Predicting molecular properties with machine learning

Md Naim Hassan Saykat

Petko Petkov

Project objective

- train machine learning models to predict molecular properties from the 3D structure of the molecules
- perform data exploration on QM9 dataset, convert the data into an appropriate format for machine learning models
- compare multiple regression models after cross-validation and hyperparameter tuning

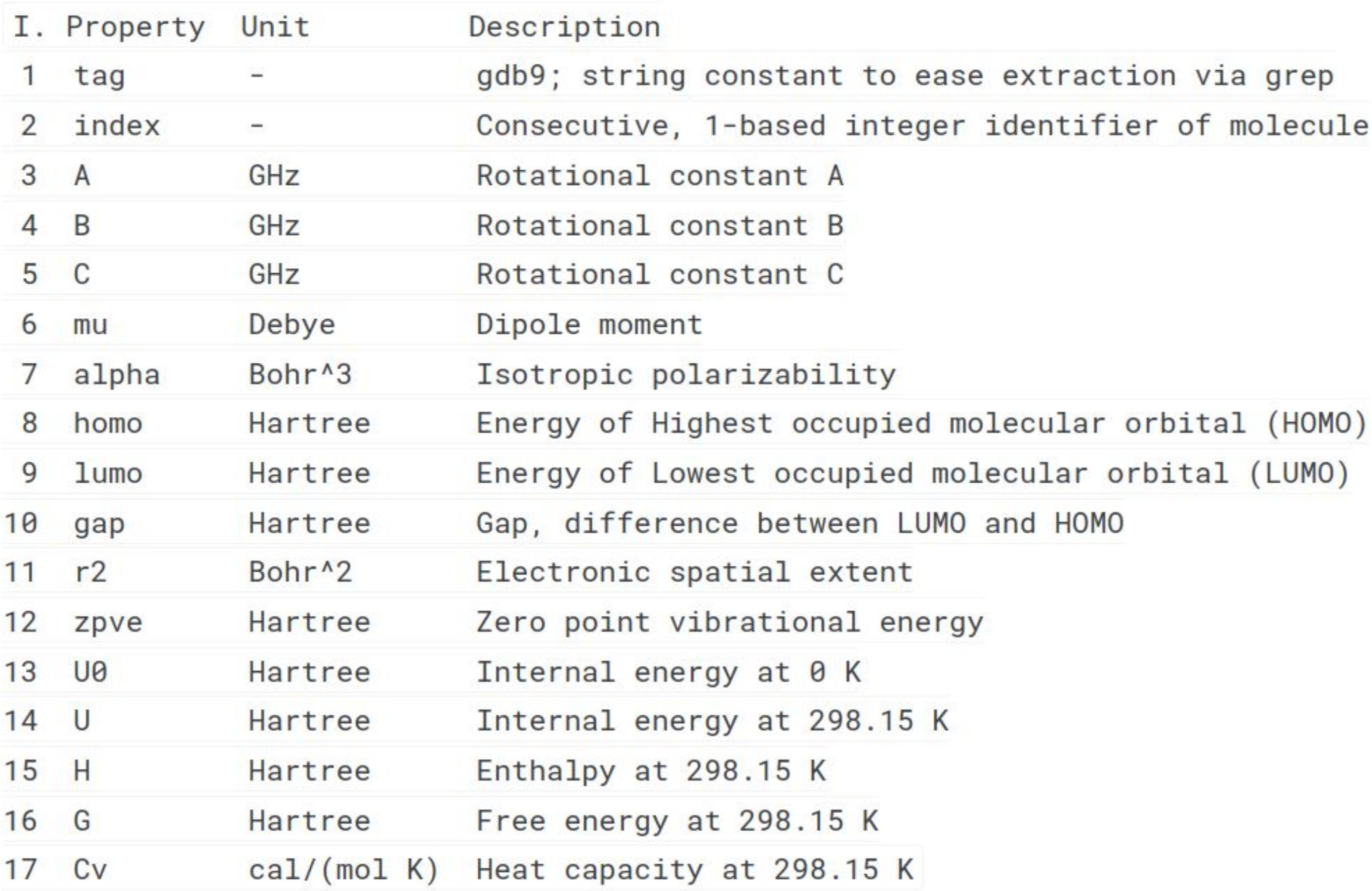
Dataset

- QM9 (quantum chemical properties of approximately 134,000 stable small organic molecules, we’re using 1000 random samples)

- includes computed geometric, energetic, electronic, and thermodynamic properties for each molecule

- contains 16 physical/chemical features (we are predicting 2 of them - heat capacity and isotropic polarizability) and Euclidean coordinates of the atoms

Dataset features

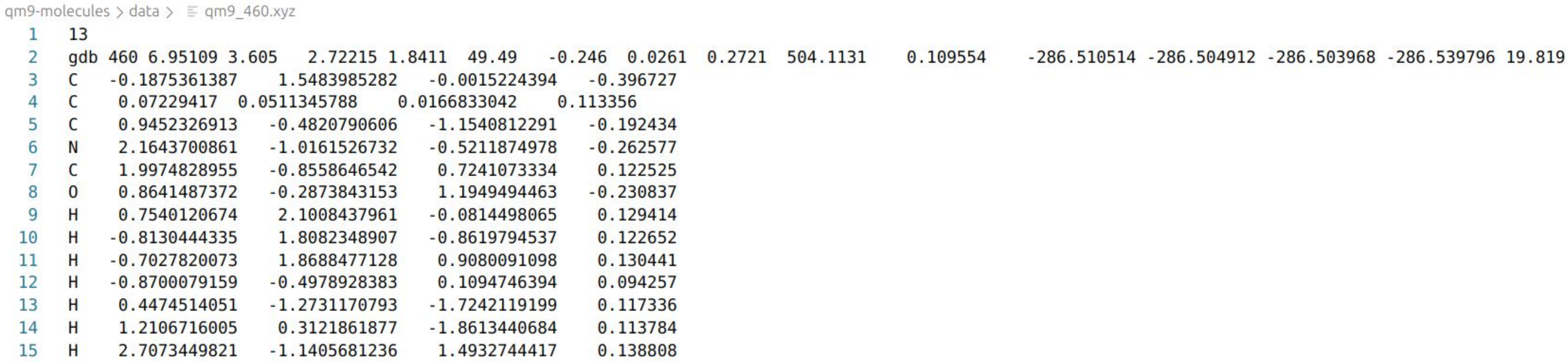


I.	Property	Unit	Description
1	tag	-	gdb9; string constant to ease extraction via grep
2	index	-	Consecutive, 1-based integer identifier of molecule
3	A	GHz	Rotational constant A
4	B	GHz	Rotational constant B
5	C	GHz	Rotational constant C
6	mu	Debye	Dipole moment
7	alpha	Bohr^3	Isotropic polarizability
8	homo	Hartree	Energy of Highest occupied molecular orbital (HOMO)
9	lumo	Hartree	Energy of Lowest occupied molecular orbital (LUMO)
10	gap	Hartree	Gap, difference between LUMO and HOMO
11	r2	Bohr^2	Electronic spatial extent
12	zpve	Hartree	Zero point vibrational energy
13	U0	Hartree	Internal energy at 0 K
14	U	Hartree	Internal energy at 298.15 K
15	H	Hartree	Enthalpy at 298.15 K
16	G	Hartree	Free energy at 298.15 K
17	Cv	cal/(mol K)	Heat capacity at 298.15 K

Preprocessing

- the dataset contains XYZ format files (coordinates of atoms and the molecular properties)
- extracting the atoms’ coordinates and features from the XYZ files

Example XYZ format file:



```
13
gdb 460 6.95109 3.605   2.72215 1.8411  49.49   -0.246  0.0261  0.2721  504.1131    0.109554    -286.510514 -286.504912 -286.503968 -286.539796 19.819
C   -0.1875361387    1.5483985282   -0.0015224394   -0.396727
C    0.07229417  0.0511345788    0.0166833042    0.113356
C    0.9452326913   -0.4820790606   -1.1540812291   -0.192434
N    2.1643700861   -1.0161526732   -0.5211874978   -0.262577
C    1.9974828955   -0.8558646542    0.7241073334    0.122525
O    0.8641487372   -0.2873843153    1.1949494463   -0.230837
H    0.7540120674    2.1008437961   -0.0814498065    0.129414
H   -0.8130444335    1.8082348907   -0.8619794537    0.122652
H   -0.7027820073    1.8688477128    0.9080091098    0.130441
H   -0.8700079159   -0.4978928383    0.1094746394    0.094257
H    0.4474514051   -1.2731170793   -1.7242119199    0.117336
H    1.2106716005    0.3121861877   -1.8613440684    0.113784
H    2.7073449821   -1.1405681236    1.4932744417    0.138808
```

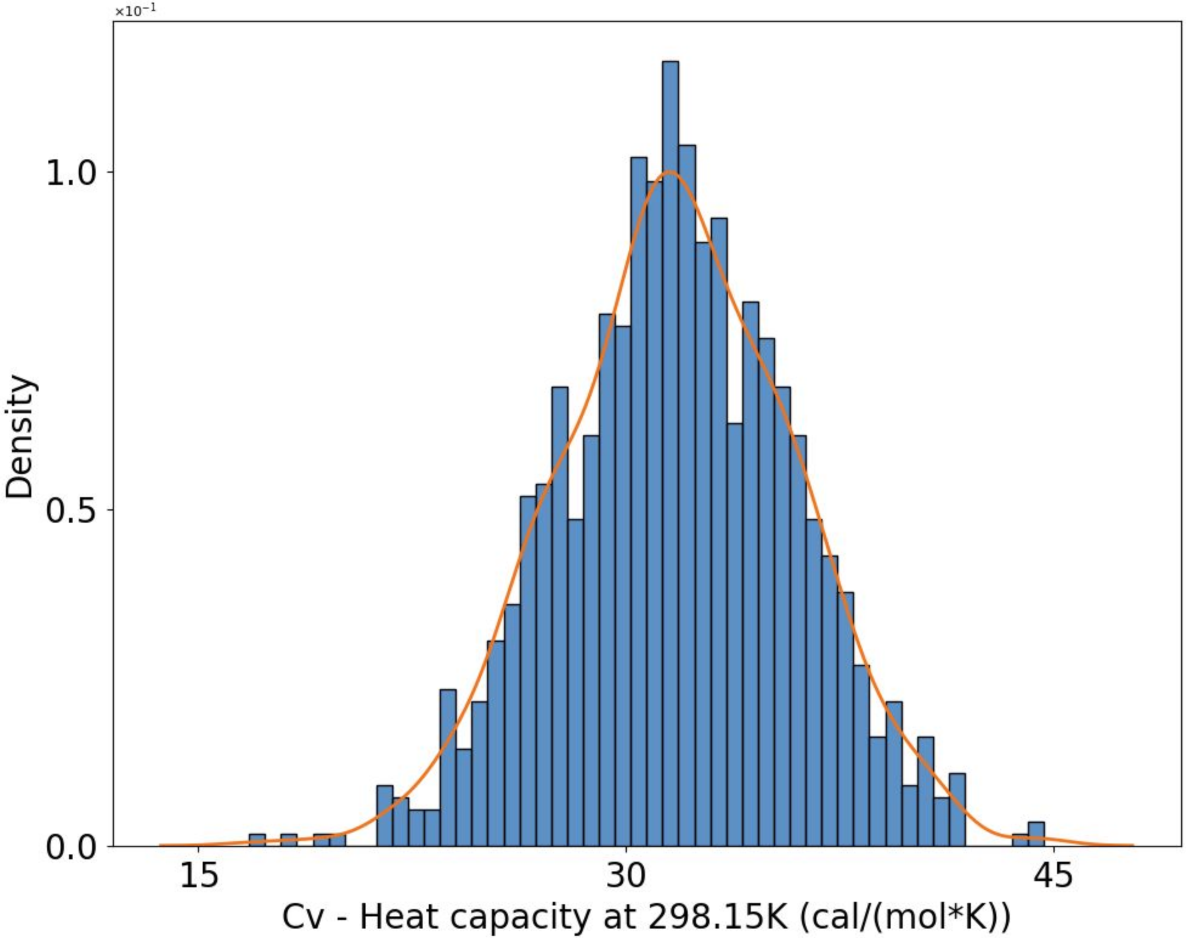
Data exploration

- visualize the data so we can get an idea of the targets we can predict

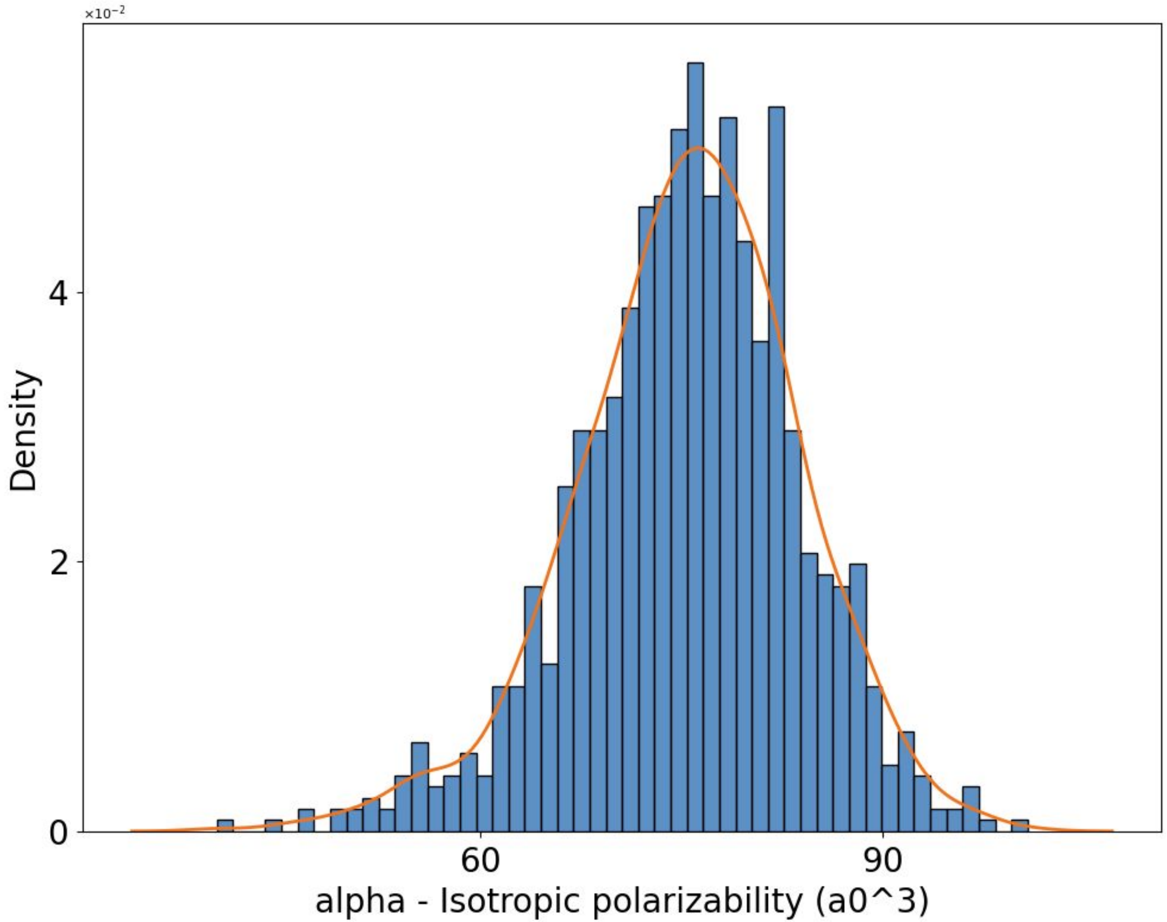
- make more informed decisions on model selection and evaluation metrics

- identify any potential outliers or skewness in the data that may require special handling during preprocessing

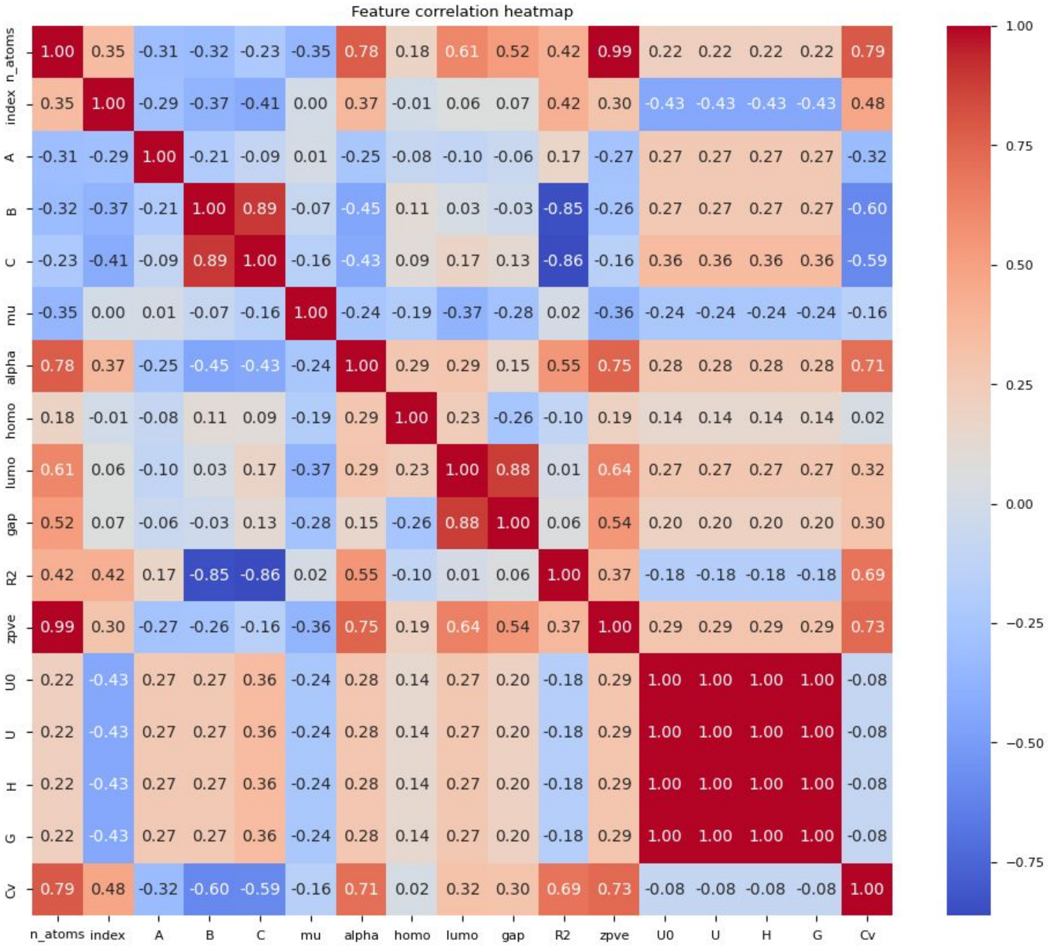
Heat capacity distribution



Isotropic polarizability distribution



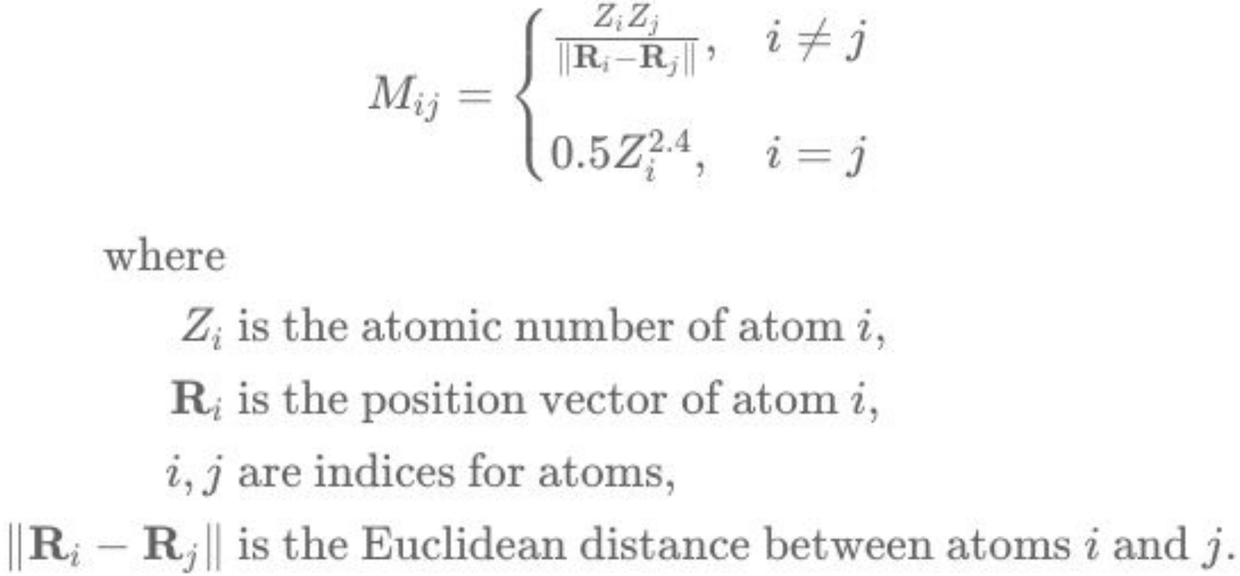
Correlation between features


Feature correlation heatmap

	n_atoms	index	A	B	C	mu	alpha	homo	lumo	gap	R2	zpve	U0	U	H	G	Cv
n_atoms	1.00	0.35	-0.31	-0.32	-0.23	-0.35	0.78	0.18	0.61	0.52	0.42	0.99	0.22	0.22	0.22	0.22	0.79
index	0.35	1.00	-0.29	-0.37	-0.41	0.00	0.37	-0.01	0.06	0.07	0.42	0.30	-0.43	-0.43	-0.43	-0.43	0.48
A	-0.31	-0.29	1.00	-0.21	-0.09	0.01	-0.25	-0.08	-0.10	-0.06	0.17	-0.27	0.27	0.27	0.27	0.27	-0.32
B	-0.32	-0.37	-0.21	1.00	0.89	-0.07	-0.45	0.11	0.03	-0.03	-0.85	-0.26	0.27	0.27	0.27	0.27	-0.60
C	-0.23	-0.41	-0.09	0.89	1.00	-0.16	-0.43	0.09	0.17	0.13	-0.86	-0.16	0.36	0.36	0.36	0.36	-0.59
mu	-0.35	0.00	0.01	-0.07	-0.16	1.00	-0.24	-0.19	-0.37	-0.28	0.02	-0.36	-0.24	-0.24	-0.24	-0.24	-0.16
alpha	0.78	0.37	-0.25	-0.45	-0.43	-0.24	1.00	0.29	0.29	0.15	0.55	0.75	0.28	0.28	0.28	0.28	0.71
homo	0.18	-0.01	-0.08	0.11	0.09	-0.19	0.29	1.00	0.23	-0.26	-0.10	0.19	0.14	0.14	0.14	0.14	0.02
lumo	0.61	0.06	-0.10	0.03	0.17	-0.37	0.29	0.23	1.00	0.88	0.01	0.64	0.27	0.27	0.27	0.27	0.32
gap	0.52	0.07	-0.06	-0.03	0.13	-0.28	0.15	-0.26	0.88	1.00	0.06	0.54	0.20	0.20	0.20	0.20	0.30
R2	0.42	0.42	0.17	-0.85	-0.86	0.02	0.55	-0.10	0.01	0.06	1.00	0.37	-0.18	-0.18	-0.18	-0.18	0.69
zpve	0.99	0.30	-0.27	-0.26	-0.16	-0.36	0.75	0.19	0.64	0.54	0.37	1.00	0.29	0.29	0.29	0.29	0.73
U0	0.22	-0.43	0.27	0.27	0.36	-0.24	0.28	0.14	0.27	0.20	-0.18	0.29	1.00	1.00	1.00	1.00	-0.08
U	0.22	-0.43	0.27	0.27	0.36	-0.24	0.28	0.14	0.27	0.20	-0.18	0.29	1.00	1.00	1.00	1.00	-0.08
H	0.22	-0.43	0.27	0.27	0.36	-0.24	0.28	0.14	0.27	0.20	-0.18	0.29	1.00	1.00	1.00	1.00	-0.08
G	0.22	-0.43	0.27	0.27	0.36	-0.24	0.28	0.14	0.27	0.20	-0.18	0.29	1.00	1.00	1.00	1.00	-0.08
Cv	0.79	0.48	-0.32	-0.60	-0.59	-0.16	0.71	0.02	0.32	0.30	0.69	0.73	-0.08	-0.08	-0.08	-0.08	1.00

Calculate Coulomb matrices

- simple global descriptor which mimics the electrostatic interaction between nuclei
- N x N dimension where N is the number of atoms. The number of eigenvalues is also equal to N



$$M_{ij} = \begin{cases} \frac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|}, & i \neq j \\ 0.5 Z_i^{2.4}, & i = j \end{cases}$$

where

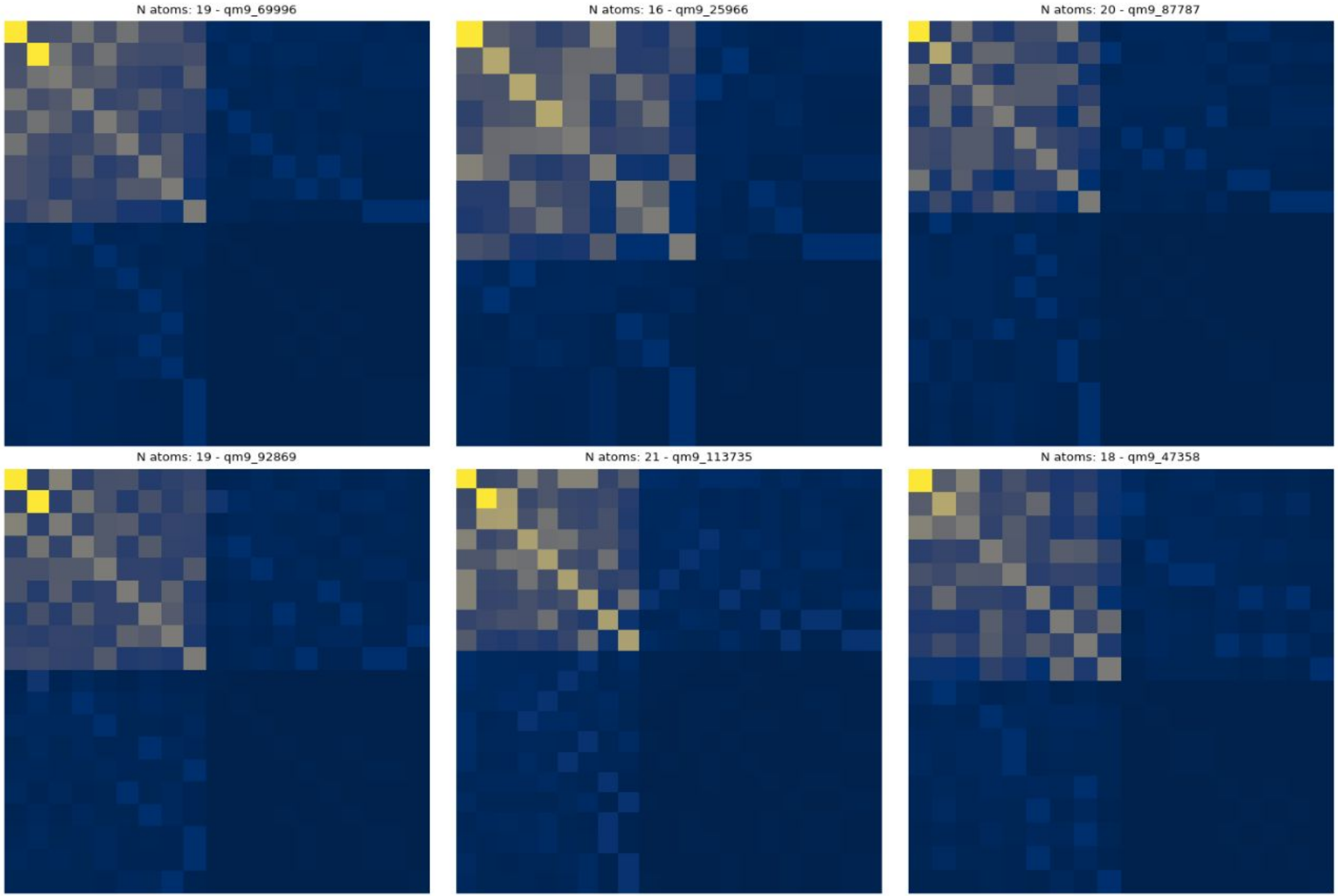
Z_i is the atomic number of atom i,

$\mathbf{R}_i$ is the position vector of atom i,

i, j are indices for atoms,

$\|\mathbf{R}_i - \mathbf{R}_j\|$ is the Euclidean distance between atoms i and j.

Example Coulomb matrices



N atoms: 19 - qm9_69996

N atoms: 16 - qm9_25966

N atoms: 20 - qm9_87787

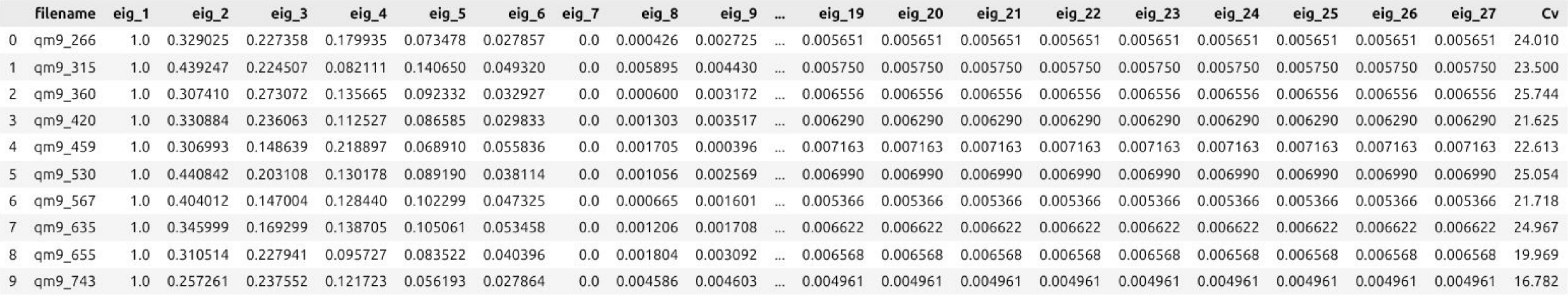
N atoms: 19 - qm9_92869

N atoms: 21 - qm9_113735

N atoms: 18 - qm9_47358

Calculation and padding of eigenvalues

- use eigenvalues of Coulomb matrix instead of the matrix itself as input to the models
- more efficient compressed version of the matrix
- apply padding to the eigenvalues to ensure that all eigenvalues have the same dimension



	filename	eig_1	eig_2	eig_3	eig_4	eig_5	eig_6	eig_7	eig_8	eig_9	...	eig_19	eig_20	eig_21	eig_22	eig_23	eig_24	eig_25	eig_26	eig_27	Cv
0	qm9_266	1.0	0.329025	0.227358	0.179935	0.073478	0.027857	0.0	0.000426	0.002725	...	0.005651	0.005651	0.005651	0.005651	0.005651	0.005651	0.005651	0.005651	0.005651	24.010
1	qm9_315	1.0	0.439247	0.224507	0.082111	0.140650	0.049320	0.0	0.005895	0.004430	...	0.005750	0.005750	0.005750	0.005750	0.005750	0.005750	0.005750	0.005750	0.005750	23.500
2	qm9_360	1.0	0.307410	0.273072	0.135665	0.092332	0.032927	0.0	0.000600	0.003172	...	0.006556	0.006556	0.006556	0.006556	0.006556	0.006556	0.006556	0.006556	0.006556	25.744
3	qm9_420	1.0	0.330884	0.236063	0.112527	0.086585	0.029833	0.0	0.001303	0.003517	...	0.006290	0.006290	0.006290	0.006290	0.006290	0.006290	0.006290	0.006290	0.006290	21.625
4	qm9_459	1.0	0.306993	0.148639	0.218897	0.068910	0.055836	0.0	0.001705	0.000396	...	0.007163	0.007163	0.007163	0.007163	0.007163	0.007163	0.007163	0.007163	0.007163	22.613
5	qm9_530	1.0	0.440842	0.203108	0.130178	0.089190	0.038114	0.0	0.001056	0.002569	...	0.006990	0.006990	0.006990	0.006990	0.006990	0.006990	0.006990	0.006990	0.006990	25.054
6	qm9_567	1.0	0.404012	0.147004	0.128440	0.102299	0.047325	0.0	0.000665	0.001601	...	0.005366	0.005366	0.005366	0.005366	0.005366	0.005366	0.005366	0.005366	0.005366	21.718
7	qm9_635	1.0	0.345999	0.169299	0.138705	0.105061	0.053458	0.0	0.001206	0.001708	...	0.006622	0.006622	0.006622	0.006622	0.006622	0.006622	0.006622	0.006622	0.006622	24.967
8	qm9_655	1.0	0.310514	0.227941	0.095727	0.083522	0.040396	0.0	0.001804	0.003092	...	0.006568	0.006568	0.006568	0.006568	0.006568	0.006568	0.006568	0.006568	0.006568	19.969
9	qm9_743	1.0	0.257261	0.237552	0.121723	0.056193	0.027864	0.0	0.004586	0.004603	...	0.004961	0.004961	0.004961	0.004961	0.004961	0.004961	0.004961	0.004961	0.004961	16.782

Models training

- compare multiple regression models - Linear regression, Random forest, SVR (Epsilon-support vector regression), K-nearest neighbors, XGBoost (Extreme gradient boosting) and Ridge regression

- perform hyperparameter optimization for each of the models in order to find the best hyperparameters

- use cross-validation to assess how well the models will generalize on unseen data

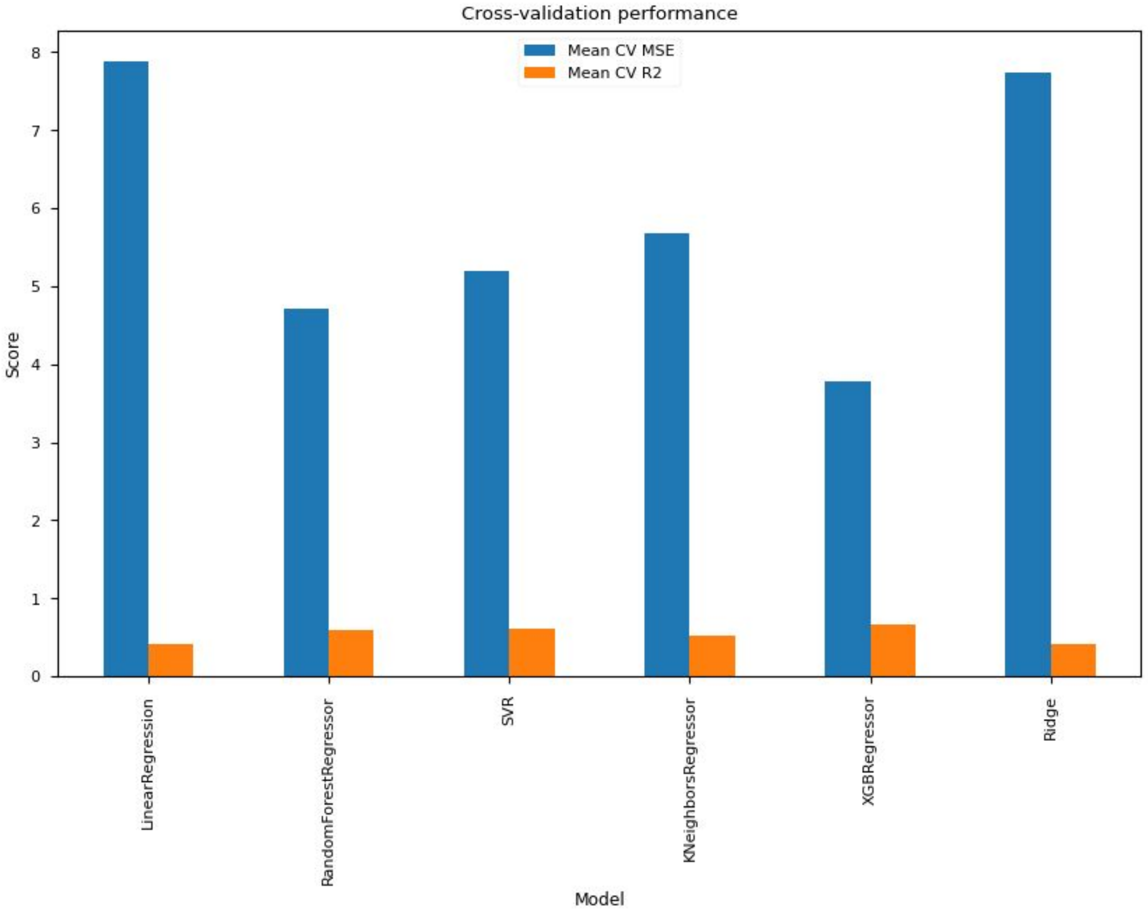
Mean squared error and R^2 loss functions

Compare models’ performance with MSE and R^2 loss functions:

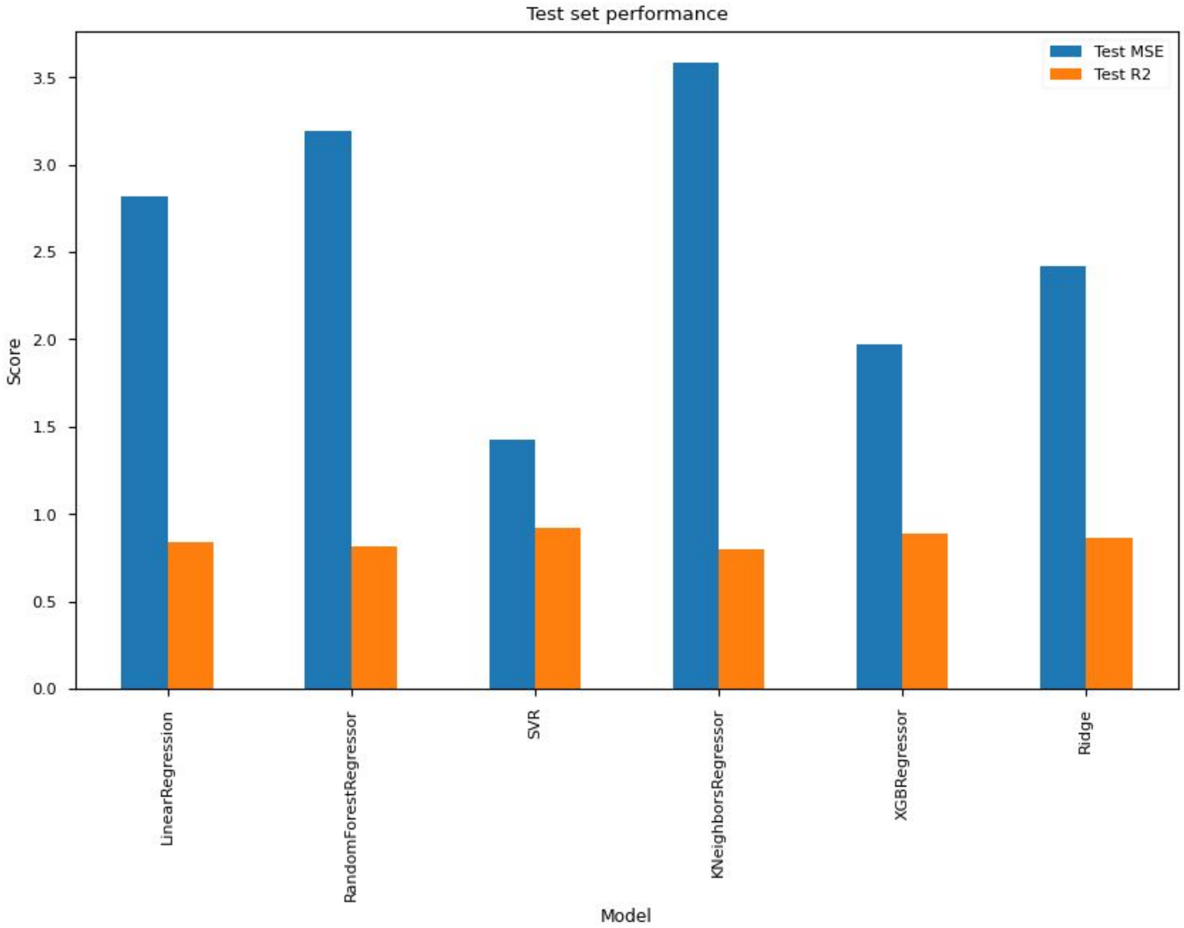
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}$$

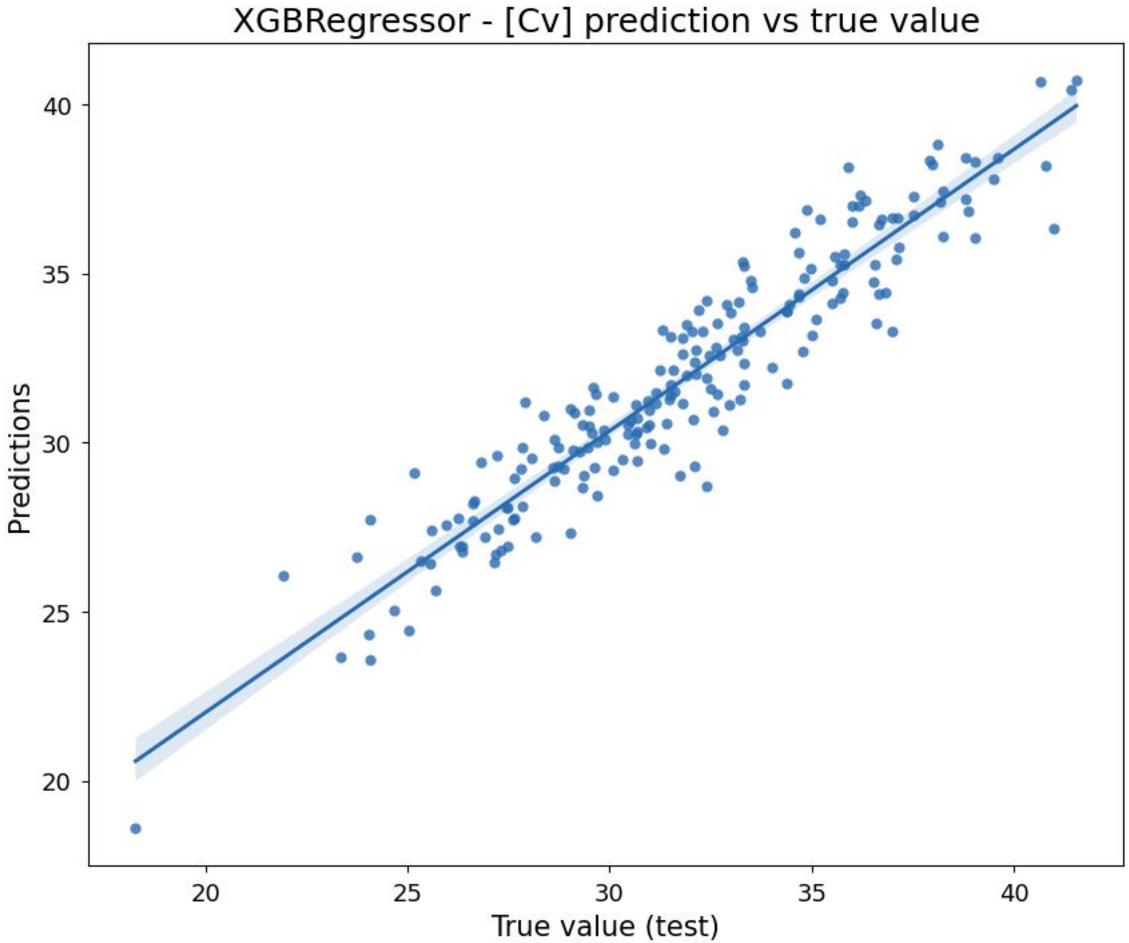
Heat capacity models cross-validation performance



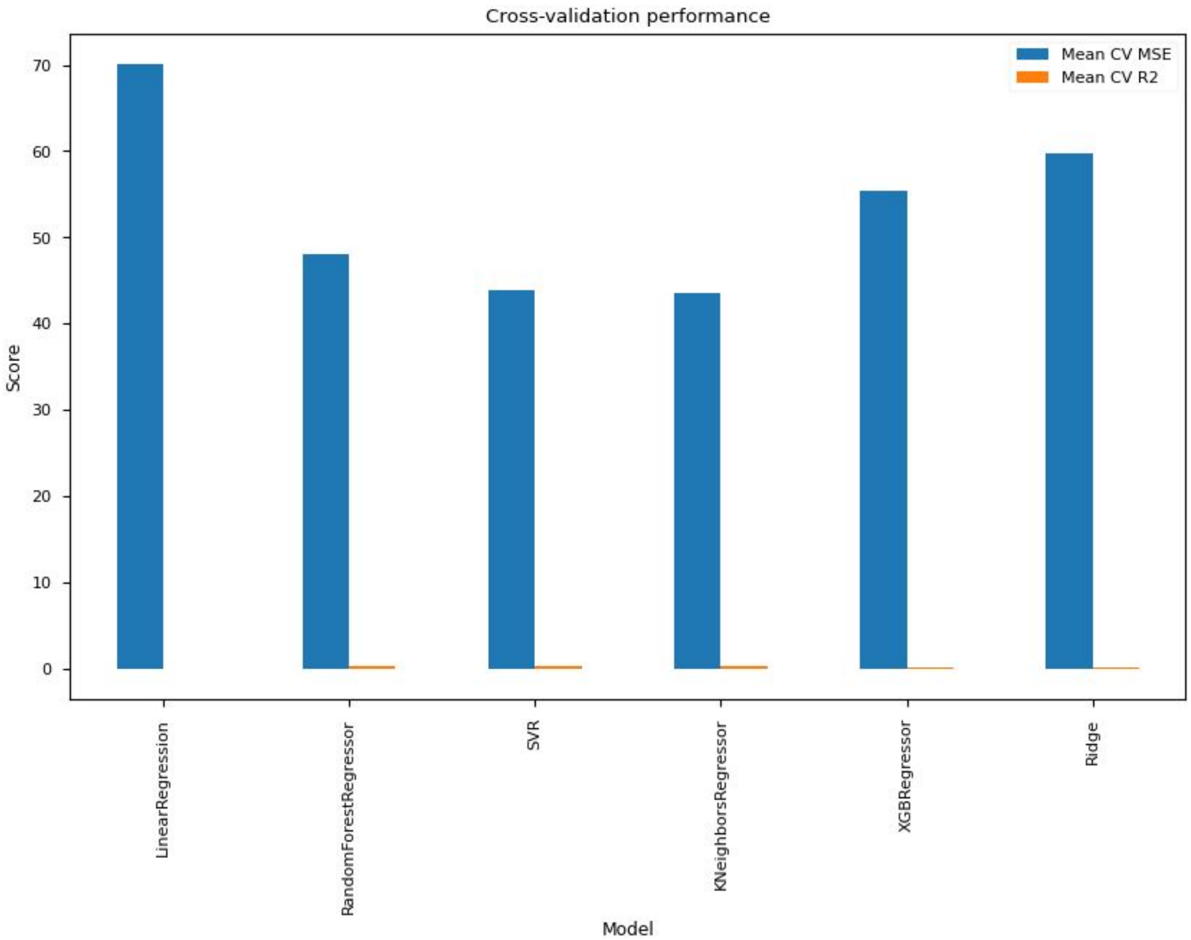
Heat capacity models test set performance



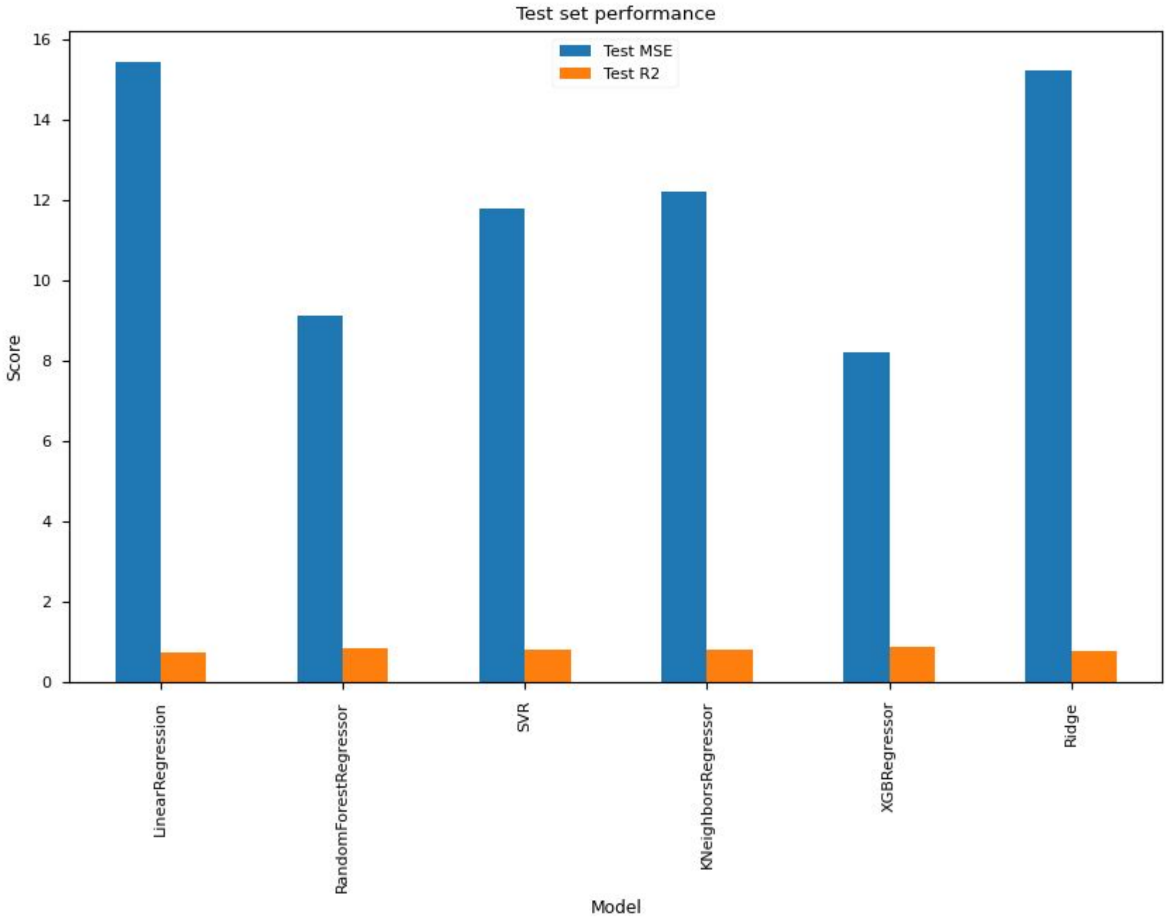
Heat capacity predictions



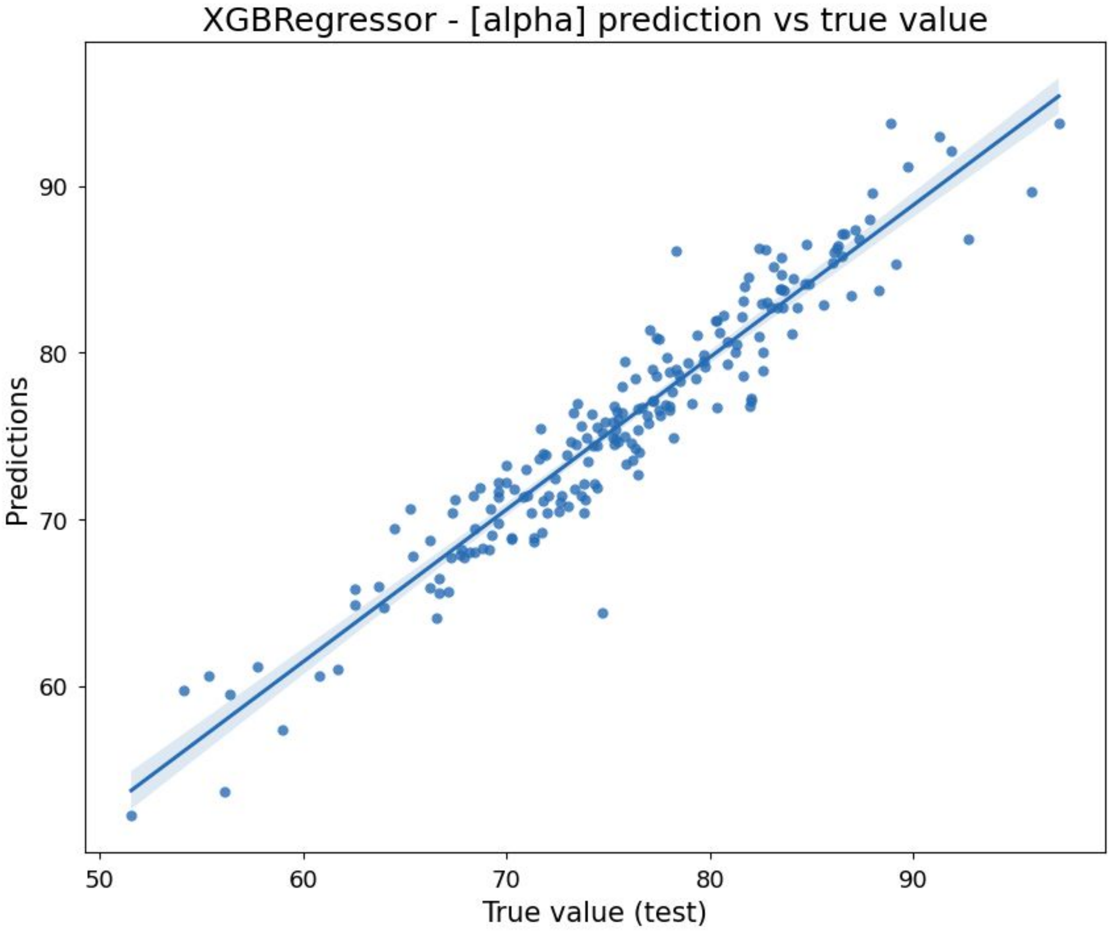
Isotropic polarizability cross-validation performance



Isotropic polarizability test set performance



Isotropic polarizability predictions



Conclusion

- data preprocessing and exploration on the QM9 dataset

- trained and compared multiple regression models on the eigenvalues of the Coulomb matrices to predict heat capacity and isotropic polarizability, performed hyperparameter tuning and cross-validation

- XGBoost is the best model for this task

- if we had more time: extend the dataset by adding molecules with more atoms and compare how these models would perform when the size of the molecules is significantly increased