Initial report

Group 210

2024-12-03

Data exploration

The GeckoQ dataset contains 26 columns. The variable log_pSat_Pa is the response variable, while the remaining columns are considered predictor variables. The ID column is not related to the response variable and is therefore excluded from the exploratory data analysis. The dataset includes 24 numerical predictors and one categorical predictor. The predictor MV is a continuous variable, while the rest of the numerical predictors are discrete variables. Parentspecies is the only categorical variable.

The following chart shows the correlation between all variables in the GeckoQ dataset. The correlation matrix calculates the linear relationship between pairs of variables. A positive value indicates a positive correlation between the variables, while a negative value indicates a negative correlation. A value of +1 represents a perfect positive correlation, whereas a value of -1 represents a perfect negative correlation. Weak correlations range from 0.1 to 0.3, moderate correlations from 0.3 to 0.5, and strong correlations from 0.5 to 1.

The chart highlights that the predictors NumHBondDonors and NumOfConf have a strong negative correlation with the response variable. Moderate negative correlations with the response variable can be observed in the variables NumOfAtoms, hyfroxyl (alkyl), carboxylic acid, and hydroperoxide. No variable exhibits a moderate or strong positive correlation.

In regression problems, collinearity between predictors can cause issues, as it becomes more difficult to isolate the individual effects of each predictor on the response variable. In the upper left corner of the correlation matrix, we observe strong correlations between several predictors. Collinearity among multiple predictors can lead to multicollinearity, where predictors are highly correlated with one another.

In addition to the correlation matrix, the Variance Inflation Factor (VIF) can be used to assess multicollinearity. The following chart displays the VIF values for all variables in the GeckoQ dataset.

Multicollinearity exists in a predictor if its VIF value exceeds 10. We can see that aldehyde, ketone, ester, and peroxide exhibit high multicollinearity. Additionally, several predictors have a VIF value of infinity (inf), which indicates perfect linear dependence. This suggests that one predictor can be entirely predicted by another predictor.

The boxplot can be used to analyze how observations are distributed for each feature. The boxplot chart can be found in the appendix (??). Several features exhibit outliers. These observations differ significantly from the rest of the dataset and may occur due to natural variability or measurement errors. Outliers can impact the results of machine learning models, and models trained on data with many outliers may produce misleading results.

In addition to the correlation matrix, scatterplots can provide insights into the relationships between variables. The correlation matrix shows the linear correlation between variables, while scatterplots reveal non-linear relationships between pairs of variables. The scatterplot, where the response variable is paired with each of the predictors, can be found in the appendix (??). The predictors x, y, and z, which had strong correlations in the correlation matrix, appear to exhibit similar non-linear relationships with the response variable. Additionally, the variable aromatic hydroxyl

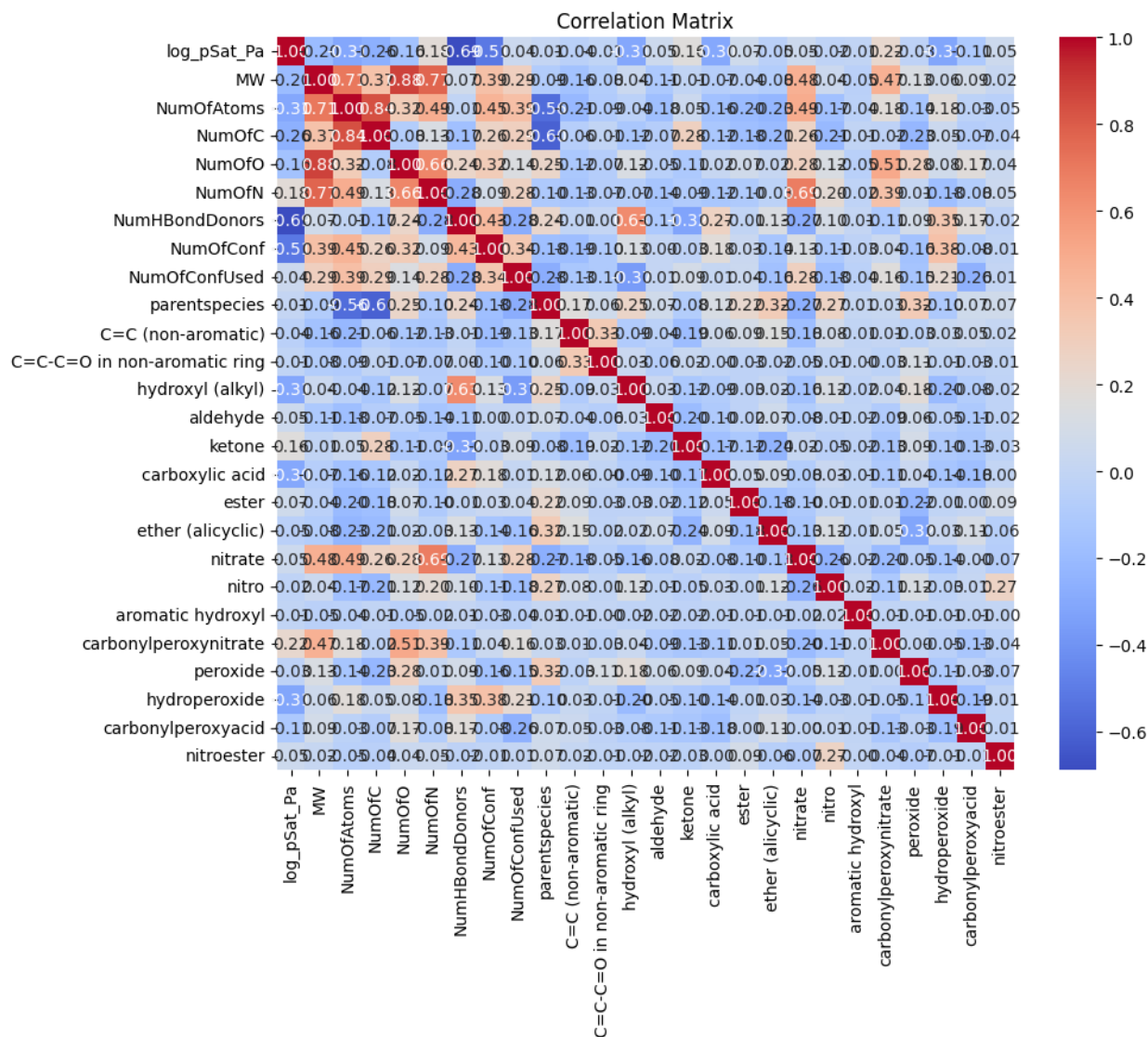Figure 1: Correlation matrix

	index	14	15	16	17	18	19	20	21	22	23	24
0	Feature	peroxide	ketone	ester	aldehyde	ether (alicyclic)	NumOfConf	parentspecies	C=C (non-aromatic)	nitroester	NumOfConfUsed	C=C-C=O in non-aromatic ring
1	VIF	19.867931	18.013132	12.24474	10.29383	5.256933	3.243523	2.961237	2.145305	2.068556	1.895979	1.191147

Figure 2: VIF-table

Pre-processing

The data was quite clean so it did not require a lot of pre-processing. The feature "parentspecies" had 210 null values in the training data and 33 null values in the test data. We replaced the null values with the mode of the feature, which was 'toluene' (?).

Since "parentspecies" was the only categorical variable and had only x unique values, we applied the One-Hot Encoding method to it. This method converts categorical data into x binary vectors, where the column representing the original value for a given row is assigned a value of 1, while all other columns are assigned 0.

Additionally, we standardized the data depending on the machine learning method being applied. Standardization ensures that features have a mean of 0 and a standard deviation of 1. This step is essential for machine learning models like SVM and k-NN, as well as dimensionality reduction techniques such as PCA. If the data contains features with different scales, the model's performance may degrade significantly.

- SVR Standardointi
- Parentspecies one-hot coding

Model selection

First, several different models were trained and evaluated for reference using the default parameters. The models were trained with all features and evaluated using a 10-fold cross-validation. A simple least squares linear regression model performed surprisingly well, but the best scores were attained by nonlinear models, such as Random Forest and Support Vector Regressor (SVR). We chose to focus on Random Forest and SVR, as we believed that these models had the most potential of the models tested. Next, feature selection and parameter tuning was performed for both models individually.

- Vertaile eka kaikki malleja niiden default parametrien kanssa
- Valitaan SVR, RF ja GradientBoosting
- Ajetaan SVR, RF ja GB 5-fold cross validation kaikilla feature selection metodeilla

Feature selection

Subset selection was performed using a forward stepwise selection approach. The features were first sorted by feature importance. The best subset of features was selected by first training a model with only the most important feature selected. Then, subsequent models were trained by adding one feature at a time in order of feature importance.
During the subset selection, each model was evaluated by performing a 5-fold cross-validation and scoring the models by the coefficient of determination.

The feature importances were computed using the permutation feature importance technique, where the values of a single feature are randomly shuffled and the importance of this feature is determined by the level of degradation of the model's performance.

Principal component analysis (PCA) was also done on the data set to see, whether selecting features this way would be
beneficial. Majority of the variance in the data was explained well without the last few components.

PVE and Cumulative PVE
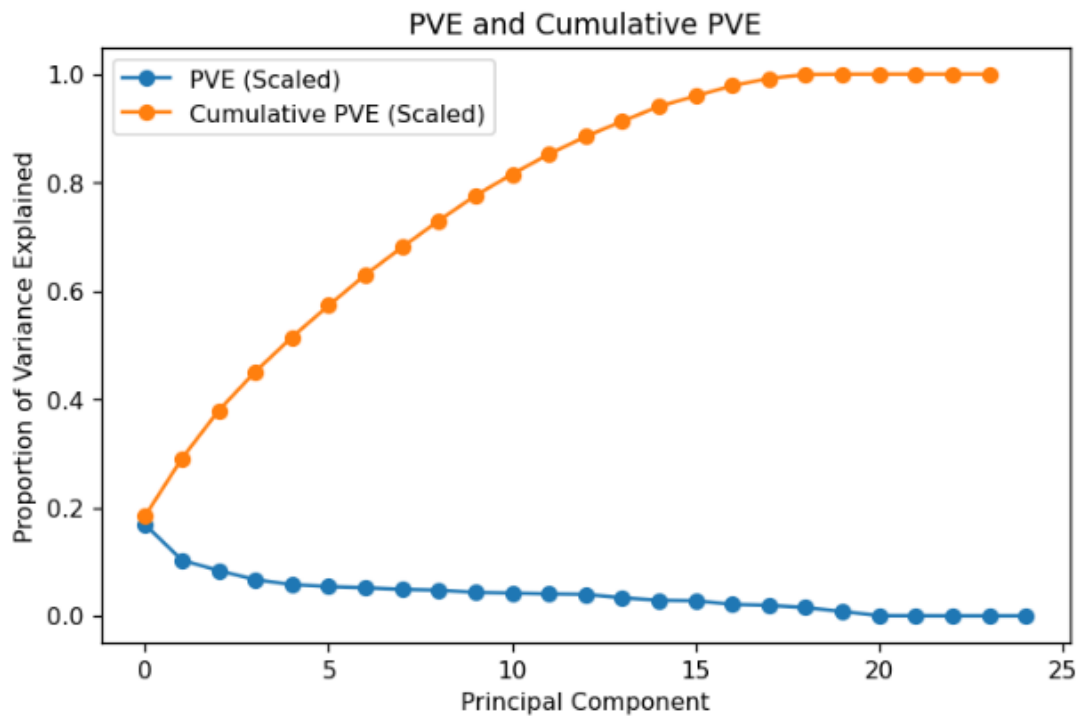
R²-scores were also tested for, by increasing number of principal components, using multiple models to see where we would start seeing diminishing returns.

R² Scores as a Function of Number of Components
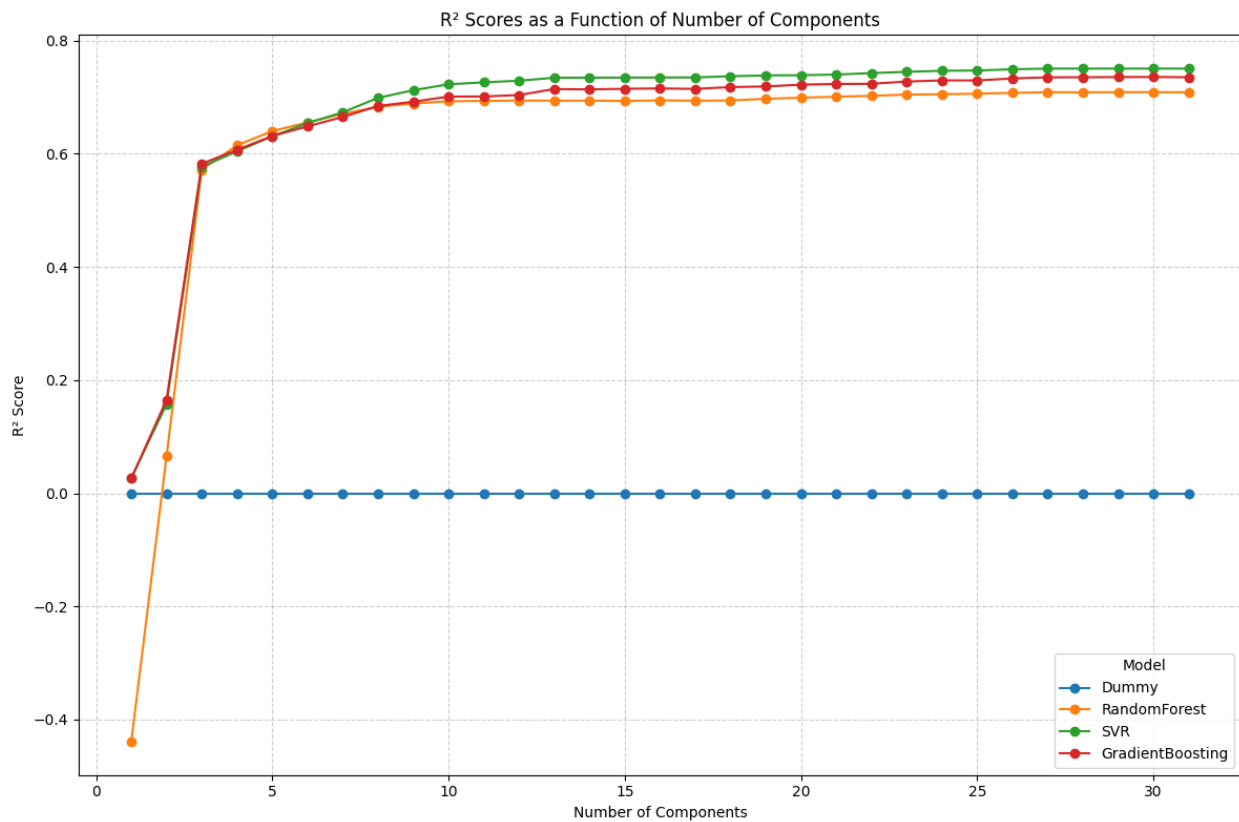
- PCA

- Stepwise selection

- Features importances

Performance estimation

Based on the testing done on the default models, three models, SVR, GB and RF, were chosen for further optimization. R^2 and MSE were chosen as the two scoring methods and each model's performance was estimated using 5-fold cross validation.

The performance of models was estimated using R^2-score and MSE. Each model was tested by performing 5-fold cross validation

on the training data set. The means of the scores of these cross-validation runs were used to compare the models.

SVR Parameter Tuning

Default SVR achieved best R2 performance of 0.7505 on PCA with 30 component and we started the hyperparameter with it. We also tried PCA with different number of components. The best result of 0.751277 was obtained with all 30 components. The exclusion of components only reduced R2 and MSE. We found it unusual that the best R2 and MSE performance was achieved by removing only one component, especially since the last 9 components explained only 1% of the model's variance. Consequently, it was decided to also hyperparameter-tune the second-best feature selection approach stepwise feature selection with 22 features. In the inial feature selection analysis, this approach achieved r2 performance of 0.7475. However, during hyperparameter tuning, the stepwise approach outperformed PCA, leading us to proceed exclusively with the stepwise approach for SVR.

Among all the default models, SVR was the second most computationally expensive model after polynomial model. Thus, hyperparameter tuning was only limited to parameters kernel, gamma, C and epsilon. RandomizedSearchCV and GridSearchCV were used to identify the optimal values for parameters. We started by comparing different kernel types, and noticed quickly that sigmoid, precomputer and linear performed badly comparing to 'rbf' and 'poly'. Kernel 'poly' was extremely sensitive to increases of gamma, and we decided to limit the gamma values to 0.5 in order to run our hyperparameter tuning in sufficient time range and avoid computational issues. In the results, rbf outperformed poly kernel. However, it is worth noting that excluding the degree parameter from the hyperparameter tuning might have impacted the performance of the 'poly' kernel.

We began optimizing the 'rbf' kernel by testing gamma values between 0.01 and 0.5, as gamma appeared to significantly affect computational performance. We observed a pattern where increasing gamma resulted in poorer R2 and MSE performance. The optimal value of gamma found range of 0.02-0.035. There was no clear of pattern or trend when altering parameter 'epsilon'. However, in one of our grid searches, an epsilon value of 0.678571 appeared in the best-performing parameter combinations. Parameter C was tested with values from range 2 to 10 and the results clearly indicated that best perfomance was achieved usually with the largest C values. The table below highlights the result from the last grid search, where parameter gamma ranged from 0.020-0.027, C parameter ranges from 8-10 and epsilon was fixed to 0.678571.

The best performance of 0.753750 was achieved with gamma was 0.022333 and C=10. In addition, we can observe that the top-performing models was same for R2 and MSE.

From all the default models, SVR was

- SVR, RF ja GB mallien parhaiten selittävät featuret valitaan hyperparametri tuunaukseen.

- Selitään kaikkien mallien hyperparametri tuunauksesta.

- Vertaillaan tuunauksen jälkeisiä parhaita malleja toisiinsa.

- Valitaan paras malli kolmen mallin väliltä.