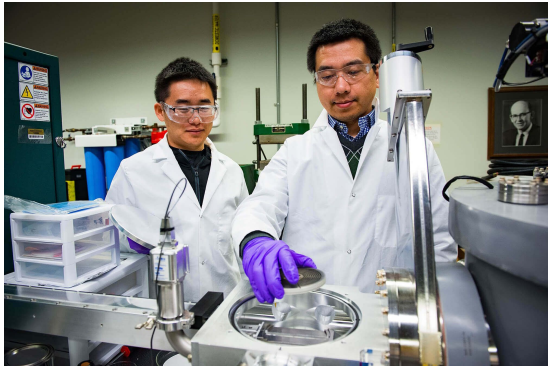
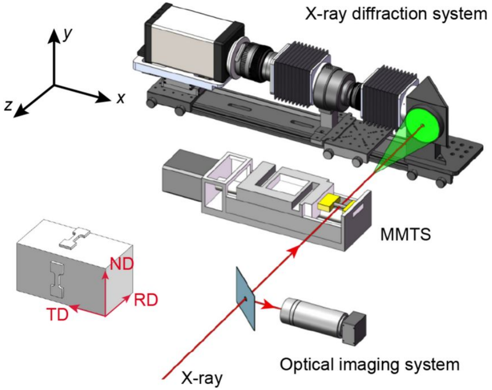
ECE 208 Project Presentation

Hannan (Danny) Wang
JD Qian

Project Description

Aim to create simple and effective ML model to predict material properties.



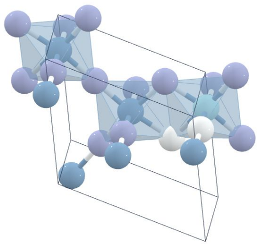


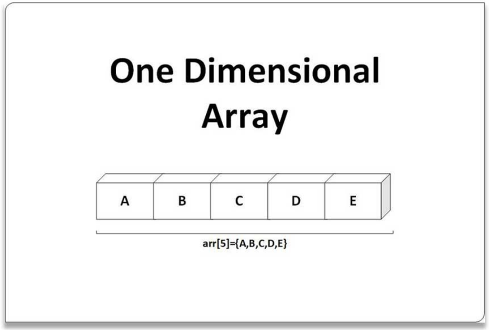
Using current published material properties dataset to train our program and compare the calculated value with experimental data

ML program workflow

Step one: Descriptor

Materials





Step two: Data set

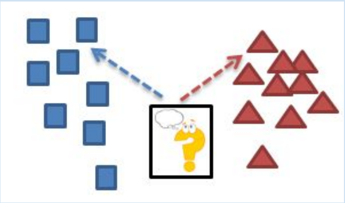
Descriptor

Material 1		
Material 2		
Material 3		
…		
Material N		

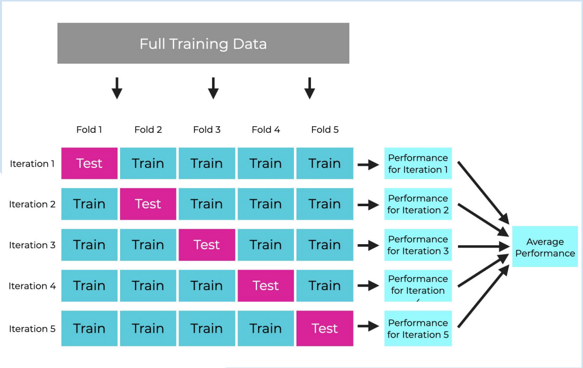
Train to map the descriptor to target properties

Step three: ML

ML to predict properties of materials



Supervised learning



Creating dataset

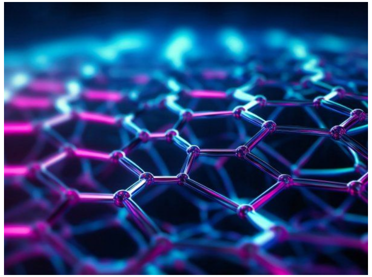
MATERIALS PROJECT

A open source materials properties dataset

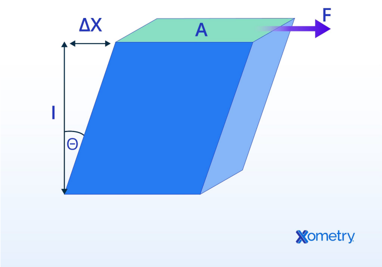
pymatgen

- open-source Python library for materials analysis
- Grabbing properties from the CIF files

Energy Band Gap

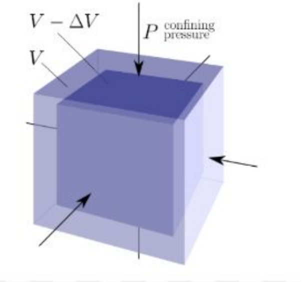


Shear Modulus



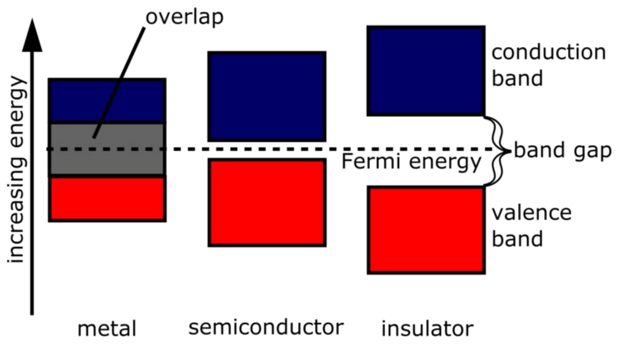
Bulk modulus

$$K = \frac{P}{\Delta V / V}$$



Band Gap

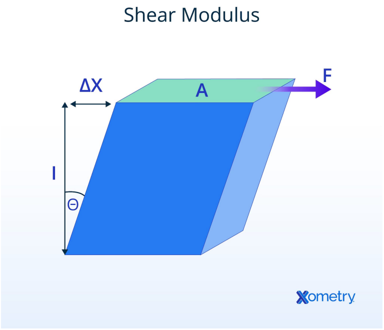
- Determine electrical conductivity of a material
- Defined as distance between valence band and conduction band
- Larger the band gap, lower the conductivity.



Band gap illustration with comparison among metal, semiconductor and insulator
[Band gap - Energy Education]

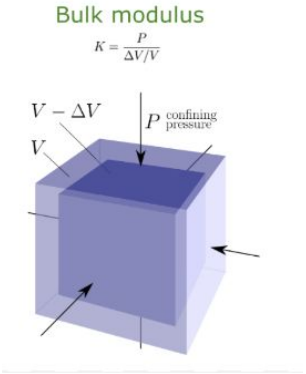
Shear modulus

- Shear modulus measures a material's resistance to shape changes when subjected to shear stress.
- A higher shear modulus means the material is stiffer against shearing deformation.
- Rubber band: Low shear modulus — easy to twist or stretch.



Bulk modulus

- Bulk modulus measures a material's resistance to uniform compression.
- Foam: Low bulk modulus — compresses easily.
- Diamond: Extremely high bulk modulus — nearly incompressible.

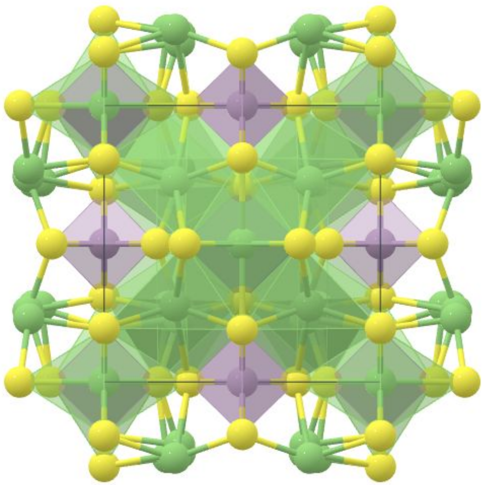


CIF file

```
# generated using pymatgen
data_Li10Ge(PS6)2
_symmetry_space_group_name_H-M   'P 1'
_cell_length_a   8.78764600
_cell_length_b   8.78764600
_cell_length_c   12.65754600
_cell_angle_alpha   90.00000000
_cell_angle_beta   90.00000000
_cell_angle_gamma   90.00000000
_symmetry_Int_Tables_number   1
_chemical_formula_structural   Li10Ge(PS6)2
_chemical_formula_sum   'Li20 Ge2 P4 S24'
_cell_volume   977.45015876
_cell_formula_units_Z   2
loop_
 _symmetry_equiv_pos_site_id
 _symmetry_equiv_pos_as_xyz
  1  'x, y, z'
loop_
 _atom_type_symbol
 _atom_type_oxidation_number
  Li+  1.0
  Ge4+  4.0
  P4+  4.0
  S2-  -2.0
  S-  -1.0
loop_
 _atom_site_type_symbol
 _atom_site_label
 _atom_site_symmetry_multiplicity
 _atom_site_fract_x
 _atom_site_fract_y
 _atom_site_fract_z
 _atom_site_occupancy
  Li+  Li0  1  0.22869800  0.27295000  0.29456300  1
  Li+  Li1  1  0.77130200  0.72705000  0.29456300  1
  Li+  Li2  1  0.27295000  0.77130200  0.79456300  1
  Li+  Li3  1  0.72705000  0.22869800  0.79456300  1
  Li+  Li4  1  0.22869800  0.72705000  0.29456300  1
  Li+  Li5  1  0.77130200  0.27295000  0.29456300  1
  Li+  Li6  1  0.27295000  0.22869800  0.79456300  1
  Li+  Li7  1  0.72705000  0.77130200  0.79456300  1
  Li+  Li8  1  0.00000000  0.00000000  0.93973000  1
  Li+  Li9  1  0.00000000  0.00000000  0.43973000  1
  Li+  Li10  1  0.50000000  0.50000000  0.54802000  1
  Li+  Li11  1  0.50000000  0.50000000  0.04802000  1
  Li+  Li12  1  0.25631800  0.72477200  0.03666300  1
  Li+  Li13  1  0.74368200  0.27522800  0.03666300  1
  Li+  Li14  1  0.27522800  0.25631800  0.53666300  1
```

```python
parser = CifParser(StringIO(cifFile))
```

```
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
---  ----  --------  --------  --------
  0  Li    0.228698  0.27295   0.294563
  1  Li    0.771302  0.72705   0.294563
  2  Li    0.27295   0.771302  0.794563
  3  Li    0.72705   0.228698  0.794563
  4  Li    0.228698  0.72705   0.294563
  5  Li    0.771302  0.27295   0.294563
  6  Li    0.27295   0.228698  0.794563
  7  Li    0.72705   0.771302  0.794563
  8  Li    0         0         0.93973
  9  Li    0         0         0.43973
 10  Li    0.5       0.5       0.54802
 11  Li    0.5       0.5       0.04802
 12  Li    0.256318  0.724772  0.036663
 13  Li    0.743682  0.275228  0.036663
 14  Li    0.275228  0.256318  0.536663
 15  Li    0.724772  0.743682  0.536663
 16  Li    0.275228  0.743682  0.536663
...
 46  S     0.5       0.707378  0.698166
 47  S     0.5       0.292622  0.698166
 48  S     0.707378  0.5       0.198166
 49  S     0.292622  0.5       0.198166
```



Using pymatgen to query material base one specific properties

mp-api

Star 130

Navigation

Project Modules

- mp_api.client.mprester
- mp_api.client.routes

Quick search

Go

mp_api.client.routes.materials

Modules

mp_api.client.routes.materials.absorption	
mp_api.client.routes.materials.alloys	
mp_api.client.routes.materials.bonds	
mp_api.client.routes.materials.charge_density	
mp_api.client.routes.materials.chemenv	
mp_api.client.routes.materials.dielectric	
mp_api.client.routes.materials.doi	
mp_api.client.routes.materials.elasticity	
mp_api.client.routes.materials.electrodes	
mp_api.client.routes.materials.electronic_structure	
mp_api.client.routes.materials.eos	
mp_api.client.routes.materials.fermi	
mp_api.client.routes.materials.grain_boundary	

```
results =
m.materials.summary.search(ban
d_gap=########)
```

Retrieving SummaryDoc documents: 100% 933/933 [00:00<00:00, 91436.18it/s]

```
[MPDataDoc<SummaryDoc>(
 material_id=MPID(mp-977360),
 fields_not_requested=['builder_meta', 'nsites', 'elements', 'nelements', 'composition', 'composition_reduced', 'formula_pretty', 'formula_anonymous',
 ),
```

Using pervious method to load the dataset

```
# get data
results = m.materials.summary.search(formula = "ABC3", fields=["band_gap","structure"])
```

Retrieving SummaryDoc documents: 100% 4700/4700 [00:05<00:00, 724.73it/s]

```
results = m.materials.elasticity.search(all_fields=False, fields=["shear_modulus","structure"])
```

Retrieving ElasticityDoc documents: 100% 13283/13283 [02:20<00:00, 733.19it/s]

```
results = m.materials.elasticity.search(all_fields=False, fields=["bulk_modulus","structure"])
```

Retrieving ElasticityDoc documents: 100% 13283/13283 [02:20<00:00, 733.19it/s]

Apply descriptors feature code from publish paper

Naturally-meaningful and efficient descriptors: machine learning of material properties based on robust one-shot ab initio descriptors

Sherif Abdulkader Tawfik & Salvy P. Russo

Journal of Cheminformatics **14**, Article number: 78 (2022) | Cite this article

This is a publish program that could extract materials properties such as bang gap, bulk modulus etc. from the loaded dataset

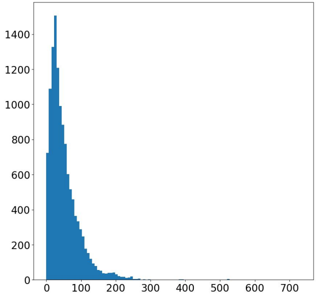
S. A. Tawfik and S. P. Russo, "Naturally-meaningful and efficient descriptors: machine learning of material properties based on robust one-shot ab initio descriptors," *Journal of Cheminformatics*, vol. 14, no. 1, Nov. 2022, doi: https://doi.org/10.1186/s13321-022-00658-9.

```
descriptors_list = atomic_numbers +\
    [Density] +\
    [alpha_parameters] +\
    [beta_parameters] +\
    [gamma_parameters] +\
    [metals_fraction] +\
    distance_matrix +\
    van_der_waals_radius +\
    electrical_resistivity +\
    velocity_of_sound +\
    reflectivity +\
    poissons_ratio +\
    molar_volume +\
    thermal_conductivity +\
```

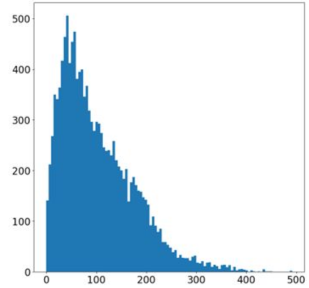
```
    melting_point +\
    critical_temperature +\
    superconduction_temperature +\
    liquid_range +\
    bulk_modulus +\
    youngs_modulus +\
    brinell_hardness +\
    rigidity_modulus +\
    vickers_hardness +\
    density_of_solid +\
    coefficient_of_linear_thermal_expansion +\
    average_ionic_radius +\
    average_cationic_radius +\
    average_anionic_radius +\
    spacegroup_numbers_list
return descriptors_list
```

	voigt	reuss	vrh
0	68.415	67.285	67.850
1	58.340	39.828	49.084
2	402.314	400.654	401.484
3	137.934	137.731	137.832
4	162.128	136.002	149.065
...	...	...	...
13076	102.137	54.015	78.076
13077	27.224	27.133	27.179
13078	10.557	8.457	9.507
13079	76.040	73.800	74.920
13080	40.735	40.659	40.697

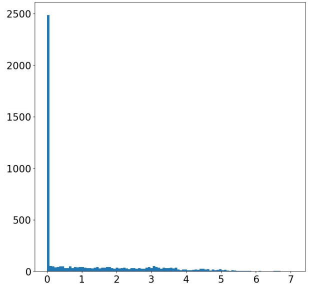
Shear modulus



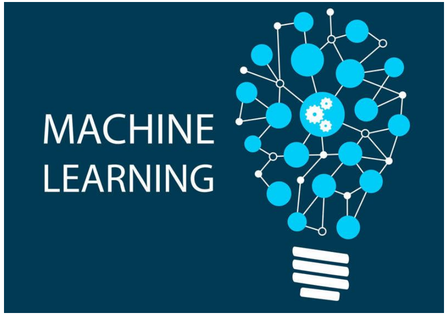
Bulk modulus



Band Gap



Feature dataset from descriptor



Linear Regression

Random Forest

Gradient Boosting Machines

Neural Network

Training and Testing Data slip

```python
X_train, X_test, y_train, y_test = train_test_split(dataset_df, band_gaps, test_size=.2, random_state=None)
```

80% training data / 20% testing data

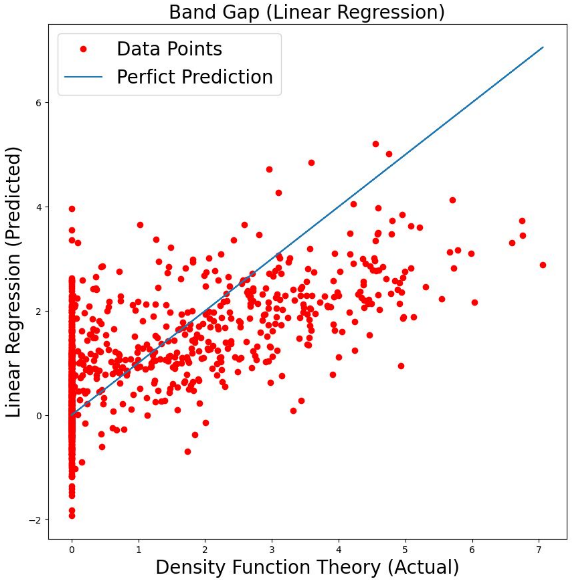
```python
descriptors_list = atomic_numbers +\
    [Density] +\
    [alpha_parameters] +\
    [beta_parameters] +\
    [gamma_parameters] +\
    [metals_fraction] +\
    distance_matrix +\
    van_der_waals_radius +\
    electrical_resistivity +\
    velocity_of_sound +\
    reflectivity +\
    poissons_ratio +\
    molar_volume +\
    thermal_conductivity +\
```

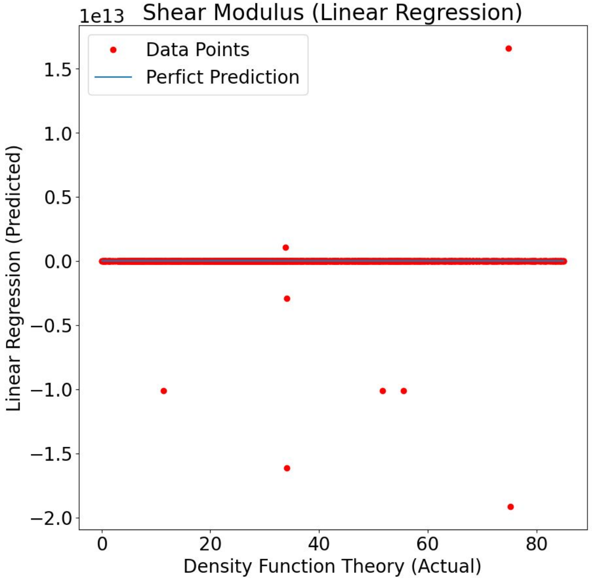
```python
    melting_point +\
    critical_temperature +\
    superconduction_temperature +\
    liquid_range +\
    bulk_modulus +\
    youngs_modulus +\
    brinell_hardness +\
    rigidity_modulus +\
    vickers_hardness +\
    density_of_solid +\
    coefficient_of_linear_thermal_expansion +\
    average_ionic_radius +\
    average_cationic_radius +\
    average_anionic_radius +\
    spacegroup_numbers_list
return descriptors_list
```

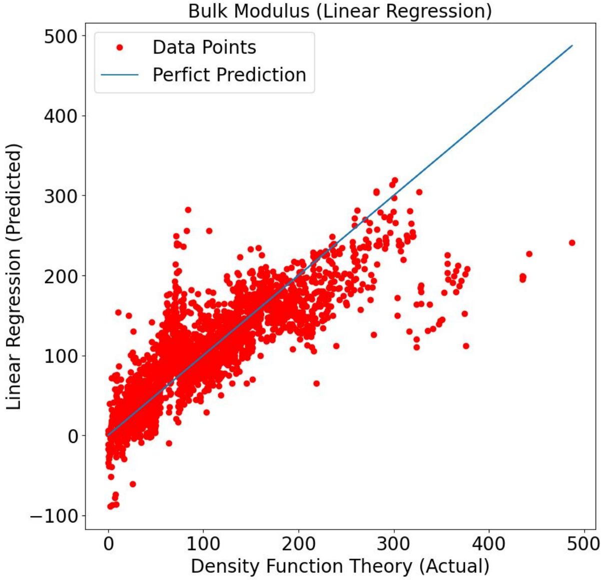
Materials properties that aim to predict

# data	Band Gap	Shear Modulus	Bulk Modulus
Training	3760	10368	10332
Testing	940	2592	2584

Linear Regression



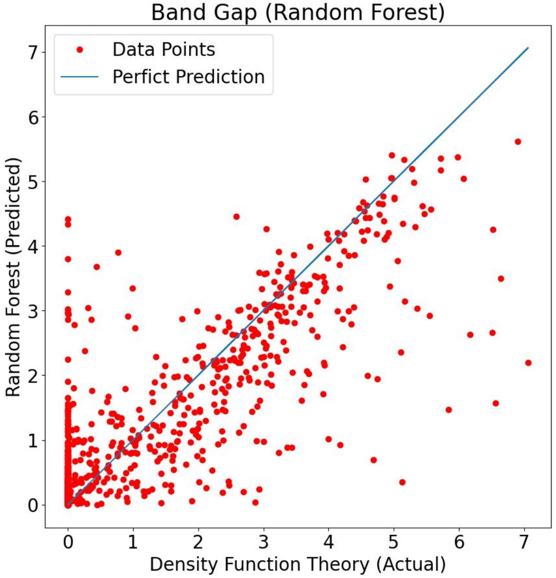




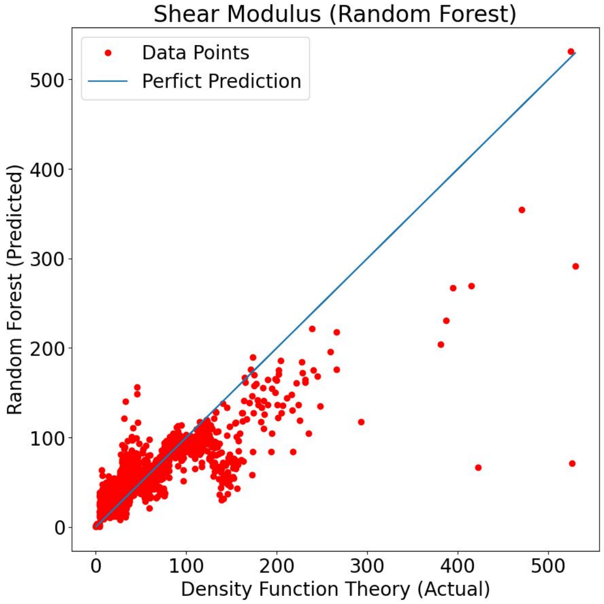
R square: 42%

R square: 67%

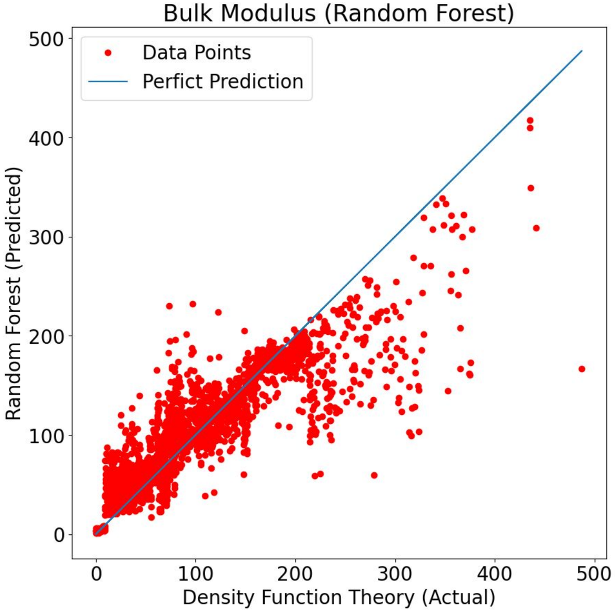
Random Forest



R square: 69%

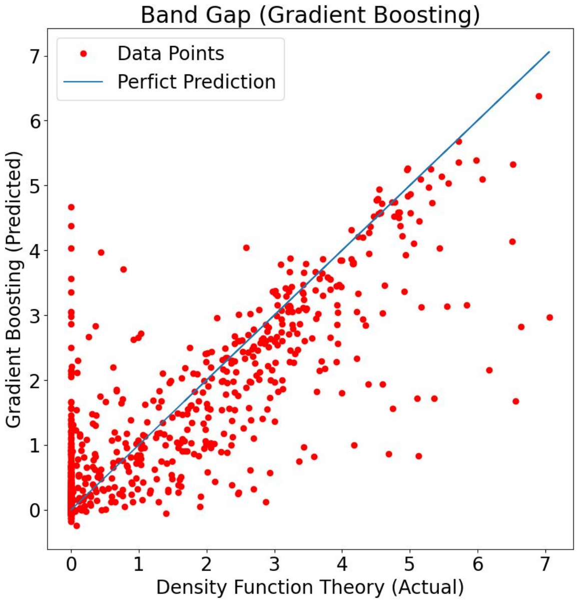


R square: 72%

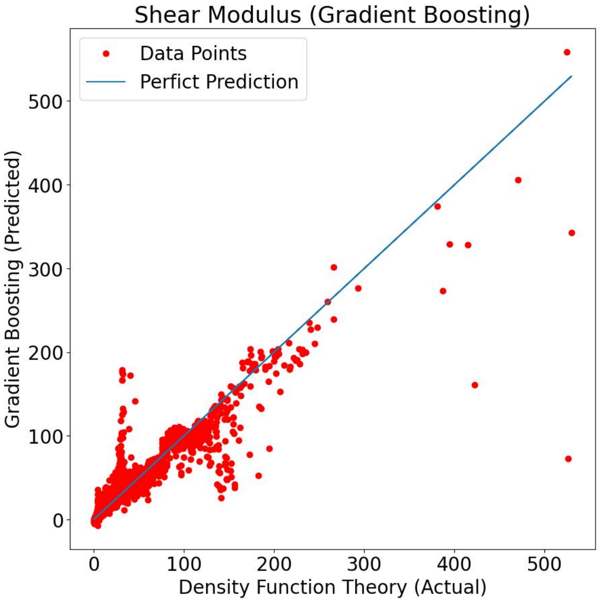


R square: 75%

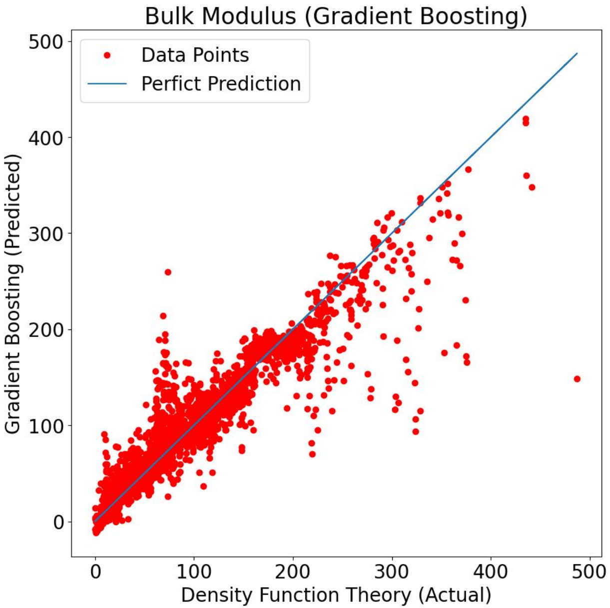
Gradient Boosting Machines



R square: 71%

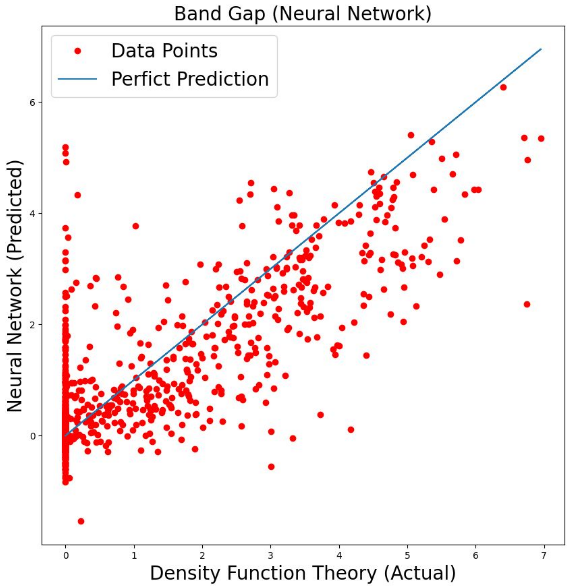


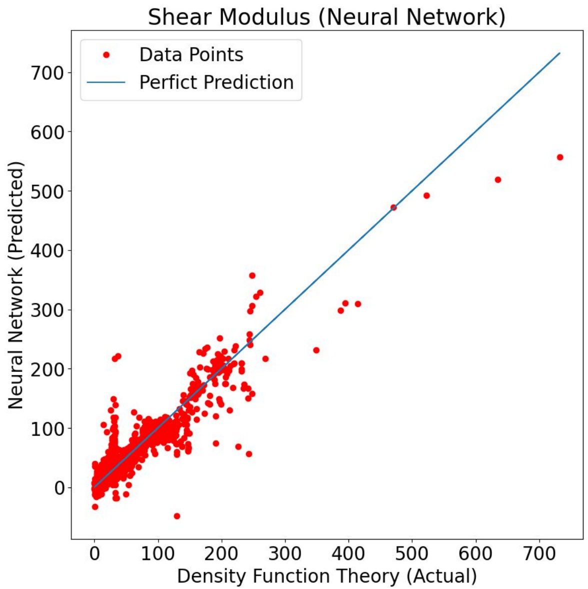
R square: 81%

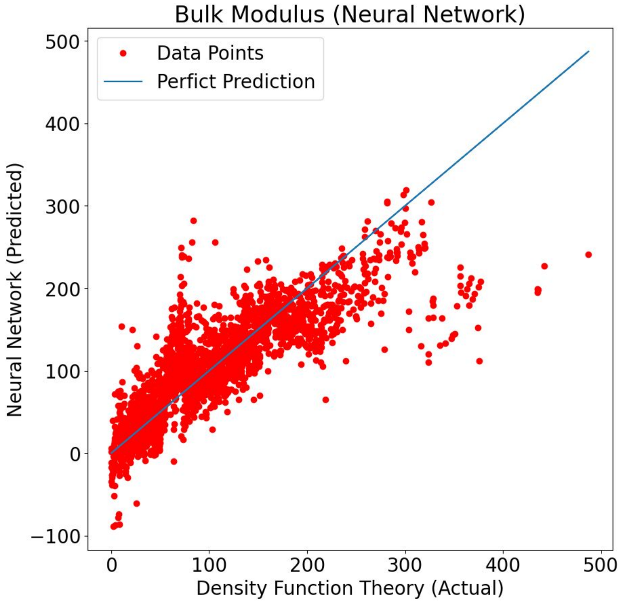


R square: 85%

Neural Network







R square: 62%

R square: 87%

R square: 83%

Conclusion

- Created a effective way to extract dataset from materials projects base on the target material property
- The ML model successfully predict the material properties (band gap, shear modulus, bulk modulus) with accuracy between 42 - 87%
- Band gap has a lower accuracy than others due the large amount of metal in the dataset
- Gradient Boosting Machines and neural networking have better performance.
    - GBMs build trees one at a time. This makes the model progressively better at capturing difficult patterns in the data. / interactions between variables
    - NNs could handle high dimension and complex data.

Future Task

- Improve the training accuracy
- Improve the training efficiency on CPU