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

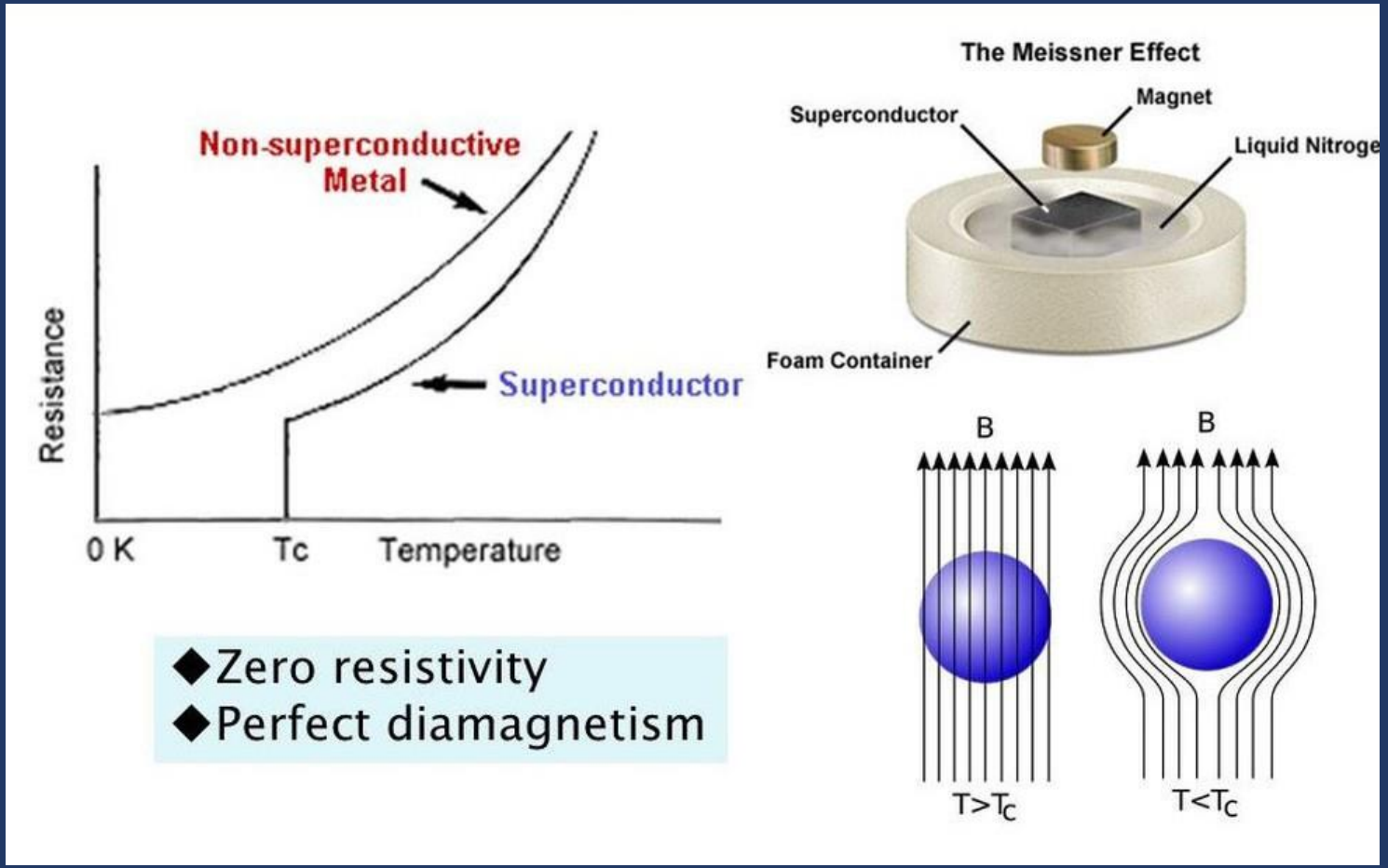
Machine Learning Approach for Prediction of Critical Temperature of Superconductor Materials

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 data saved in the file ''train.csv, which are the result calculations on the data of the file 'unique.csv'.

- **Continous Target Variable**

- **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

Content

- **Exploratory Data Analysis**
  - Missing and Categorical Data
  - Scatter Plot Matrix and Correlation Matrix
- **Data Preprocessing**
  - Feature Scaling
  - Dimensionality Reduction
  - Cross Validation
- **Evaluation and Tuning**
  - Scoring
  - Hyperparameter Tuning

Regression models

- Linear Regression
- Non-linear Regression
- Ensamble

- **Conclusions**
  - Results
  - Running Time

Exploratory Data Analysis

- No Missing Data
- No Categorical Data

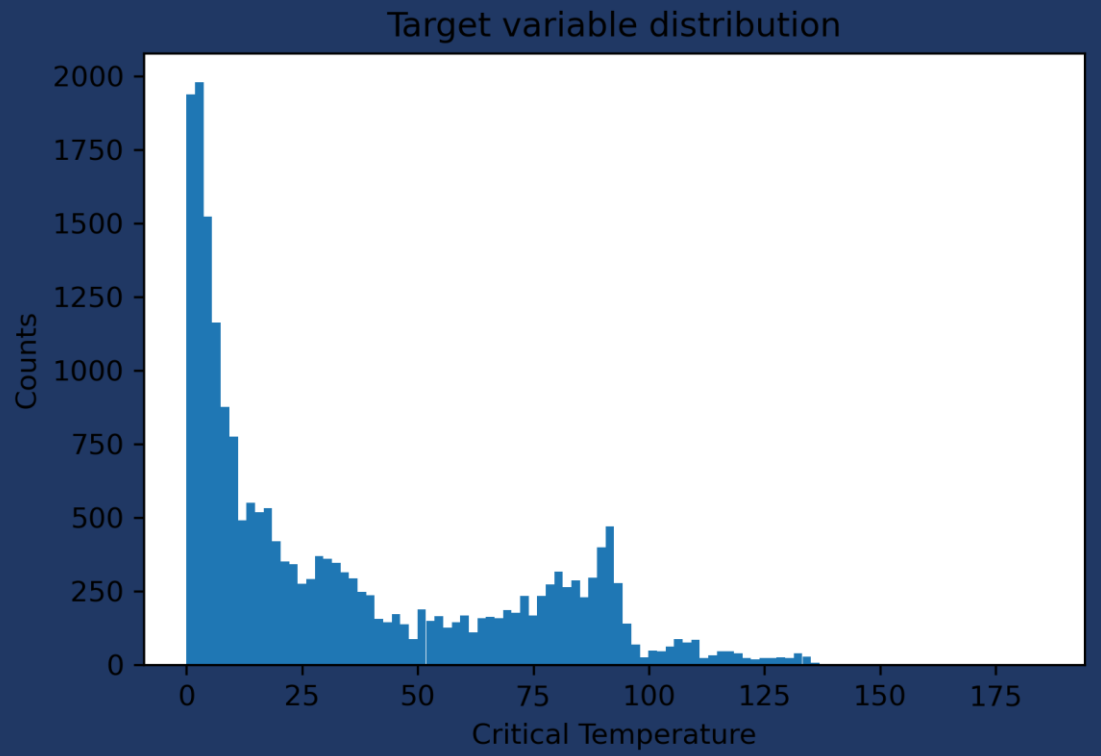
Stats	Formula
MEAN	$\mu = \sum_{i=1}^{n} \frac{t_i}{i}$
WEIGHTED MEAN	$\nu = \sqrt{\sum_{i=1}^{n} p_i t_i}$
GEOMETRIC MEAN	$= \prod_{i}^{n} t_i^{\frac{1}{n}}$
WEIGHTED GEOMETRIC MEAN	$= \prod_{i}^{n} t_i^{p_i}$

Stats	Formula
ENTROPY	$= -\sum_{i=1}^{n} w_i \log(w_i)$
WEIGHTED ENTROPY	$= -\sum_{i=1}^{n} A_i \log(A_i)$
RANGE	$t_{max} - t_{min}$
WEIGHTED RANGE	$p(t_{max})t_{max} - p(t_{min})t_{min}$
STANDARD DEVIATION	$\sqrt{\frac{1}{2}\sum_{i}^{n}(t_i - \mu)^2}$
WEIGHTED STANDARD DEVIATION	$\sqrt{p_i \sum_{i}^{n}(t_i - \mu)^2}$

Exploratory Data Analysis

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

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



Correlation Matrices

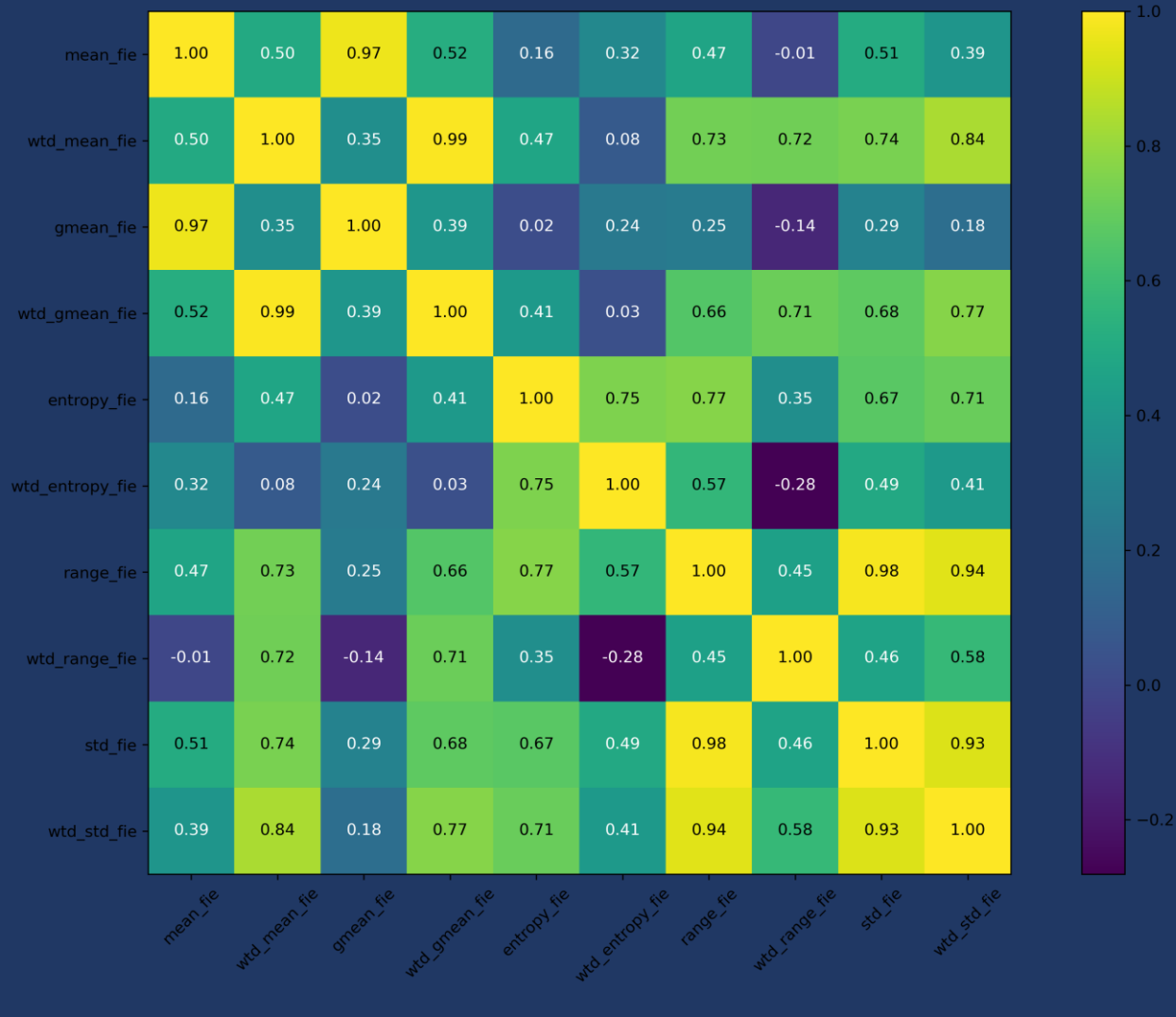
Pearson Correlation Coefficients:

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

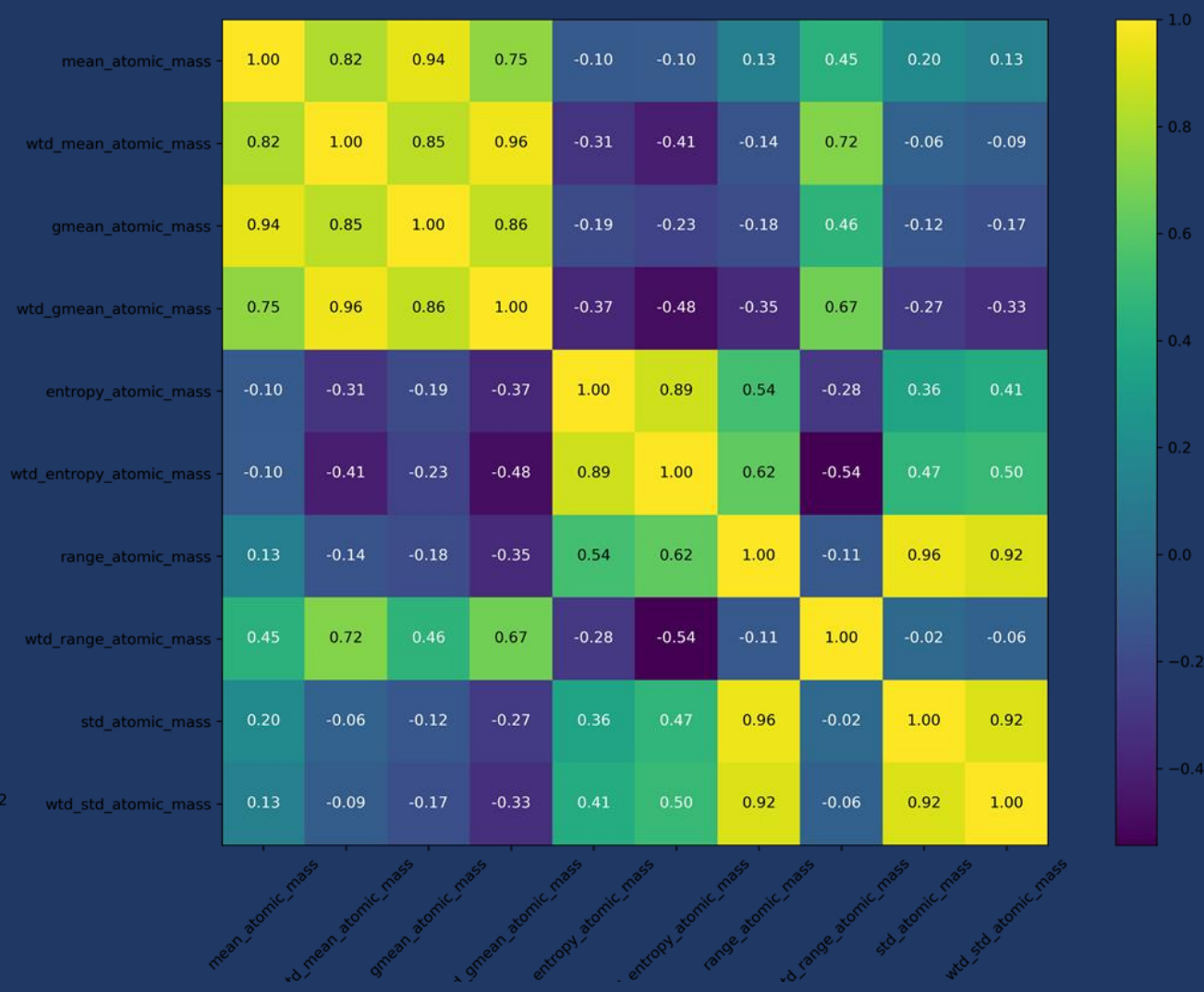
- It measures the linear dependence between pairs of features
- $-1 \leq r \leq 1$
- $r = 0$: no correlation
- $r = 1$: positive correlation
- $r = -1$: negative correlation

Relationship between the statistics of the same physical property

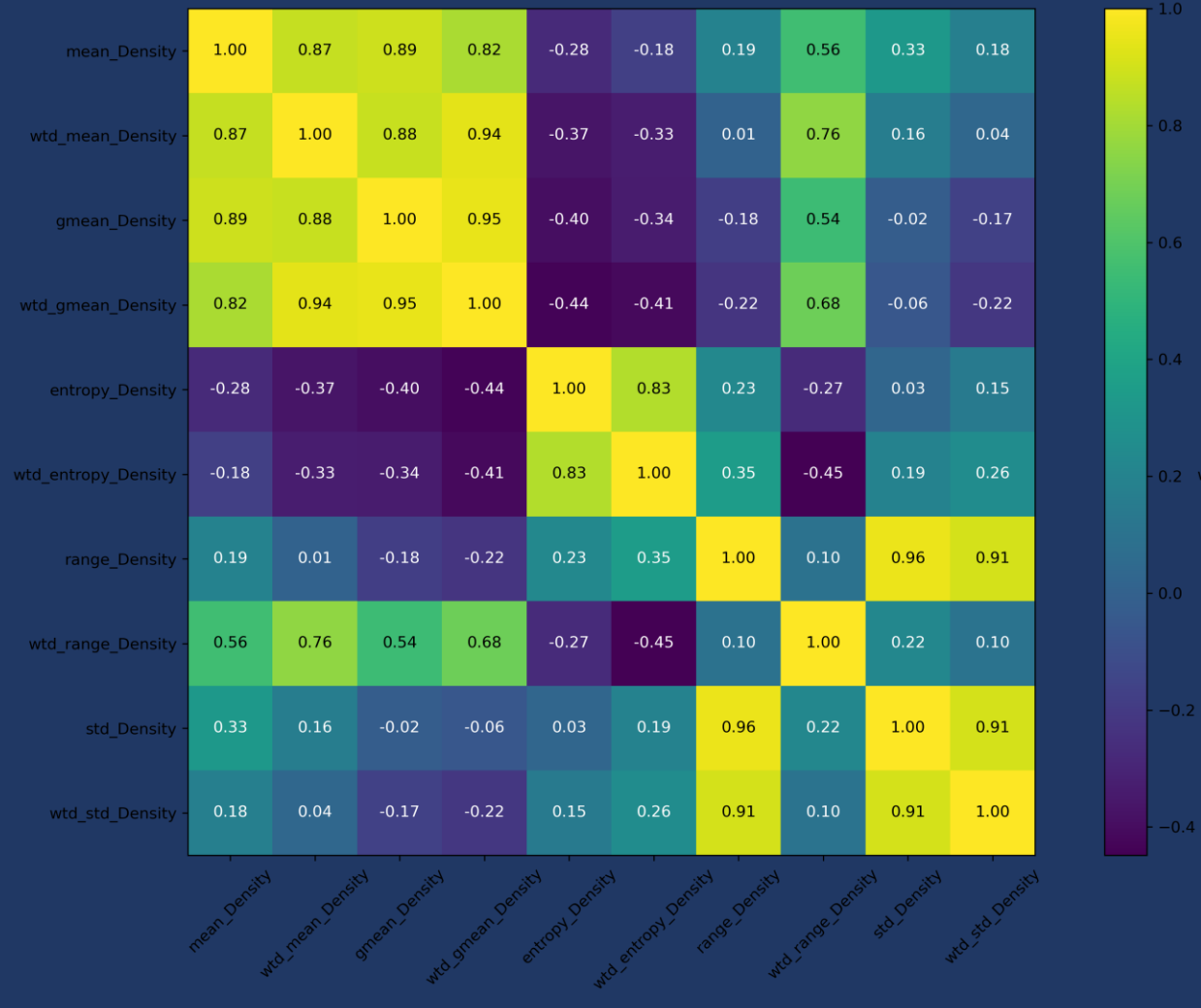
First Ionization Energy



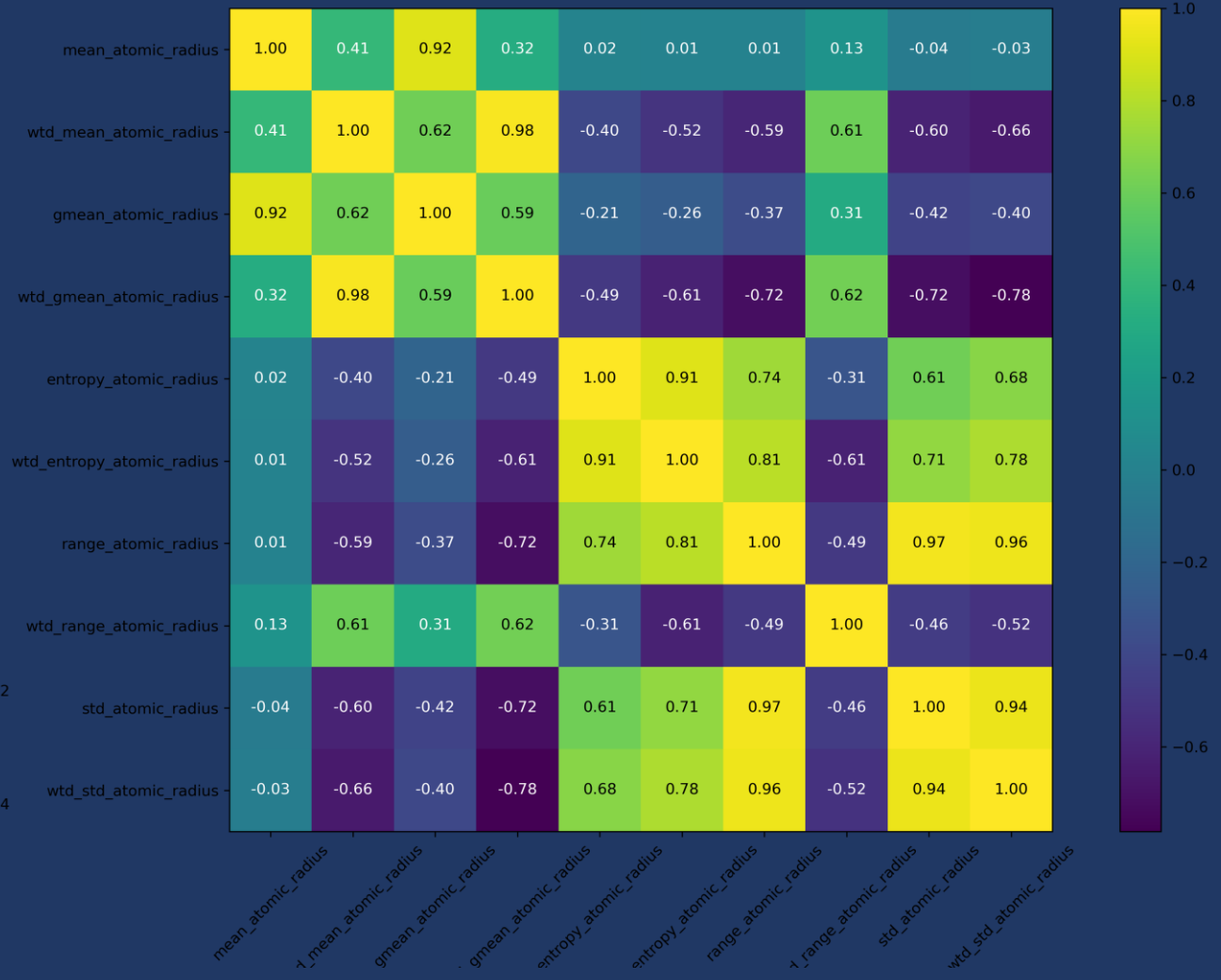
Atomic Mass



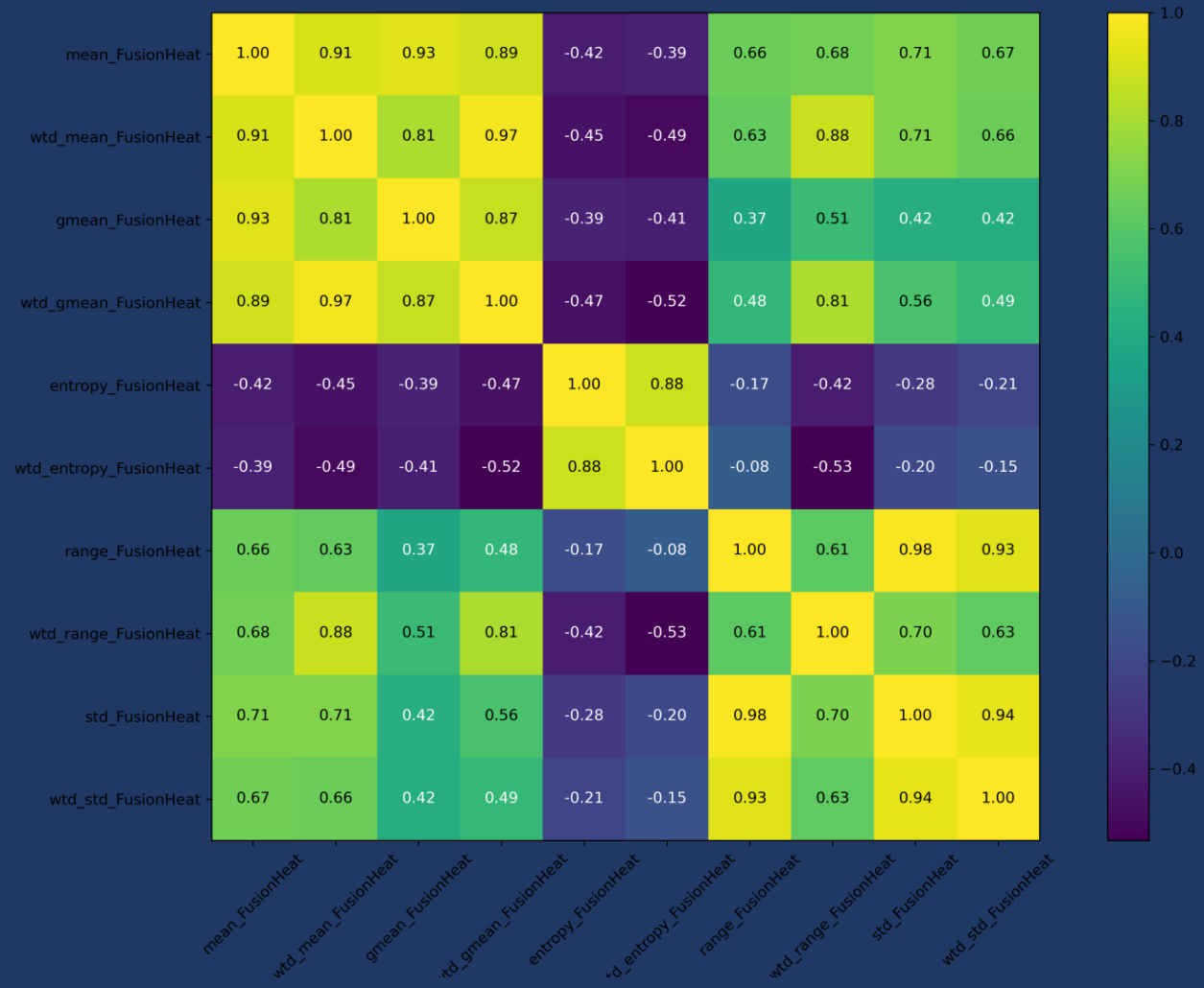
Density



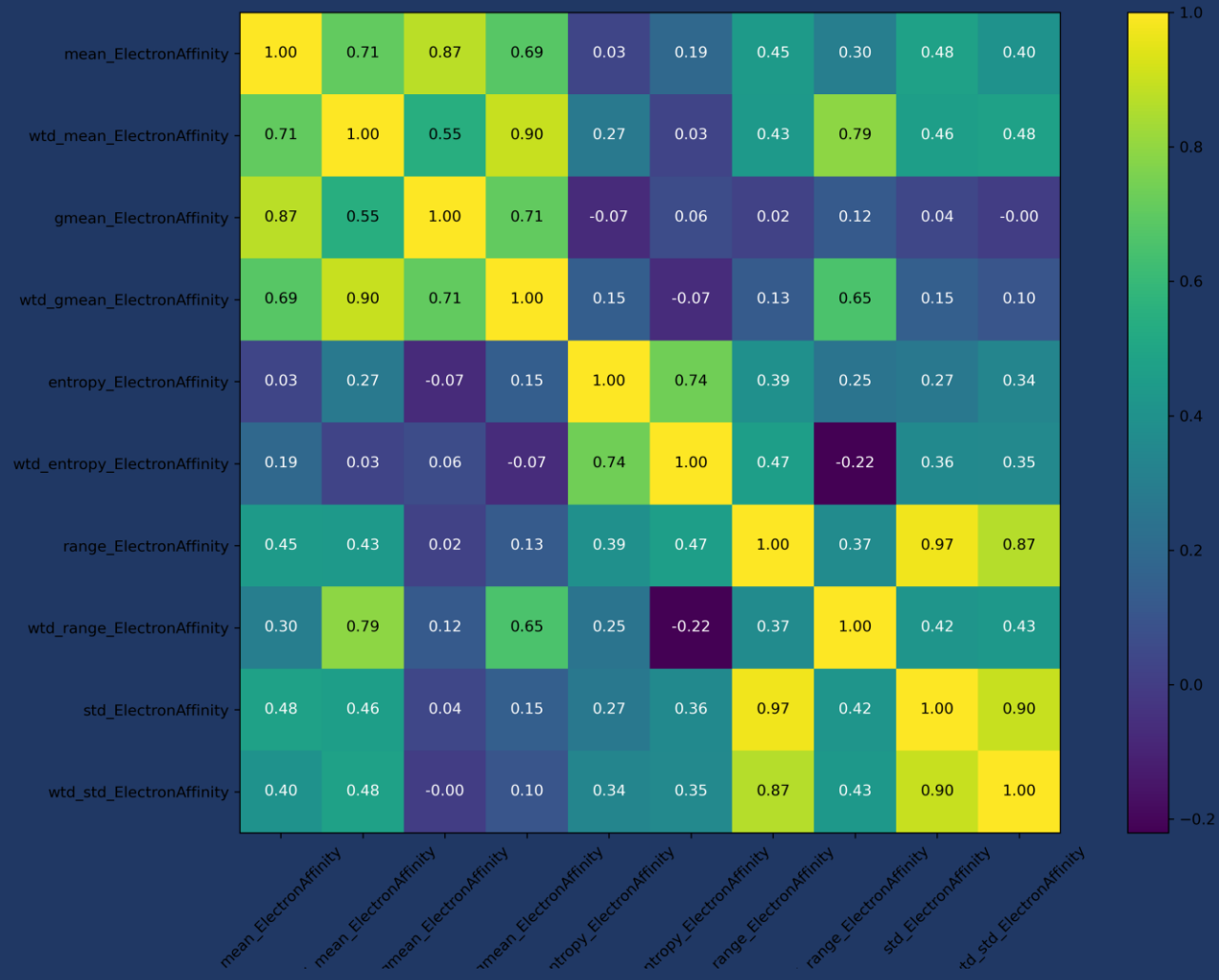
Atomic Radius



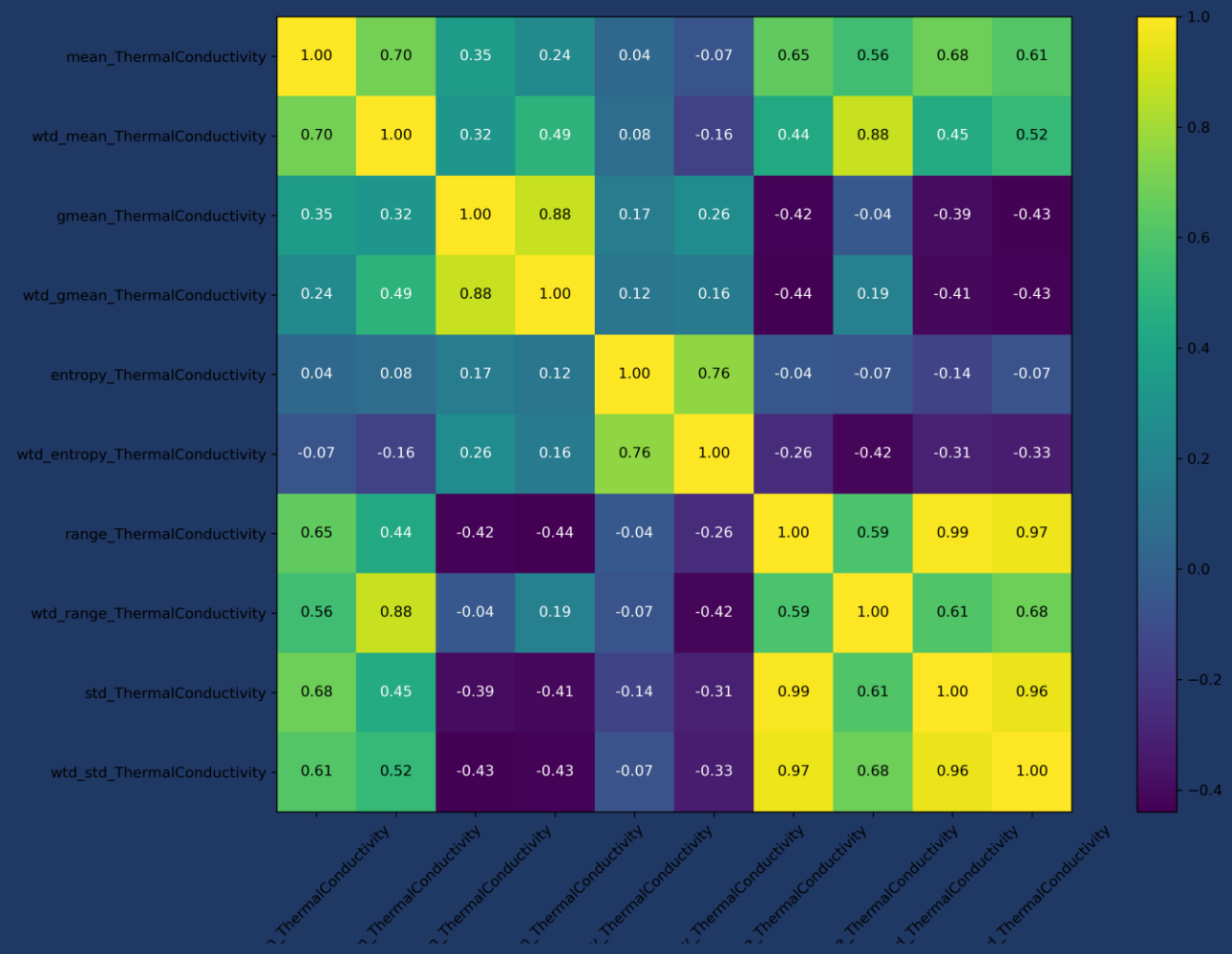
Fusion Heat



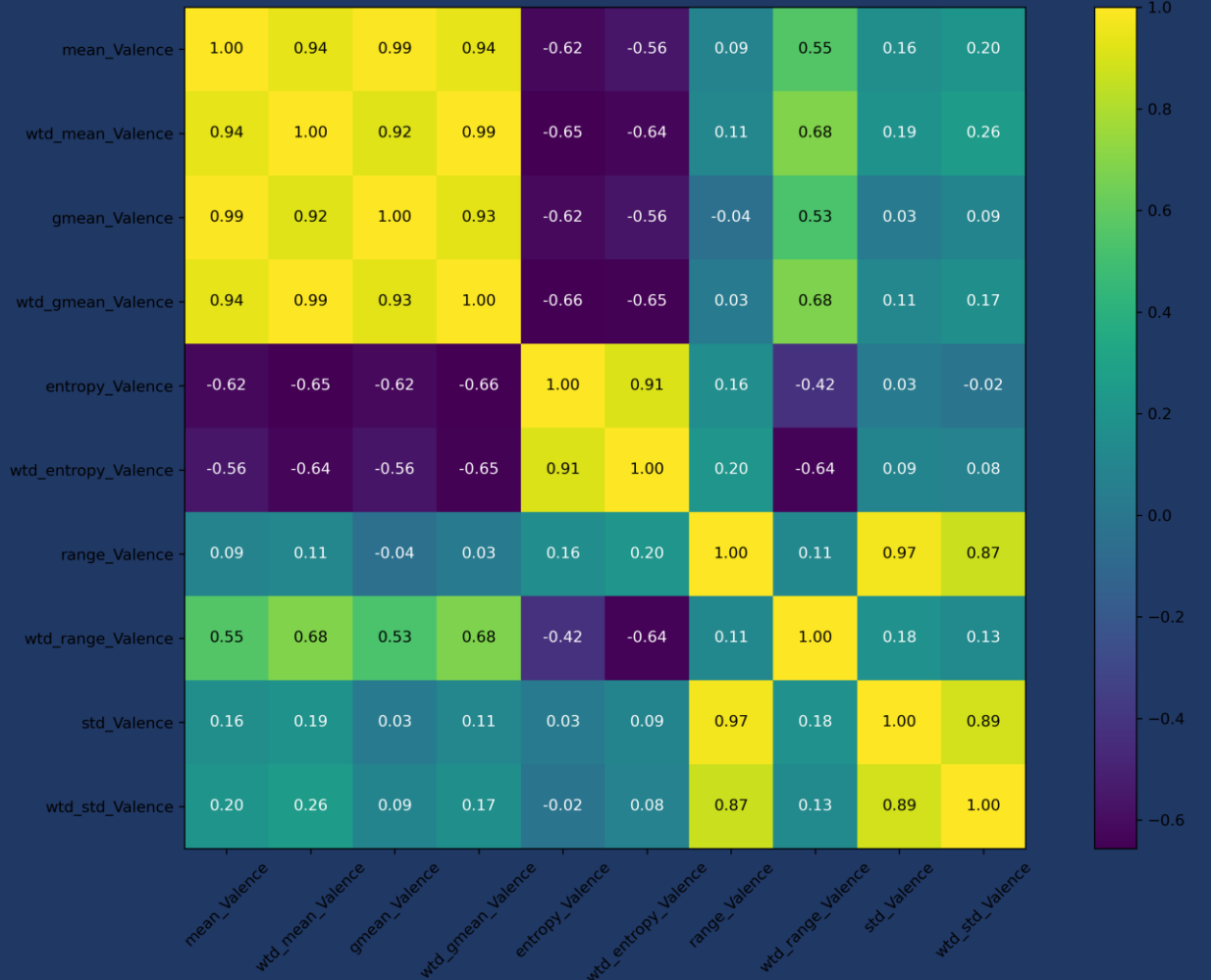
Electron Affinity



Thermal Conductivity



Valence



Scatter plot Matrix

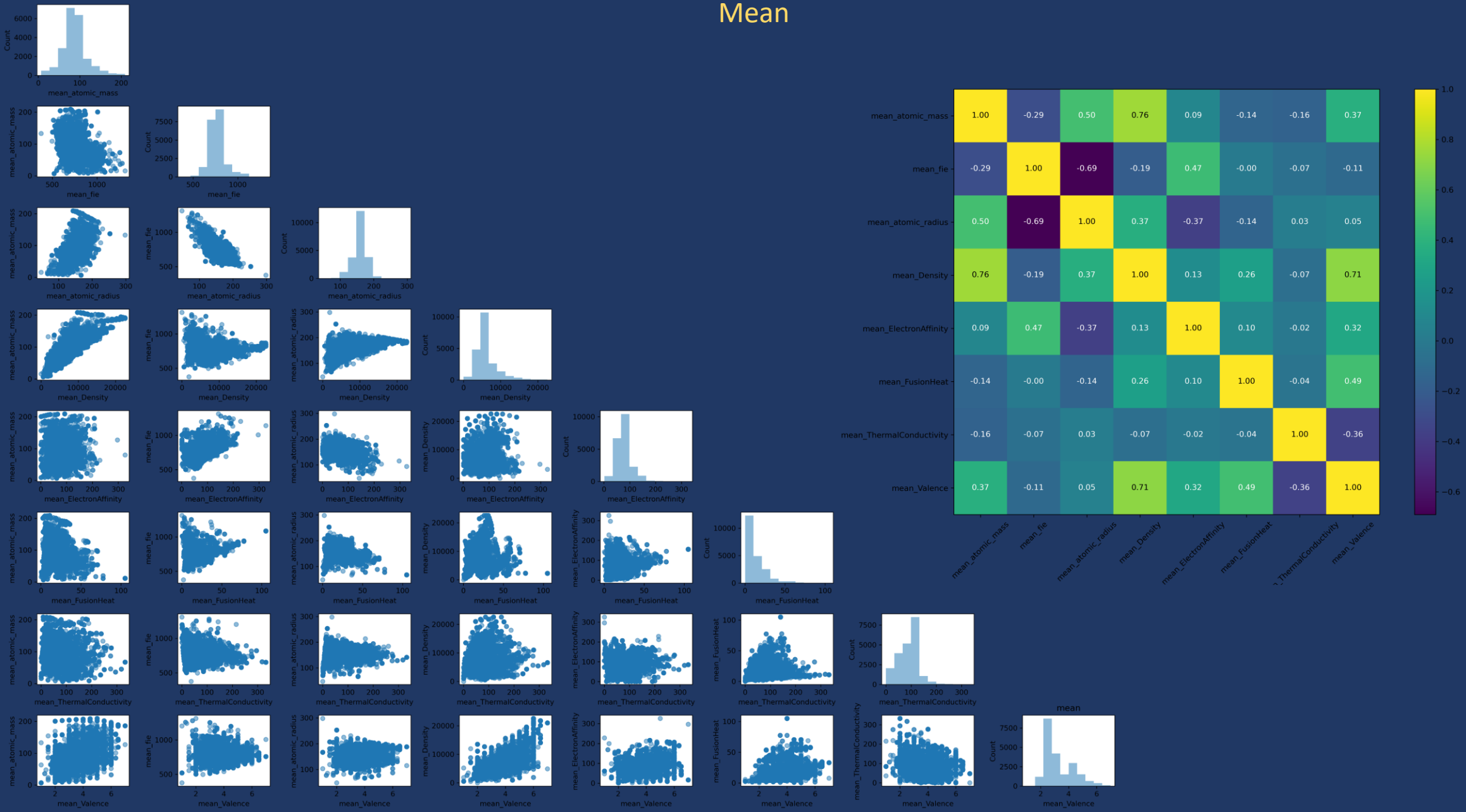
- Given a set of variables $x_1, x_2, \dots, x_n$, it contains all the pairwise scatter plots of the variables on a single page in a matrix format. The i-th row and j-th column of this matrix is a plot of x_i vs $\mathrm{x_j}$.

- Since x_i vs $\mathrm{x_j}$ is equivalent to x_j vs x_i with the axes reversed, we prefer to omit the plots above the diagonal.

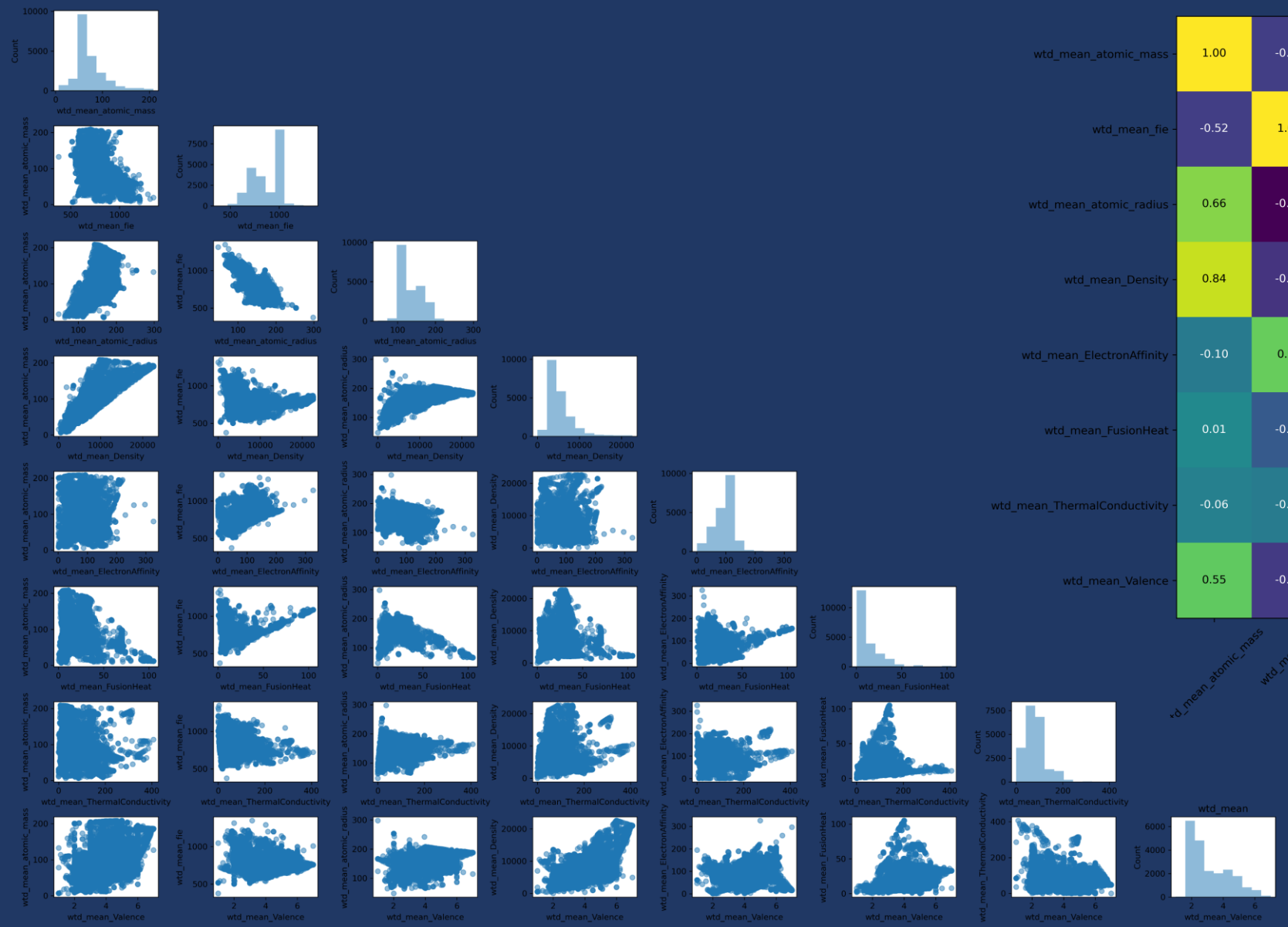
- On the diagonal, instead of graphing a trivial relationship, we plot the univariate histogram.

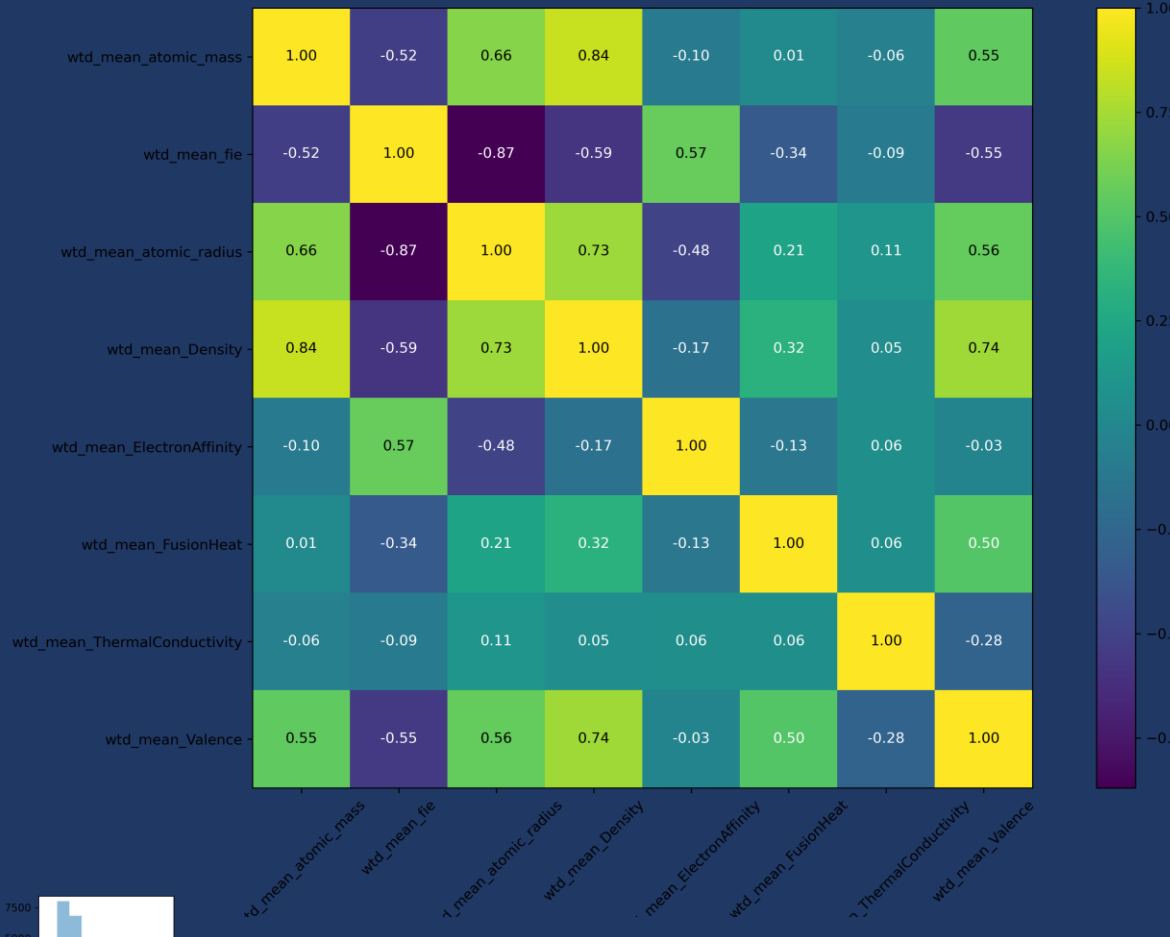
Now we will explore the relationship between feature that rapresents different physical properties, fixing a single statistic.

Mean

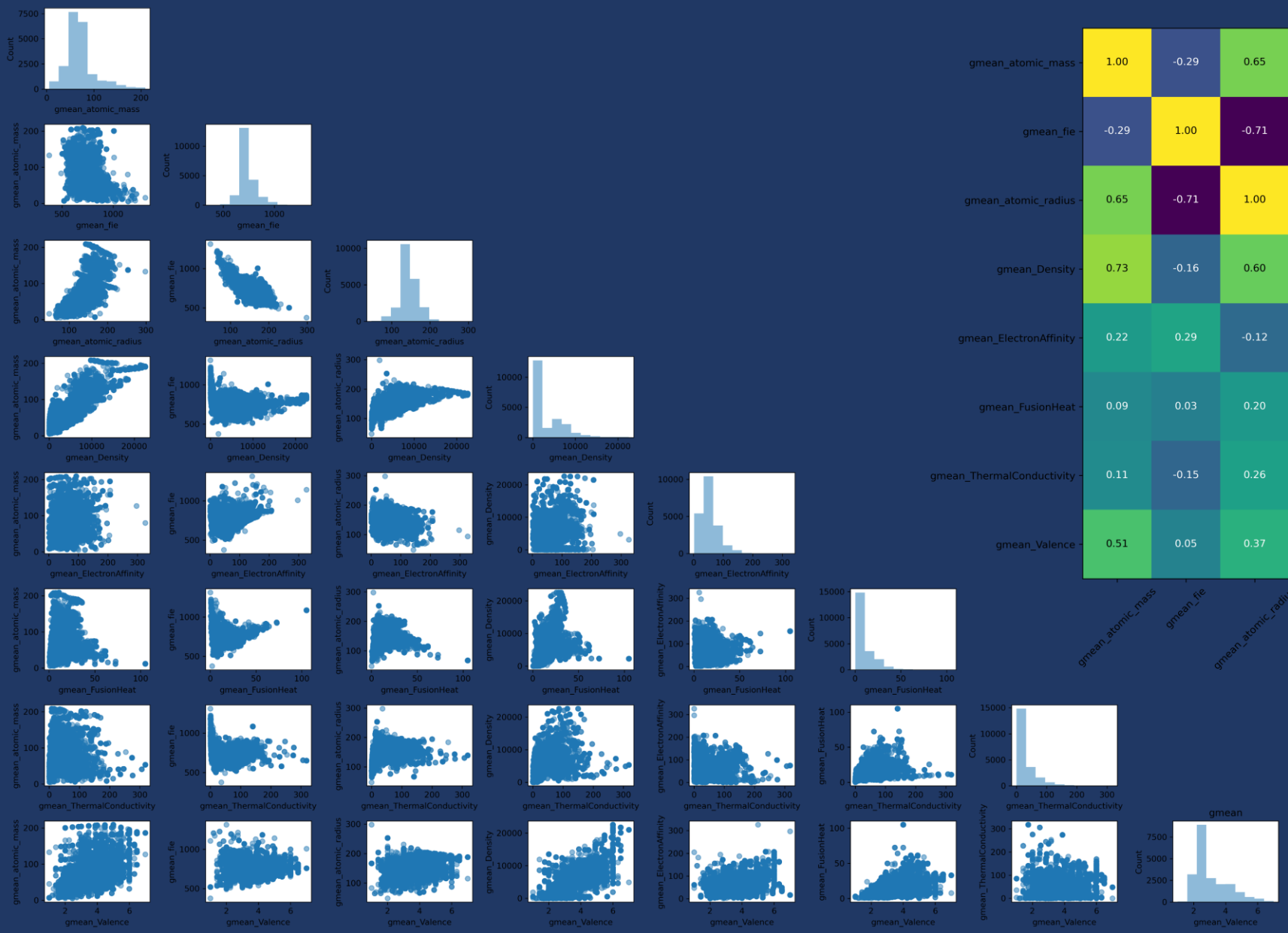

mean_atomic_mass
mean_fie
mean_atomic_radius
mean_Density
mean_ElectronAffinity
mean_FusionHeat
mean_ThermalConductivity
mean_Valence
Count
mean
1.00 -0.29 0.50 0.76 0.09 -0.14 -0.16 0.37
-0.29 1.00 -0.69 -0.19 0.47 -0.00 -0.07 -0.11
0.50 -0.69 1.00 0.37 -0.37 -0.14 0.03 0.05
0.76 -0.19 0.37 1.00 0.13 0.26 -0.07 0.71
0.09 0.47 -0.37 0.13 1.00 0.10 -0.02 0.32
-0.14 -0.00 -0.14 0.26 0.10 1.00 -0.04 0.49
-0.16 -0.07 0.03 -0.07 -0.02 -0.04 1.00 -0.36
0.37 -0.11 0.05 0.71 0.32 0.49 -0.36 1.00

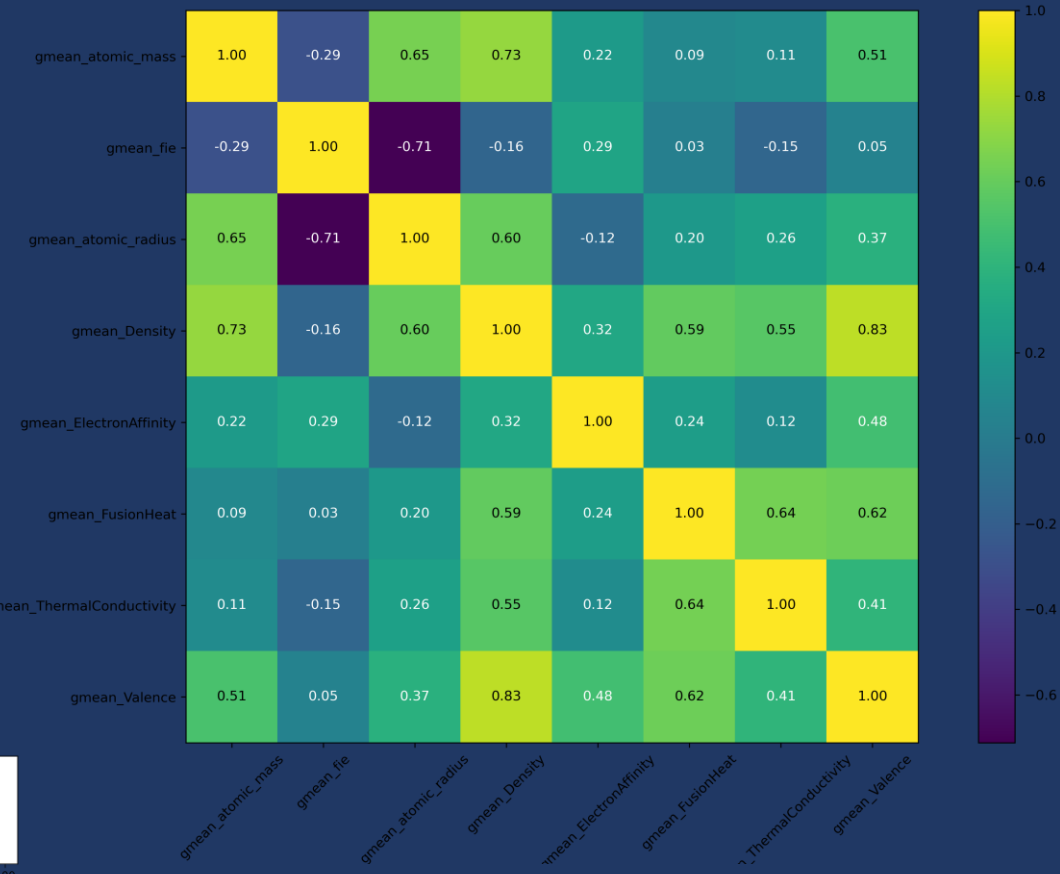
Weighted Mean



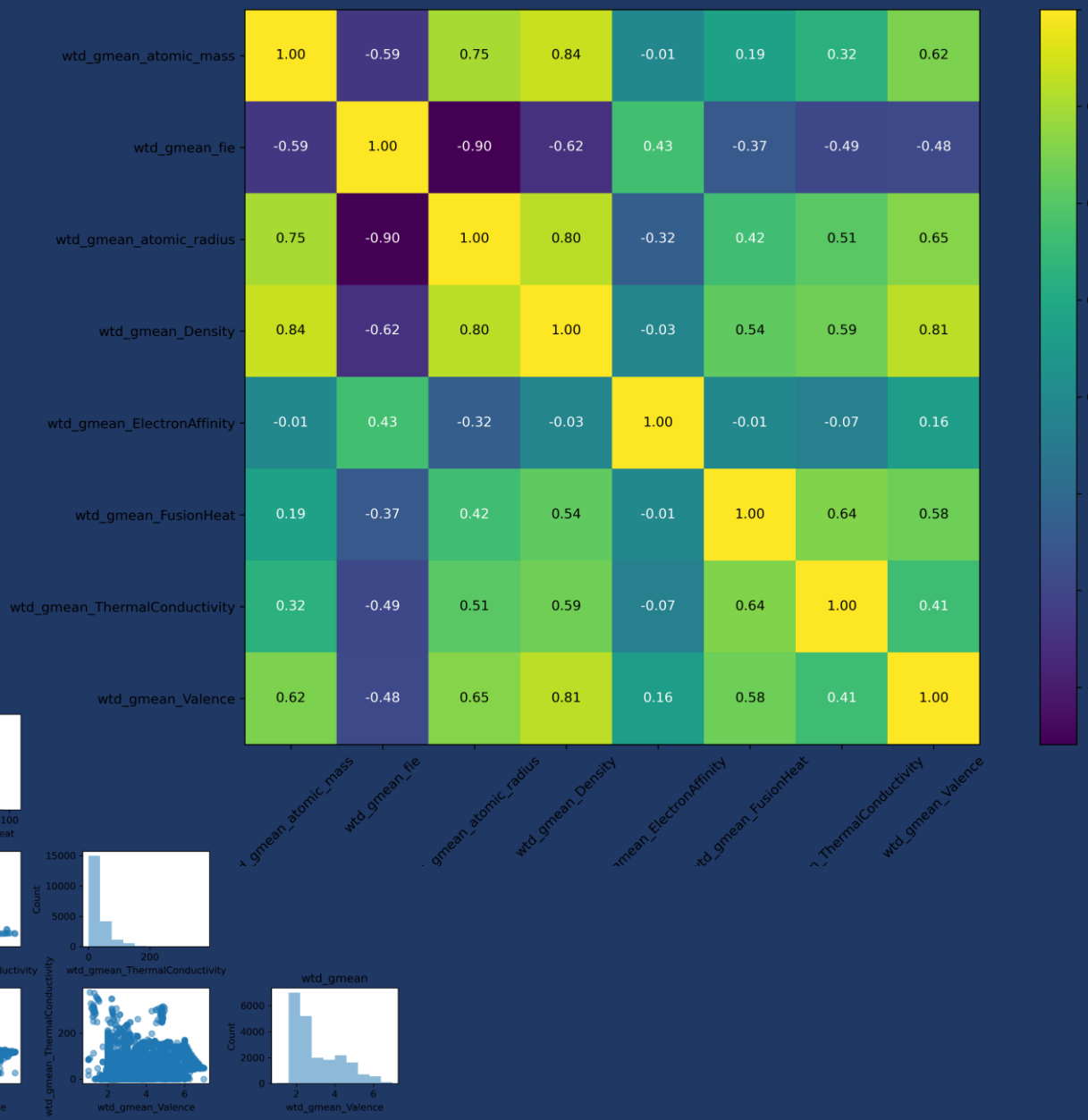


Geometric Mean

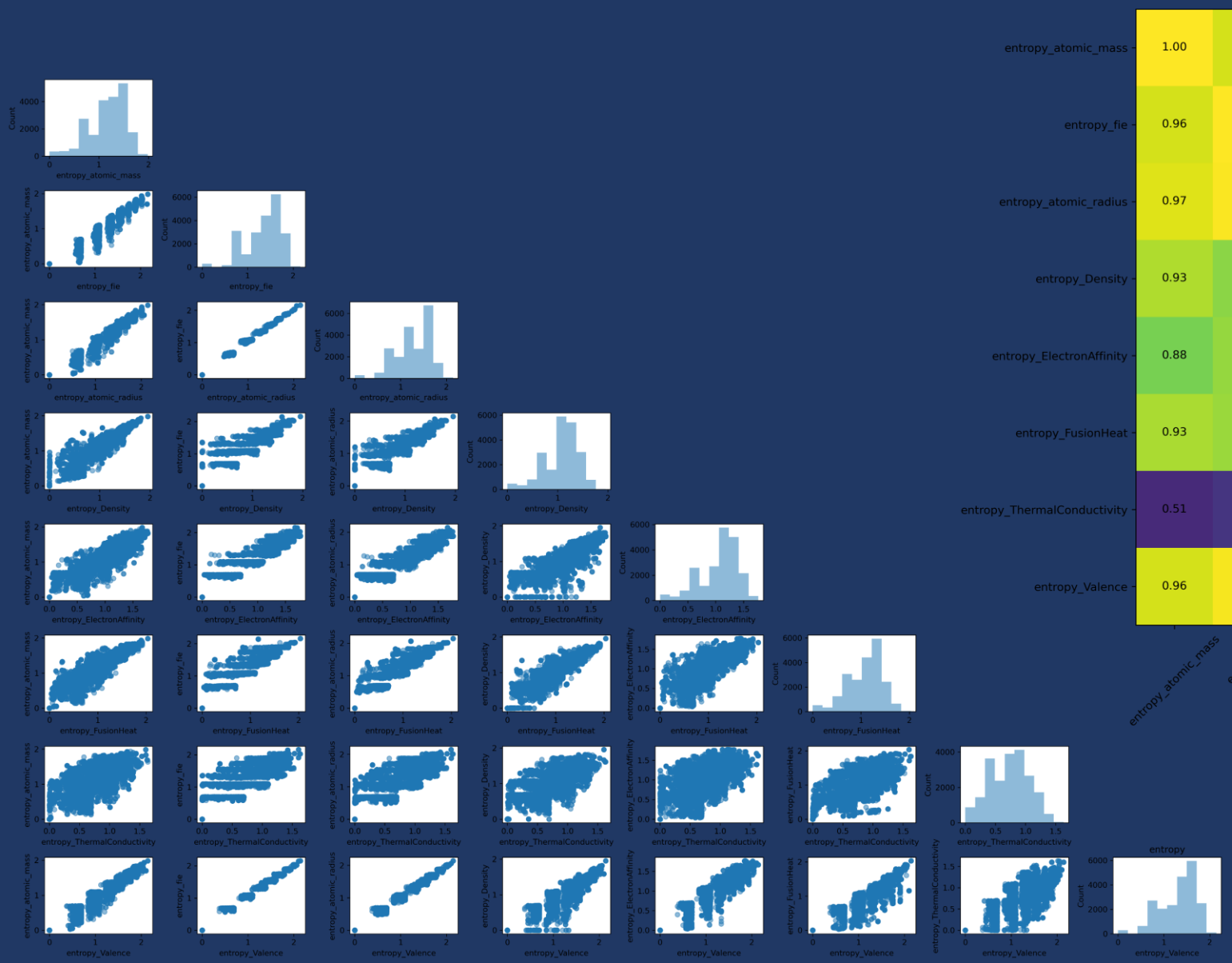


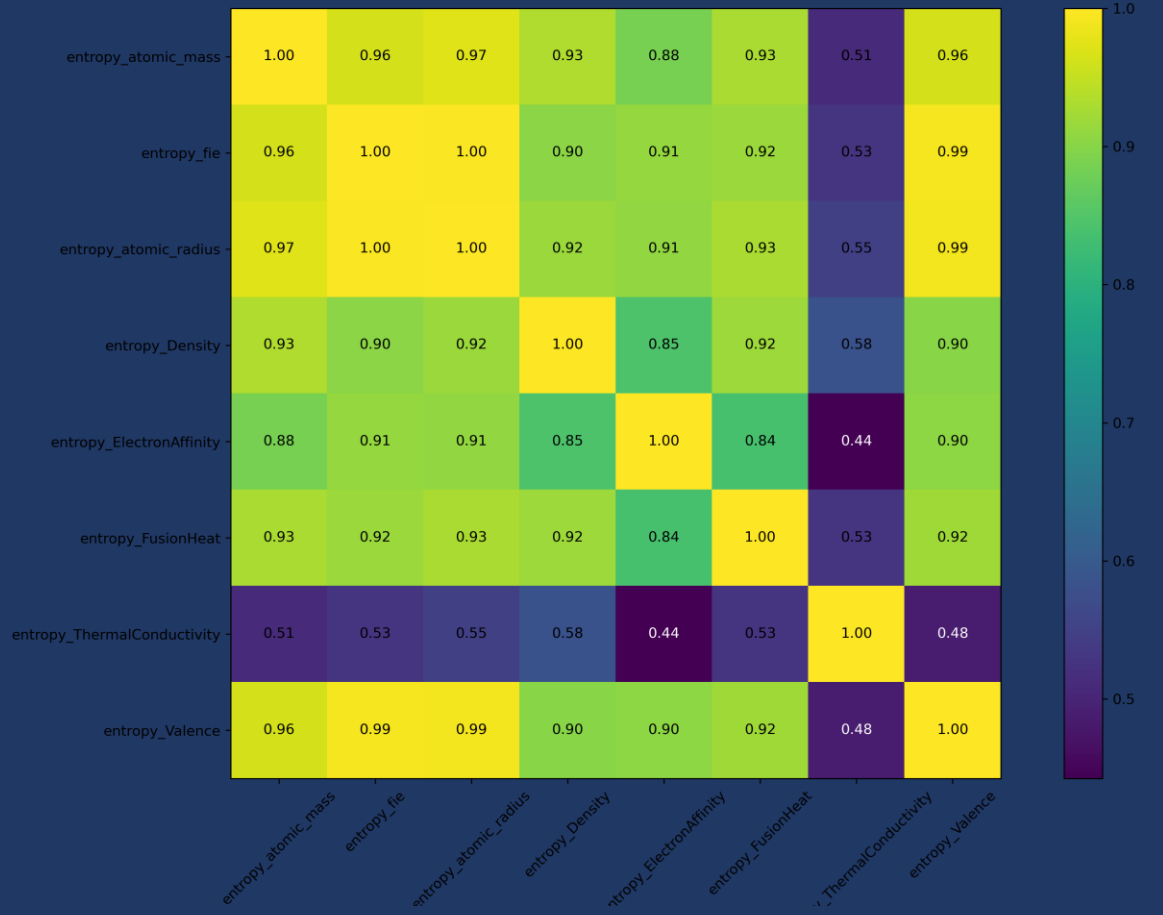


Weighted Geometric Mean

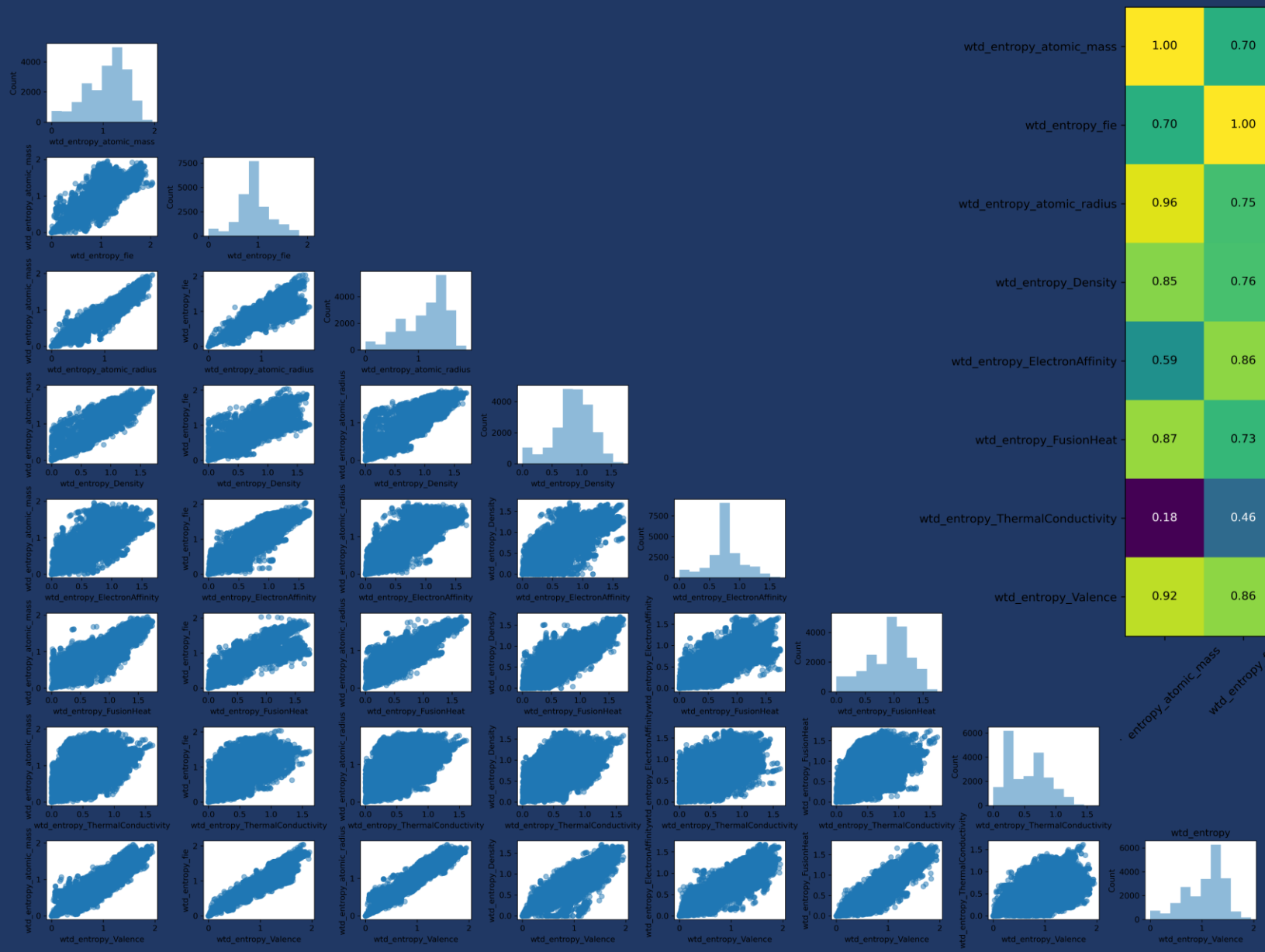


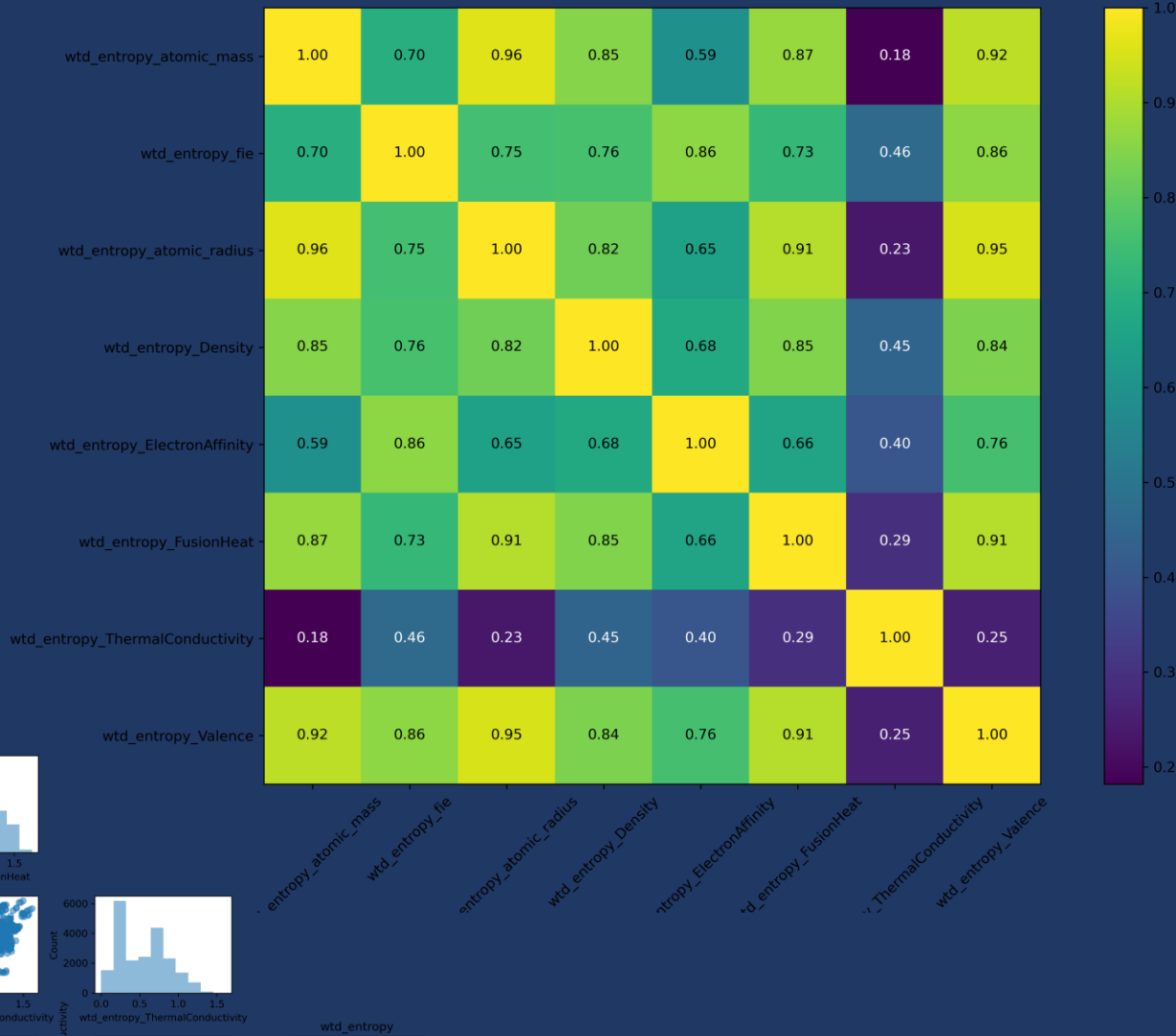
Entropy


entropy_atomic_mass
entropy_fie
entropy_atomic_radius
entropy_Density
entropy_ElectronAffinity
entropy_FusionHeat
entropy_ThermalConductivity
entropy_Valence
Count

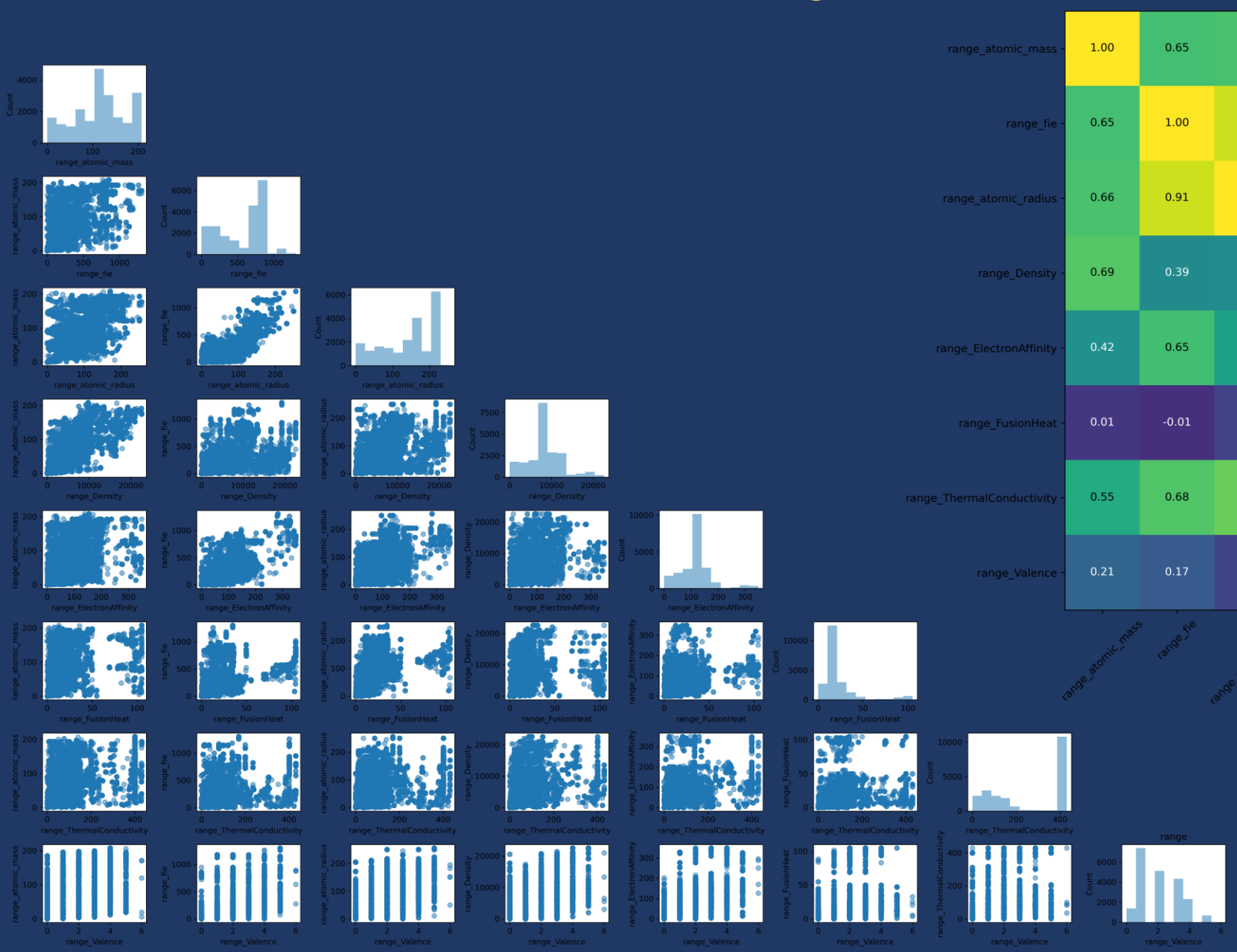

entropy_atomic_mass 1.00 0.96 0.97 0.93 0.88 0.93 0.51 0.96
entropy_fie 0.96 1.00 1.00 0.90 0.91 0.92 0.53 0.99
entropy_atomic_radius 0.97 1.00 1.00 0.92 0.91 0.93 0.55 0.99
entropy_Density 0.93 0.90 0.92 1.00 0.85 0.92 0.58 0.90
entropy_ElectronAffinity 0.88 0.91 0.91 0.85 1.00 0.84 0.44 0.90
entropy_FusionHeat 0.93 0.92 0.93 0.92 0.84 1.00 0.53 0.92
entropy_ThermalConductivity 0.51 0.53 0.55 0.58 0.44 0.53 1.00 0.48
entropy_Valence 0.96 0.99 0.99 0.90 0.90 0.92 0.48 1.00
1.0
0.9
0.8
0.7
0.6
0.5

Weighted Entropy



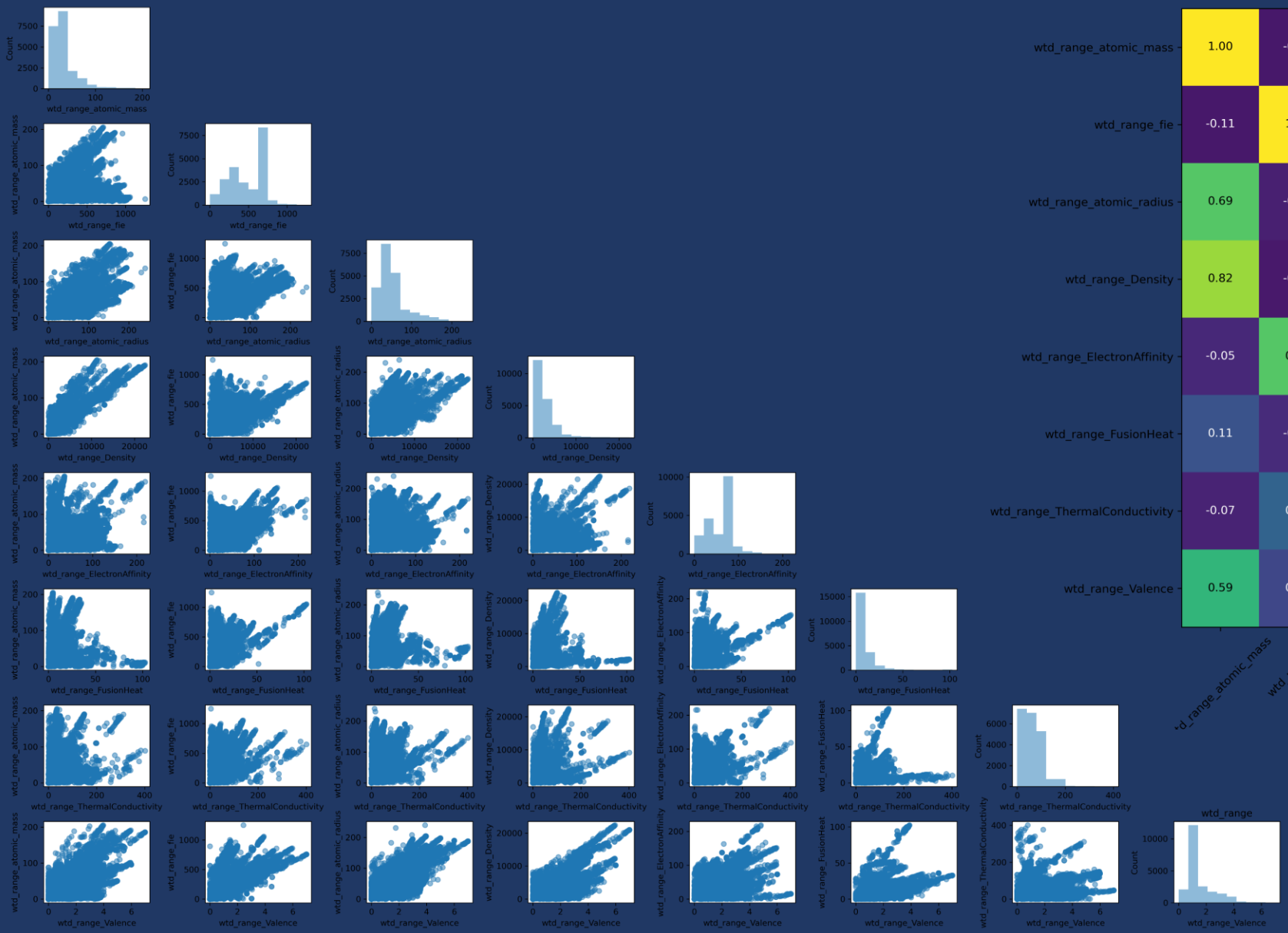


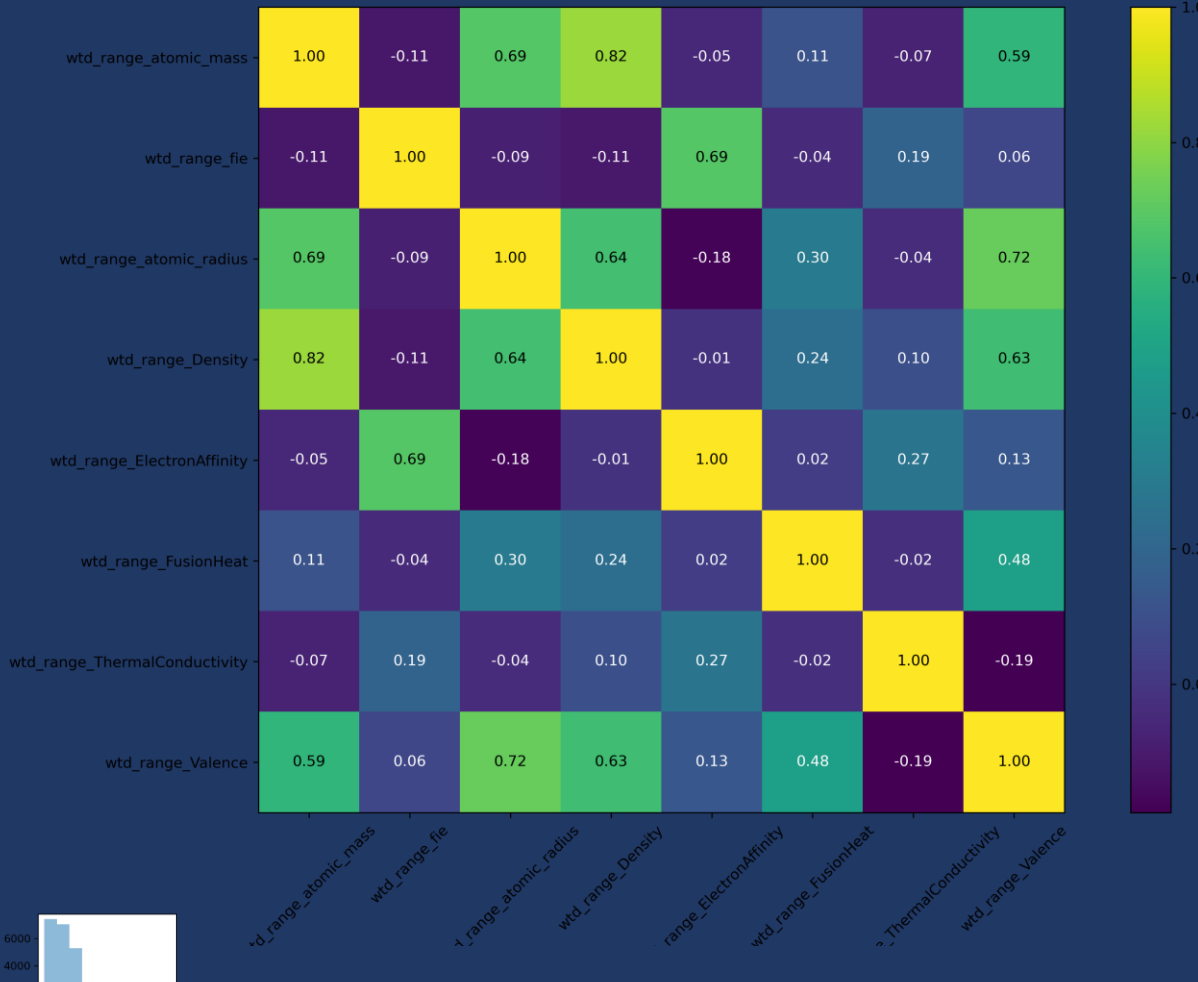
Range


range_atomic_mass
range_fie
range_atomic_radius
range_Density
range_ElectronAffinity
range_FusionHeat
range_ThermalConductivity
range_Valence
Count

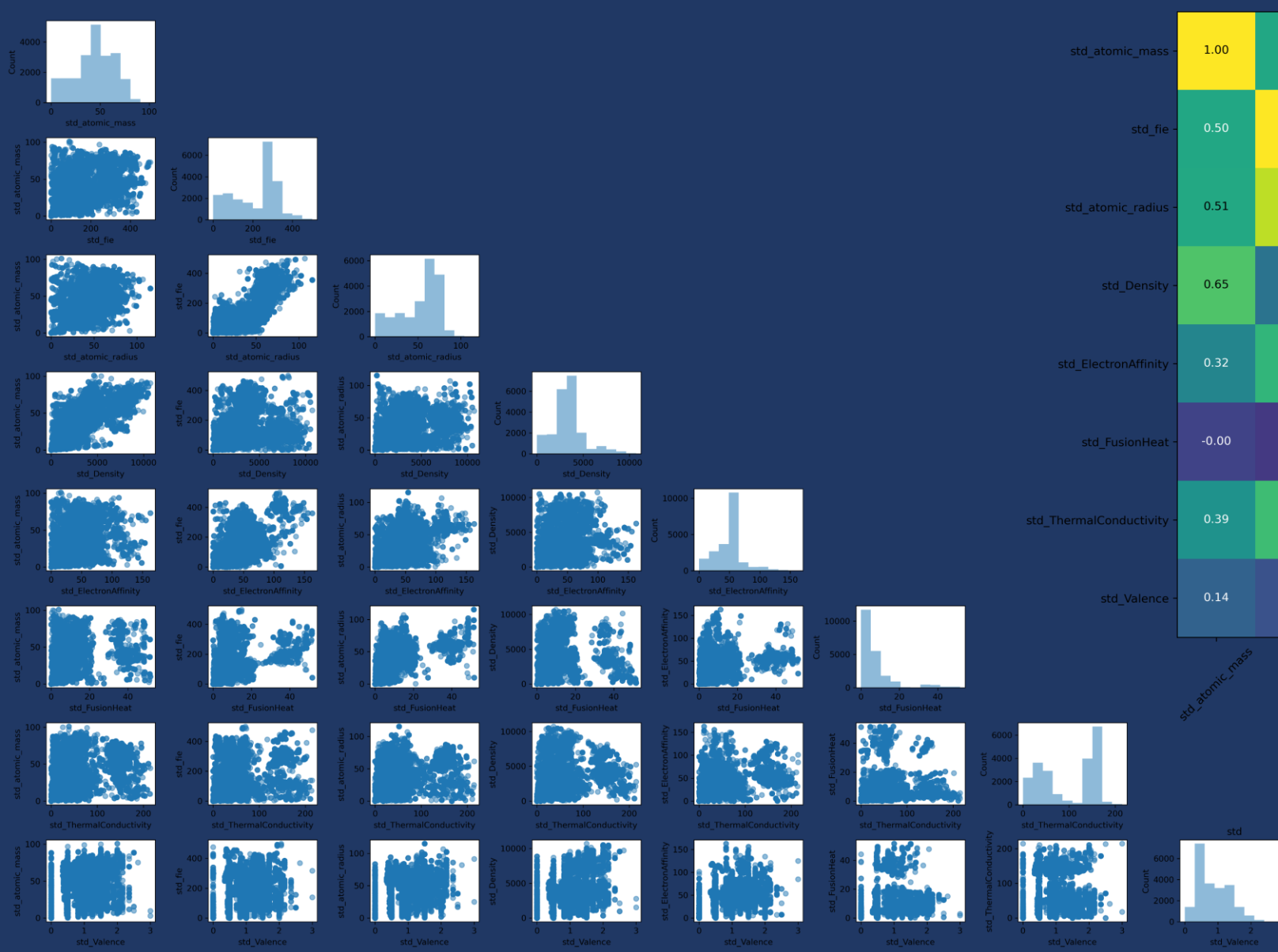

range_atomic_mass 1.00 0.65 0.66 0.69 0.42 0.01 0.55 0.21
range_fie 0.65 1.00 0.91 0.39 0.65 -0.01 0.68 0.17
range_atomic_radius 0.66 0.91 1.00 0.40 0.49 0.06 0.74 0.07
range_Density 0.69 0.39 0.40 1.00 0.34 0.02 0.35 0.27
range_ElectronAffinity 0.42 0.65 0.49 0.34 1.00 0.12 0.32 0.37
range_FusionHeat 0.01 -0.01 0.06 0.02 0.12 1.00 -0.17 0.19
range_ThermalConductivity 0.55 0.68 0.74 0.35 0.32 -0.17 1.00 -0.14
range_Valence 0.21 0.17 0.07 0.27 0.37 0.19 -0.14 1.00
1.0
0.8
0.6
0.4
0.2
0.0

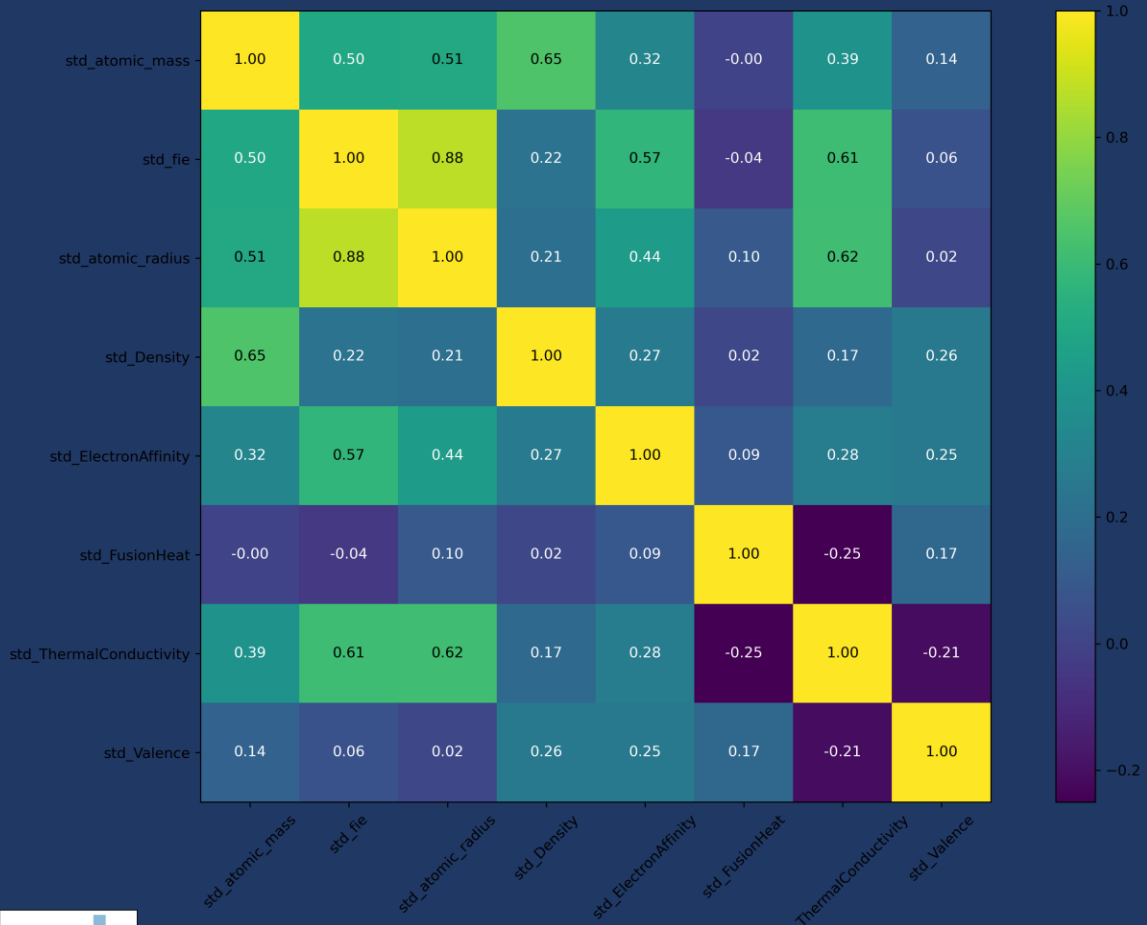
Weighted Range



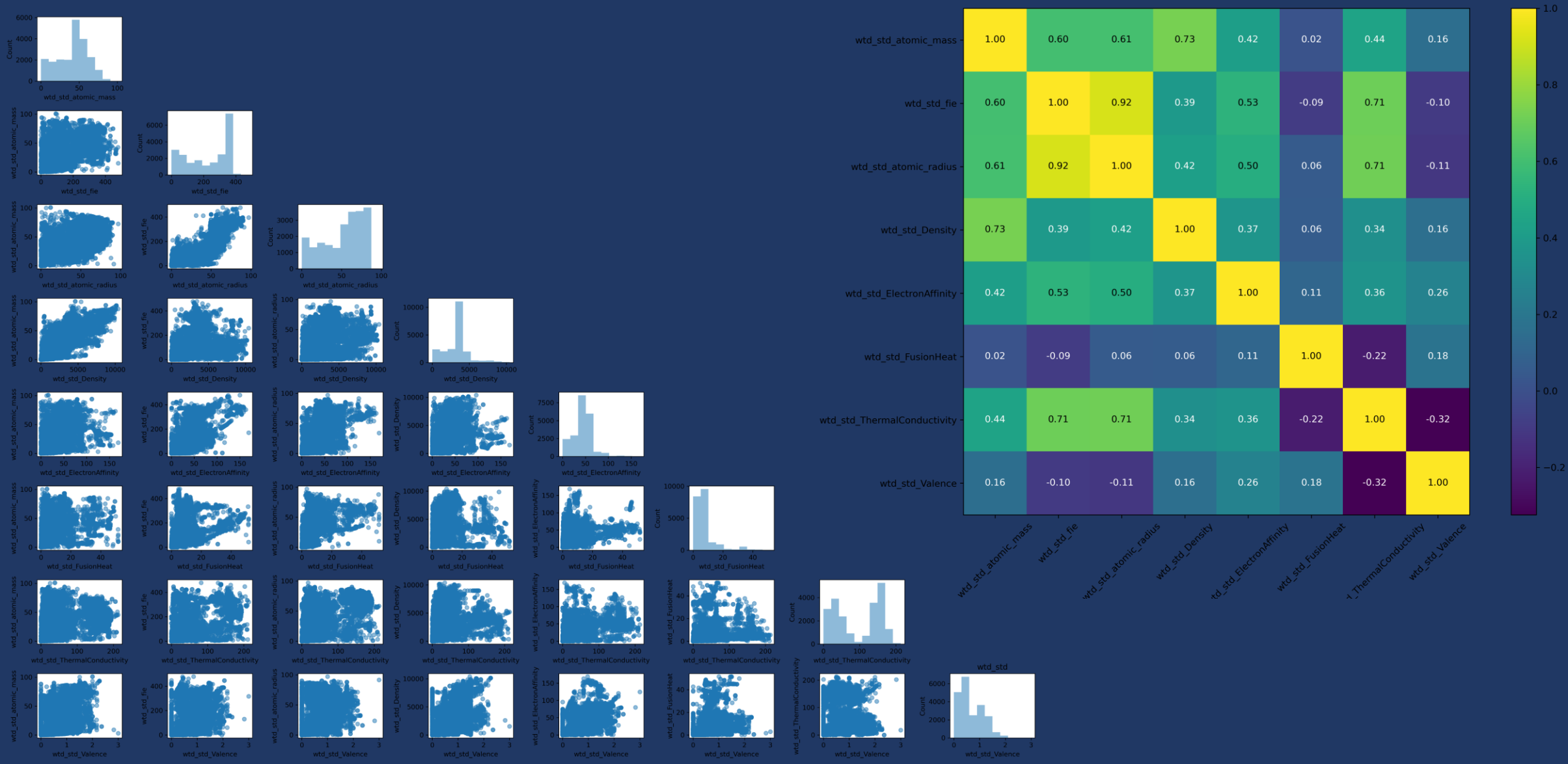


Standard Deviation





Weighted Standard Deviation



Feature Scaling

PROBLEMS

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

SOLUTIONS

- Feature Standardization:

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

- Feature Normalization:

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

Dimensionality Reduction

PROBLEMS

- Multicolliniarity
- Overfitting
- High Computational Cost

SOLUTIONS

- Feature selection
- Feature extraction

Feature Selection

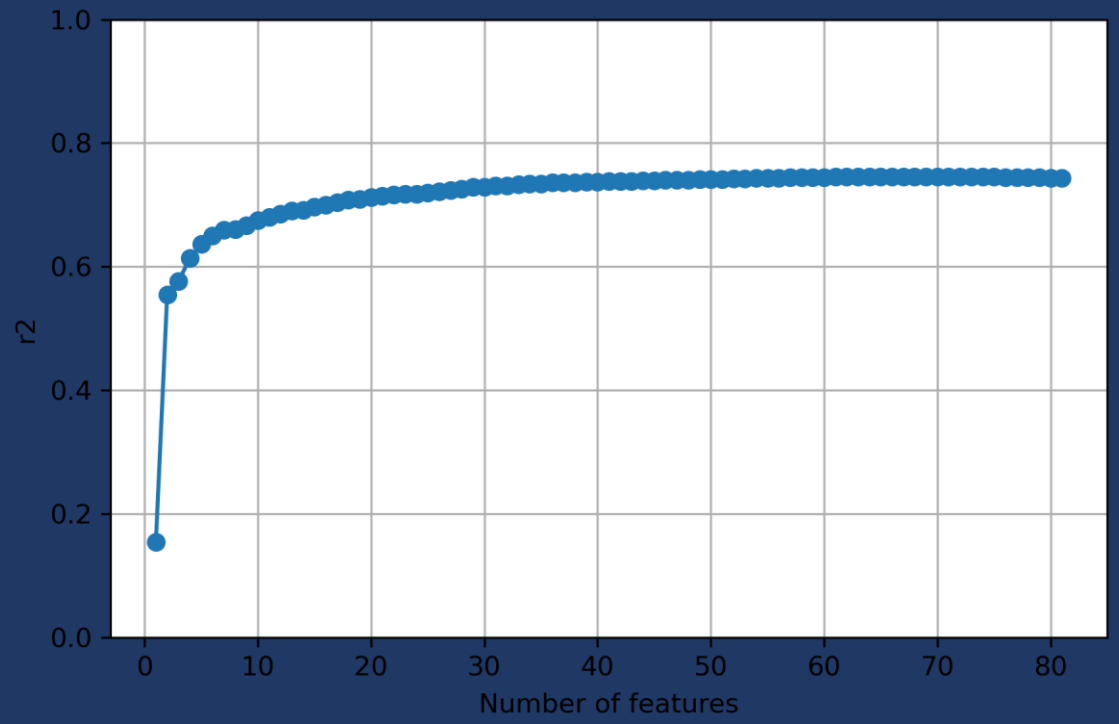
- Variance Threshold Selection
  - It removes all features whose variance is lower than a certain threshold.

$$\sigma^2 = \sum_{i=0}^{n-1} \frac{(x_i - \mu)^2}{n}$$

- Sequential Backward Selection
  - Gridy Search algorithm
  - Useful for algorithms that don’t support regularization
- Random Forest
  - Feature importances

Feature Selection on Superconductors Dataset

SBS on X_train_std



30 features selected

Dropping low variance features

Variance Threshold	Feature selected
10	55

Feature Extraction

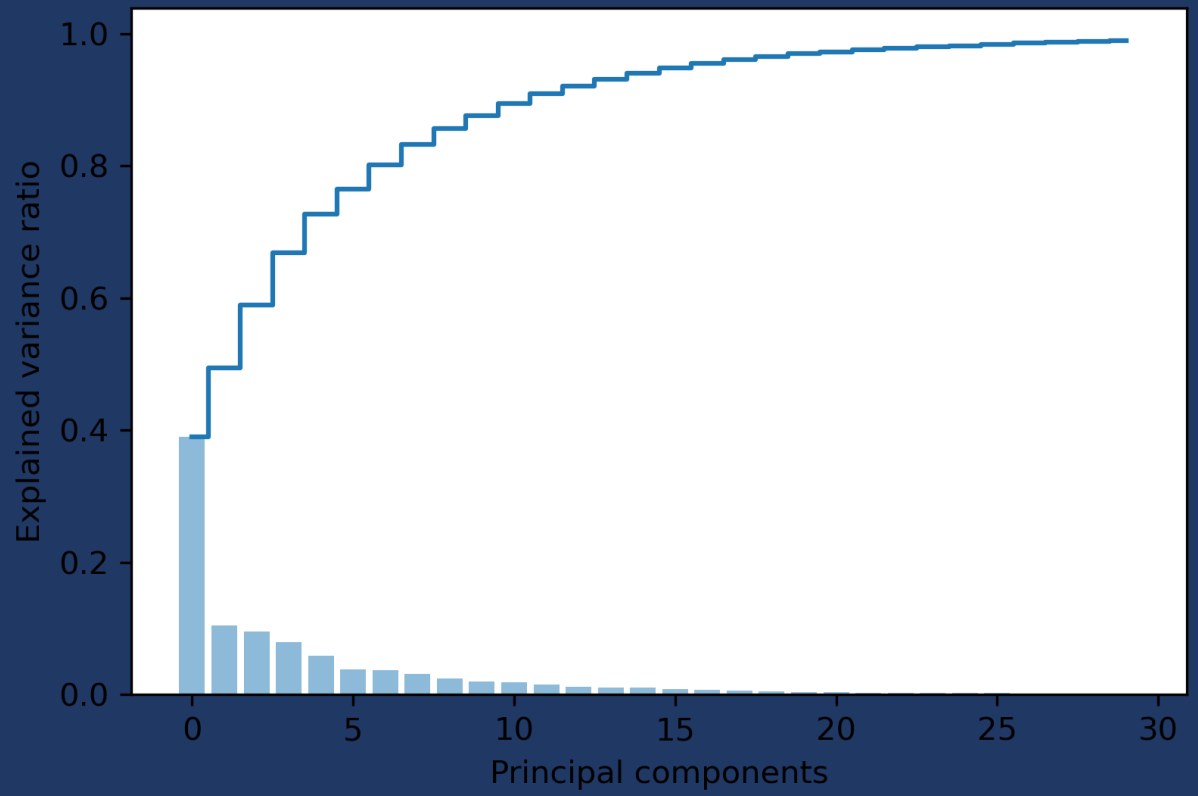
- Feature extraction transform or project the data onto a new feature space of lower dimensionality than the original one.
- Using feature selection algorithms we maintain the original features, but on the contrary using feature selection we create new features.
- The best-known example is the Principal Component Analysis algorithm.

Principal Component Analysis

PCA can be summarised in the following steps:

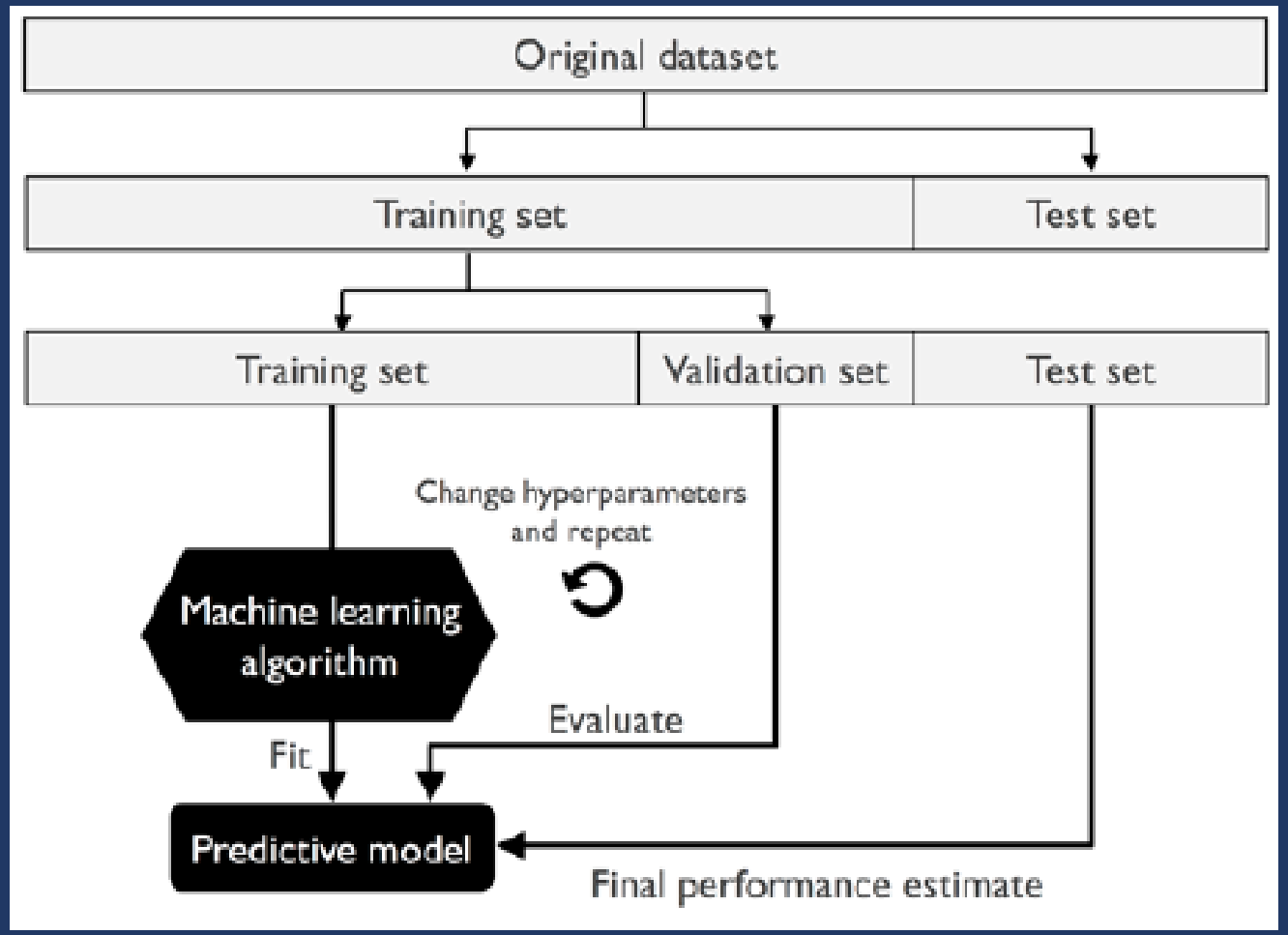
1. Standardize the d-dimensional dataset.
2. Construct the covariance matrix.
3. Decompose the covariance matrix into its eigenvectors and eigenvalues.
4. Sort the eigenvalues by decreasing order to rank the corresponding eigenvectors.
5. Select k eigenvectors which correspond to the k largest eigenvalues, where k is the dimensionality of the new feature subspace(k<d).
6. Construct a projection matrix W from the ”top” k eigenvectors.
7. Transform the d-dimensional input dataset X using the projection matrix W to obtain the new k-dimensional feature subspace

PCA on Superconductors Dataset



- 30 features selected.
- Explained variance ratio with 30 components: 0,99.

Model Performance



- PARAMETER VS HYPERPARAMETER

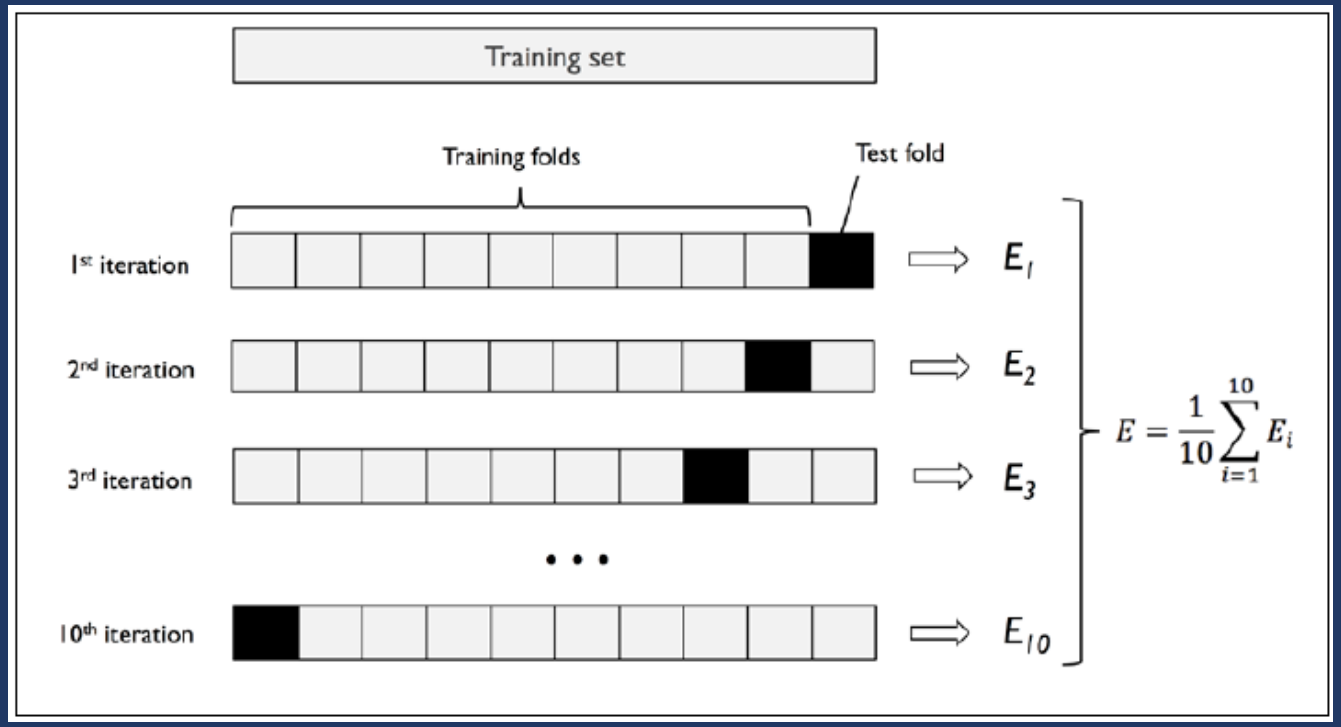
Cross Validation

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.

Cross Validation on Superconductors dataset

- HOLDOUT
  - 20% Test Dataset
  - 80% Training Dataset (of which 20% is the Validation dataset)
- 5-FOLD CV



HYPERPARAETER TUNING

Grid Search CV

- Brute-force exhaustive search paradigm
- Sklearn.model_selection.GridSearchCV

Random Search CV

- Randomized search for sampling different parameter combinations
- sklearn.model_selection.RandomizedSearchCV

PARAMETERS:

- estimator: algorithm of which you want to optimize the hyperparameters
- param_grid : collection of hyperparameters to be tested in the algorithm
- scoring: Strategy to evaluate the performance of the cross-validated model on the test set
- n_jobs: Number of jobs to run in parallel
- refit: Refit an estimator using the best found parameters on the whole dataset
- n_iter :Number of parameter settings that are sampled (only for Random Search CV)

Validation Curves

- Validation curves are a useful tool for improving the performance of a model by addressing issues such as overfitting or underfitting.
- Validation curves are plots of the training and test scores as functions of the model parameters, such as the max_depth of decision tree.

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

Supervised Learning Techniques for Regression Problem

- Multiple Linear Regression
- Regularized Methods
  - Lasso
  - Ridge
  - ElasticNet
- RANSAC

Sklearn.linear_model

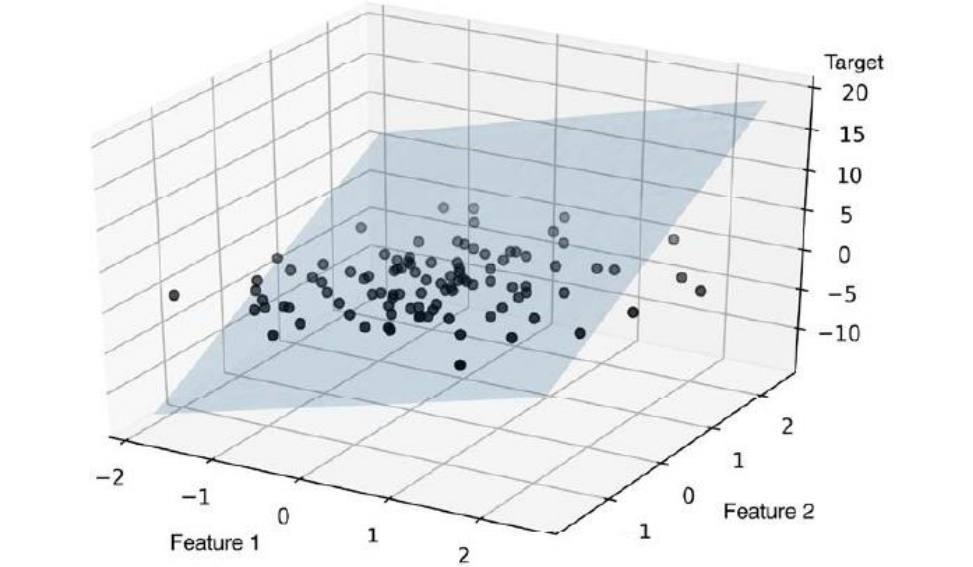
- Decision Tree → Sklearn.tree
- Random Forest → Sklearn.ensemble
- XGB → xgboost

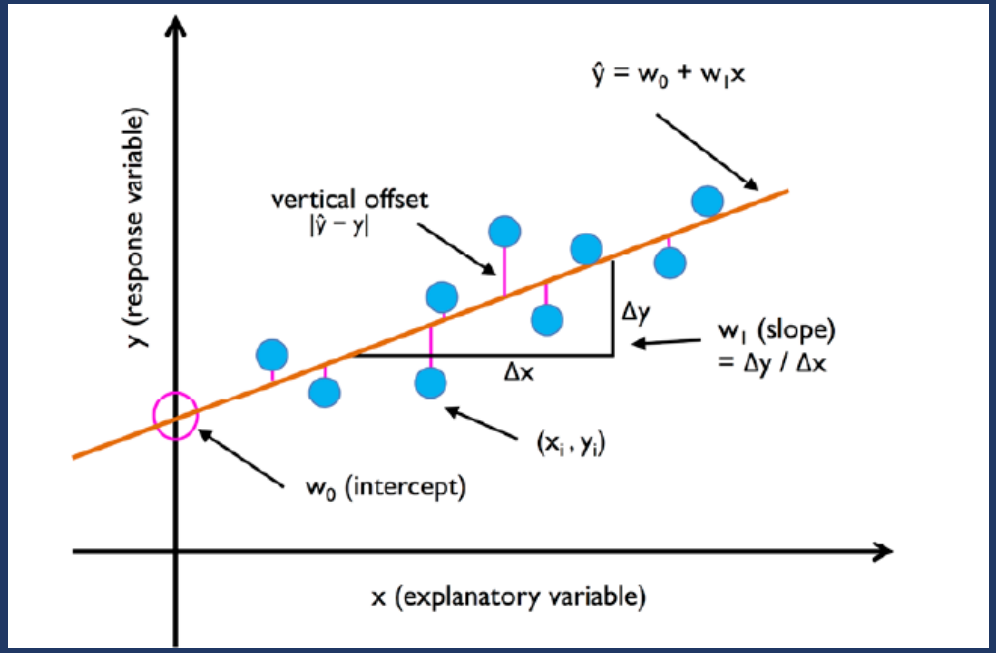
Multiple Linear Regression

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

$$y = w_0 + w_1 x$$

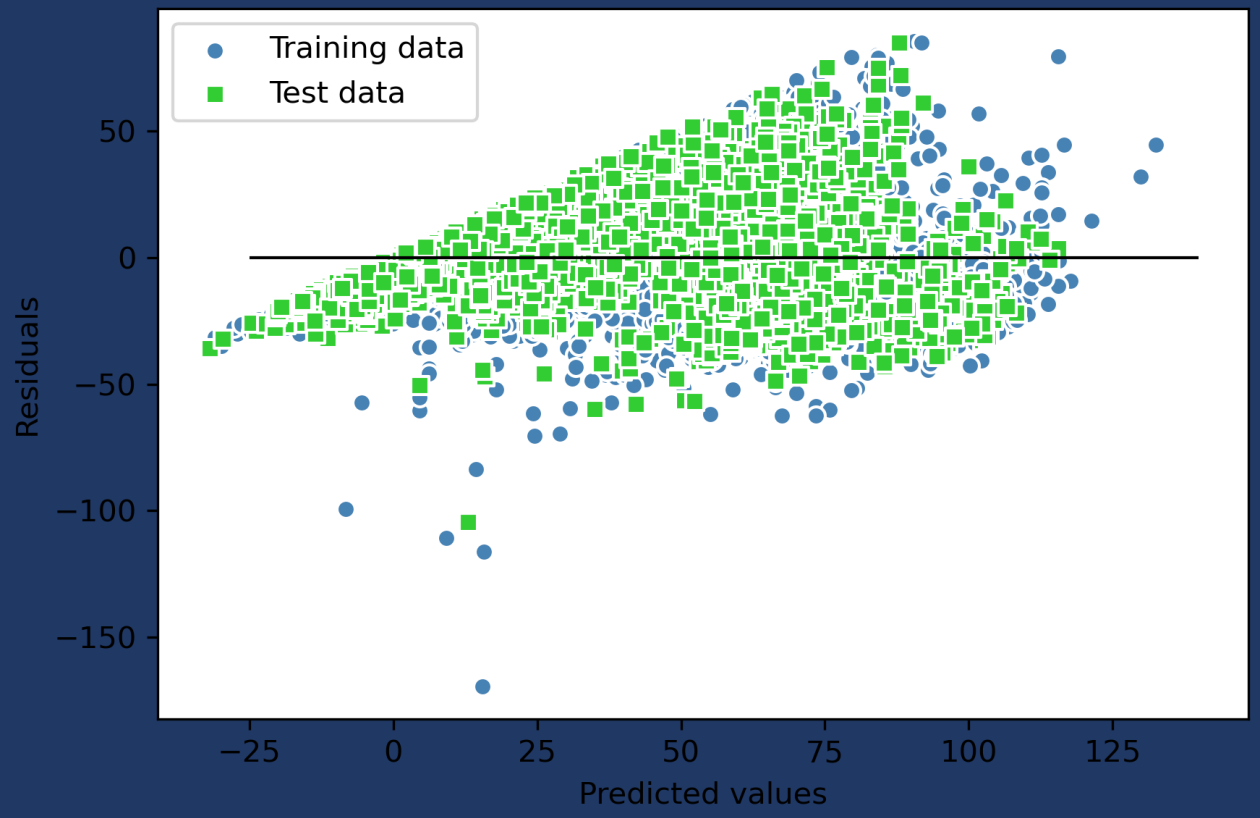
x = explanatory variable(s)
y = response variable
w_0 = intercept
w_1 = slope(s)





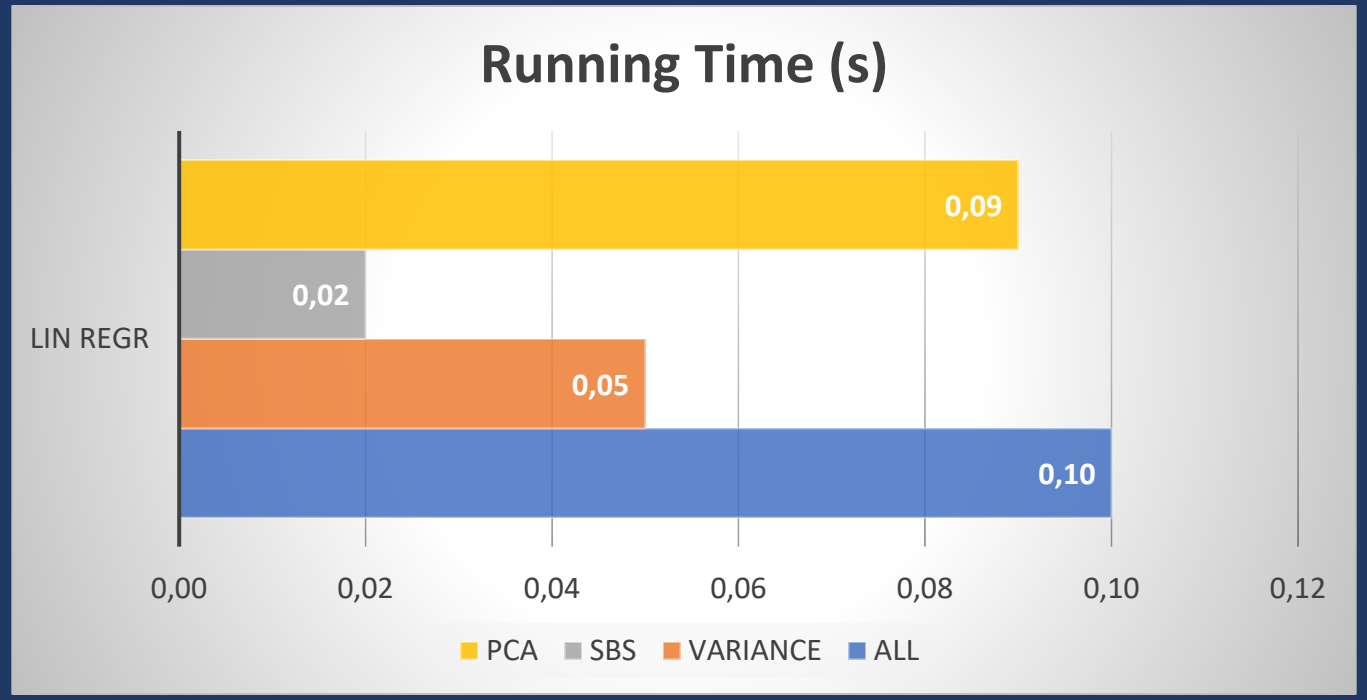
PROS	CONS
Simple and Rapid algorithm	Heavy impact by the presence of outliers

RESIDUAL PLOT



- Residues follow a fairly regular growing pattern, indicating that the data is not "very linear"

METRICS	ALL	VARIANCE	SBS	PCA
R2 TRAIN	0,74	0,72	0,720	0,74
R2 TEST	0,73	0,710	0,710	0,73



Regularized Methods for Regression

Lasso

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

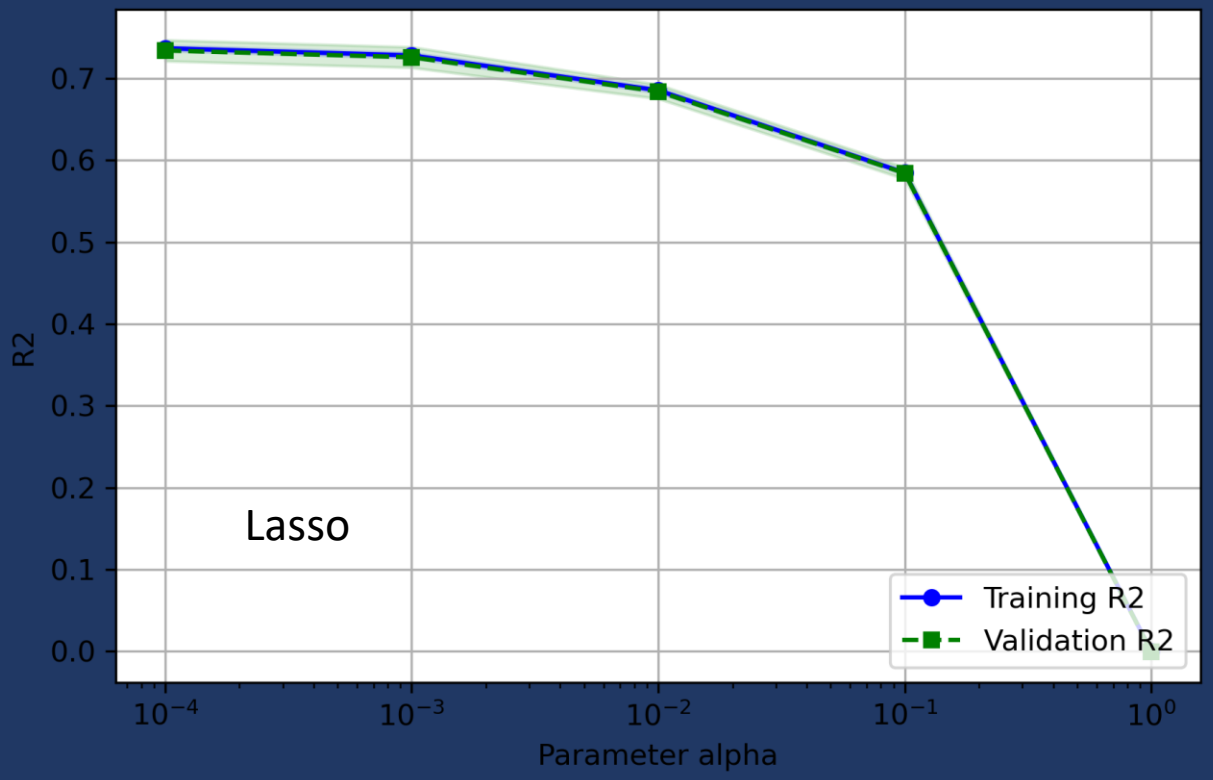
Ridge

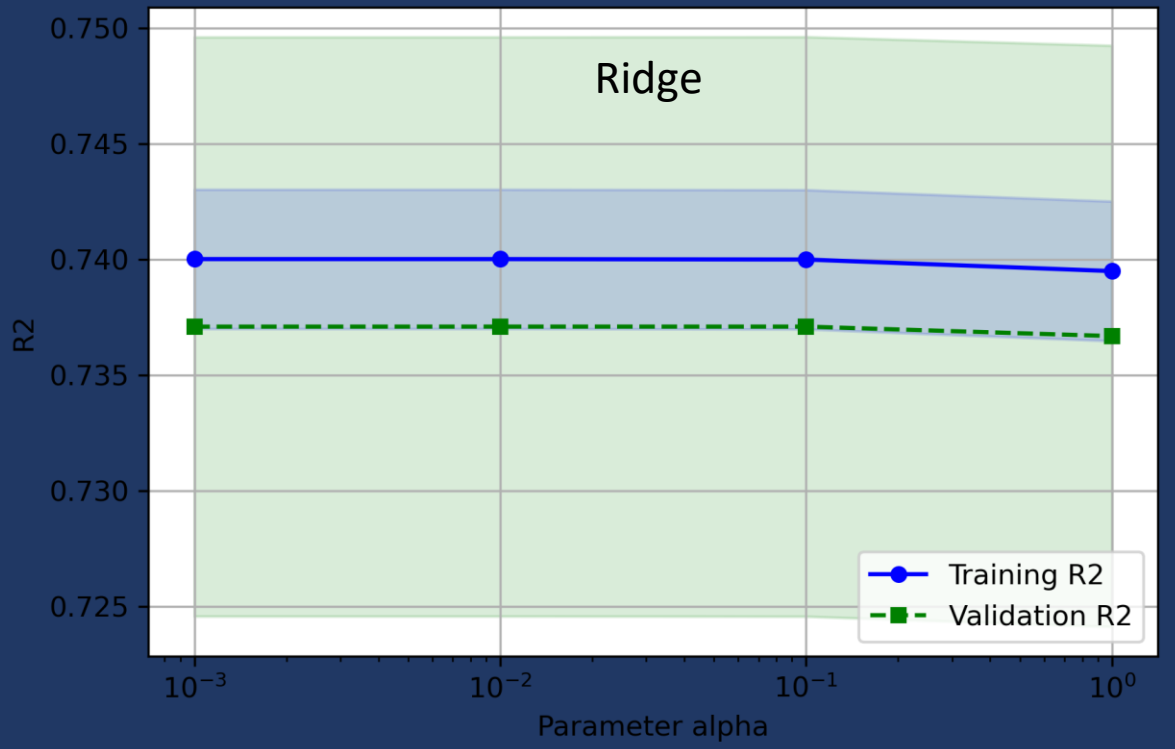
- L2 penalized model
- $J(w)_{RIDGE} = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda |w|^2_2$
- Parameter: 'alpha'

ElasticNet

- Compromise between the two previous models
- $J(w)_{ELNET} = \sum_{\{i=1\}}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda_1 \sum_{j=1}^{m} w_j^2 + \lambda_2 \sum_{j=1} |w_j|$
- Parameters: 'alpha', 'l1_ratio'

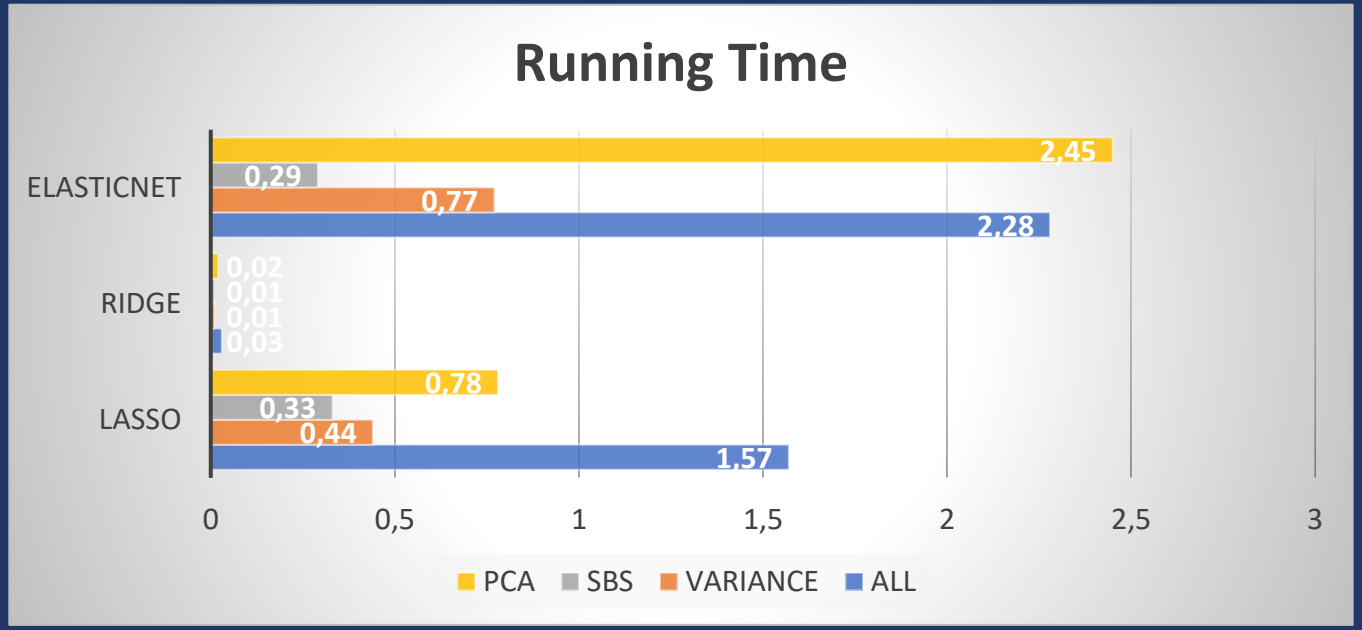
Tuning





ELASTICNET BEST PARAMETERS: ‘l1’=0,001, ‘alpha’=0,001

	METRICS	ALL	VARIANCE	SBS	PCA
LASSO	R2 TRAIN	0,62	0,63	0,62	0,62
	R2 TEST	0,62	0,62	0,62	0,62
RIDGE	R2 TRAIN	0,74	0,72	0,72	0,74
	R2 TEST	0,73	0,71	0,71	0,73
ELASTICNET	R2 TRAIN	0,61	0,58	0,59	0,61
	R2 TEST	0,61	0,58	0,59	0,61



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.

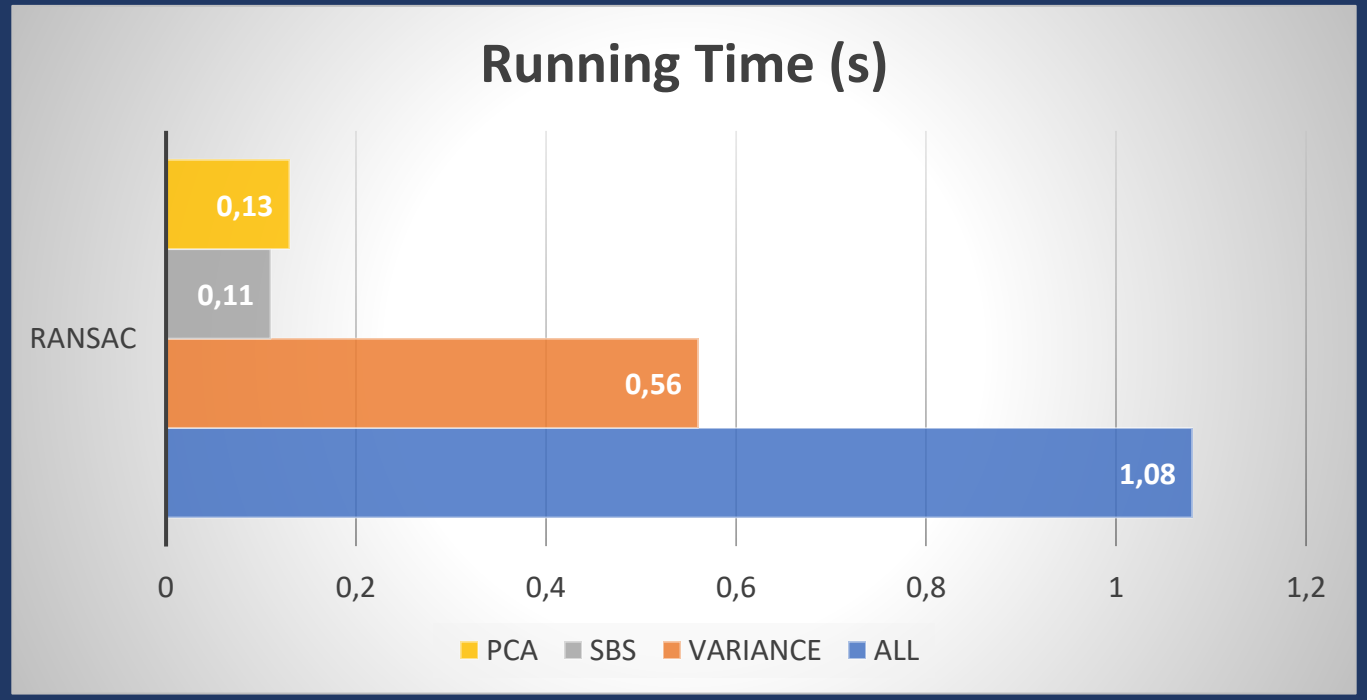
PARAMETERS

- **min_samples**: Minimum number of samples chosen randomly from original data
- **residual_threshold**: Maximum residual for a data sample to be classified as an inlier
- **max_trials**: Maximum number of iterations for random sample selection
- **loss**: ‘absolute_error’ and ‘squared_error’ are supported which find the absolute error and squared error per sample respectively

BEST PARAMETERS: min_samples=1000, residual_trhreshold = 100, max_trials = 1000, loss= ‘squared_error.

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

METRICS	ALL	VARIANCE	SBS	PCA
R2 TRAIN	0,72	0,71	0,71	0,74
R2 TEST	0,71	0,69	0,70	0,73



Decision Tree Regressor

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

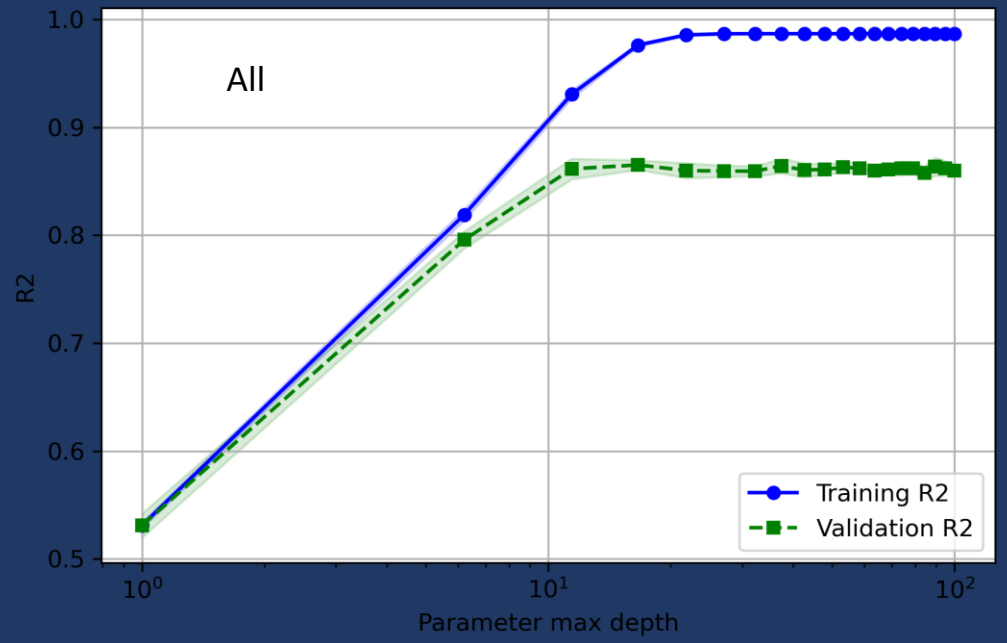
$$\mathrm{I(t)} = \frac{1}{N_t} \sum_{i \in D_t}^{c} \left(y^{(i)} - \hat{y}_t\right)^2$$

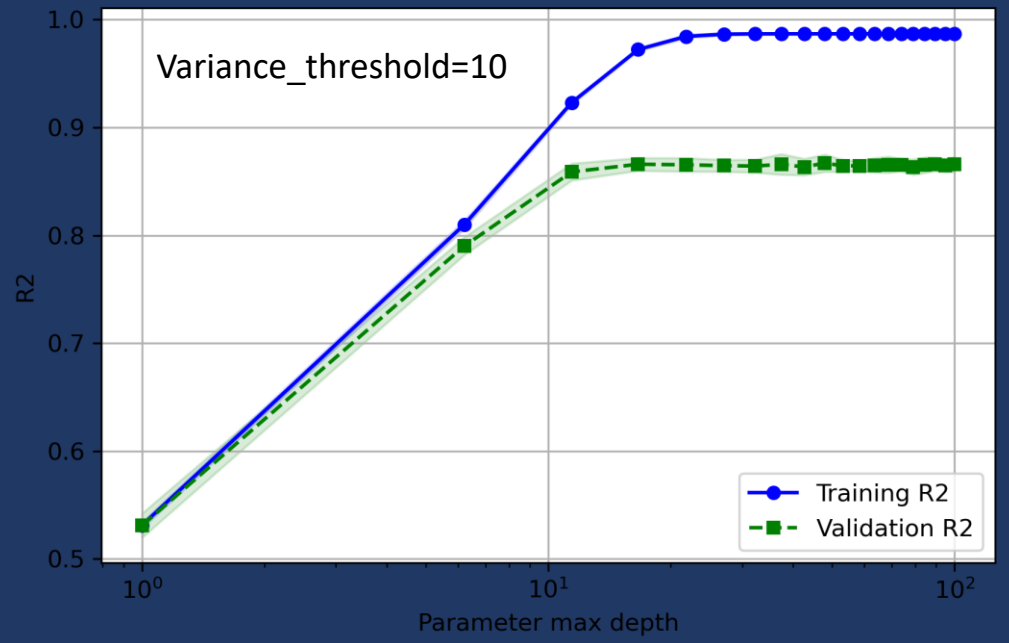
PROS	CONS
• Understanding • No standardization needed • Low Computaional cost	• Overfitting • High noise sensibility • Hyperparameters

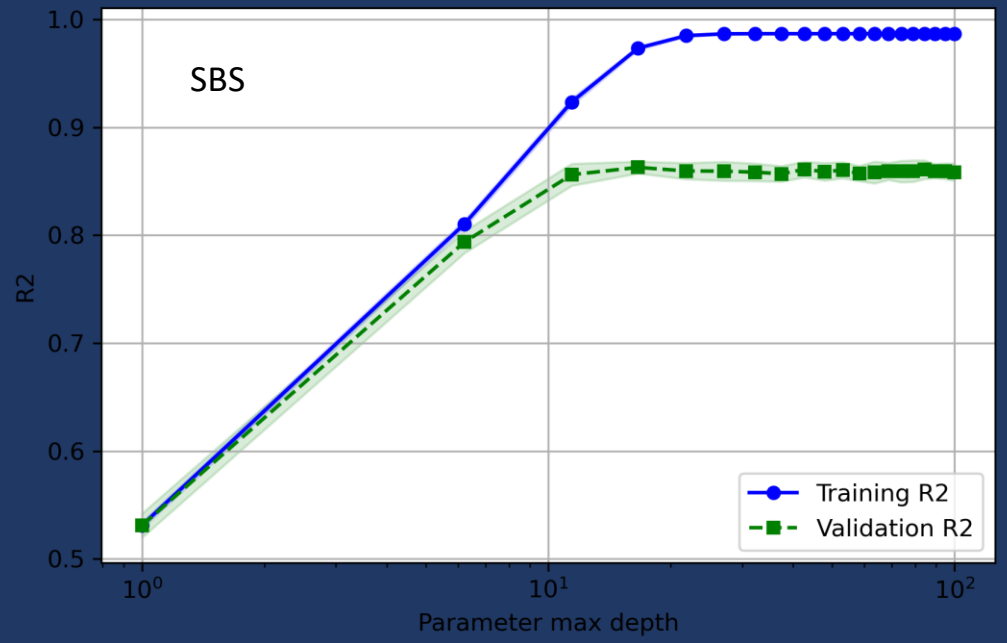
PARAMETERS

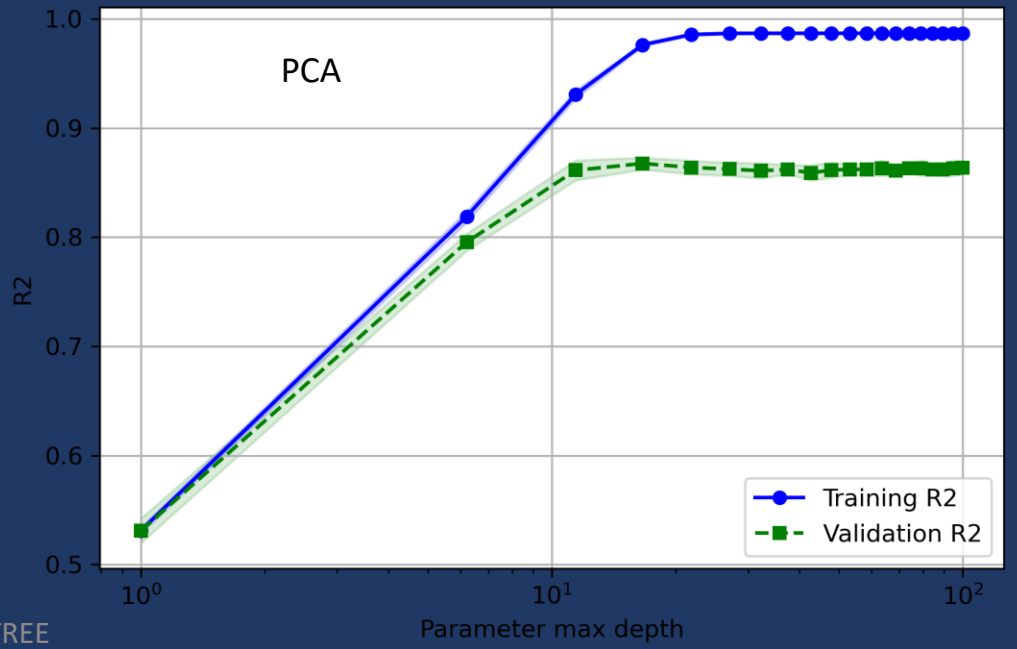
- max_depth: The maximum depth of the tree
- criterion: The function to measure the quality of a split
- splitter: The strategy used to choose the split at each node
- min_samples_split: The minimum number of samples required to split an internal node:
- min_samples_leaf: The minimum number of samples required to be at a leaf node.

BEST PARAMETERS: max_depth=12,, criterion= ‘squared_error’, splitter= ‘ best’.

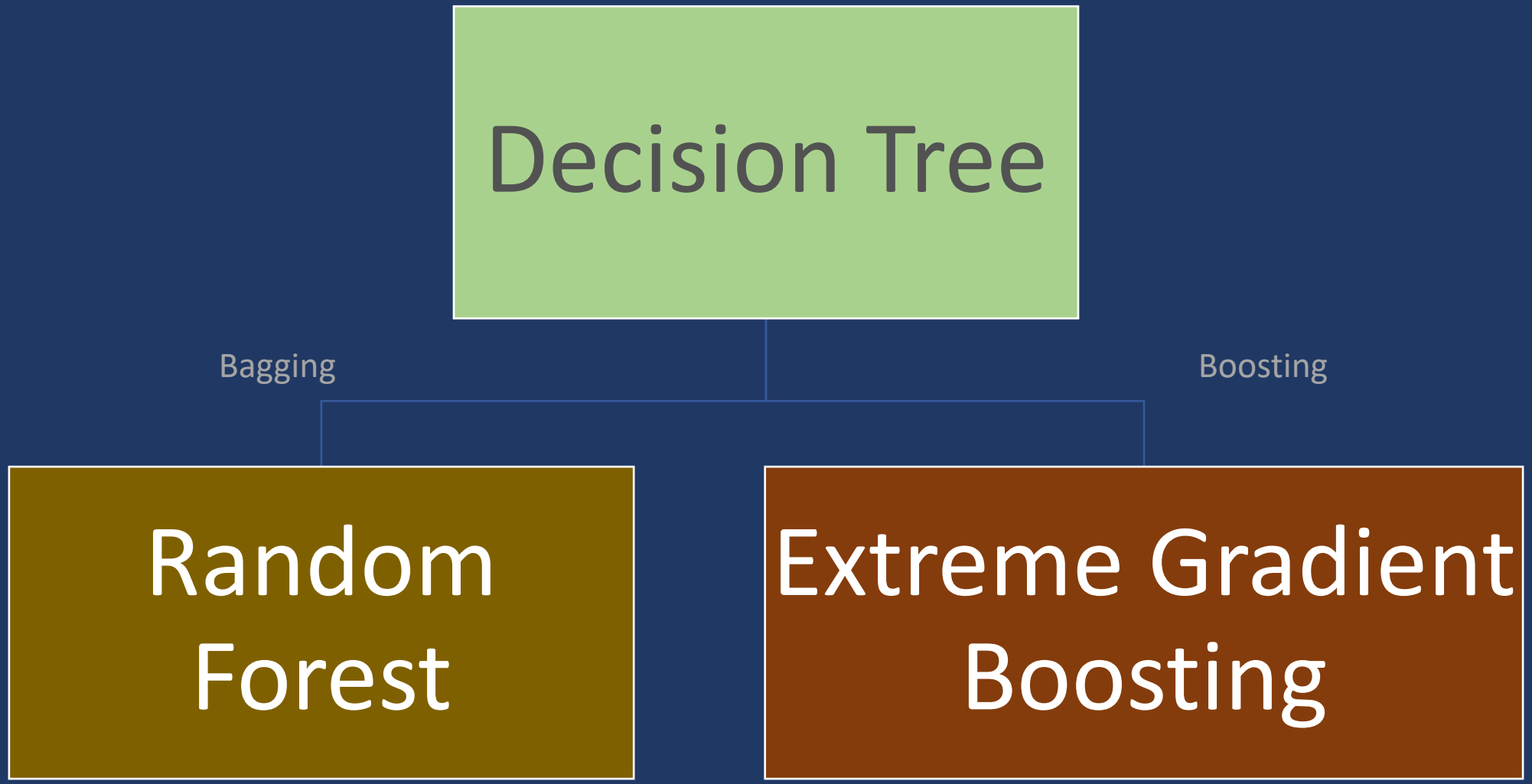








DECISION TREE


Decision Tree
Bagging
Boosting
Random Forest
Extreme Gradient Boosting

Random Forest Regressor

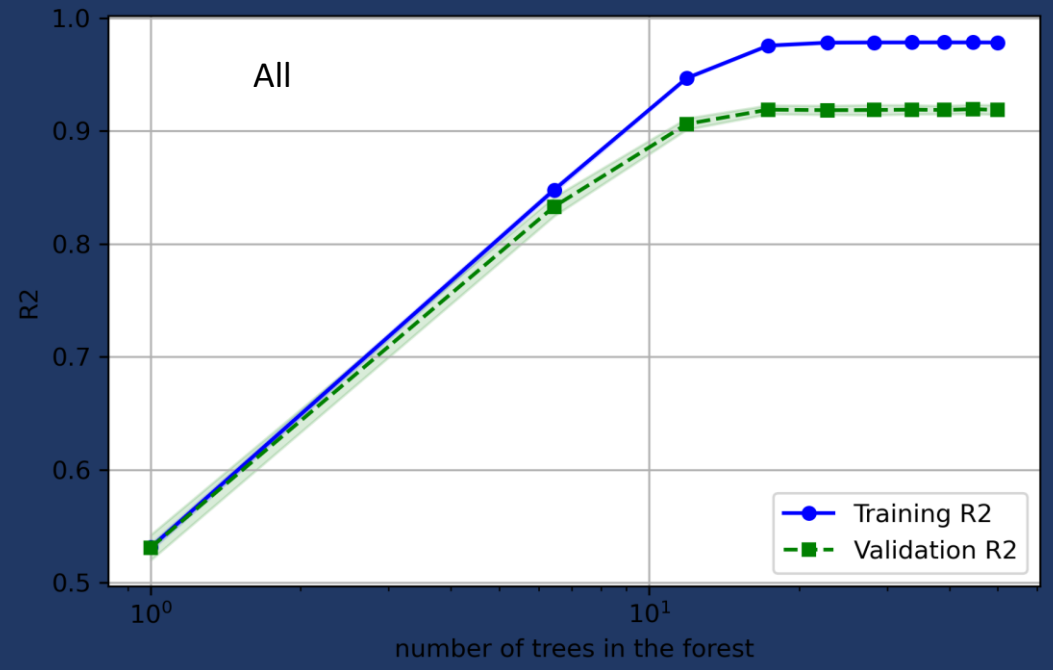
- Ensamble Method (Bagging)
- Decision Tree ‘’in parallel’’

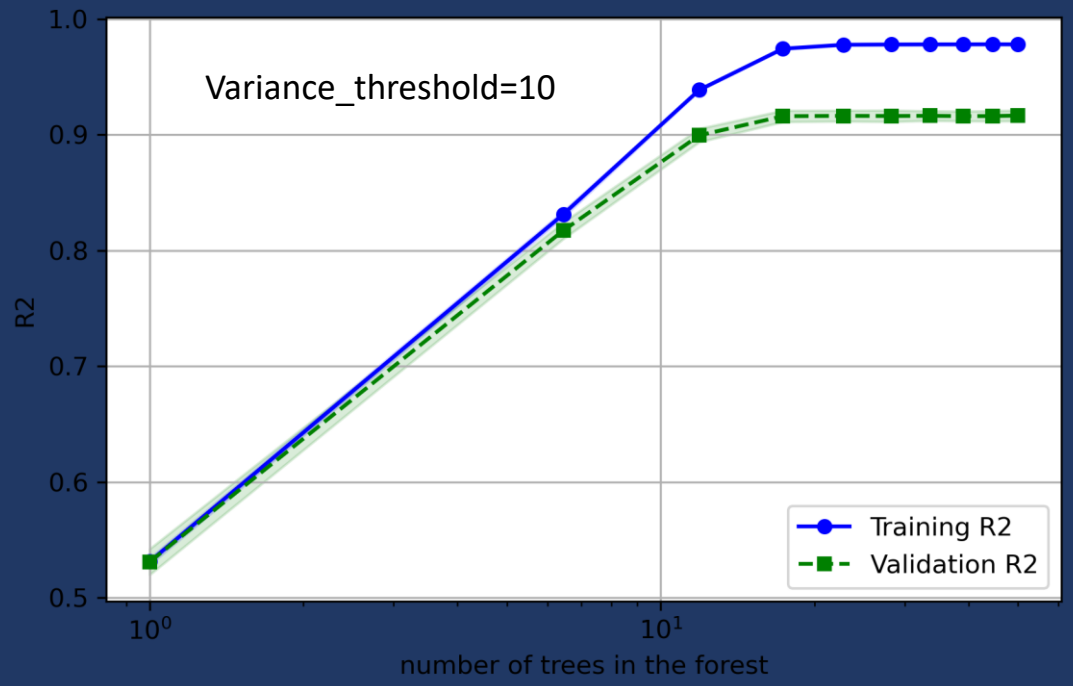
PROS	CONS
• Invariant scale • Best generalization performance • Simple tuning	Overfitting

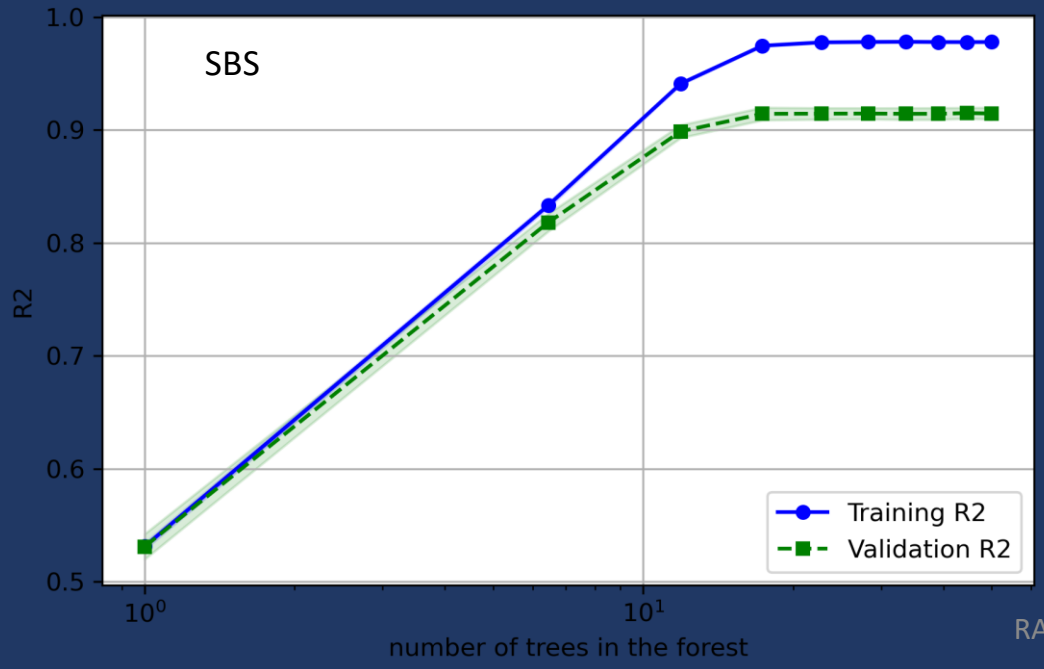
PARAMETERS

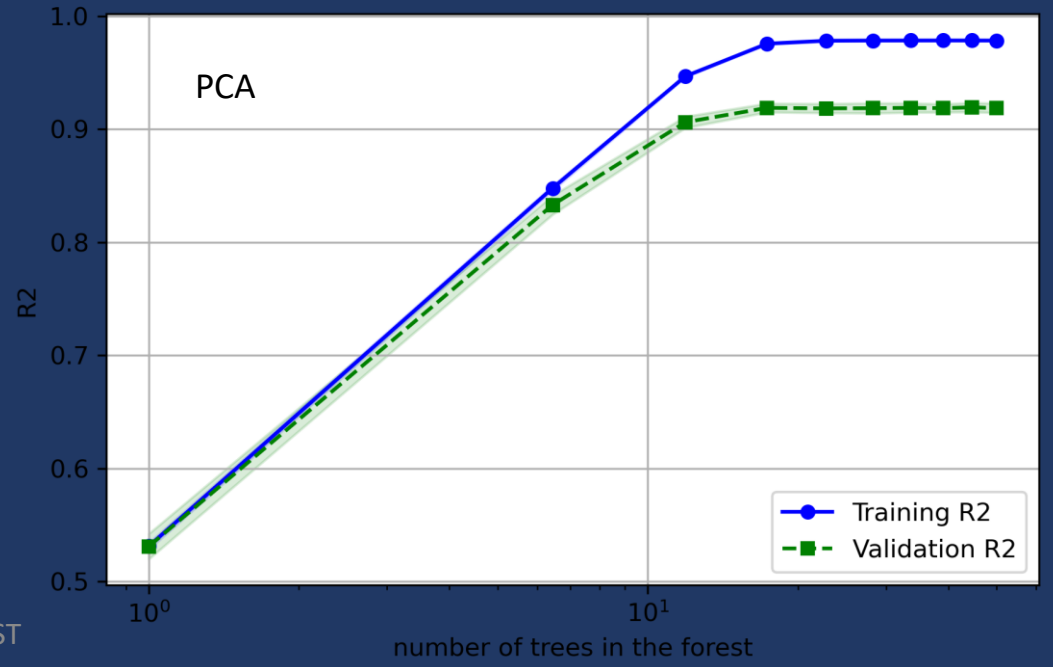
- **n_estimators**: The number of trees in the forest
- **max_depth**: The maximum depth of the tree
- **criterion**: The function to measure the quality of a split
- **min_samples_split**: The minimum number of samples required to split an internal node:
- **min_samples_leaf**: he minimum number of samples required to be at a leaf node.

BEST PARAMETERS: n_estimators=100, max_depth = 15, criterion= ‘squared_error’.


All
R2
number of trees in the forest
Training R2
Validation R2


Variance_threshold=10
R2
number of trees in the forest
Training R2
Validation R2


SBS
R2
number of trees in the forest
Training R2
Validation R2


PCA
R2
number of trees in the forest
Training R2
Validation R2

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.

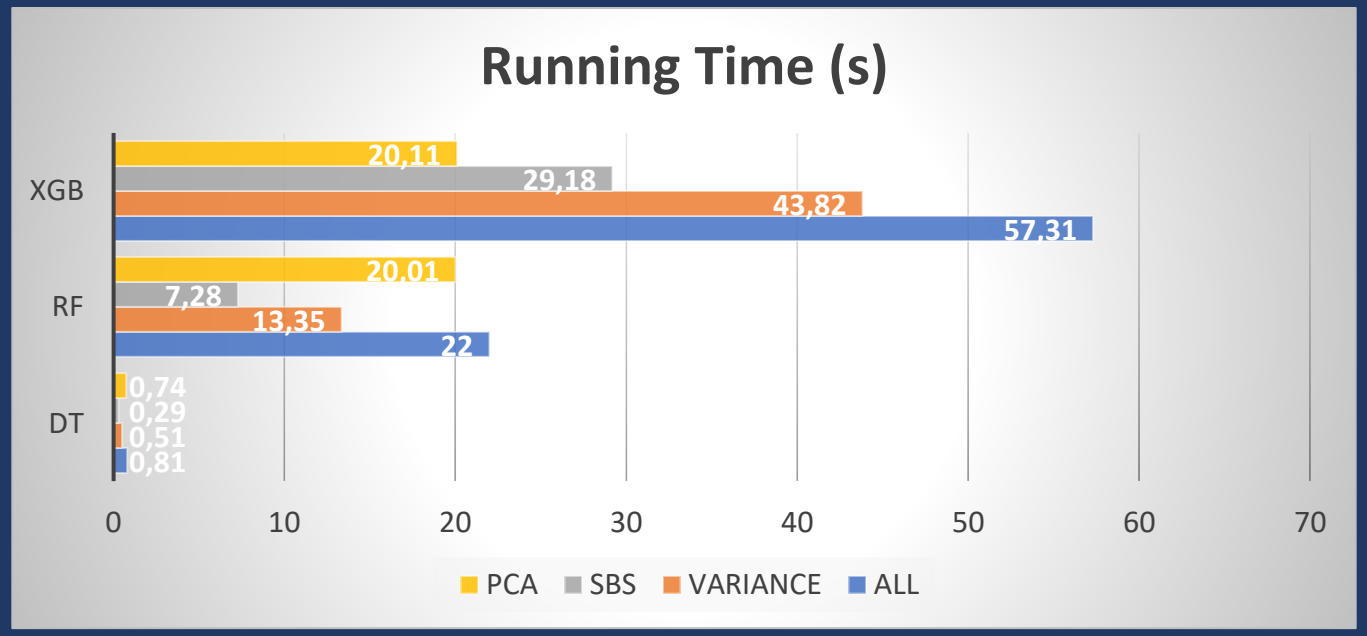
PROS	CONS
• Reduce Variance and Bias with respect to DT	• Compuationally hard

PARAMETERS

- **n_estimators**: Number of gradient boosted trees
- **eta**: Step size shrinkage used in update to prevents overfitting
- **max_depth**: Maximum depth of a tree
- **subsample**: Subsample ratio of the training instances

BEST PARAMETERS: n_estimators=350, eta = 0,02, max_depth = 15, subsample= 0,5. (Grid Search)

	METRICS	ALL	VARIANCE	SBS	PCA
DT	R2 TRAIN	0,94	0,94	0,94	0,94
	R2 TEST	0,86	0,86	0,86	0,86
RF	R2 TRAIN	0,97	0,97	0,97	0,97
	R2 TEST	0,92	0,91	0,91	0,92
XGB	R2 TRAIN	0,98	0,98	0,98	0,97
	R2 TEST	0,92	0,92	0,93	0,92



DECISION TREE, RANDOM, FOREST, XG BOOST

Multiple Polynomial Regression

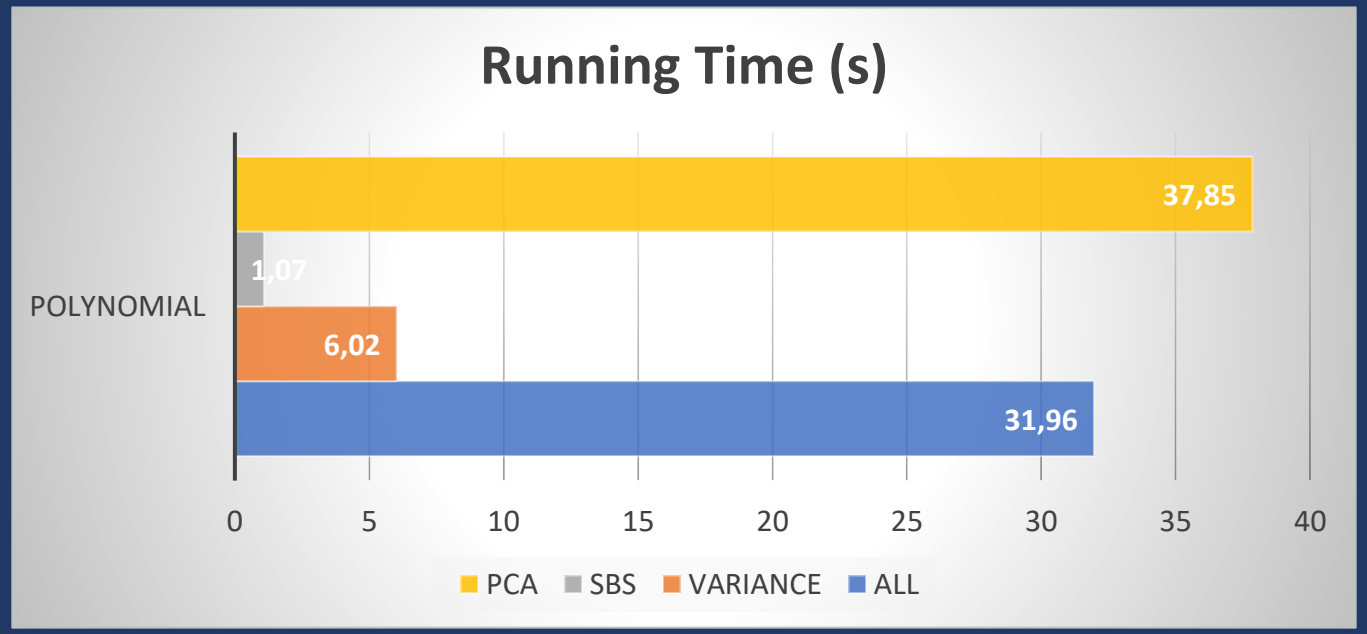
- This algorithm allows us to describe the data with a polynomial of degree n :

$$y = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$$

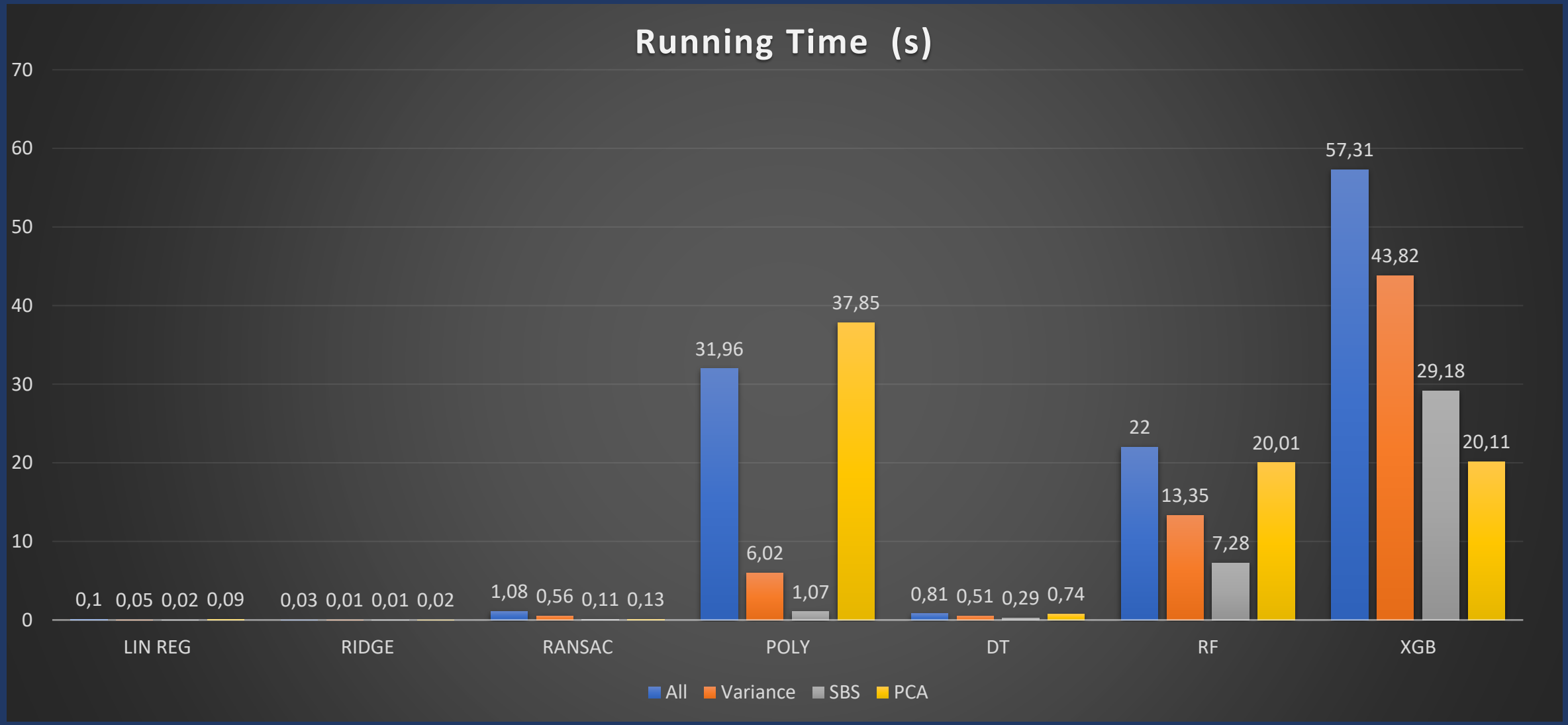
- Together with this model the algorithm was quite efficient.

- On Superconductors dataset, the best degree for the polynomial fitting curve is n=2.

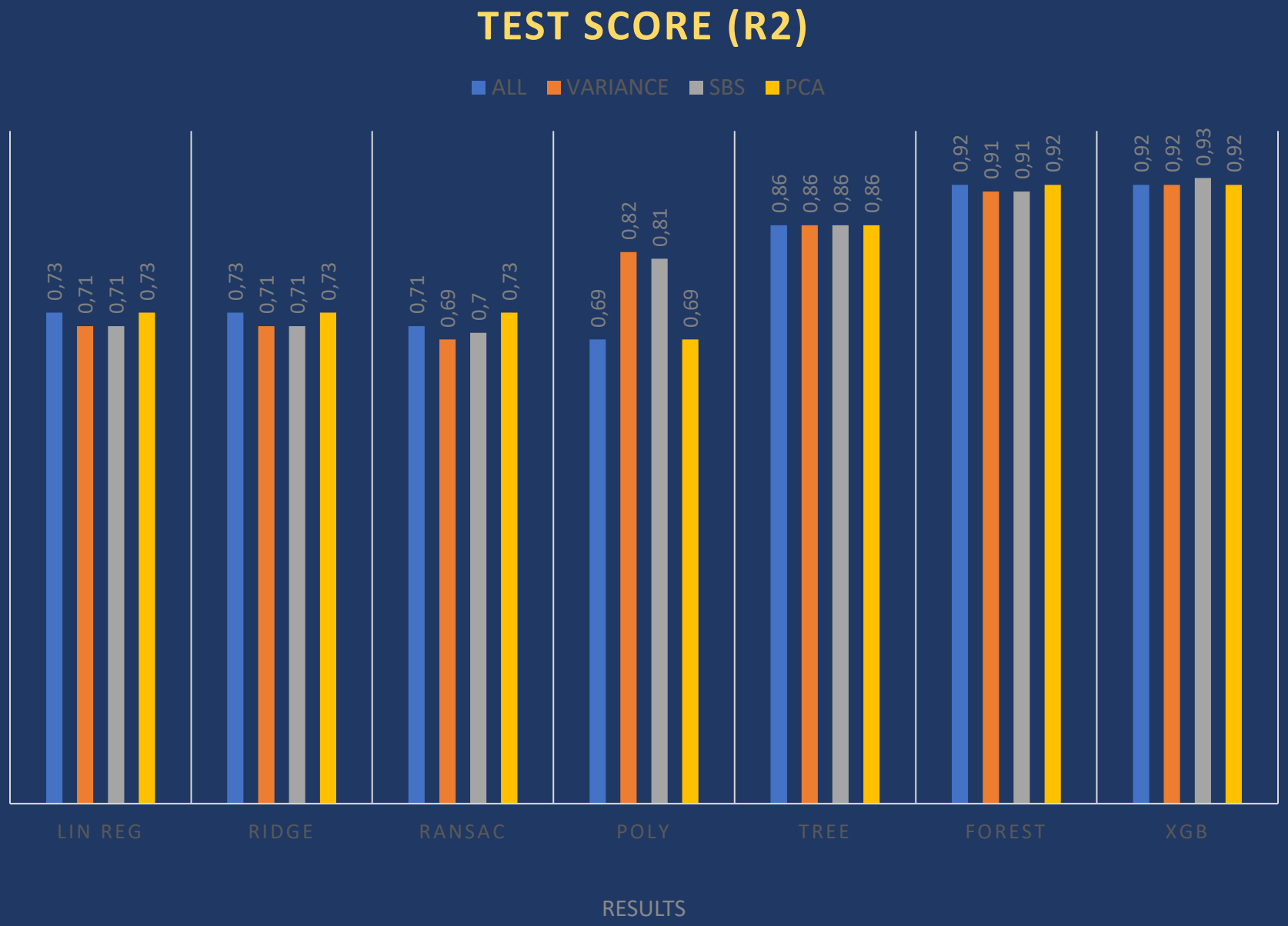
METRICS	ALL	VARIANCE	SBS	PCA
R2 TRAIN	0,91	0,87	0,84	0,91
R2 TEST	0,69	0,82	0,81	0,69



Results


Running Time (s)
70
60
50
40
30
20
10
0
0,1
0,05
0,02
0,09
0,03
0,01
0,01
0,02
1,08
0,56
0,11
0,13
31,96
6,02
1,07
37,85
0,81
0,51
0,29
0,74
22
13,35
7,28
20,01
57,31
43,82
29,18
20,11
LIN REG
RIDGE
RANSAC
POLY
DT
RF
XGB
All
Variance
SBS
PCA

Results


TEST SCORE (R2)
ALL
VARIANCE
SBS
PCA
0,73
0,71
0,71
0,73
0,73
0,71
0,71
0,73
0,71
0,69
0,7
0,73
0,69
0,82
0,81
0,69
0,86
0,86
0,86
0,86
0,92
0,91
0,91
0,92
0,92
0,92
0,93
0,92
LIN REG
RIDGE
RANSAC
POLY
TREE
FOREST
XGB
RESULTS

Conclusions

- We have managed to achieve resourceful results with most models, but the best in terms of performance is the Extreme Gradient Boost.
- The model that represents the best compromise between algorithm execution time and results is the Random Forest Regressor on the features obtained through SBS .
- It is important to note that with this model it is not possible to predict whether a material is superconductive or not. However, if we have a superconductor material, we can estimate its critical temperature by knowing some chemical-physical characteristics.
- Taking RMSE as error estimate, we can say that we are able to predict the critical temperature of a superconductor with an error of about 9,4 Kelvin.