Code used for bandgap prediction

```
from google.colab import drive
drive.mount('/content/gdrive',force_remount=True)
```

- With the help of mount attaching data file in to the google colab

```
import pandas as pd
df =pd.read_csv('/content/gdrive/My Drive/bandgap_prediction_Sheet1.csv')
df
```

- Panda is used to read the data file from the gdrive

```
! pip install lazypredict
```

- Installing lazy predict it helps to build several models at a time without using much code also which helps to understand which model works better.

```
dfx=df['Sites','Energy above hull(ev/atom)','Formation energy','Volume A^3','Number of atoms','Density(g-cm^3)','a (angstrom)','b (angstrom)','c (angstrom)','bond length (M-X)']
dfx
```

- Importing feature parameters as x-axis

```
dfy=df['Bandgap(eV)']
dfy
```

- importing target parameter as y-axis

```
import pandas as pd
import seaborn as seaborn
```

```
from sklearn.model_selection import train_test_split
import lazypredict
from lazypredict.Supervised import LazyRegressor
```

- Determining suitable model from R^2 also from root mean square error(RMSE)

```
# perform data splitting using 70/30 ratio
x_train, x_test, y_train, y_test =  train_test_split(dfx, dfy, test_size=0.3
```

- Performing training data and testing data by splitting in the ratio of 70/30

```
clf=LazyRegressor()
model_train,predictions_train=clf.fit(x_train,x_train,y_train,y_train)
model_test,predictions_test=clf.fit(x_train,x_test,y_train,y_test)
predictions_train
```

- Training data for prediction

```
import numpy as np
npx=np.array(x_test).reshape(-1,10)
x_test
```

- Testing the data

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
models=RandomForestRegressor(n_estimators=100)
models.fit(x_train,y_train)
ypred=models.predict(npx)
```

```
sns.set(color_codes=True)
sns.set_style("white")
ax=sns.regplot(y=y_test, x=ypred,scatter_kws={'alpha':0.4})
ax.set_xlabel('Predicted Bandgap(eV)',fontsize='large',fontweight='bold')
ax.set_ylabel('Experimental Bandgap(eV)',fontsize='large',fontweight='bold')
ax.set_xlim(-1, 12)
ax.set_ylim(-1, 12)
ax.figure.set_size_inches(5, 5)
plt.show
```

- Fitting the graph between experimental and predicted bandgap

```
models.score(npx,ypred)
```

- For final model prediction