (band gap computed with the GLLB-SC functional),
  - `e_form` (formation energy),
  - `mu_b` (magnetic moment),
  - `structure` information (as pymatgen Structure objects).
- In this initial study, we focus on predicting the band gap using compositional and elemental features extracted from the chemical formula. While the dataset also contains structural information, we defer the use of structure-based features or graph neural networks (GNNs) to future work due to current technical limitations.
- Our goals are:
  - To explore how traditional machine learning models (e.g., linear regression, random forest) can be used for band gap prediction,
  - To identify key compositional descriptors correlated with the band gap,
  - And to build a foundation for future extensions using more complex models and feature types.
- Data Source: Matminer `castelli_perovskites` dataset.

Data Science Methods

- In this project, we used classical supervised learning techniques to predict the electronic band gap (`gap gllbsc`) of inorganic perovskite materials.
- Tools and Libraries:
  - `matminer`: dataset loading
  - `pymatgen`: chemical formula handling
  - `pandas`, `numpy`, `seaborn`, `matplotlib`: data handling and visualization

    - `scikit-learn`: modeling and evaluation

- Feature Engineering:

    - We converted the chemical formula to atomic fraction vectors using `pymatgen.Composition`.
    - The `structure` column was excluded due to data complexity and limitations in this phase.

- Modeling:

    - Linear Regression and Random Forest Regressor were trained.

- Evaluation

    - 5-fold cross-validation (metrics: MAE, R^2).
    - Permutation importance was used to interpret model results.

- Graph Construction and Modeling (Planned for Update 2)

    - Although not yet implemented in the current update, we plan to incorporate structure-based graph representations using the MEGNet framework in the next phase of the project (Update 2). In this approach:
    - Graph Construction:
        - Each perovskite material will be converted into a graph, where atoms are represented as nodes and edges correspond to interatomic interactions within a specified cutoff radius.
        - We will utilize MEGNet's built-in graph converter, which accepts `pymatgen.Structure` objects directly and transforms them into model-ready graph inputs.
    - Modeling:
        - The MEGNet model will be used as a graph neural network (GNN) for regression, predicting the `gap gllbsc` value (band gap) from structural information.
        - This method allows learning directly from the atomic environment and spatial relationships, which are often critical for electronic property prediction.
    - This enhancement aims to address current limitations of composition-only models by incorporating richer physical structure into the learning process.

Exploratory Data Analysis

Explanation of your data set

- We analyzed the `castelli_perovskites` dataset from `matminer`, which contains 18,928 perovskite samples and 66 columns after atomic fraction features were included(Table.1).

- Original features include:

    - `e_form` (formation energy)
    - `gap gllbsc` (band gap via GLLB-SC functional)
    - `mu_b` (magnetic moment)
    - `fermi level`, `fermi width`, `cbm`, `vbm`, etc.
    - `structure`: pymatgen Structure object (not used in this analysis)

- Additional features were generated by decomposing `formula` into elemental atomic fractions, resulting in 50+ extra columns.

```
# Convert formula to comoposition
from pymatgen.core.composition import Composition
from tqdm import tqdm
tqdm.pandas()

def get_atomic_fraction_dict(formula):
    try:
        comp = Composition(formula)
        return comp.fractional_composition.as_dict()
    except Exception as e:
        print(f"Error at: {formula}, {e}")
        return {}

# Apply
df_atomic_frac = df["formula"].progress_apply(get_atomic_fraction_dict)

df_atomic_frac = pd.DataFrame(df_atomic_frac.tolist()).fillna(0)

# Merged into the original DataFrame
df = pd.concat([df, df_atomic_frac], axis=1)

# Basic information on the data set
print("Number of rows:", df.shape[0])
print("Number of variables (columns):", df.shape[1])
print("\nData types:\n", df.dtypes.value_counts())
print("\nMissing values per column:\n", df.isnull().sum())
print("\nUnique values in categorical columns:\n", df.drop(columns=
["structure"]).select_dtypes(include=["object", "bool"]).nunique())
```

Item	Value
Number of rows	18,928
Number of columns	66
Data types	float64 (63), object (2), bool (1)
Missing values	None
Categorical variables	`gap is direct`: 2 values, `formula`: 18,928 unique

- Table.1:Basic information on the data set

Data Cleaning

- Dropped the `structure` column due to incompatibility with standard ML pipelines (non-hashable object).
- Removed `cbm` and `vbm` from the modeling phase, as they are mathematically related to `gap gllbsc`.

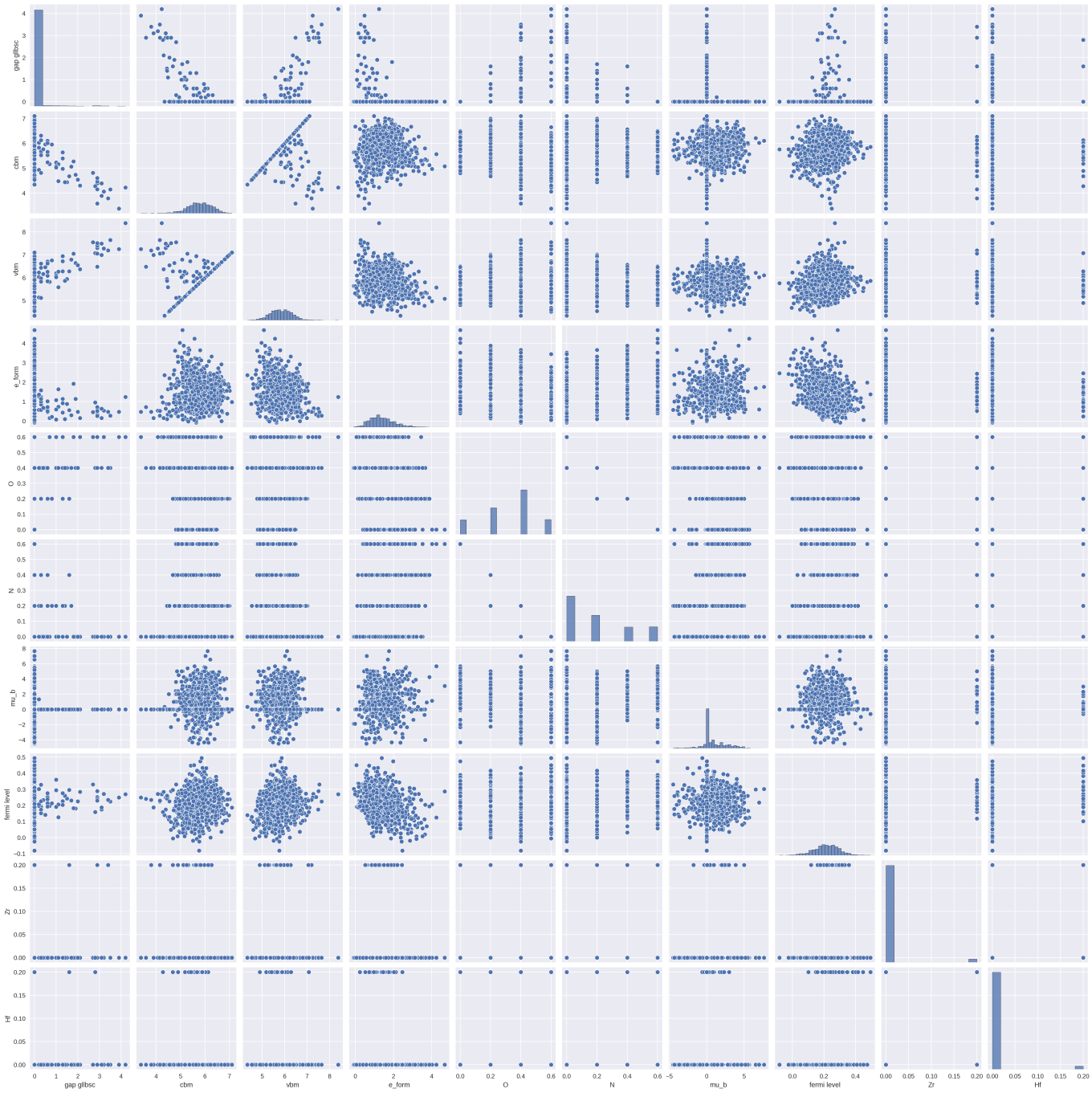
- Verification:

```
diff = df["cbm"] - df["vbm"]
is_equal = np.isclose(diff, df["gap gllbsc"])
print(is_equal.sum() / len(df))  # → About 96% are exact matches.
```

Data Vizualizations

- To better understand pairwise relationships and feature distributions, we generated a pairplot for the top 10 features. This helped identify trends, outliers, and non-linear dependencies(Fig.1).
  - Pairwise trends between `gap gllbsc`, `e_form`, and atomic fractions were weakly non-linear.
  - Many element fraction features (e.g., `O`, `N`, `Zr`) showed sparse or binary distributions.

```
# Select numeric columns only
num_cols = df_top_features.select_dtypes(include="number").columns

# Limit to 1000 samples as too many samples are heavy
sns.pairplot(df_top_features[num_cols].sample(1000, random_state=0))
plt.show()
```



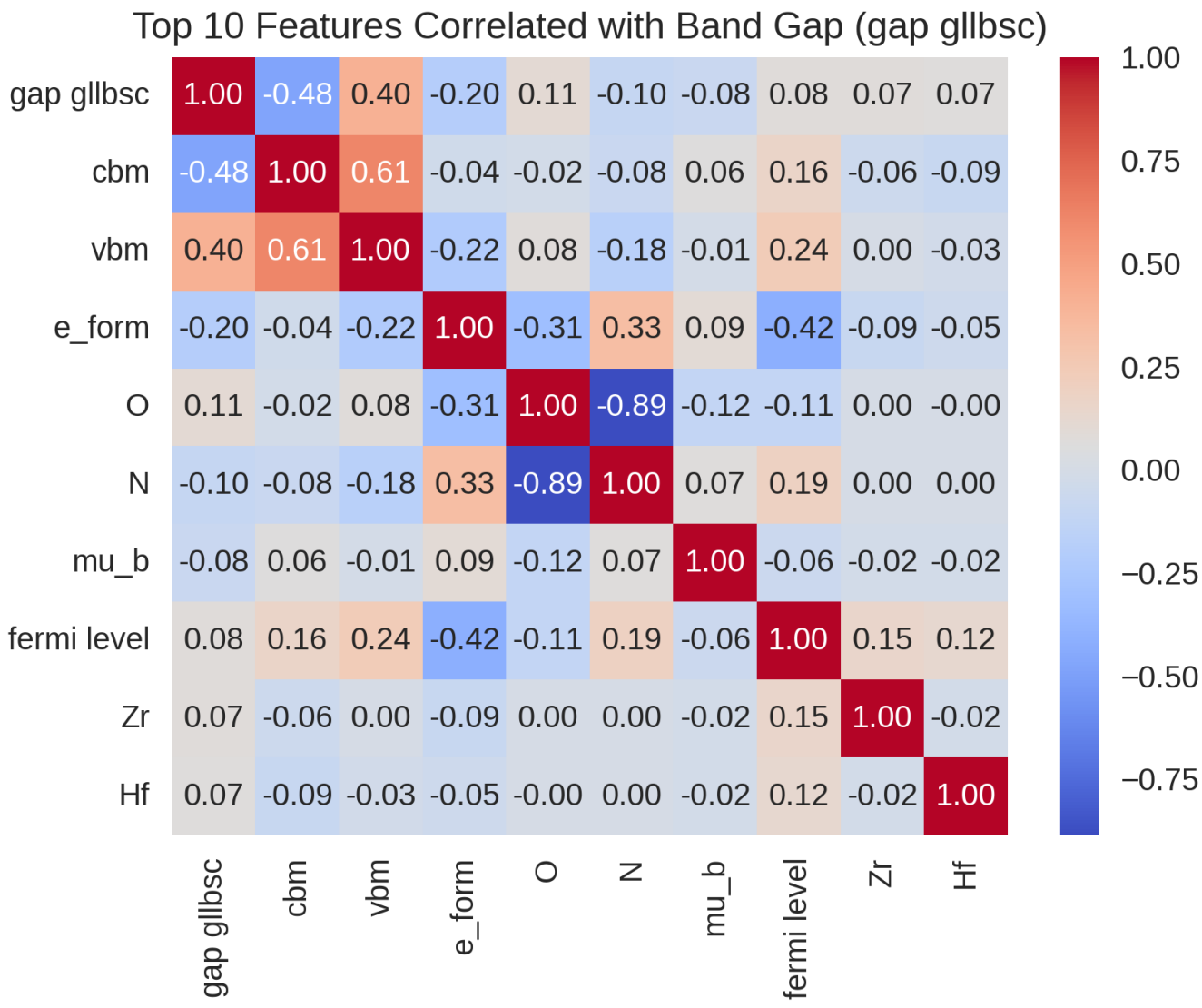
- Fig.1: Pairplot of top 10 features related to band gap.

Variable Correlations

- A correlation matrix was computed to identify the features most associated with the band gap (Fig.2). The 10 most correlated features include:
- `cbm`, `vbm` (positively correlated)
- `O`, `N`, `mu_b`, `e_form`, `fermi level`, etc.

```
# Correlation heat map
import numpy as np

# Target variable.
target_col = "gap gllbsc"
```

```
# Obtain the absolute value of the correlation coefficient and select the top n
correlations with the objective variable
top_n = 10
corr_with_target = df_features[numeric_cols].corr()
[target_col].abs().sort_values(ascending=False)
top_features = corr_with_target.head(top_n).index.tolist()

df_top_features = df_features[top_features]

# Visualisation of correlation matrices (limited to upper-level features)
sns.heatmap(df_top_features.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title(f"Top {top_n} Features Correlated with Band Gap (gap gllbsc)")
plt.show()
```



- Fig.2: Top 10 features correlated with `gap gllbsc`.

Statistical Learning: Modeling & Prediction

- To explore the relationship between material features and the band gap (gap gllbsc), we applied two types of supervised learning models: a simple linear regression and a non-linear ensemble model

(Random Forest Regressor).

Model selection

- Linear Regression
  - A baseline model was constructed using a standard linear regression on all numerical features, including physical properties (e_form, mu_b, etc.) and atomic fractions derived from composition.
- Random Forest Regression
  - To capture potential non-linear relationships, we trained a Random Forest model with default parameters, using 5-fold cross-validation to ensure generalization.

Cross-validation, Predictive R2

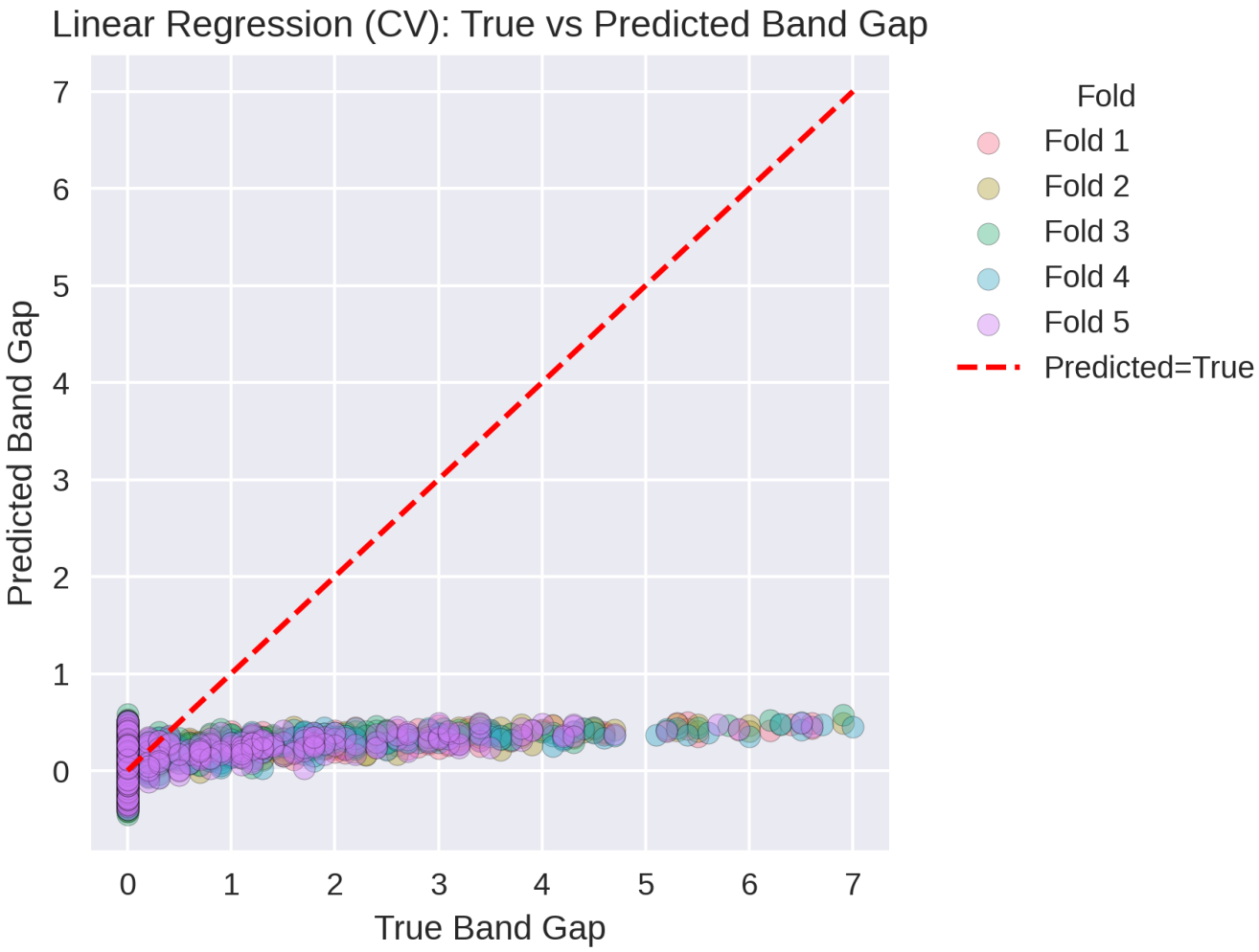
```
# Definition of features and target variables
X = df_features.drop(columns=["gap gllbsc", "formula", "gap is direct", "cbm",
"vbm", "structure"])
y = df_features["gap gllbsc"]
feature_names = X.columns

# Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Preparing models and cross-validation
model = RandomForestRegressor(n_estimators=100, random_state=42, n_jobs=-1)
cv = KFold(n_splits=5, shuffle=True, random_state=42)

# For saving results
y_true_all, y_pred_all, fold_ids = [], [], []
importances_list = []
mae_list, r2_list = [], []

# Color palette for folds
colors = sns.color_palette("husl", n_colors=cv.get_n_splits())

# CV loop
for fold, (train_idx, val_idx) in enumerate(cv.split(X_scaled)):
    X_train, X_val = X_scaled[train_idx], X_scaled[val_idx]
    y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

    model.fit(X_train, y_train)
    y_pred = model.predict(X_val)

    y_true_all.extend(y_val)
    y_pred_all.extend(y_pred)
    fold_ids.extend([fold] * len(y_val))  # Store fold info for each point

    # Score output
    mae = mean_absolute_error(y_val, y_pred)
    r2 = r2_score(y_val, y_pred)
    mae_list.append(mae)
```

```
    r2_list.append(r2)
    print(f"Fold {fold+1} - MAE: {mae:.3f}, R²: {r2:.3f}")

    # Permutation importance (by val data)
    result = permutation_importance(model, X_val, y_val, n_repeats=5,
random_state=42, n_jobs=-1)
    importances_list.append(result.importances_mean)

# Average score output
print(f"\nAverage MAE: {np.mean(mae_list):.3f} ± {np.std(mae_list):.3f}")
print(f"Average R²: {np.mean(r2_list):.3f} ± {np.std(r2_list):.3f}")

# Scatterplot with fold-wise color & legend
for fold in range(cv.get_n_splits()):
    indices = [i for i, f in enumerate(fold_ids) if f == fold]
    plt.scatter(
        np.array(y_true_all)[indices],
        np.array(y_pred_all)[indices],
        color=colors[fold],
        alpha=0.4,
        label=f"Fold {fold+1}",
        edgecolor='k',
        linewidth=0.2
    )
plt.plot([min(y_true_all), max(y_true_all)], [min(y_true_all), max(y_true_all)],
'r--',label="Predicted=True")
plt.xlabel("True Band Gap")
plt.ylabel("Predicted Band Gap")
plt.title("Random Forest (CV): True vs Predicted Band Gap")
plt.legend(title="Fold", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

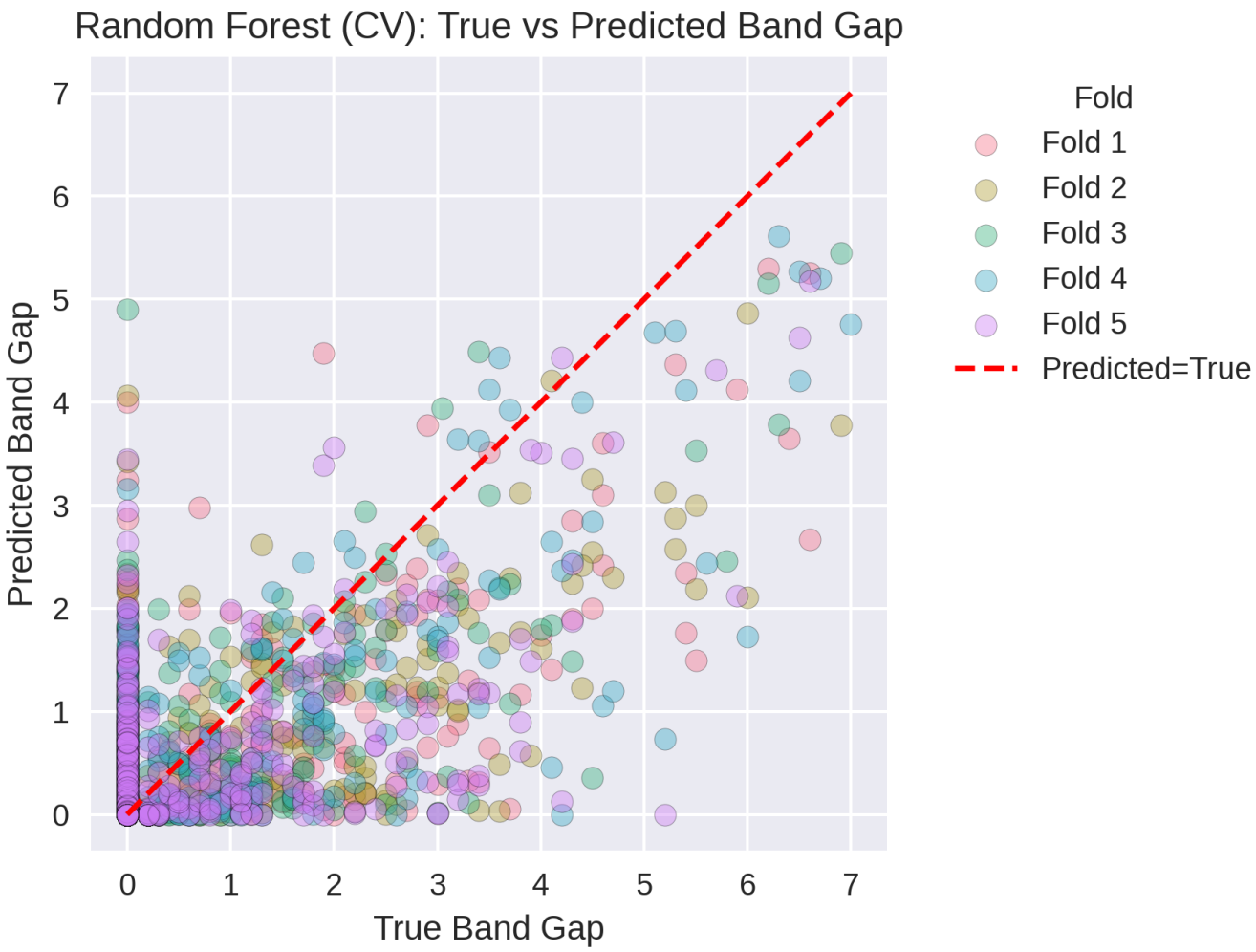
- We used 5-fold cross-validation to evaluate both models, computing the Mean Absolute Error (MAE) and R^2 score for each fold. The predictive performance is summarized below:
- Random Forest significantly outperformed the linear model(Fig.3,Fig.4), highlighting the importance of non-linear modeling in materials property prediction.

Model	MAE	R^2 Score
Linear Regression	0.173 ± 0.005	0.084 ± 0.012
Random Forest	0.068 ± 0.006	0.539 ± 0.059

- Table.2:Scores for each model in 5-fold CV



- Fig.3:Linear Regression (CV): True vs Predicted Band Gap

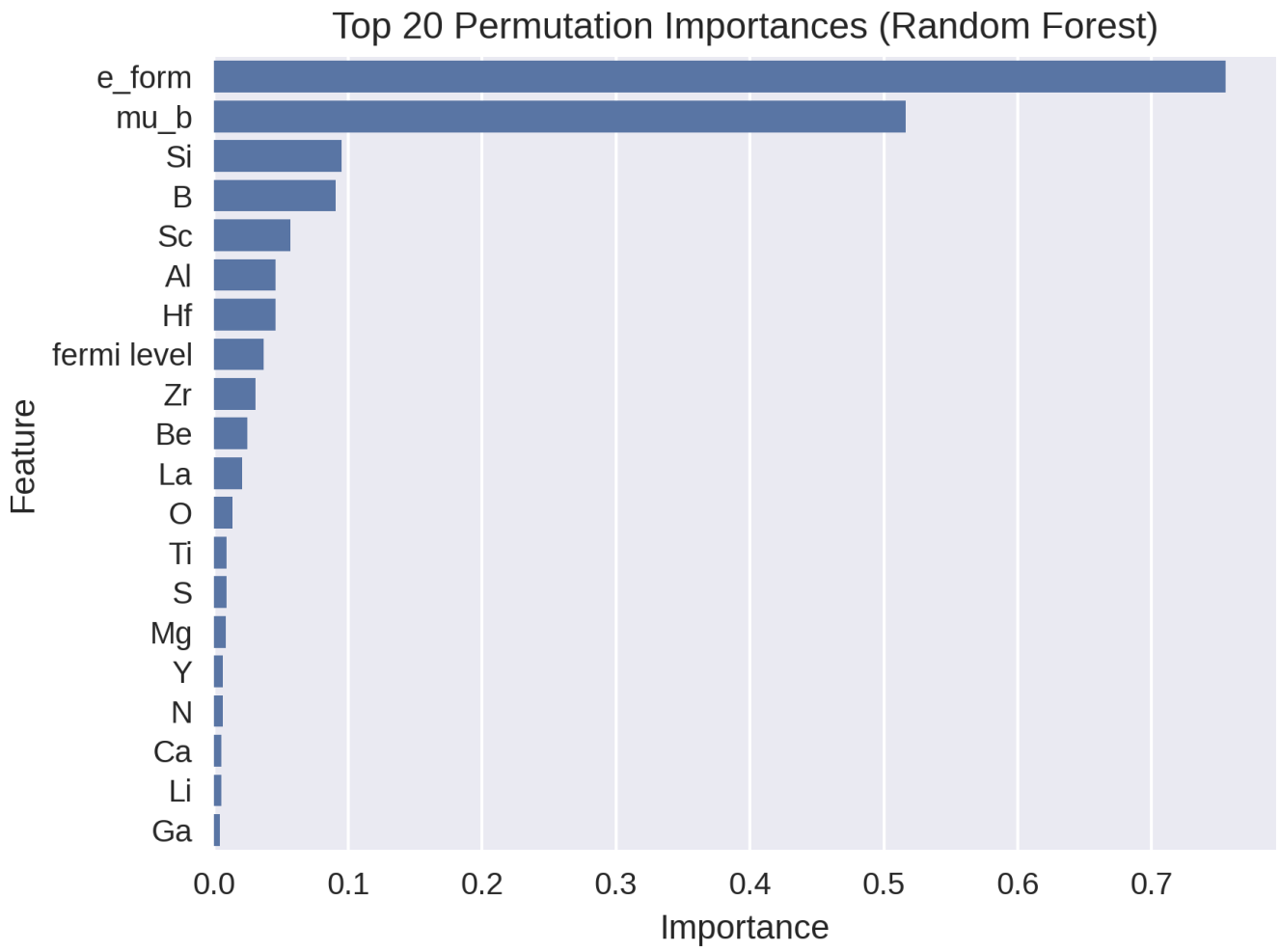


- Fig.4:Random Forest (CV): True vs Predicted Band Gap

Interpret results

```
# Permutation importance
avg_importance = np.mean(importances_list, axis=0)
imp_df = pd.DataFrame({"Feature": feature_names, "Importance": avg_importance})
imp_df = imp_df.sort_values("Importance", ascending=False).head(20)

sns.barplot(data=imp_df, x="Importance", y="Feature")
plt.title("Top 20 Permutation Importances (Random Forest)")
plt.tight_layout()
plt.show()
```

- To interpret model behavior, we applied Permutation Importance analysis to the trained Random Forest model(Fig.5). The most influential features were:
  - e_form: formation energy
  - mu_b: magnetic moment
  - Atomic fractions of elements such as Si, B, Sc, Al
- These features likely encode critical physical and compositional information tied to electronic structure and stability.



- Fig.5:Top 20 Permutation Importances (Random Forest)

Discussion

- The central question of this study was whether the electronic band gap of perovskite-like materials could be reliably predicted using only compositional features and classical ML methods.
- Our findings suggest several key insights:
    - Non-linearity is critical: Linear regression performed poorly ($R^2 \approx 0.08$), suggesting that band gap formation cannot be described by a simple linear combination of elemental fractions or physical properties.
    - Random Forests capture complexity: The Random Forest model achieved a moderate $R^2 \approx 0.54$, indicating that it successfully modeled non-linear interactions among features. However, substantial variance remains unexplained.
    - Feature importance is physically meaningful: Formation energy (`e_form`) and magnetic moment (`mu_b`) were identified as dominant features, which aligns with known factors influencing electronic structure.
    - Atomic fractions encode chemical diversity: Even without explicit structural data, features like the presence of B, Si, Al, and Sc were strongly correlated with band gap variations. This shows that compositional information alone captures valuable patterns, making this approach promising for rapid screening.

- That said, the following limitations constrain predictive accuracy:
    - Lack of structural descriptors: Since band gap is strongly dependent on local bonding, orbital overlap, and symmetry, excluding the `structure` column likely removed critical explanatory variables.
    - Feature collinearity: Some elemental fractions may introduce redundancy, affecting model stability.
    - Uncertainty estimation is missing: No probabilistic assessment of prediction confidence was made, which is important for real-world material discovery workflows.

Conclusions

- This project demonstrated that machine learning models trained on compositional features can partially predict electronic band gaps in perovskite materials. Our main conclusions are:
    - Random Forest models significantly outperform linear regression, underscoring the importance of modeling non-linear effects in materials property prediction.
    - Physically grounded features such as formation energy and magnetic moment consistently emerge as important, validating their relevance in determining electronic structure.
    - Composition-only models are effective for initial screening, particularly when structural information is unavailable.
- Future directions to improve model performance and utility include:
    - Incorporating structural features (e.g., space group, coordination environment, graph-based representation).
    - Applying GNNs such as MEGNet to directly learn from atomic structures.
    - Introducing uncertainty quantification, which is crucial for experimental prioritization.
    - Using SHAP or other model-agnostic explanation tools to better understand interactions among features.
- This study lays a foundation for more advanced modeling and paves the way for integrating structure-aware deep learning techniques in future updates.

Acknowledgments

- We would like to thank Professor Pawan Tripathi for organizing the course Materials Informatics and for providing valuable guidance throughout the project.
- We also acknowledge the contributions of Ivano E. Castelli et al. for making the `castelli_perovskites` dataset publicly available via the `matminer` library.
- This project was conducted as part of the coursework at materials data science course offered in spring of 2025 at Tohoku University, Japan, and we appreciate the collaborative discussions within the class and group.

References

1. Castelli, I. E., Landis, D. D., Thygesen, K. S., Dahl, S., Chorkendorff, I., Jaramillo, T. F., & Jacobsen, K. W. (2012).
   New cubic perovskites for one- and two-photon water splitting using the computational materials repository.
   Energy & Environmental Science, **5**(10), 9034–9043. https://doi.org/10.1039/C2EE22341D

2. Matminer documentation.
   http://hackingmaterials.lbl.gov/matminer/

3. Sklearn documentation.
   https://scikit-learn.org/stable/