UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II - DIPARTIMENTO DI
FISICA "ETTORE PANCINI"

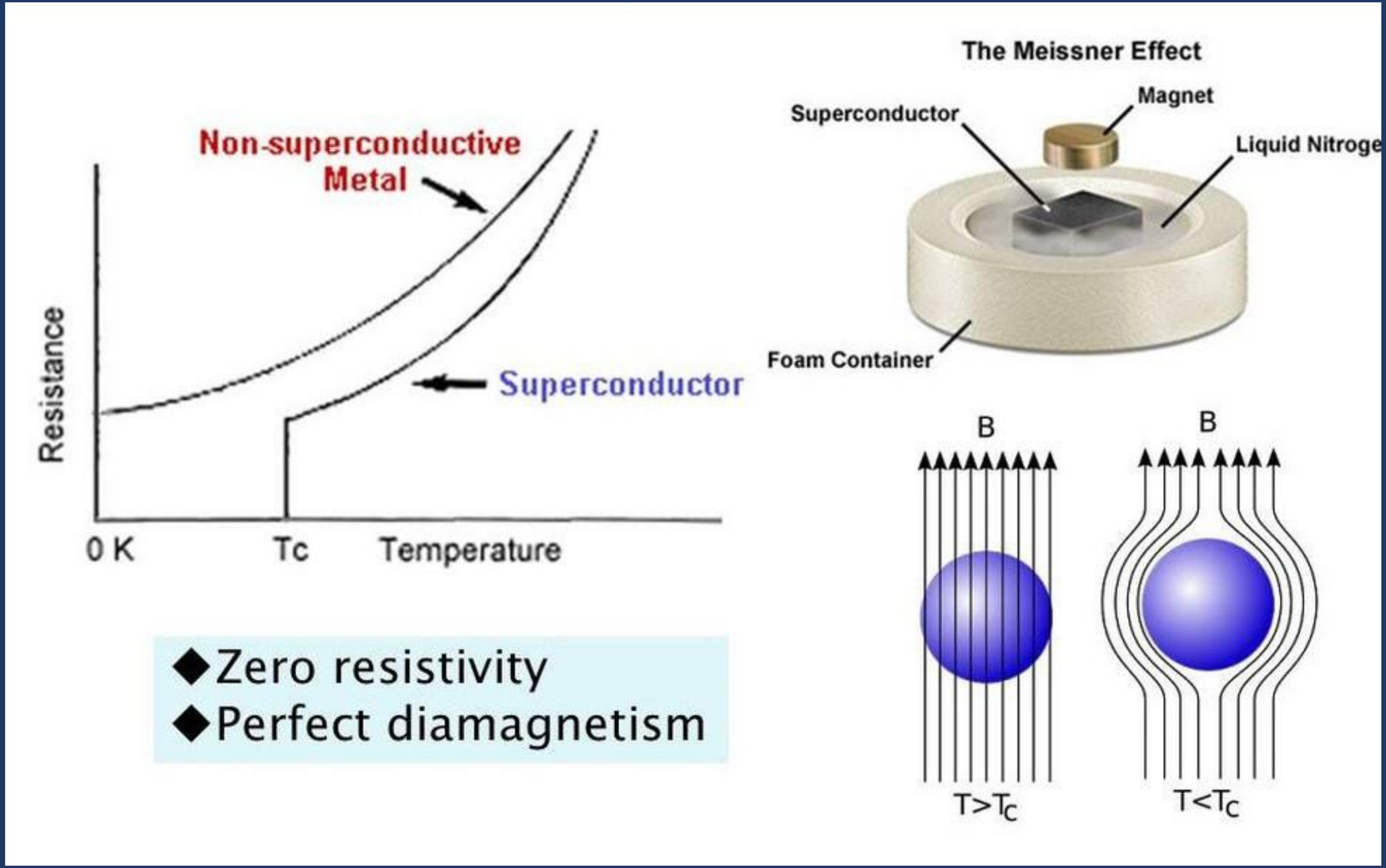
Machine Learning Approach for Prediction of Critical Temperature of Superconductor Materials

METODI DI APPRENDIMENTO AUTOMATICO PER LA FISICA 2021/2022

Vincenzo Lipardi
N94000636

What is Superconductivity?

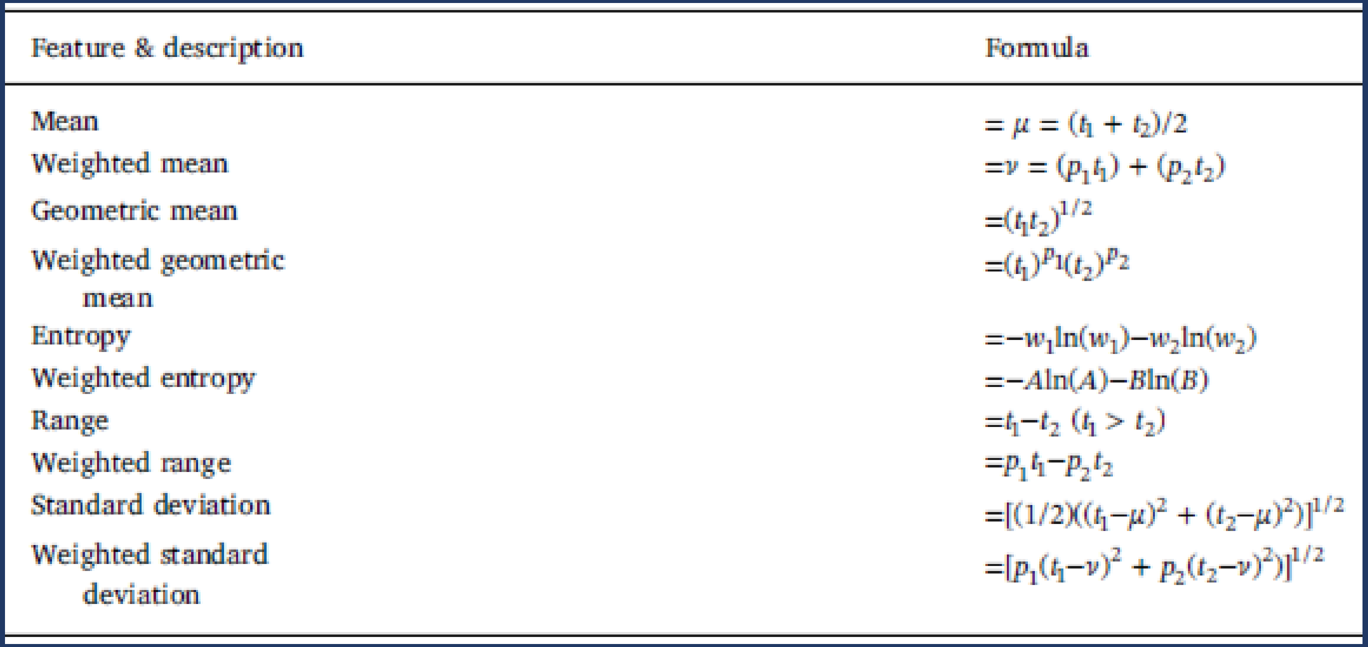
- Discovered by Onnes in 1911
- Zero Resistivity, Meissner Effect, Fase Transition
- Low Temperature VS High Temperature Superconductors
- BCS Theory
- Applications: Quantum Computers, Magnetic Levitation, Electromagnets for Engineering and many others.



Dataset Introduction

- **21263 Samples**
  - SuperCon Online Database of NIMS (Japan’s National Institute for Materials Science)
  - Link: https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data
  - We are going to study the preprocessed data stored in

- **81 Feature Real Variables**
  1. Number of Elements
  2. Atomic Mass
  3. Atomic Radius
  4. First Ionization Energy
  5. Density
  6. Electron Affinity
  7. Fusion Heat
  8. Thermal Conductivity
  9. Valence



Feature & description	Formula
Mean	$= \mu = (t_1 + t_2)/2$
Weighted mean	$= \nu = (p_1 t_1) + (p_2 t_2)$
Geometric mean	$= (t_1 t_2)^{1/2}$
Weighted geometric mean	$= (t_1)^{p_1} (t_2)^{p_2}$
Entropy	$= -w_1 \ln(w_1) - w_2 \ln(w_2)$
Weighted entropy	$= -A \ln(A) - B \ln(B)$
Range	$= t_1 - t_2 \ (t_1 > t_2)$
Weighted range	$= p_1 t_1 - p_2 t_2$
Standard deviation	$= [(1/2)((t_1 - \mu)^2 + (t_2 - \mu)^2)]^{1/2}$
Weighted standard deviation	$= [p_1 (t_1 - \nu)^2 + p_2 (t_2 - \nu)^2)]^{1/2}$

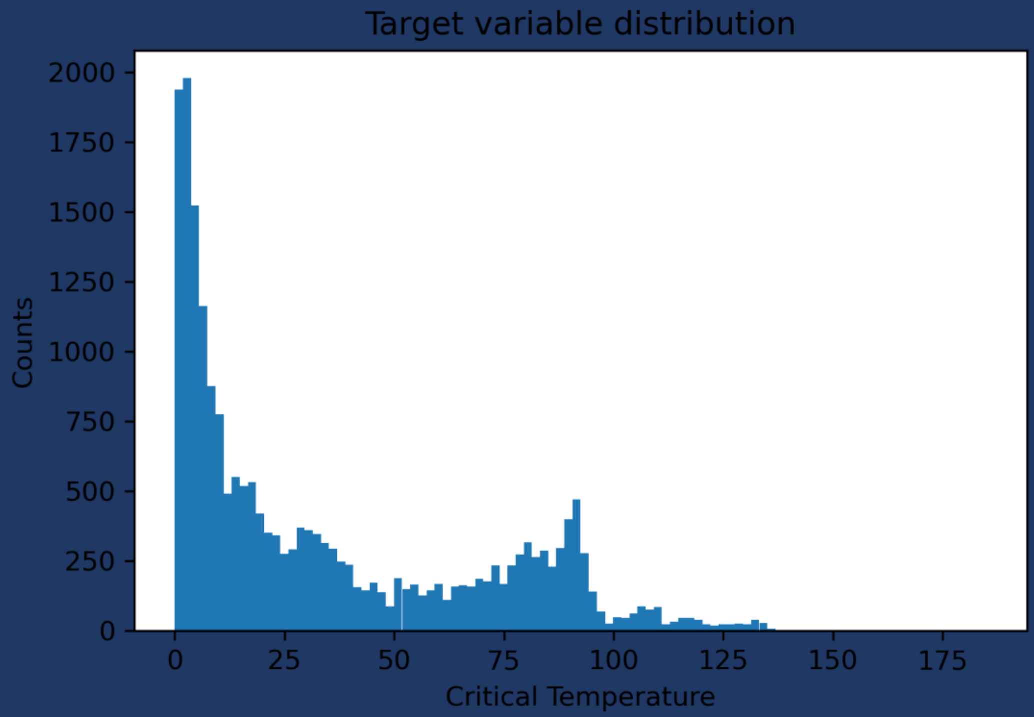
- **Continous Target Variable: Critical Temperature**
  - Regression problem

Content

Exploratory Data Analysis

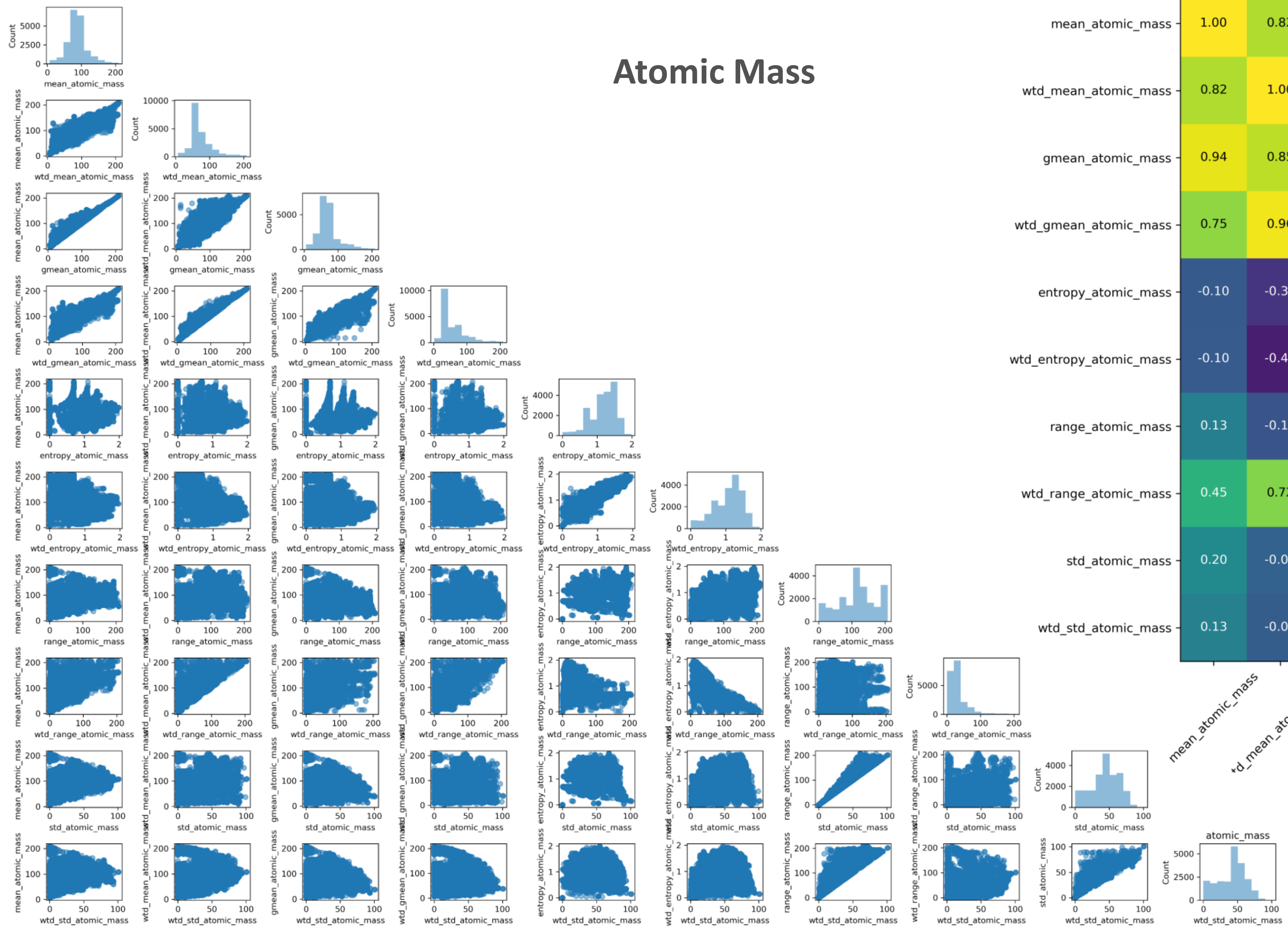
- **No Missing Data**
- **No Categorical Data**

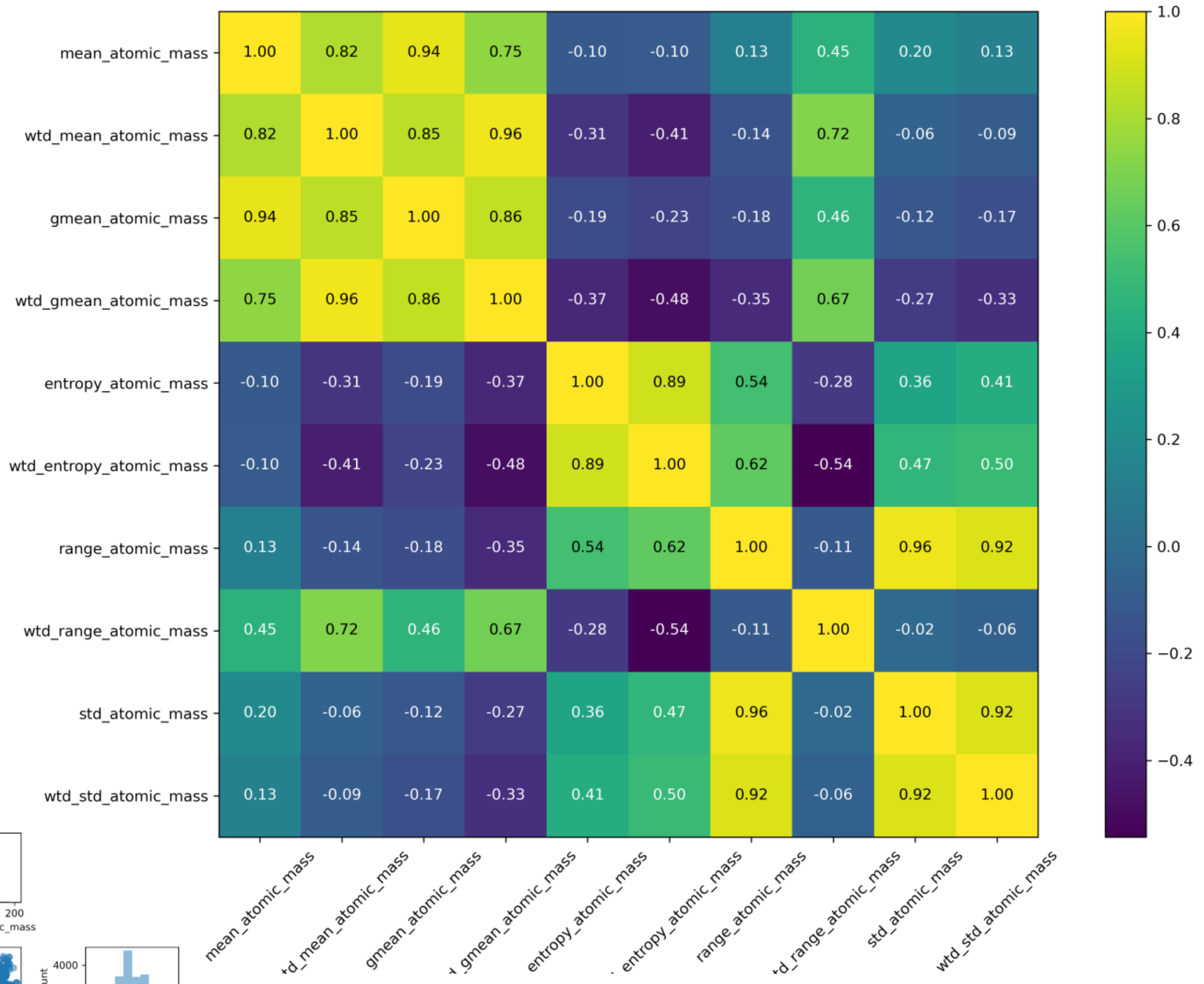
	CRITICAL TEMPERATURE
MEAN	34,4
STD	34,2
MIN	0,0
MAX	185,0



Scatter Plot and Correlation Matrices

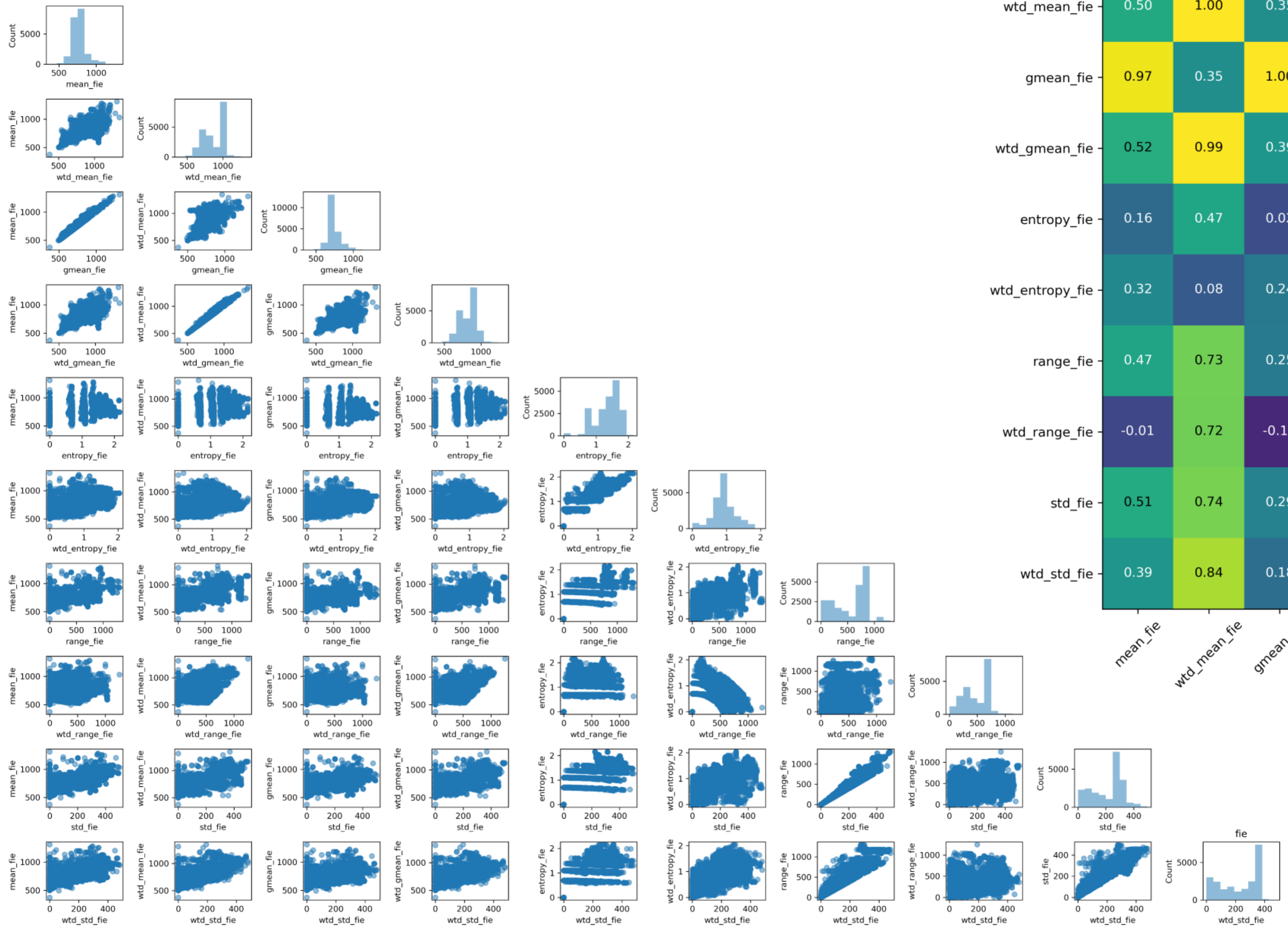
Atomic Mass

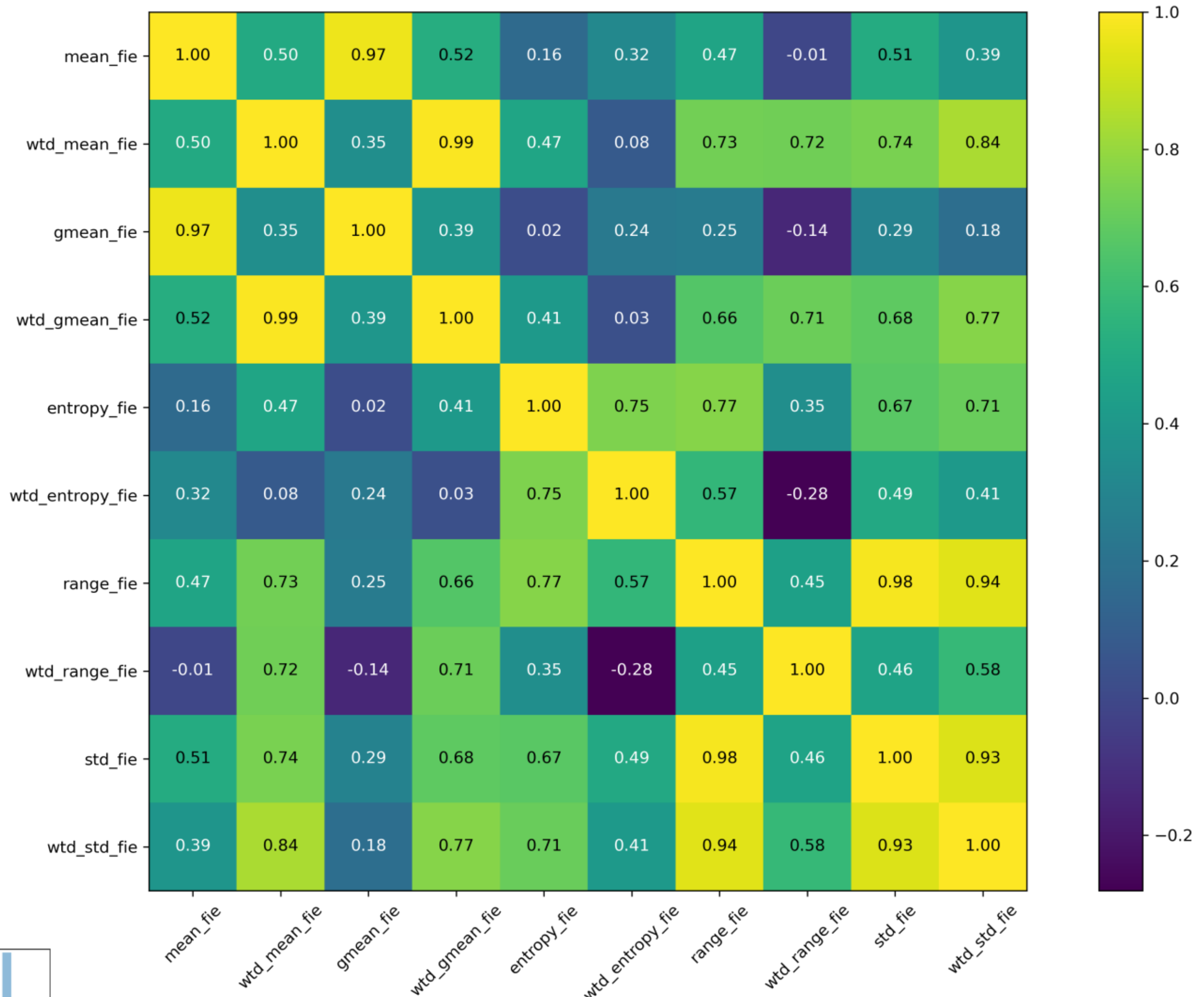




	mean_atomic_mass	wtd_mean_atomic_mass	gmean_atomic_mass	wtd_gmean_atomic_mass	entropy_atomic_mass	wtd_entropy_atomic_mass	range_atomic_mass	wtd_range_atomic_mass	std_atomic_mass	wtd_std_atomic_mass
mean_atomic_mass	1.00	0.82	0.94	0.75	-0.10	-0.10	0.13	0.45	0.20	0.13
wtd_mean_atomic_mass	0.82	1.00	0.85	0.96	-0.31	-0.41	-0.14	0.72	-0.06	-0.09
gmean_atomic_mass	0.94	0.85	1.00	0.86	-0.19	-0.23	-0.18	0.46	-0.12	-0.17
wtd_gmean_atomic_mass	0.75	0.96	0.86	1.00	-0.37	-0.48	-0.35	0.67	-0.27	-0.33
entropy_atomic_mass	-0.10	-0.31	-0.19	-0.37	1.00	0.89	0.54	-0.28	0.36	0.41
wtd_entropy_atomic_mass	-0.10	-0.41	-0.23	-0.48	0.89	1.00	0.62	-0.54	0.47	0.50
range_atomic_mass	0.13	-0.14	-0.18	-0.35	0.54	0.62	1.00	-0.11	0.96	0.92
wtd_range_atomic_mass	0.45	0.72	0.46	0.67	-0.28	-0.54	-0.11	1.00	-0.02	-0.06
std_atomic_mass	0.20	-0.06	-0.12	-0.27	0.36	0.47	0.96	-0.02	1.00	0.92
wtd_std_atomic_mass	0.13	-0.09	-0.17	-0.33	0.41	0.50	0.92	-0.06	0.92	1.00

Fie





	mean_fie	wtd_mean_fie	gmean_fie	wtd_gmean_fie	entropy_fie	wtd_entropy_fie	range_fie	wtd_range_fie	std_fie	wtd_std_fie
mean_fie	1.00	0.50	0.97	0.52	0.16	0.32	0.47	-0.01	0.51	0.39
wtd_mean_fie	0.50	1.00	0.35	0.99	0.47	0.08	0.73	0.72	0.74	0.84
gmean_fie	0.97	0.35	1.00	0.39	0.02	0.24	0.25	-0.14	0.29	0.18
wtd_gmean_fie	0.52	0.99	0.39	1.00	0.41	0.03	0.66	0.71	0.68	0.77
entropy_fie	0.16	0.47	0.02	0.41	1.00	0.75	0.77	0.35	0.67	0.71
wtd_entropy_fie	0.32	0.08	0.24	0.03	0.75	1.00	0.57	-0.28	0.49	0.41
range_fie	0.47	0.73	0.25	0.66	0.77	0.57	1.00	0.45	0.98	0.94
wtd_range_fie	-0.01	0.72	-0.14	0.71	0.35	-0.28	0.45	1.00	0.46	0.58
std_fie	0.51	0.74	0.29	0.68	0.67	0.49	0.98	0.46	1.00	0.93
wtd_std_fie	0.39	0.84	0.18	0.77	0.71	0.41	0.94	0.58	0.93	1.00

Scatter Plot and Correlation Matrices

Electron Affinity

Fusion Heat

Thermal Conductivity

Valence

Atomic Radius

Density

Scatter Plot and Correlation Matrices

Feature Scaling

PROBLEMS

- Distorted feature importances on some models:
- Overfitting
- Long Running time

SOLUTION

- Feature Standardization

$$x' = \frac{x - \mu}{\sigma}$$

- Feature Normalization

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Feature Selection

PROBLEMS

- Multicolliniarity
- Overfitting
- Long Running time

SOLUTIONS

- Variance Threshold Selection
  - 42 Features
- Sequential Backward Selection
  - 2 Features
- Random Forest: feature importances

Feature Extraction

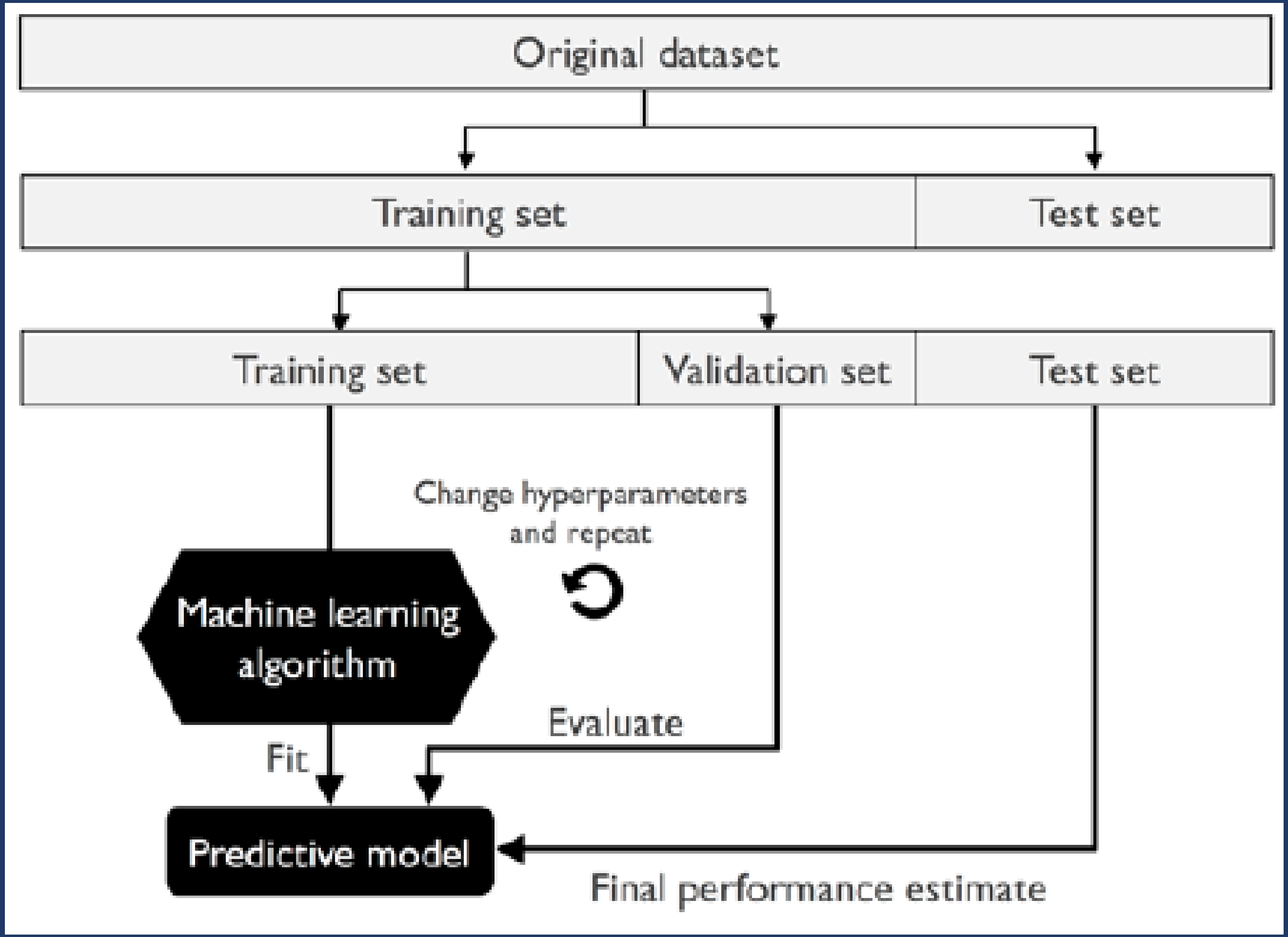
PROBLEMS

- Multicolliniarity
- Overfitting
- Long Running time

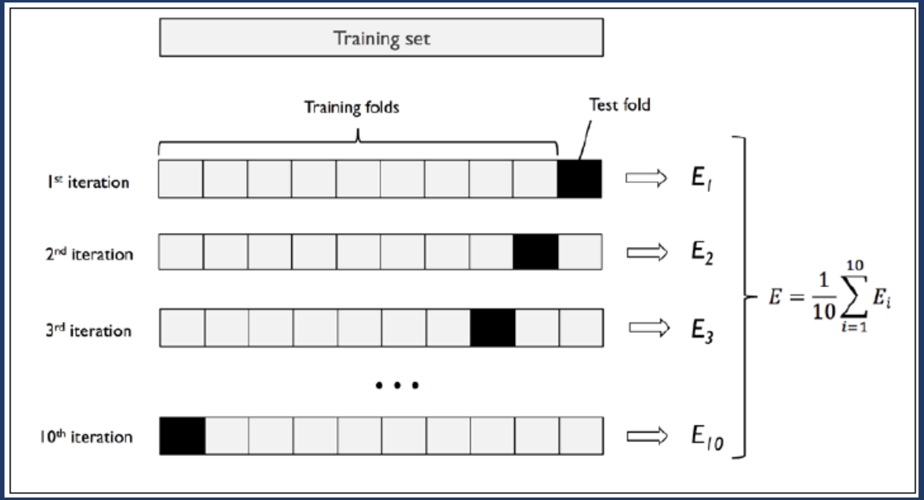
SOLUTIONS

- PCA

Model Performance



- PARAMETER VS HYPERPARAMETER



Cross Validation

It estimates the generalization performance of machine learning models

CROSS VALIDATION ALGORITHM	DESCRIPTION
HOLDOUT	• It split the initial dataset into separate training and test datasets or better into 3 parts: training, validation and test datasets.
K - FOLD	• The performance estimate is less sensitive to the sub-partitioning of the training data. • It randomly split the training dataset into *k* folds without replacement, where *k* – 1 folds are used for the model training, and one fold is used for performance evaluation.
LEAVE - p - OUT	• It's a k-fold CV generalization, where you choose the number p of folds used for the model training, Then, n-p folds are used for the evaluation.

HYPERPARAETER TUNING

Grid Search CV

Random Search CV

Brute-force exhaustive search paradigm

Randomized search for sampling different parameter combinations

Evaluating the Performance of Regression Models

$$MSE = \frac{1}{n} \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Mean Squared Error

$$SSE = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Sum of Squares Error

$$SST = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \mu_y\right)^2$$

Sum of Squares Total

$$R^2 = 1 - \frac{SSE}{SST}$$

Coefficient of Determination

For training data: $0 \leq R^2 \leq 1$, *but for test data it can became negative*

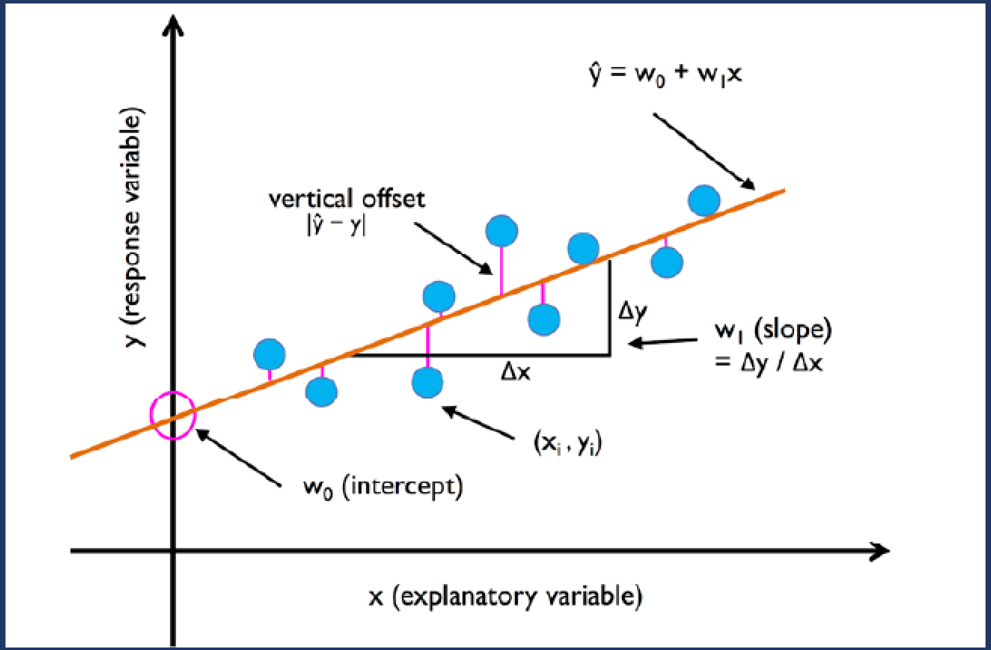
Residual Plot

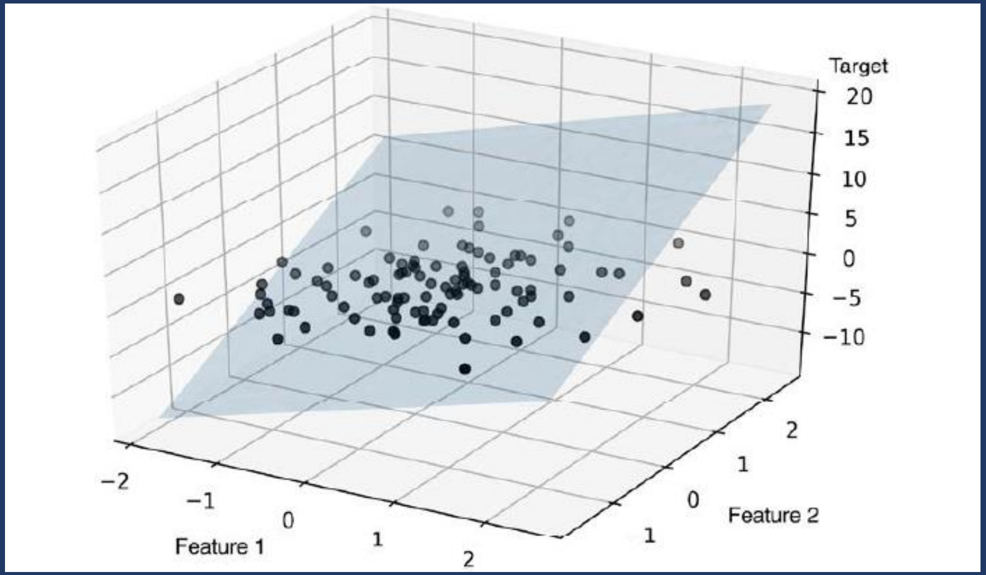
Multiple Linear Regression

The goal of linear regression is to model relationship between *one or multiple* features and a **continuous** target variable.

$$y = w_0 + w_1 x$$

x = explanatory variable(s)
y = response variable
w_0 = intercept
w_1 = slope(s)





PROS	CONS
Simple and Rapid algorithm	Heavy impact by the presence of outliers

Regularized Methods for Regression

Lasso

- L1 penalized model
- $J(w)_{LASSO} = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda \| w \|_1$

Ridge

- L2 penalized model
- $J(w)_{RIDGE} = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda \| w \|^2{}_2$

ElasticNet

- Compromise between the two previous models
- $J(w)_{ELNET} = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda_1 \sum_{j=1}^{m} w_j^2 + \lambda_2 \sum_{j=1} |w_j|$

RANdom SAmple Consensus

ALGORITHM

1) Select a random number of examples to be inliers and fit the model.
2) Test all other data points against the fitted model and add those points that fall within a user-given tolerance to the inliers.
3) Refit the model using all inliers.
4) Estimate the error of the fitted model versus the inliers.
5) Terminate the algorithm if the performance meets a certain user-defined threshold or if a fixed number of iterations were reached; go back to step 1 otherwise.

PROS	CONS
Reduce the potential effect of outliers	There are many hyperparameters to set

Decision Tree

INFORMATION GAIN

$$IG(D_p) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

IMPURITY METRIC

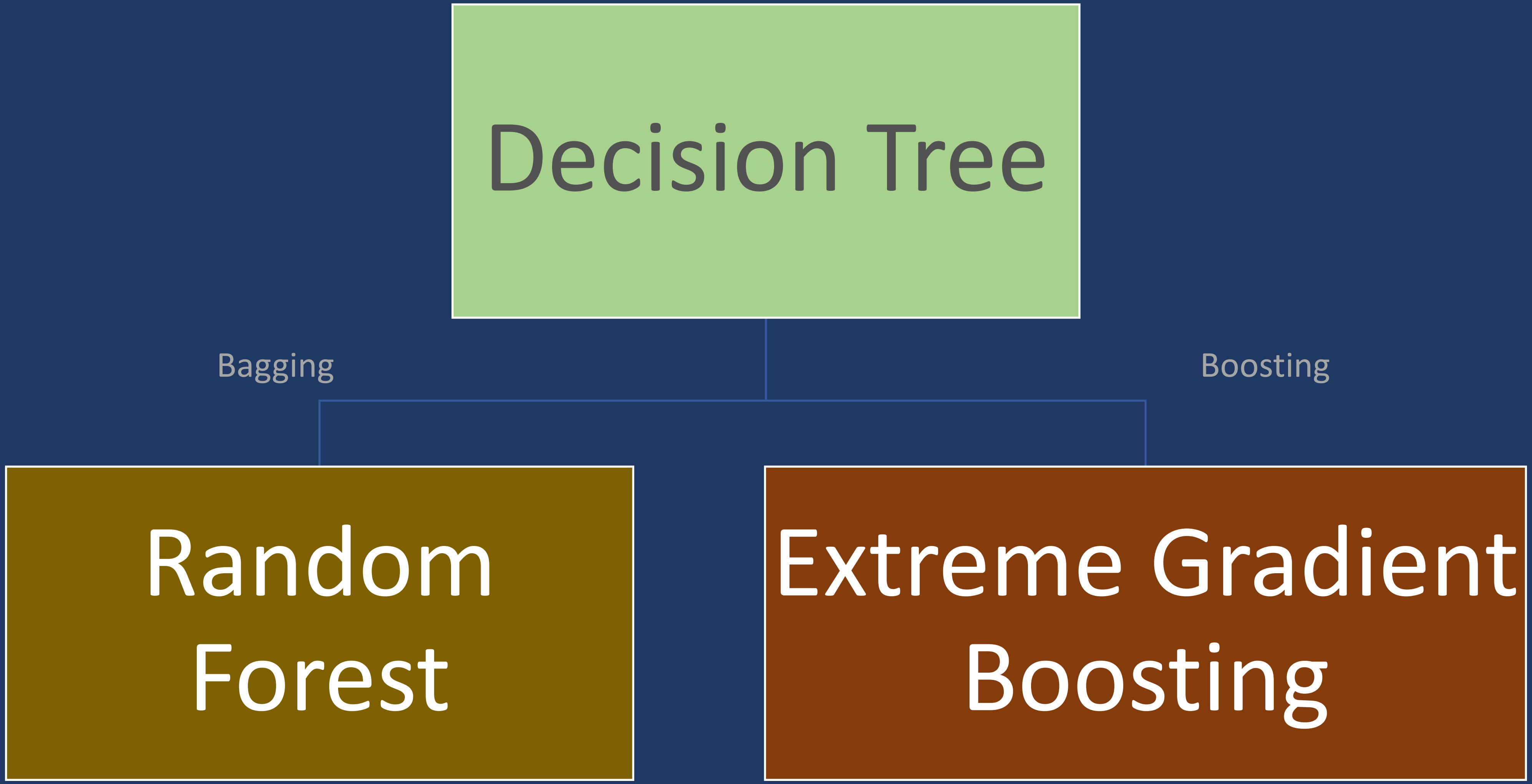
- Mean Squared Error

$$\mathrm{I(t)} = \frac{1}{N_t} \sum_{i \in D_t}^{c} \left(y^{(i)} - \hat{y}_t\right)^2$$

HYPERPARAMETERS

- max_depth
- criterion
- splitter

PROS	CONS
• Understanding • No standardization needed • Low Computaional cost	• Overfitting • High noise sensibility • A lot of hyperparameters


Decision Tree
Bagging
Boosting
Random Forest
Extreme Gradient Boosting

Random Forest

- Ensamble Method (Bagging)
- Decision Tree in parallel
- Scale Invariant
- One hyperparameter: max_depth

PROS	CONS
• Invariant scale • Best generalization performance • Simple tuning	Overfitting

Extreme Gradient Boosting Regressor

- Ensamble Method (Boosting)

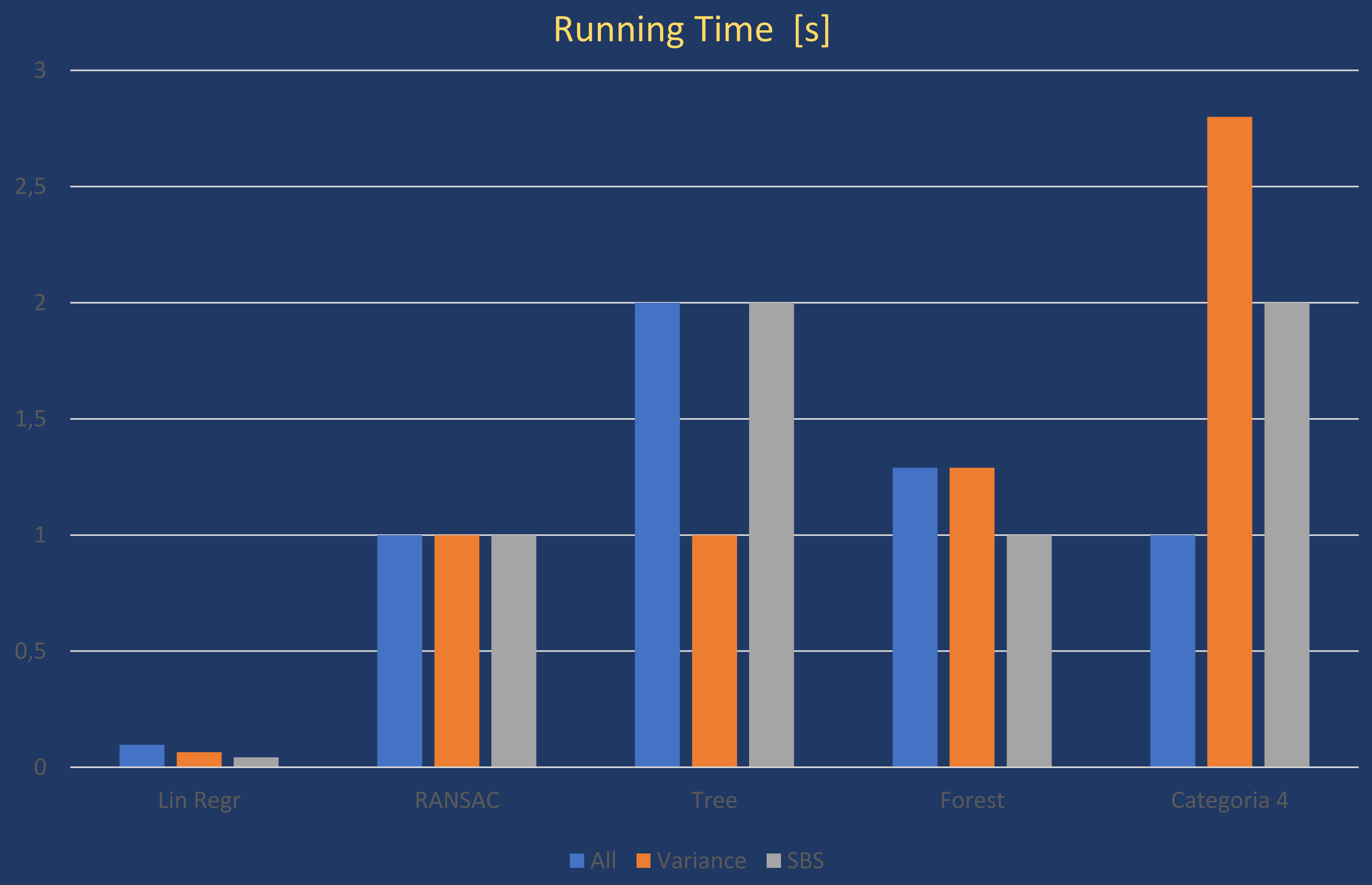
- Gradient boosting involves three elements:

  1. A loss function (MSE) to be optimized.
  2. A weak learner(Tree) to make predictions.
  3. An additive model to add weak learners to minimize the loss function.

PROS	CONS
• Reduce Variance and Bias with respect to DT	• Compuationally hard

Neural Networks

Results



Running Time [s]
3
2,5
2
1,5
1
0,5
0
Lin Regr
RANSAC
Tree
Forest
Categoria 4
All
Variance
SBS

ALL	LIN REG	RANSAC	TREE	FOREST	
MSE					
RMSE					
R2					