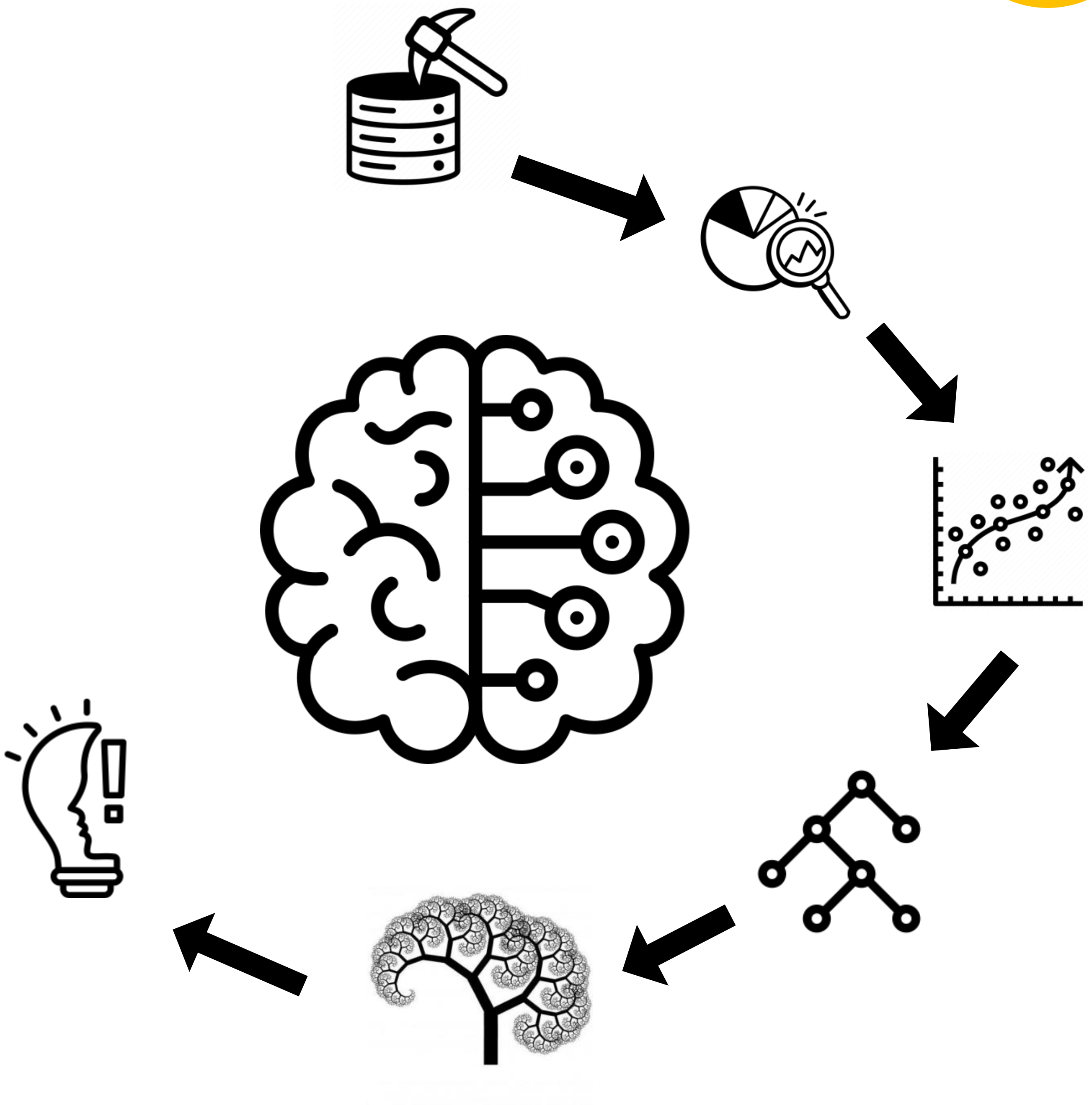
Machine Learning Project: Stream Sediment Samples

Khizer Zakir & Rodrigo Brust Santos

08/12/2023

Overview

1. Dataset Explanation
2. Regression Models
3. Random Forest
4. XGBoost
5. Final Considerations

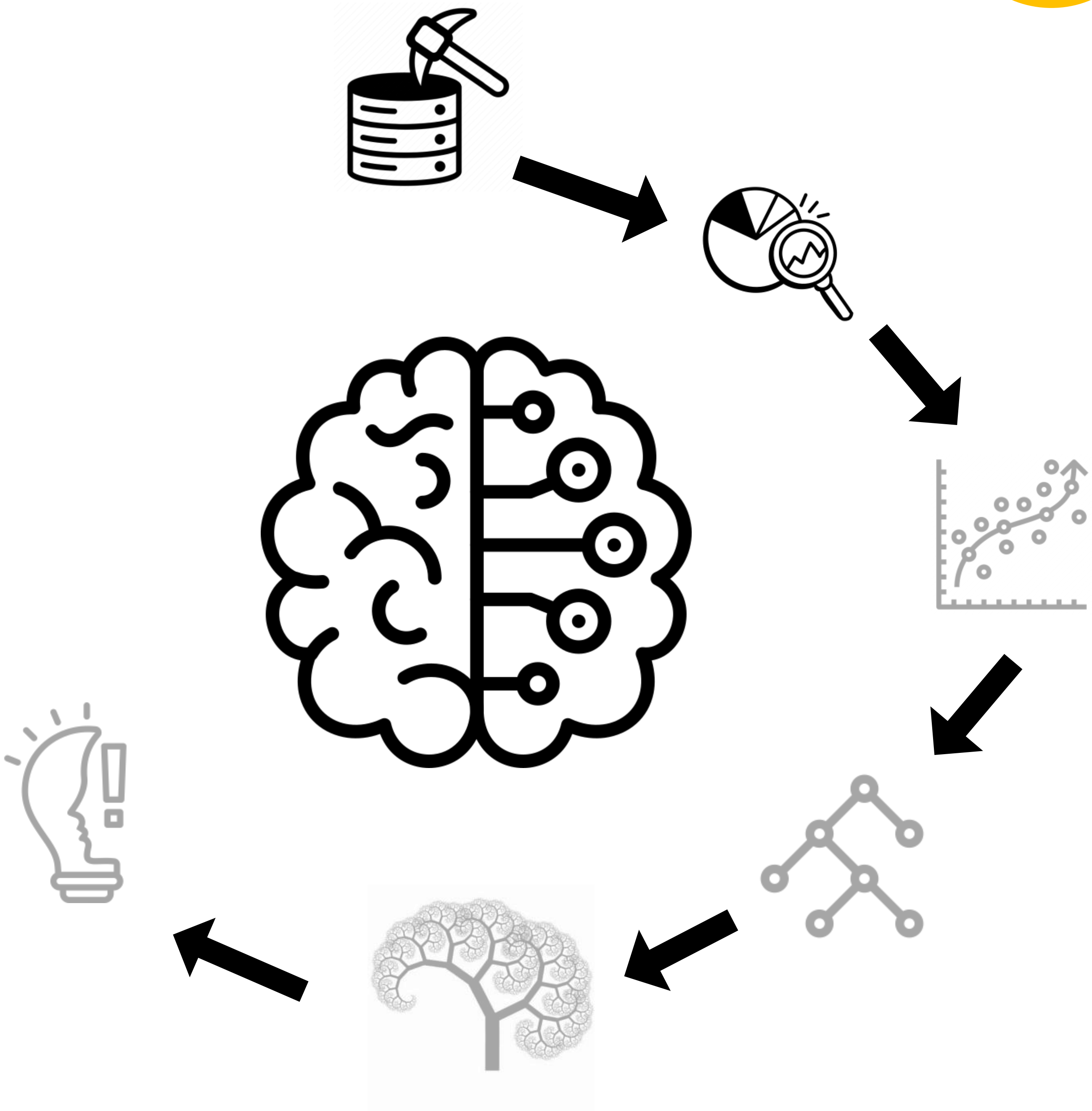


Objectives

To design a regression machine-learning pipeline to process and predict Zn (ppm) concentration from stream sediment samples dataset

Overview

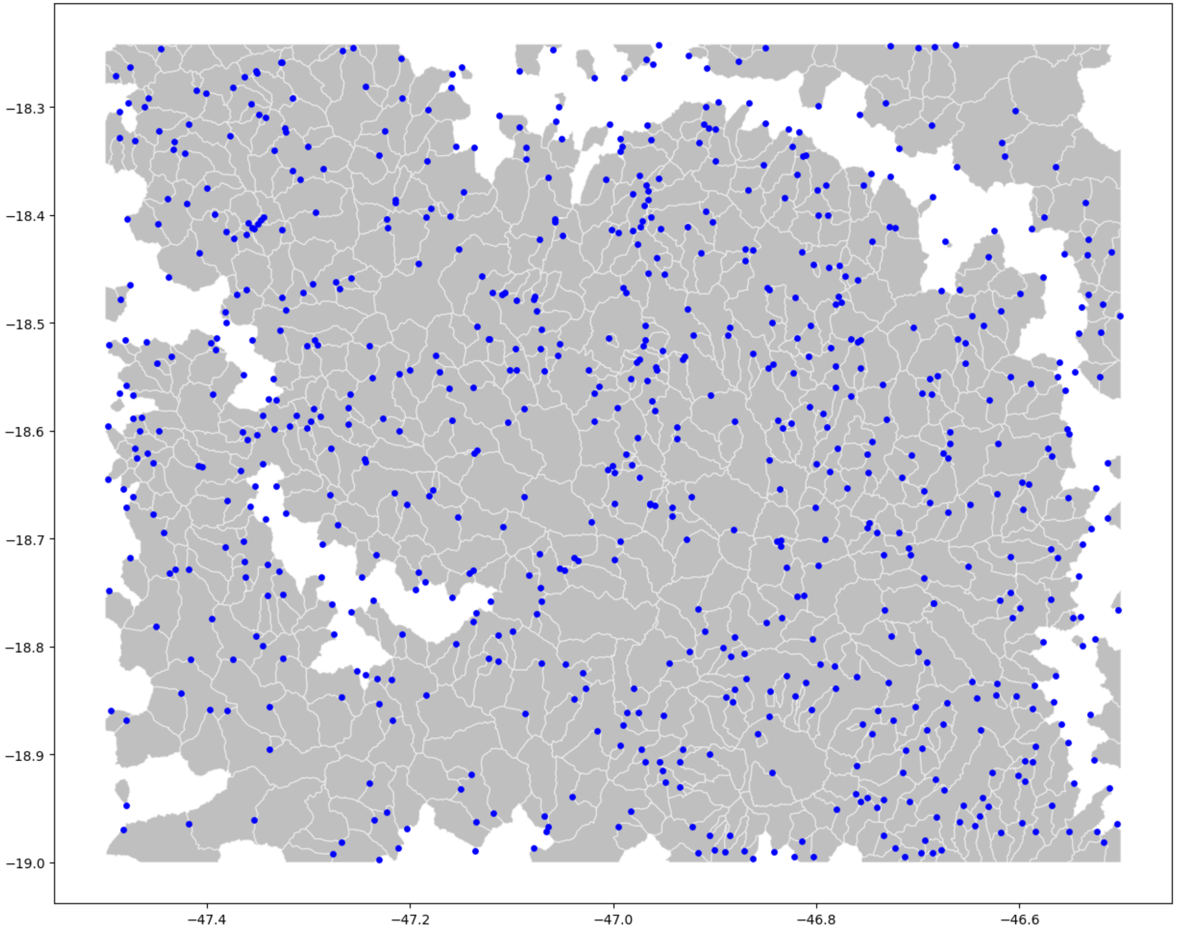
1. **Dataset Explanation**
2. Regression Models
3. Random Forest
4. XGBoost
5. Final Considerations



1. Dataset Explanation (1)

- **706 samples**
- **52 columns**
  - 5 were information about each sample (*Map IDX, Lab processed, x, y*)
  - 47 columns of elements concentration
    - We want to predict the concentration of **Zinc (Zn)** in PPM (parts-per-million)

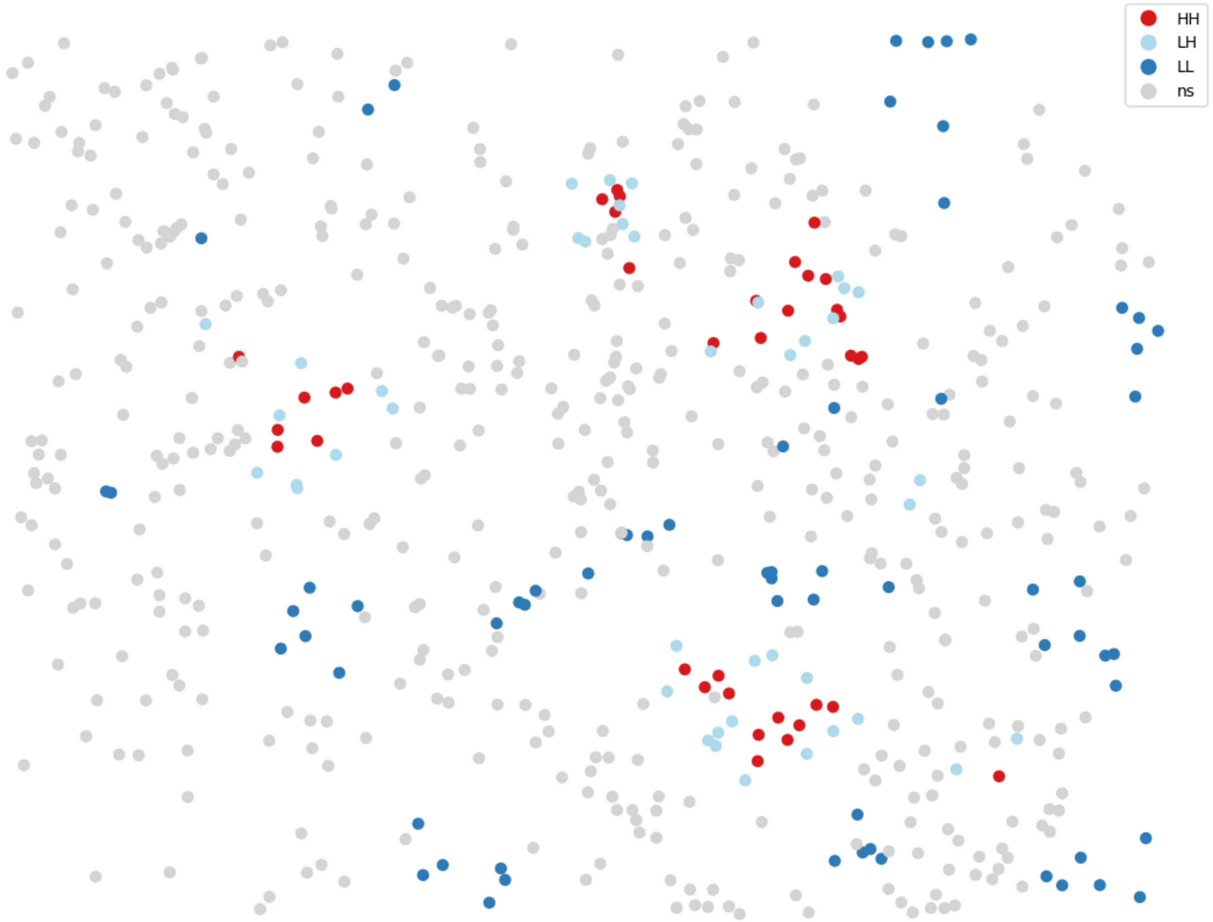
Data source = «*Brazilian Geological Survey* »



Samples are **blue** dots, in **gray** are the watersheds.

1. Dataset Explanation (2)

- **Pre-processing**
  - Normalization
  - Estimating Correlation
- The dataset has **spatial autocorrelation**
  - Regular train-test-split won't work for us!
- We decided to do train-test-split using **GroupKFolds**, where the groups were the watershed ID.



Moran's I Spatial Autocorrelation among samples

1. Dataset Explanation - Metrics

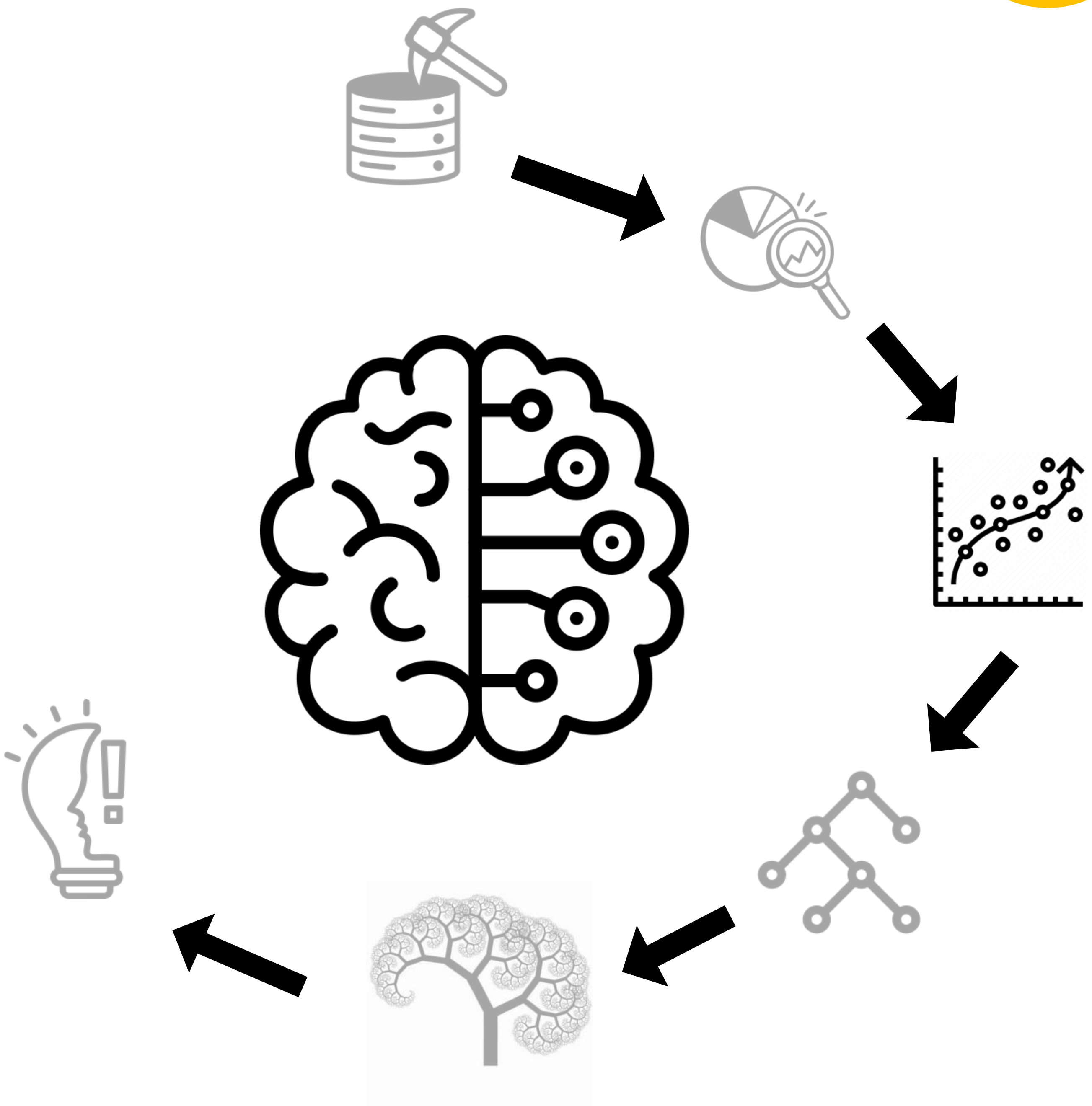
- Chosen metrics:

  - **R^2**: It gives us an estimate of the variance in the independent variable that can be explaned by the dependent variable

  - **RMSE**: It gives us a metric in ppm for Zinc, were it's possible to identify not only the performance but how much the model is being penalized.

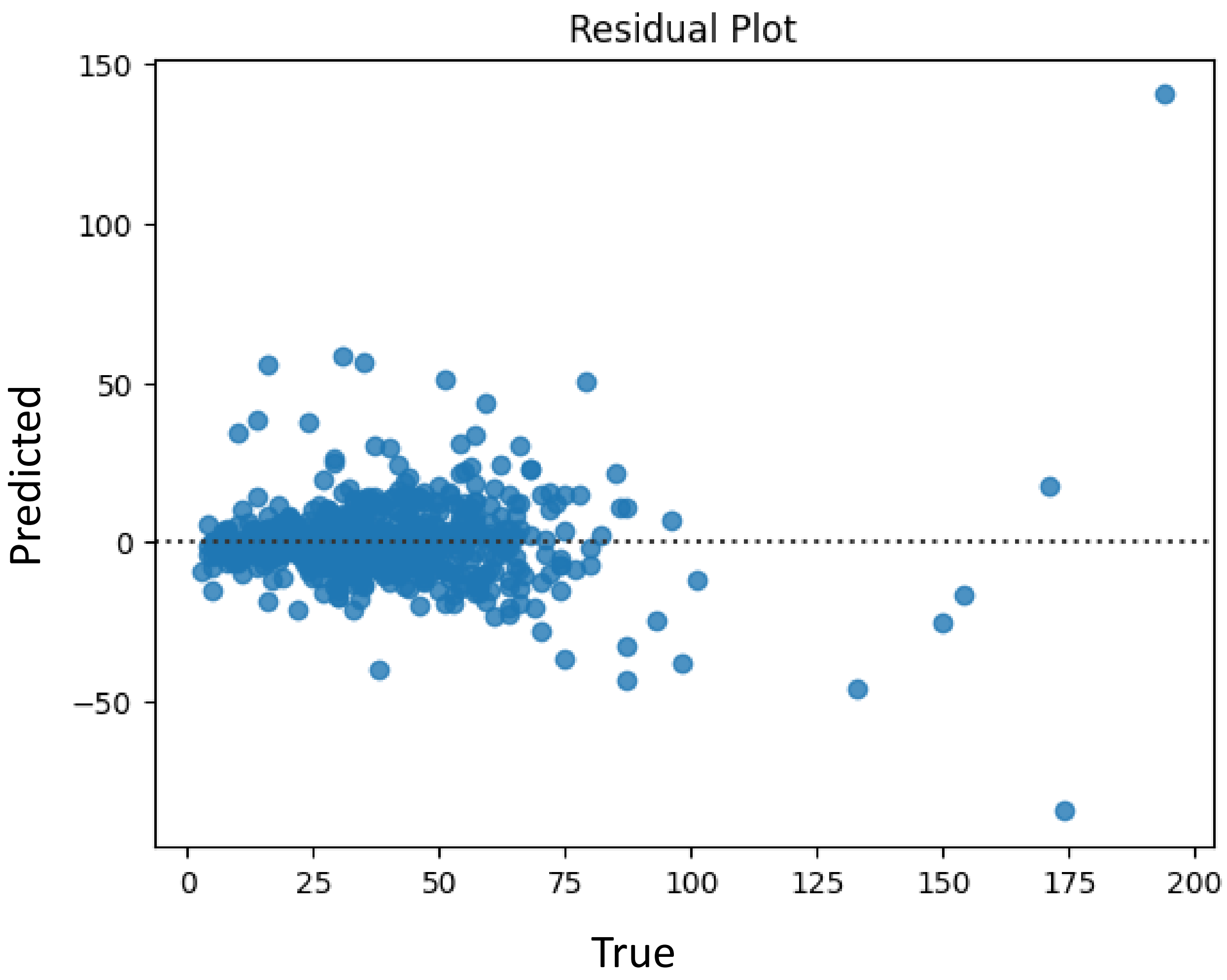
Overview

1. Dataset Explanation
2. **Regression Models**
3. Random Forest
4. XGBoost
5. Final Considerations



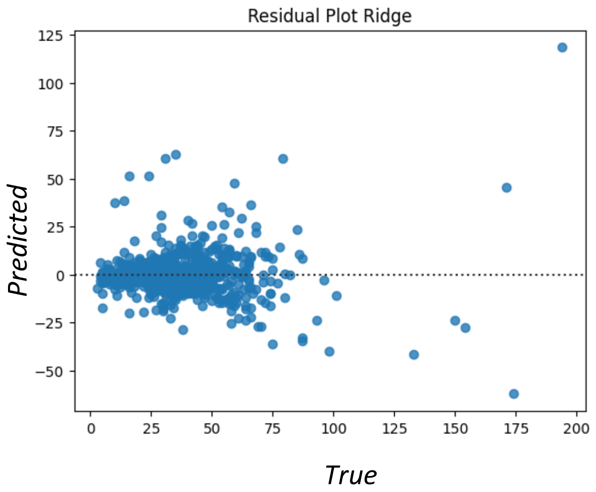
2. Regression Models

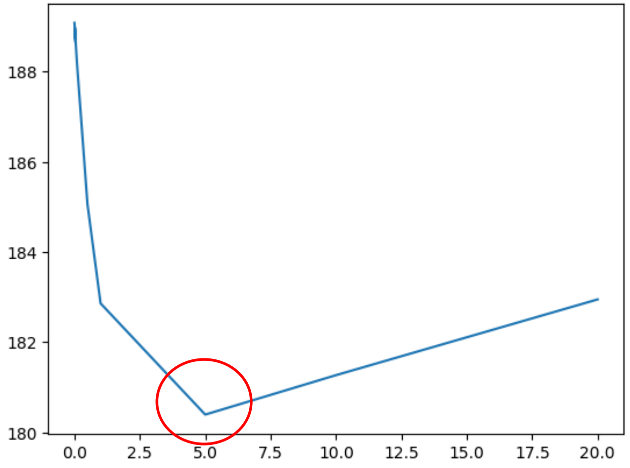
- Multiple Linear Regression
  - All elements to predict Zn
  - CV GroupKFold
    - RMSE: 13.75
    - R^2: 0.566

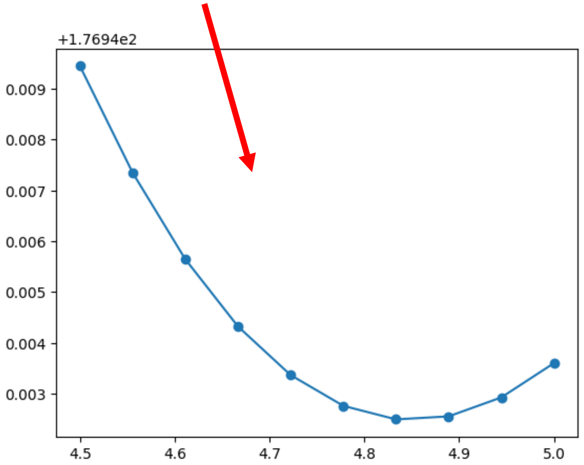


2. Regression Models: Regularization

- Ridge Regression
  - All elements to predict Zn
  - Regularization
    - Slight increase in bias to reduce a lot variance
  - **Optimal LR at 4.83**
    - R^2: 0.59
    - RMSE: 13.43
  - Reduction of variables that don't have importance in the model.

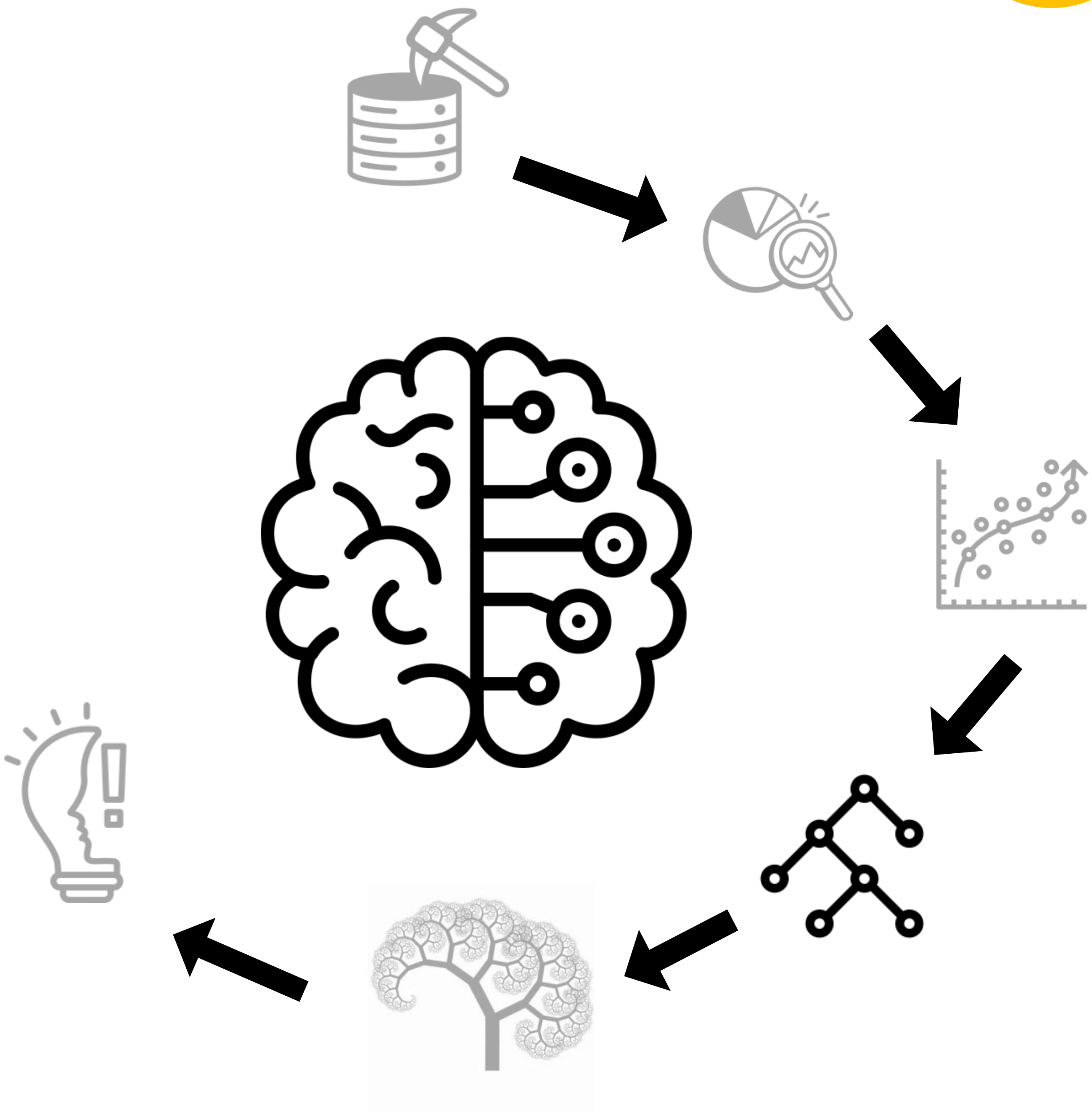






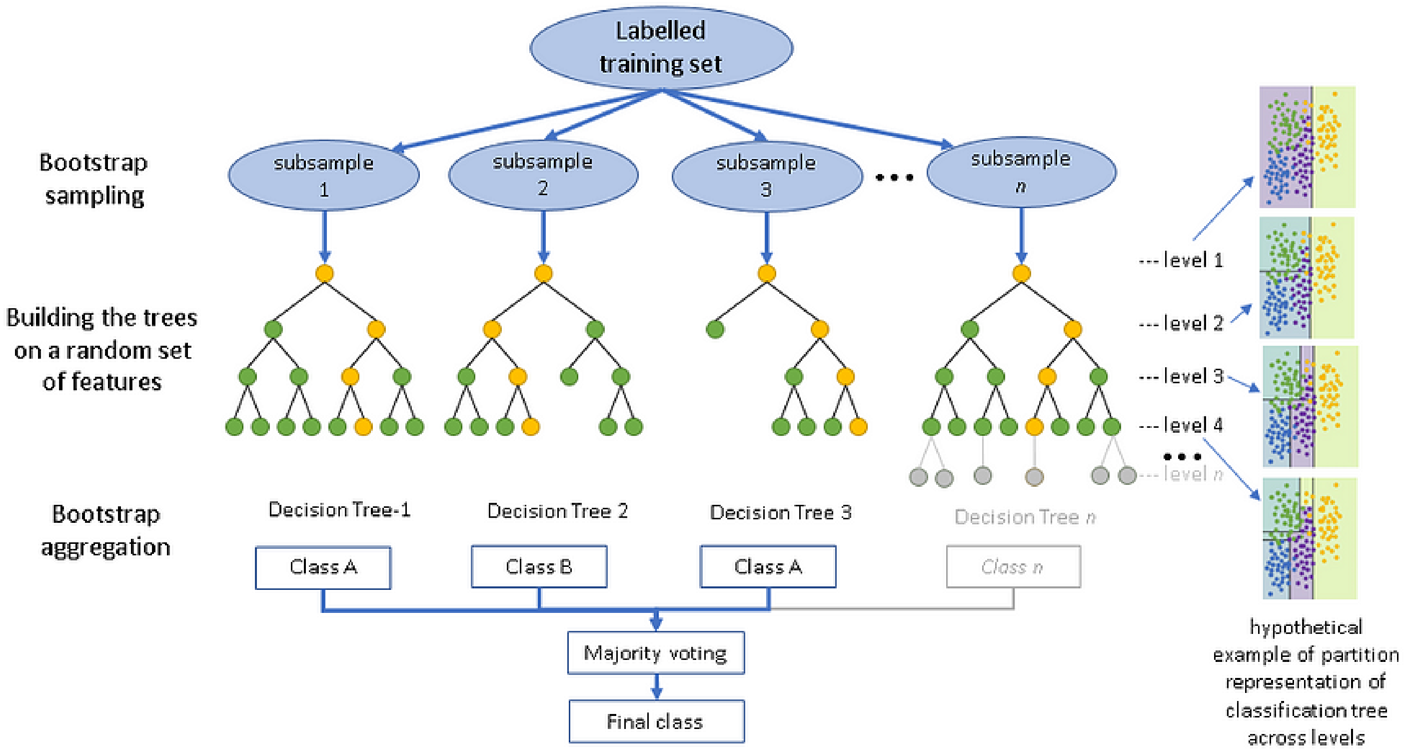
Overview

1. Dataset Explanation
2. Regression Models
3. **Random Forest**
4. XGBoost
5. Final Considerations



3. Random Forest

- RandomForest Hyperparameter Setting:
  - `Bootstrapping + feature sampling`
  - `max_features = features in each tree`
  - `n_estimators = number of trees`
- Propose a protocol that relies on the out-of-bag (OOB) error



Reference: https://medium.com/nerd-for-tech/random-forest-sturdy-algorithm-d60b9f9140d4

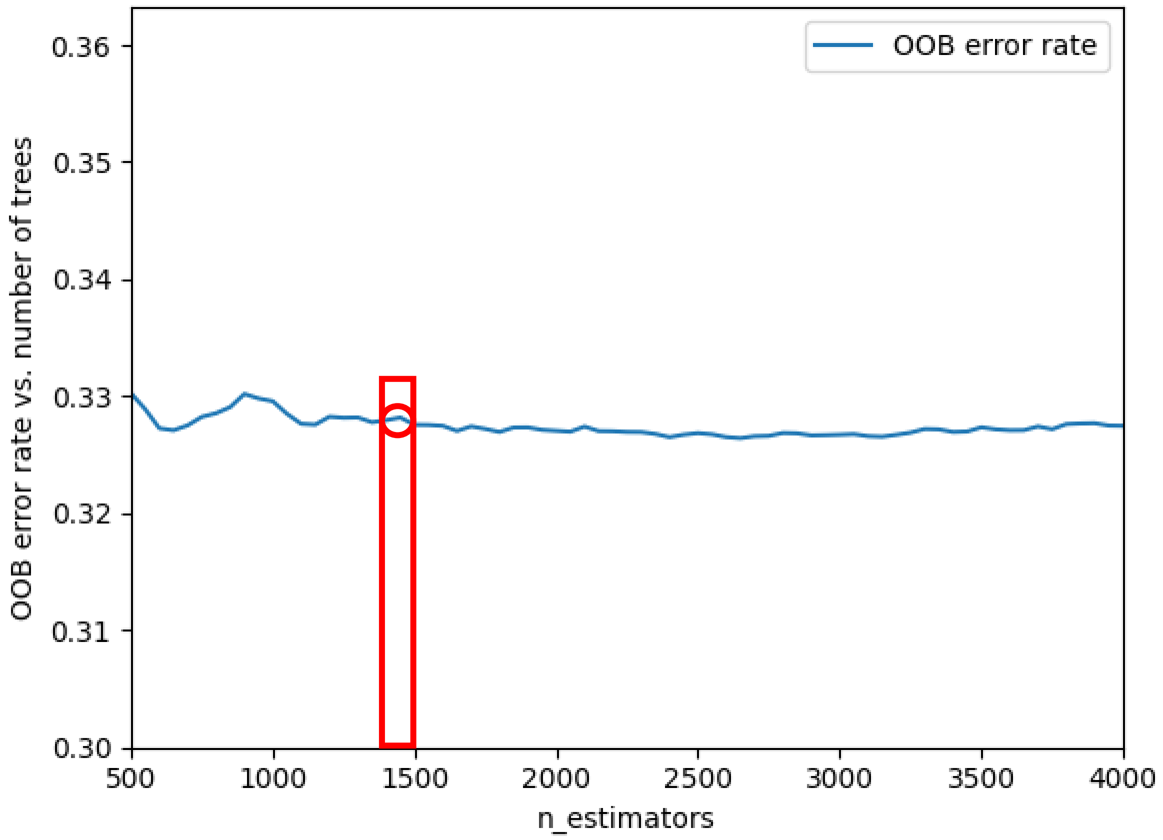
Hyperparameter tuning using **OOB** error

max_features

params	mean_test_score	std_test_score
{'max_features': 0.1}	0.643443	0.051902
{'max_features': 0.2}	0.657936	0.051251
{'max_features': 0.3}	0.660969	0.053202
{'max_features': 0.4}	0.658543	0.057185
{'max_features': 0.5}	0.656734	0.058457
{'max_features': 0.6}	0.651456	0.062576
{'max_features': 0.7}	0.646503	0.066120
{'max_features': 0.8}	0.641971	0.070297
{'max_features': 0.9}	0.635826	0.074615

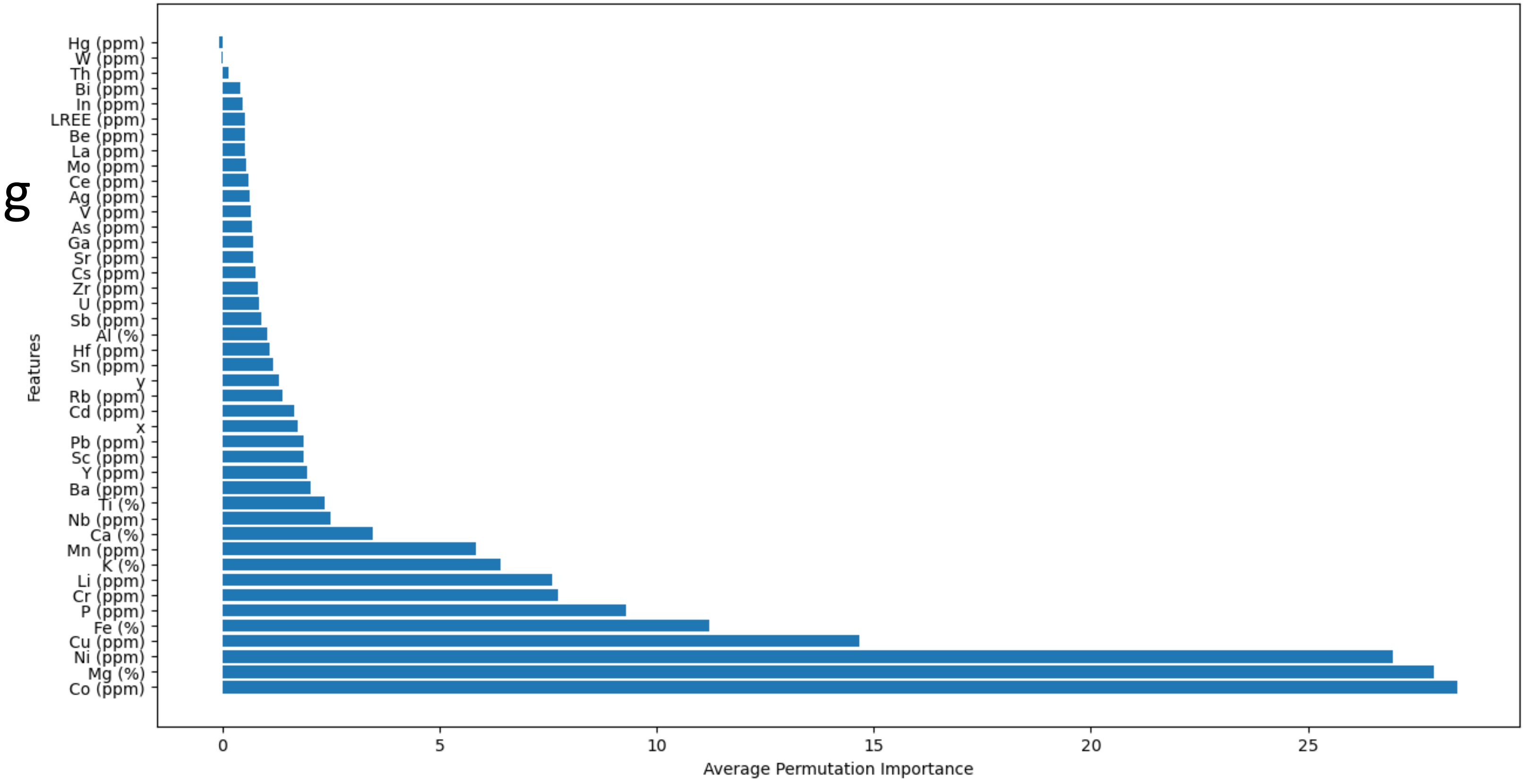
- Model Evaluation:
  - **R^2**: 0.62
  - **RMSE**: 13.83

n_estimators



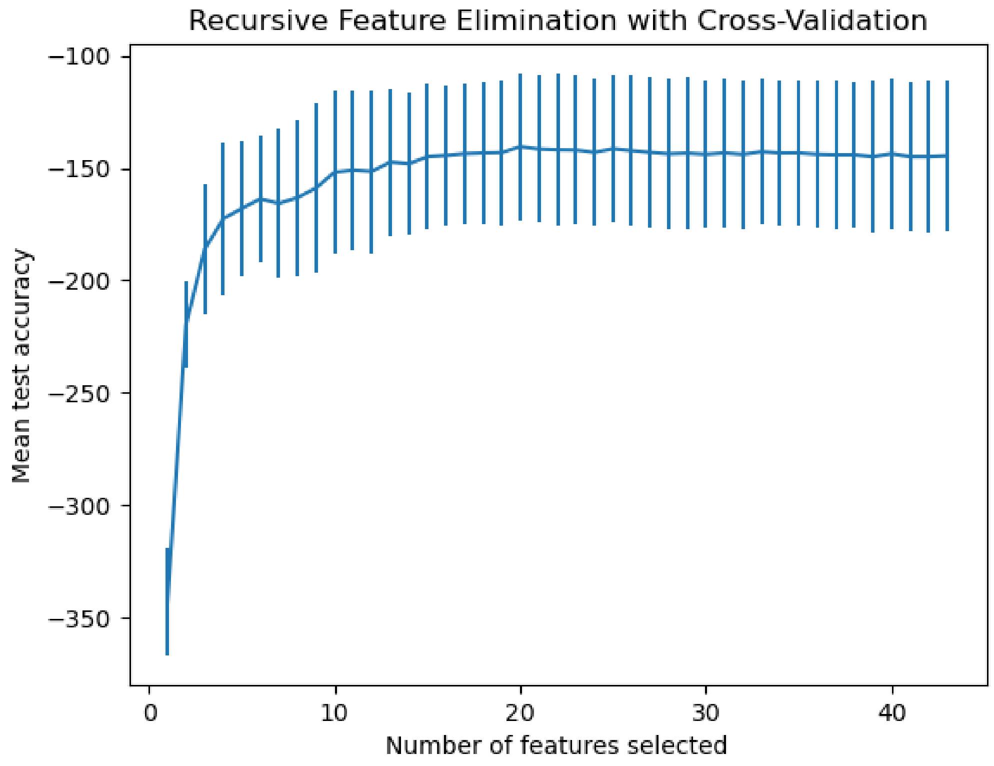
Feature Selection: Permutation importance

- **Randomly shuffle** the values of a specific feature in the dataset while keeping the other features **unchanged**.
- `"mean_test_score"`
  - **High** = important
  - **low** = not so important



REFCV - Recursive Feature Elimination with Cross-Validation

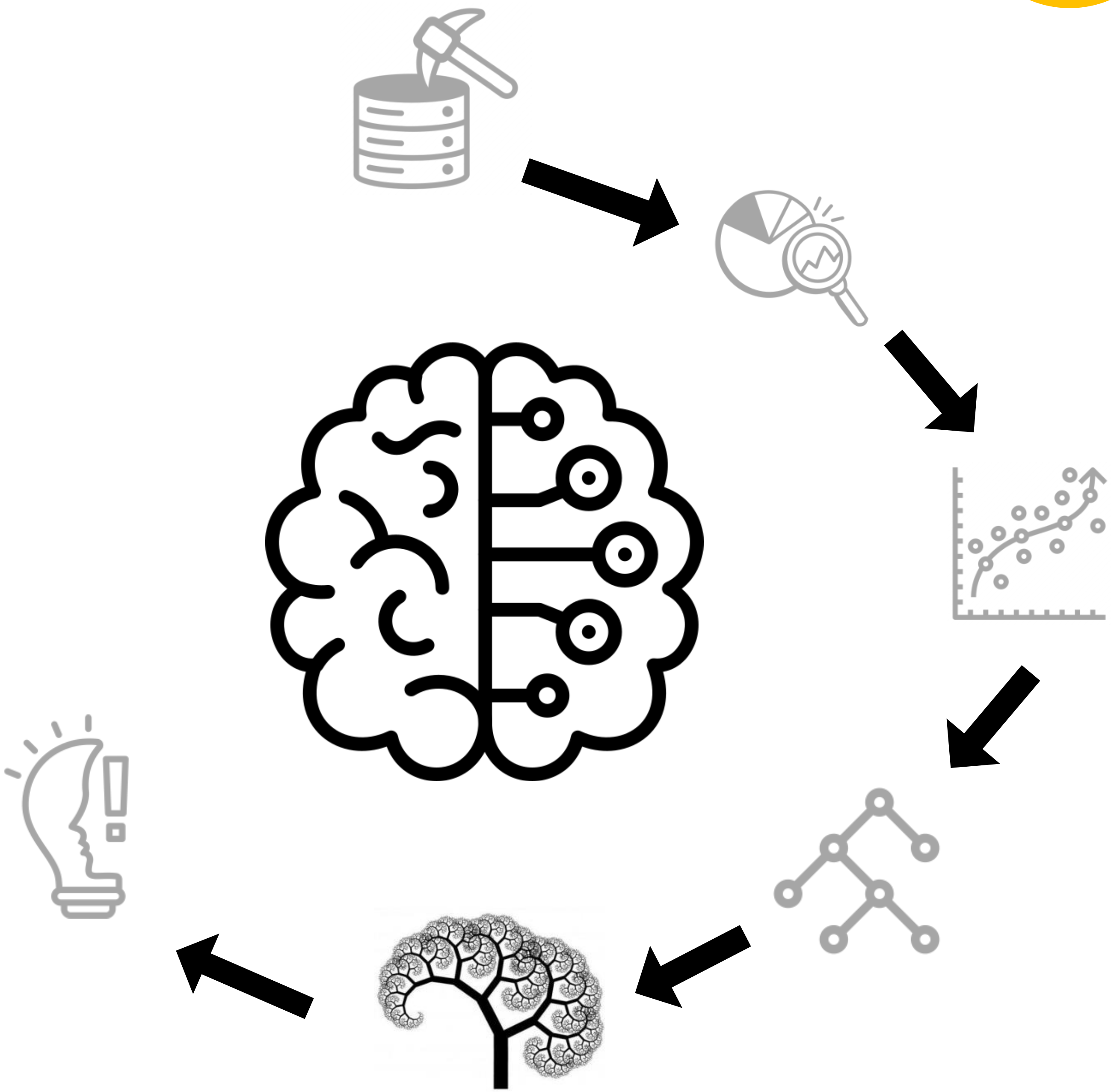
- Eliminates one feature or a small set of features at a time using cross-validation (CV)
- **Optimal Feature Subset**
- Improved performance
  - RMSE: 11.76
  - R^2: 0.67




```
Selected features: ['x', 'y', 'Ba (ppm)', 'Ca (%)', 'Co (ppm)', 'Cr (ppm)', 'Cu (ppm)', 'Fe (%)', 'K (%)', 'Li (ppm)', 'Mg (%)']
```

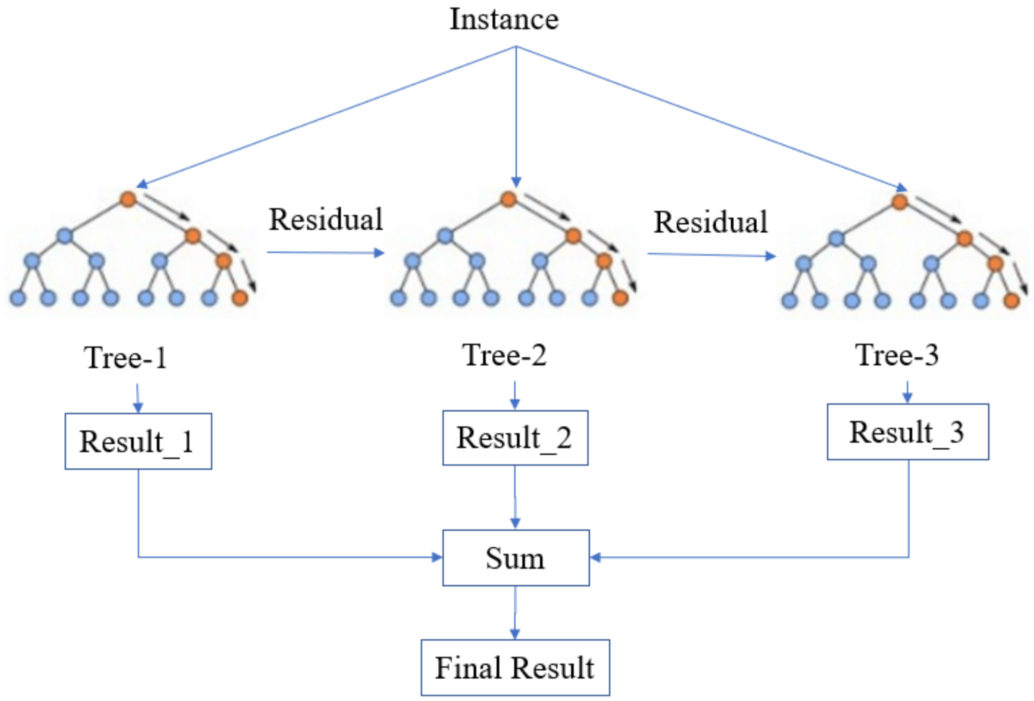
Overview

1. Dataset Explanation
2. Regression Models
3. Random Forest
4. **XGBoost**
5. Final Considerations



4. XGBoost

- XGBoost (Extreme Gradient Boost) is a machine learning algorithm introduced by Chen and Guestrin (2016).
- The base of XGBoost is the **Gradient Tree Boosting** machine learning algorithm that builds **an ensemble decision tree, where each tree attempts to correct errors from previous trees.**
- XGBoost uses **Gradient Descent** when trying to minimize the loss function when adding new models to the existing ones.



Wang et al, 2020

4. XGBoost

Hyperparameters

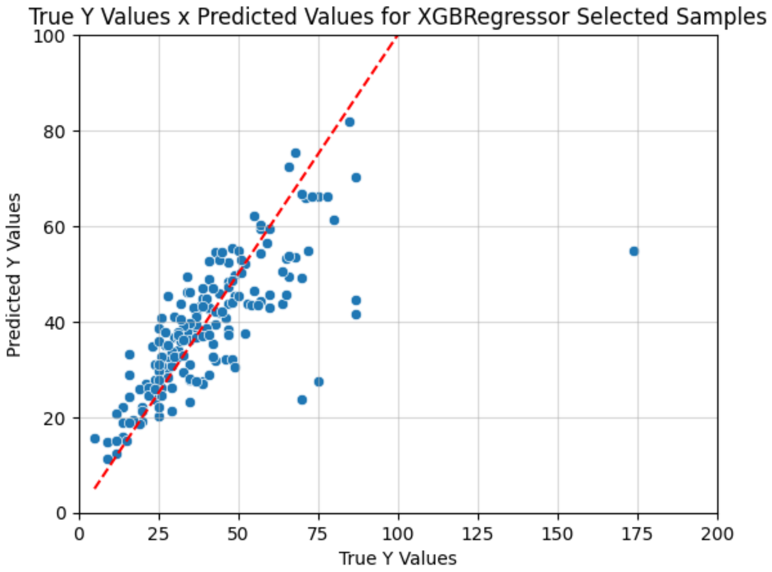
- max_depth[1]
- n_estimators
- learning_rate[3]
- subsample[2]
- colsample_bytree[2]
- max_delta_step
  - Help convergence and deal with unbalanced data

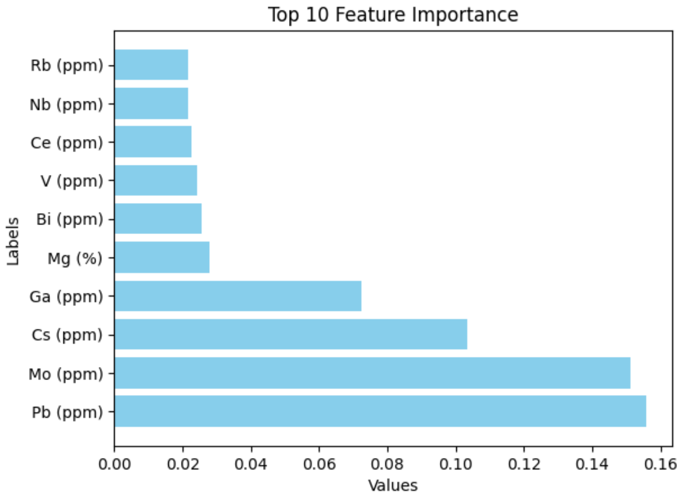
[1] - Controls model complexity

[2] - Add randomness to help with noise.

[3] - Avoid overshooting

R^2: 0.64 RMSE: 13.47

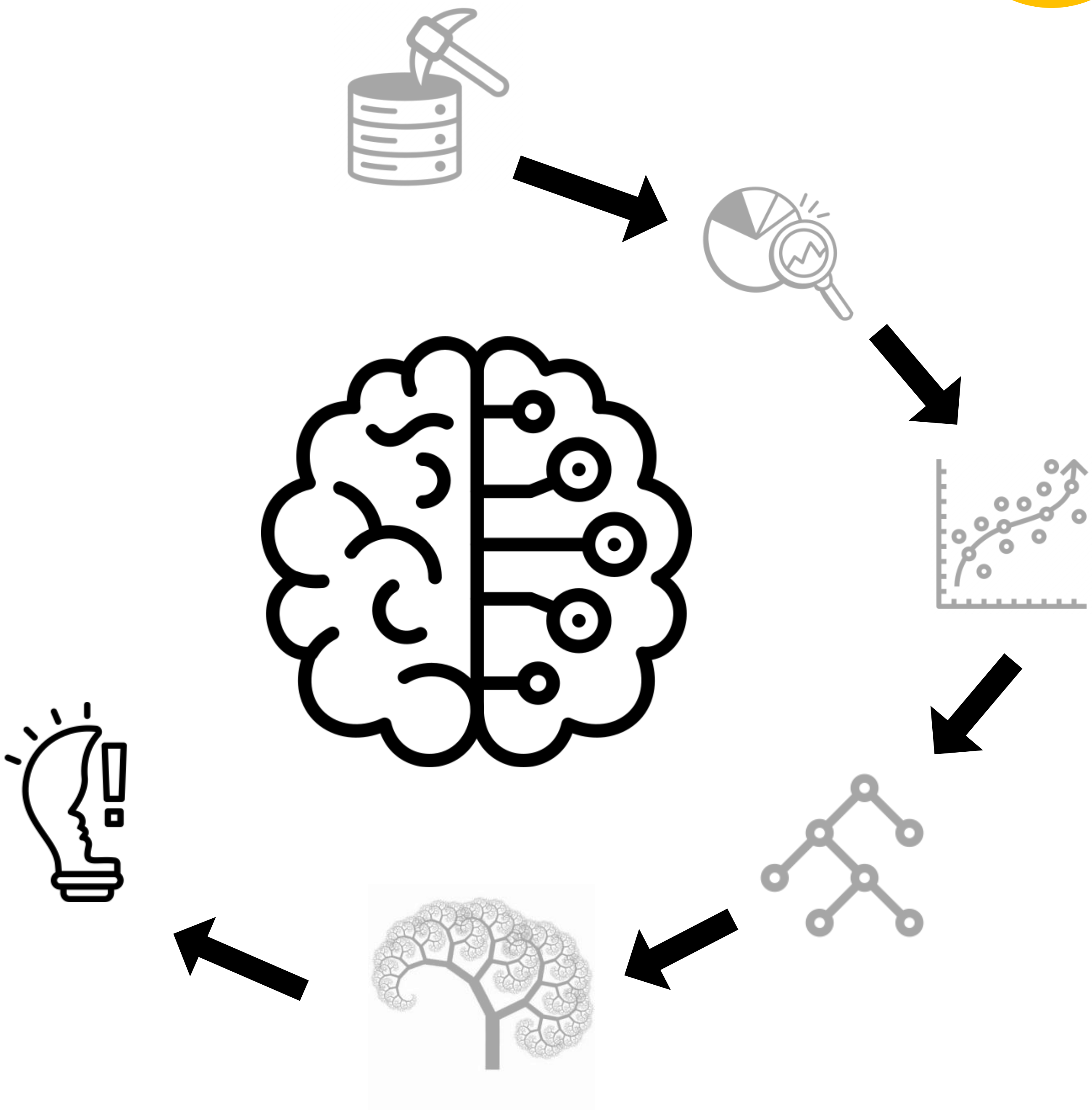




R^2: 0.58 RMSE: 14.48

Overview

1. Dataset Explanation
2. Regression Models
3. Random Forest
4. XGBoost
5. **Final Considerations**



5. Final Considerations: Model Comparison

Metric/Model	Multi Linear Regression	Ridge Regression	Random Forest	Random Forest Feature Selection	XGBoost	XGBoost Feature Selection
R^2	0.57	0.59	0.62	0.67*	0.64	0.58
RMSE	13.75	13.41	13.83	11.76*	13.47	14.48

5. Final Considerations: Model Comparison

- Overall, models perfomed slightly well
  - Ideally, a good model should have >=75%
- Possible solutions:
  - Removal of outliers (makes the model simpler)
  - Increase the amount of data to train the model (expensive)
  - Look for a geostatistical methodology that tries to take into account these outliers and still guarantee the model's good performance.

  - GitHub Repository

References

- Chugh, A. (2022, March 16). MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared—Which Metric is Better? Analytics Vidhya. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e
- How Spatial Autocorrelation (Global Moran's I) works—ArcGIS Pro | Documentation. (n.d.). Retrieved October 24, 2023, from https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/h-how-spatial-autocorrelation-moran-s-i-spatial-st.htm
- How to Create a Residual Plot in Python—GeeksforGeeks. (n.d.). Retrieved October 23, 2023, from https://www.geeksforgeeks.org/how-to-create-a-residual-plot-in-python/
- Kulcsar, L. (n.d.). Correlation: What is it? How to calculate it? .Corr() in pandas. Retrieved October 22, 2023, from https://data36.com/correlation-definition-calculation-corr-pandas/
- Regularization in Machine Learning || Simplilearn. (n.d.). Simplilearn.Com. Retrieved October 24, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning
- Saif, J. (2023, June 26). Correlation in data analytics: Medium. https://medium.com/@JaveriaSaif/correlation-in-data-analytics-75fec1f2147d
- Tabrez, S. (n.d.). Distribution of Test Data vs. Distribution of Training Data. Retrieved October 22, 2023, from https://www.tutorialspoint.com/distribution-of-test-data-vs-distribution-of-training-data
- https://machinelearningmastery.com/an-introduction-to-feature-selection/
- https://machinelearningmastery.com/feature-selection-machine-learning-python/
- https://www.datatechnotes.com/2022/10/feature-selection-example-with-rfecv-in.html#:~:text=RFECV%20(Recursive%20Feature%20Elimination%20with,features%20in%20a%20given%20dataset.
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html
- https://scikit-learn.org/stable/glossary.html#term-CV-splitter
- Stackoverflow support: https://stackoverflow.com/questions/44487654/build-a-random-forest-regressor-with-cross-validation-from-scratch
- https://medium.com/wicds/feature-importance-feature-selection-acac802ba565

Machine Learning Project: Stream Sediment Samples

Khizer Zakir & Rodrigo Brust Santos

08/12/2023