MODEL REPORT

Dataset Description:

We are provided with test and train dataset of various minerals, we have to predict the refractive index (RI) of test materials based on their given properties. Identify the top features that have the greatest impact on RI, and develop an optimal machine learning model for accurately predicting RI on test material data. Determine suitable performance metrics for measuring the model's performance.

Features given -- Name Crystal Structure Diaphaneity Specific Gravity Dispersion Optical Mohs Hardness Refractive Index Molecular data

Note: results may vary given the stochastic nature of the algorithms. Consider running the example a few times and compare the average outcome.

Approach and transformation that was done:

- Data cleaning, check for null values etc., checking the columns or features which doesn't provide any information means they do not have two or more distinct values, mostly if will be seen in the columns of different elements(their presence), further dropping columns such Name of mineral and more as shown in code file.
- Further we check for multicollinearity, threshold was not decided as per satisfaction (0.85), and then the features which showed collinearity above 0.85 were removed.
- After above cleaning the number of features got reduced from 139 to 63
- After that I decided to perform some transformation on numerical features (not target variable yet), I plotted histogram of all numerical features and performed log1p transformation, those features which were skewed before got normal and rest showed little to no change in distribution. COLUMNS LOG TRANSFORMED – Molar mass and calculated density.
- After that one hot encoded categorical variables (Crystal Structure,Diaphaneity,Optical)
- After that I standardized the dataset by minmax scaler
- Now I transformed the target variable, as the variable distribution was not easy to transform to normal distribution, I performed BOXCOX Transformation and got lambda value as **-1.7679645341555315**
- After that I checked out for any outliers and found none.

Model Selection and Hyperparameter tuning:

Primary metric – RMSE Secondary metric - R2 Score

I firstly used Randomforest regressor as my algorithm as they don't get affected by outliers much and they also don't assume underlying distribution of the dataset.

RESULTS – rmse- **0.0784** r2 score—**0.751**, got decent rmse to start with.

Then I used XGBoost as my algorithm and got predictions for my test set.

RESULTS – rmse- **0.0996** r2 score—**0.599**, results were not improved from previous algorithm.

Then I used pycaret automated library to get the best algorithm, the results are follows: (**Note**: The results vary from actual results as pycaret automatically perform more than 60 transformation and uses default parameters which vary highly but still it gives the idea for further work)

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
br	Bayesian Ridge	0.2438	1.876000e-01	0.4079	6.044000e-01	0.2199	0.1089	0.0250
lightgbm	Light Gradient Boosting Machine	0.2479	1.896000e-01	0.4082	5.867000e-01	0.2164	0.1191	0.0715
huber	Huber Regressor	0.2098	1.901000e-01	0.3971	5.820000e-01	0.2175	0.0858	0.0970
ridge	Ridge Regression	0.2510	1.926000e-01	0.4144	5.924000e-01	0.2238	0.1128	0.0190
omp	Orthogonal Matching Pursuit	0.2272	1.934000e-01	0.4078	5.890000e-01	0.2119	0.1047	0.0175
rf	Random Forest Regressor	0.2201	2.161000e-01	0.4265	5.144000e-01	0.2273	0.0941	0.3720
gbr	Gradient Boosting Regressor	0.2411	2.167000e-01	0.4353	5.190000e-01	0.2344	0.1047	0.1100

So I used two more algorithms Bayesian Ridge and LightGBM.

Firstly I used lgbm and got the following results:

RESULTS – rmse- **0.0807** r2 score—**0.7367** significant increase from Random forest regressor
then I performed hyperparameter tuning using **verstack** library got the tuned parameter and then I made prediction on test data and got following results:
RESULTS – rmse- **0.0767** r2 score—**0.76205** , got better results

Then I performed Bayesian ridge regression and got the following results

RESULTS – rmse- **0.0742** r2 score—**0.777,** better results than LightGBM, then hyperparamter tuning didn't yield any better results.

FINAL MODEL Then finally performed ensembling technique and combined the prediction of lightgbm and bayesain ridge, the weights were 0.5 and 0.5 respectively.

RESULTS – rmse- **0.0741** r2 score—**0.778,** a bit better than Bayesian ridge alone and this this our final prediction for test set.

Feature Importance

I performed feature importance using mutual info. Regression Mutual information (MI) [1] between two random variables is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependence.
(**Note:** Feature importance given by verstack library is a bit different due to its different approach to

calculate feature importance and apart from that correlation between features can also cause this type of situation).

So the importance of feature is as following: (top 6)

Specific Gravity	0.730207
Optical_0	0.323516
Mohs Hardness	0.171194
Optical_4	0.170562
Diaphaneity_0	0.157820
Calculated Density	0.085484

Out of them specific gravity both by MI and verstack is given highest importance then comes Optical_0 which implies if the material is anisotropic refractive index will be higher and same reason for Optical_4, then comes Mohs_hardness which implied the hardness of mineral according to mohs scale then comes Diaphaneity_0 whi ch means if mineral is opaque, is mineral is opaque it affects the refractive index of that material going by logic it implies the refractive index of material will very low or close to 0.
other basically have their own reason and also have correlation with above given features.