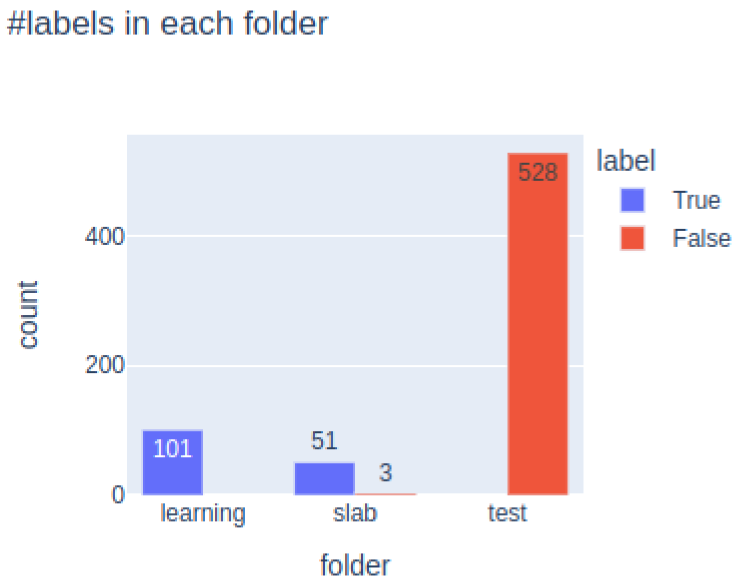
```
|___docs
|______data_understanding.pdf
|___images
|___notebooks
|______data_understanding.ipynb
|______ML.ipynb
|______requirements.txt
|___output
|______classification
|______regression
|___README.md
```

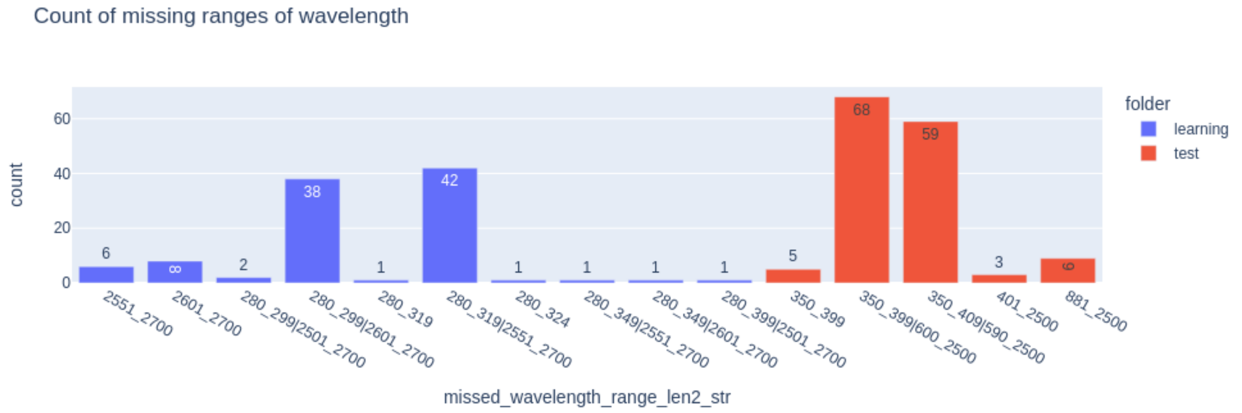
Data Overview

- There are 3 folders containing data
  - learning
  - slab
  - test
- Each sample is a json file which is composed of
  - spectrum
    - wavelength
    - reflectance
    - error
  - grain_size
  - abundances
  - Description
  - Version
- In total there are 683 samples in which 152 samples with labels (`abundances`) and 531 without labels.

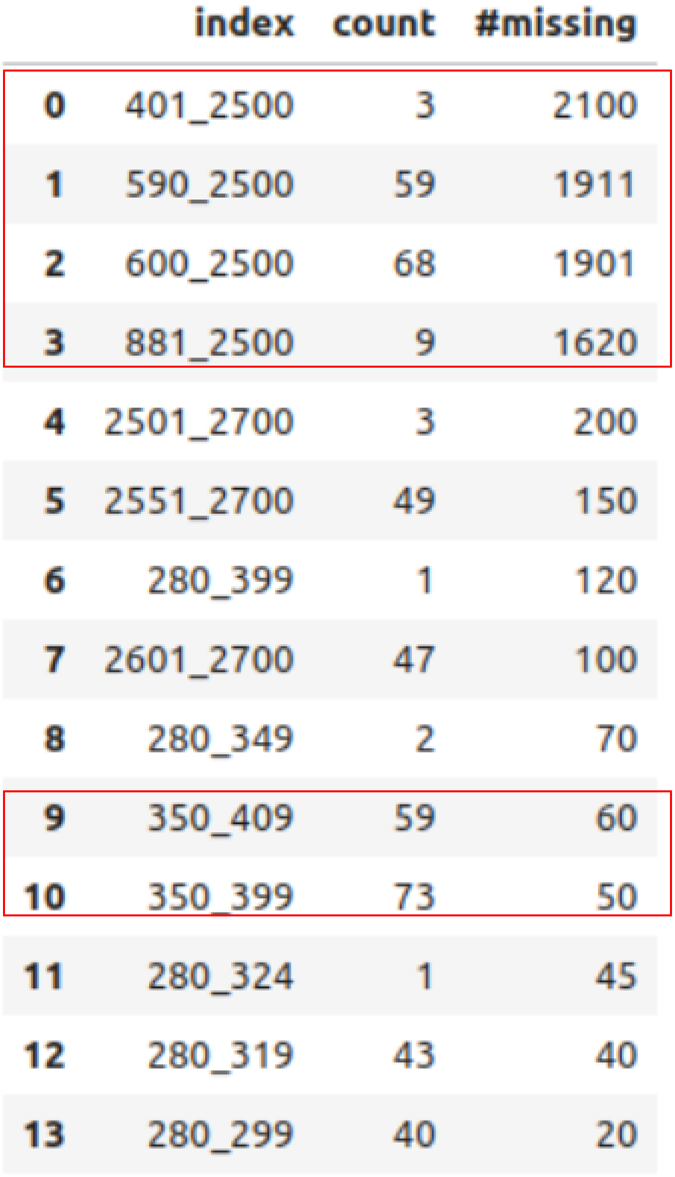


- There are 7 different classes
    - Basalt
    - Clinopyroxene
    - Glass
    - Olivine
    - Orthopyroxene
    - Plagioclase
    - graphite

- The number of samples which doesn't have consecutive wavelengths (and its `reflectance`) is depicted as the following chart.



- In numbers the table is re-written as below



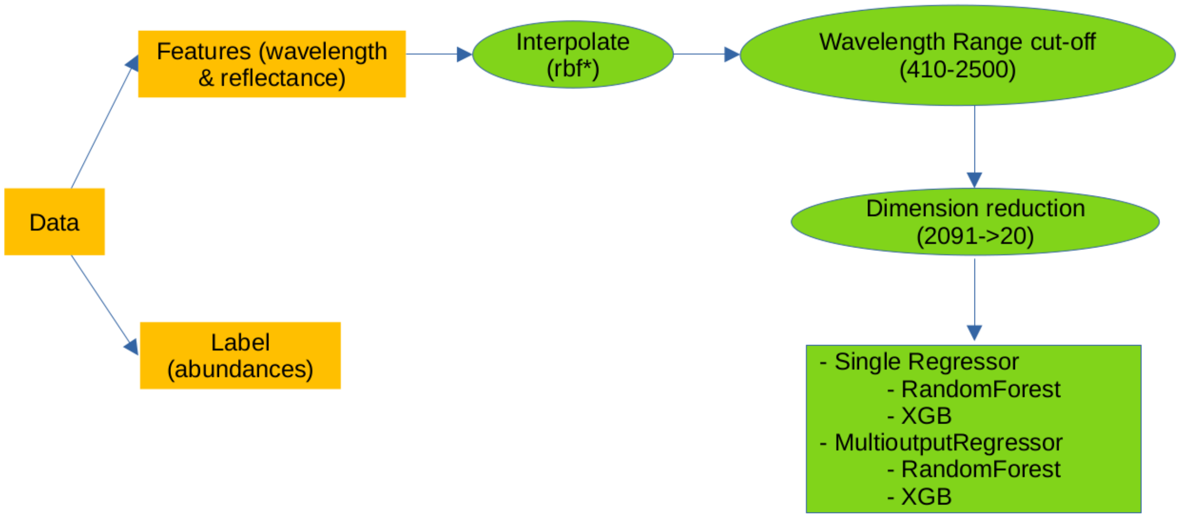
	index	count	#missing
0	401_2500	3	2100
1	590_2500	59	1911
2	600_2500	68	1901
3	881_2500	9	1620
4	2501_2700	3	200
5	2551_2700	49	150
6	280_399	1	120
7	2601_2700	47	100
8	280_349	2	70
9	350_409	59	60
10	350_399	73	50
11	280_324	1	45
12	280_319	43	40
13	280_299	40	20

- Since the missing ranges before 410 and after 2500 are dominated, in our machine learning task, the wavelengths in these ranges will be removed.

Machine Learning

Pipeline

- This is a multi-output classification or multi-output regression problem where the input is the obtained reflectance and the output is a composition of different mineral phase names.
    - In regression task, the output is a list of real numbers ranging from 0 to 100 whose sum must be 100.
    - In classification task, the output is a list of binary values (0 or 1) which indicates that phase name exists (1) or not (0)



*https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RBFInterpolator.html#scipy.interpolate.RBFInterpolator

Data Preparation

- Features: `reflectance`
    - Not only wavelength ranges are missing, single values of wavelength are also not available for all samples. To fill these gaps for reflectance values, an interpolation method is applied. Here we're using rbf interpolator (https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RBFInterpolator.html#scipy.interpolate.RBFInterpolator)
- Labels: abundances
    - For Regression task: all values are kept the same
    - For Classification task: the non-zero values are treated as 1 and 0 otherwise
- Wavelength Cut-Off: due to the missing wavelengths mentioned above, the range of 410 to 2500 is used for input.
- Dimension Reduction: there are 2091 features (values) for each samples. To simplify this but still to make sure data is not lost so much, `Principal Component Analysis (PCA)` is used to reduce the number of features from 2091 to 20, for both training and test sets.

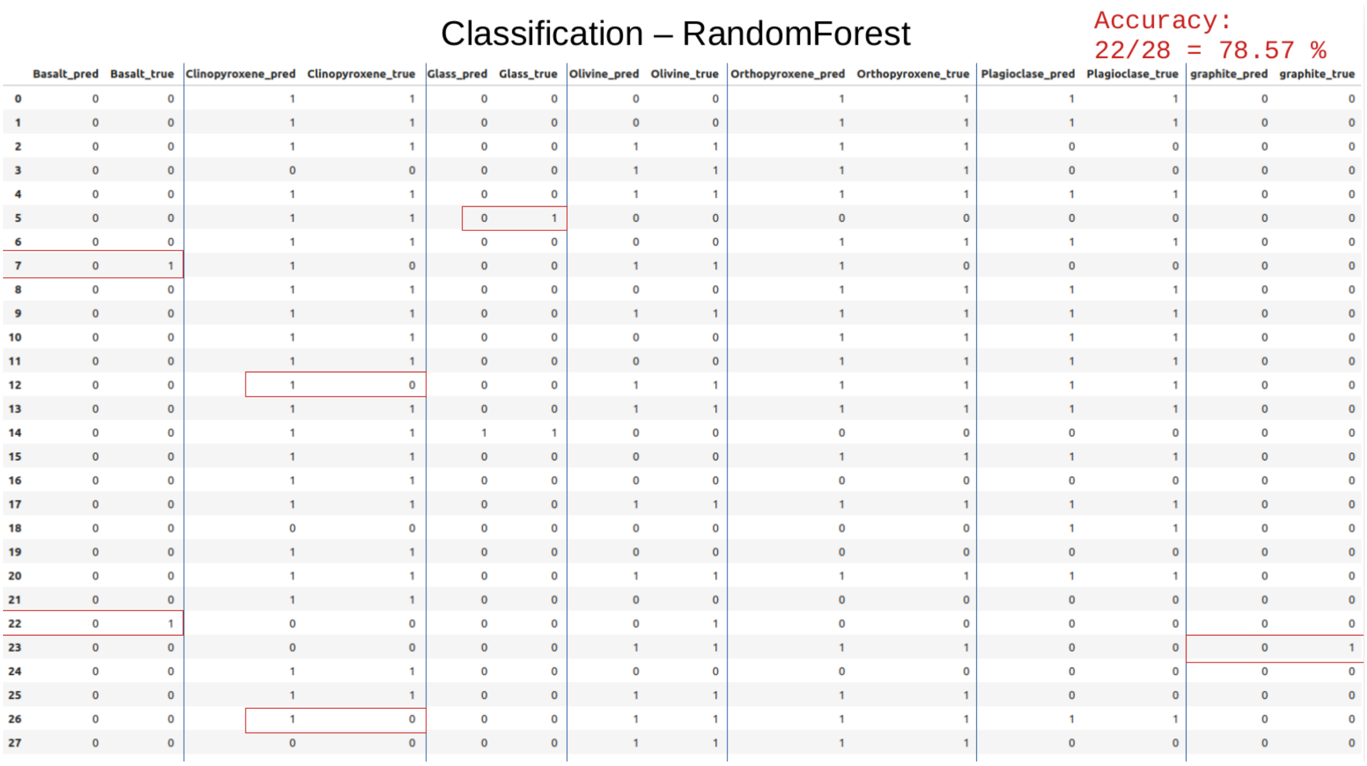
Algorithm Evaluation

- The training set is split into 2 sets: learning and evaluation. The learning set is input into the algorithm and the evaluation set is used to evaluation and optimize the algorithm.
- In our use case, 20% (28 samples) of the training data (139 samples - json files) is separated for evaluation purpose.

Results

Evaluation - Classification

Random Forest (single Classifier)



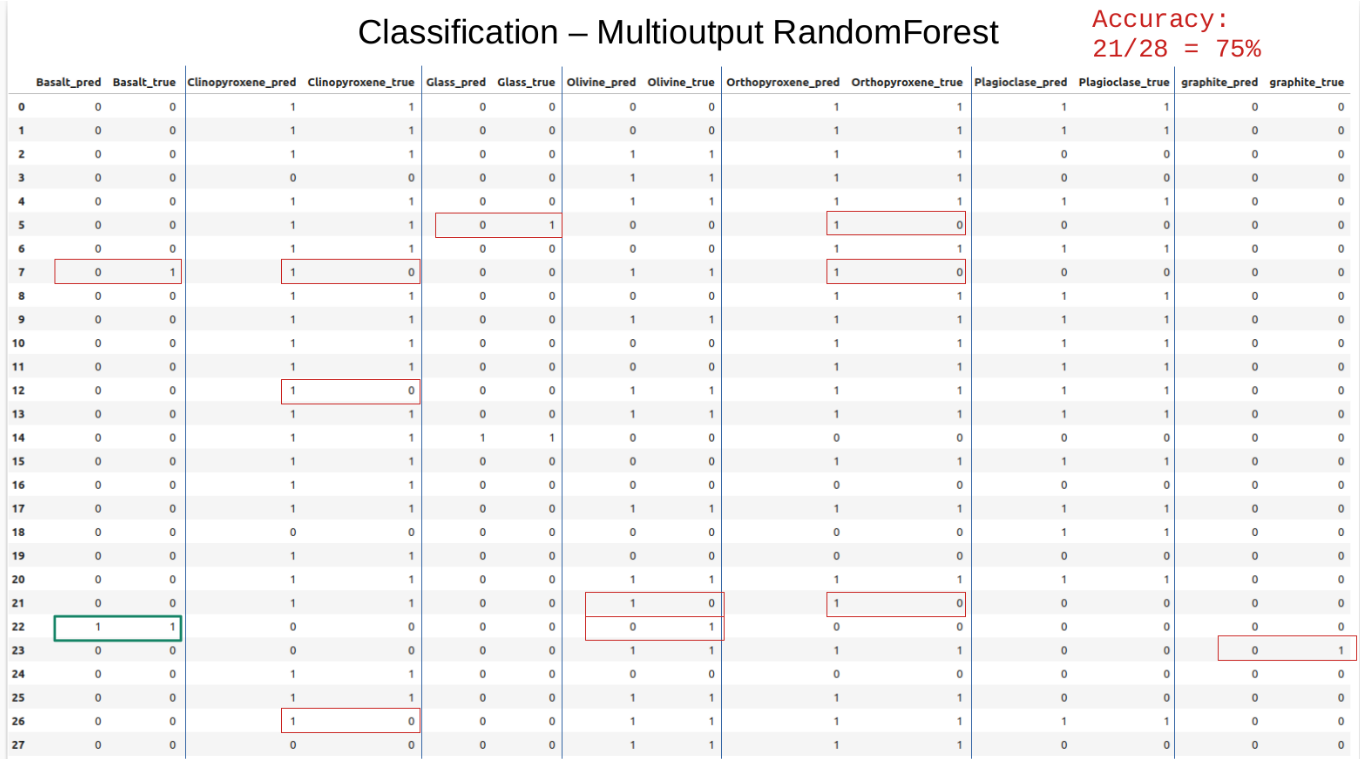
Classification – RandomForest

Accuracy:
22/28 = 78.57 %

	Basalt_pred	Basalt_true	Clinopyroxene_pred	Clinopyroxene_true	Glass_pred	Glass_true	Olivine_pred	Olivine_true	Orthopyroxene_pred	Orthopyroxene_true	Plagioclase_pred	Plagioclase_true	graphite_pred	graphite_true
0	0	0	1	1	0	0	0	0	1	1	1	1	0	0
1	0	0	1	1	0	0	0	0	1	1	1	1	0	0
2	0	0	1	1	0	0	1	1	1	1	0	0	0	0
3	0	0	0	0	0	0	1	1	1	1	0	0	0	0
4	0	0	1	1	0	0	1	1	1	1	1	1	0	0
5	0	0	1	1	0	1	0	0	0	0	0	0	0	0
6	0	0	1	1	0	0	0	0	1	1	1	1	0	0
7	0	1	1	0	0	0	1	1	1	0	0	0	0	0
8	0	0	1	1	0	0	0	0	1	1	1	1	0	0
9	0	0	1	1	0	0	1	1	1	1	1	1	0	0
10	0	0	1	1	0	0	0	0	1	1	1	1	0	0
11	0	0	1	1	0	0	0	0	1	1	1	1	0	0
12	0	0	1	0	0	0	1	1	1	1	1	1	0	0
13	0	0	1	1	0	0	1	1	1	1	1	1	0	0
14	0	0	1	1	1	1	0	0	0	0	0	0	0	0
15	0	0	1	1	0	0	0	0	1	1	1	1	0	0
16	0	0	1	1	0	0	0	0	0	0	0	0	0	0
17	0	0	1	1	0	0	1	1	1	1	1	1	0	0
18	0	0	0	0	0	0	0	0	0	0	1	1	0	0
19	0	0	1	1	0	0	0	0	0	0	0	0	0	0
20	0	0	1	1	0	0	1	1	1	1	1	1	0	0
21	0	0	1	1	0	0	0	0	0	0	0	0	0	0
22	0	1	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	1	1	1	1	0	0	0	1
24	0	0	1	1	0	0	0	0	0	0	0	0	0	0
25	0	0	1	1	0	0	1	1	1	1	0	0	0	0
26	0	0	1	0	0	0	1	1	1	1	1	1	0	0
27	0	0	0	0	0	0	1	1	1	1	0	0	0	0

- Only 6 samples are predicted wrong and only one class is wrong within 7 in total in each sample
- This classifier achieved almost 80% in accuracy

Multi-output Random Forest



Classification – Multioutput RandomForest

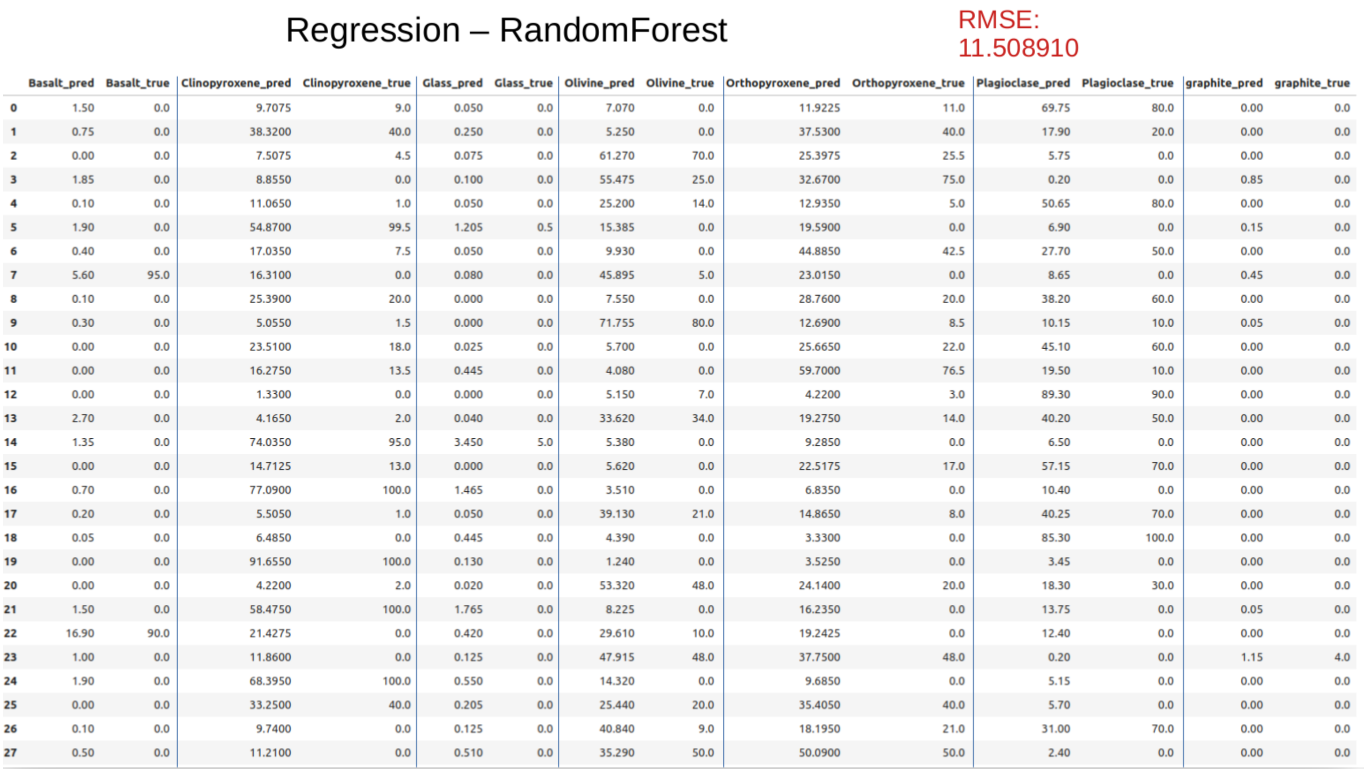
Accuracy:
21/28 = 75%

	Basalt_pred	Basalt_true	Clinopyroxene_pred	Clinopyroxene_true	Glass_pred	Glass_true	Olivine_pred	Olivine_true	Orthopyroxene_pred	Orthopyroxene_true	Plagioclase_pred	Plagioclase_true	graphite_pred	graphite_true
0	0	0	1	1	0	0	0	0	1	1	1	1	0	0
1	0	0	1	1	0	0	0	0	1	1	1	1	0	0
2	0	0	1	1	0	0	1	1	1	1	0	0	0	0
3	0	0	0	0	0	0	1	1	1	1	0	0	0	0
4	0	0	1	1	0	0	1	1	1	1	1	1	0	0
5	0	0	1	1	0	1	0	0	1	0	0	0	0	0
6	0	0	1	1	0	0	0	0	1	1	1	1	0	0
7	0	1	1	0	0	0	1	1	1	0	0	0	0	0
8	0	0	1	1	0	0	0	0	1	1	1	1	0	0
9	0	0	1	1	0	0	1	1	1	1	1	1	0	0
10	0	0	1	1	0	0	0	0	1	1	1	1	0	0
11	0	0	1	1	0	0	0	0	1	1	1	1	0	0
12	0	0	1	0	0	0	1	1	1	1	1	1	0	0
13	0	0	1	1	0	0	1	1	1	1	1	1	0	0
14	0	0	1	1	1	1	0	0	0	0	0	0	0	0
15	0	0	1	1	0	0	0	0	1	1	1	1	0	0
16	0	0	1	1	0	0	0	0	0	0	0	0	0	0
17	0	0	1	1	0	0	1	1	1	1	1	1	0	0
18	0	0	0	0	0	0	0	0	0	0	1	1	0	0
19	0	0	1	1	0	0	0	0	0	0	0	0	0	0
20	0	0	1	1	0	0	1	1	1	1	1	1	0	0
21	0	0	1	1	0	0	1	0	1	0	0	0	0	0
22	1	1	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	1	1	1	1	0	0	0	1
24	0	0	1	1	0	0	0	0	0	0	0	0	0	0
25	0	0	1	1	0	0	1	1	1	1	0	0	0	0
26	0	0	1	0	0	0	1	1	1	1	1	1	0	0
27	0	0	0	0	0	0	1	1	1	1	0	0	0	0

- Multioutput random forest classifier has more errors than the single one but it does well in prediction of the minor class `Basalt` (sample 22 in the table)
- With 7 wrong prediction, this multi-output classifier brought the accuracy of 75%

Evaluation - Regression

Radom Forest Regression



Regression – RandomForest

RMSE:
11.508910

	Basalt_pred	Basalt_true	Clinopyroxene_pred	Clinopyroxene_true	Glass_pred	Glass_true	Olivine_pred	Olivine_true	Orthopyroxene_pred	Orthopyroxene_true	Plagioclase_pred	Plagioclase_true	graphite_pred	graphite_true
0	1.50	0.0	9.7075	9.0	0.050	0.0	7.070	0.0	11.9225	11.0	69.75	80.0	0.00	0.0
1	0.75	0.0	38.3200	40.0	0.250	0.0	5.250	0.0	37.5300	40.0	17.90	20.0	0.00	0.0
2	0.00	0.0	7.5075	4.5	0.075	0.0	61.270	70.0	25.3975	25.5	5.75	0.0	0.00	0.0
3	1.85	0.0	8.8550	0.0	0.100	0.0	55.475	25.0	32.6700	75.0	0.20	0.0	0.85	0.0
4	0.10	0.0	11.0650	1.0	0.050	0.0	25.200	14.0	12.9350	5.0	50.65	80.0	0.00	0.0
5	1.90	0.0	54.8700	99.5	1.205	0.5	15.385	0.0	19.5900	0.0	6.90	0.0	0.15	0.0
6	0.40	0.0	17.0350	7.5	0.050	0.0	9.930	0.0	44.8850	42.5	27.70	50.0	0.00	0.0
7	5.60	95.0	16.3100	0.0	0.080	0.0	45.895	5.0	23.0150	0.0	8.65	0.0	0.45	0.0
8	0.10	0.0	25.3900	20.0	0.000	0.0	7.550	0.0	28.7600	20.0	38.20	60.0	0.00	0.0
9	0.30	0.0	5.0550	1.5	0.000	0.0	71.755	80.0	12.6900	8.5	10.15	10.0	0.05	0.0
10	0.00	0.0	23.5100	18.0	0.025	0.0	5.700	0.0	25.6650	22.0	45.10	60.0	0.00	0.0
11	0.00	0.0	16.2750	13.5	0.445	0.0	4.080	0.0	59.7000	76.5	19.50	10.0	0.00	0.0
12	0.00	0.0	1.3300	0.0	0.000	0.0	5.150	7.0	4.2200	3.0	89.30	90.0	0.00	0.0
13	2.70	0.0	4.1650	2.0	0.040	0.0	33.620	34.0	19.2750	14.0	40.20	50.0	0.00	0.0
14	1.35	0.0	74.0350	95.0	3.450	5.0	5.380	0.0	9.2850	0.0	6.50	0.0	0.00	0.0
15	0.00	0.0	14.7125	13.0	0.000	0.0	5.620	0.0	22.5175	17.0	57.15	70.0	0.00	0.0
16	0.70	0.0	77.0900	100.0	1.465	0.0	3.510	0.0	6.8350	0.0	10.40	0.0	0.00	0.0
17	0.20	0.0	5.5050	1.0	0.050	0.0	39.130	21.0	14.8650	8.0	40.25	70.0	0.00	0.0
18	0.05	0.0	6.4850	0.0	0.445	0.0	4.390	0.0	3.3300	0.0	85.30	100.0	0.00	0.0
19	0.00	0.0	91.6550	100.0	0.130	0.0	1.240	0.0	3.5250	0.0	3.45	0.0	0.00	0.0
20	0.00	0.0	4.2200	2.0	0.020	0.0	53.320	48.0	24.1400	20.0	18.30	30.0	0.00	0.0
21	1.50	0.0	58.4750	100.0	1.765	0.0	8.225	0.0	16.2350	0.0	13.75	0.0	0.05	0.0
22	16.90	90.0	21.4275	0.0	0.420	0.0	29.610	10.0	19.2425	0.0	12.40	0.0	0.00	0.0
23	1.00	0.0	11.8600	0.0	0.125	0.0	47.915	48.0	37.7500	48.0	0.20	0.0	1.15	4.0
24	1.90	0.0	68.3950	100.0	0.550	0.0	14.320	0.0	9.6850	0.0	5.15	0.0	0.00	0.0
25	0.00	0.0	33.2500	40.0	0.205	0.0	25.440	20.0	35.4050	40.0	5.70	0.0	0.00	0.0
26	0.10	0.0	9.7400	0.0	0.125	0.0	40.840	9.0	18.1950	21.0	31.00	70.0	0.00	0.0
27	0.50	0.0	11.2100	0.0	0.510	0.0	35.290	50.0	50.0900	50.0	2.40	0.0	0.00	0.0

Test set

- `result_cl_test_multirf.csv`: classification for test set using multi-output random forest
- `result_cl_test_rf.csv`: classification for test set using normal random forest
- `result_regr_test_rf.csv`: regression for test set using random forest
- `result_regr_test_rf.zip`: combine of result and original data

Dependency Installation

- To run the notebooks, install `requirements.txt` into the python environment using `pip`

```
pip install -r requirements.txt
```