Term project initial report

Predicting Saturation Vapour Pressure From Molecular Properties

Topias Pesonen 015251420

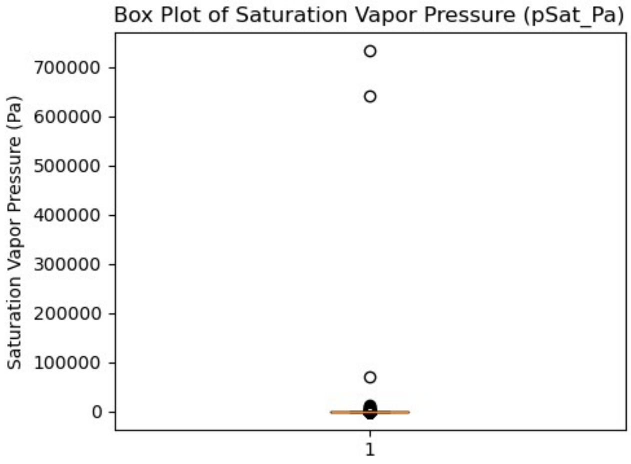
Project purpose, data exploration and preprocessing

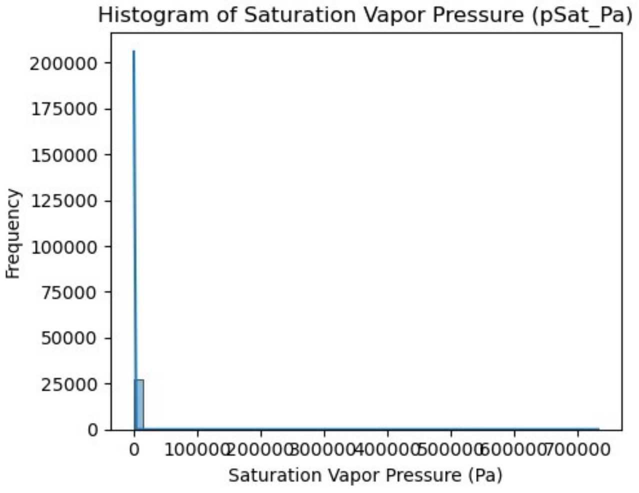
The purpose of the project is to predict the saturation vapor pressure using a regression model on the GeckoQ dataset. The dataset contains 26 columns, including the Id and the target variable. This leaves 24 columns as independent variables to predict the dependent variable, pSat_pa. We decided to go with the standard version of the project, without the topological fingerprints.

The tools for this project are the following: Python 3.10. with the packages pandas, numpy, matplotlib, seaborn, sklearn, scipy, xgboost.

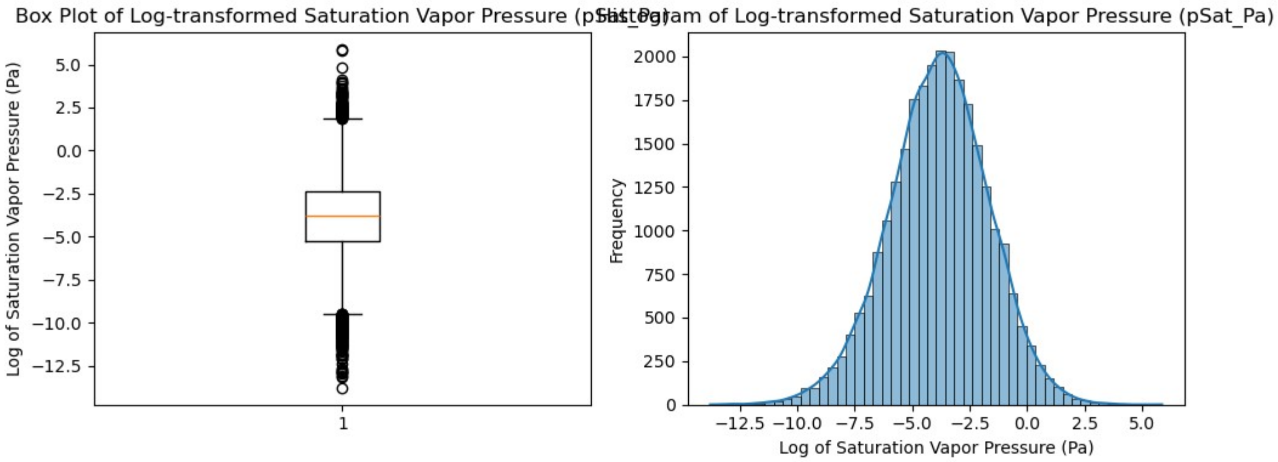
To start off, we inspected the shape and features of the data, and quickly found that the data consists of numerical variables and one categorical variable – parentspecies. This categorical variable had some nulls, which were filled as ‘None’ to avoid dropping them entirely. After this, we one-hot encoded the data, leaving us with dummy variables in place of the categorical parentspecies.

Next, we plotted the target variable:





There seem to be some outliers, and after checking the instructions, we found that using log10 values for the target is to be expected. After this transformation, the plots look a lot more nicer to deal with:



Now we have the target variable, pSat_Pa_log.

After some test runs, we decided to check for more outliers and if we could make the model a bit better. We decided to use Z-score to find some outlier rows, and by limiting the Z-score withing [-3,3], we have the final dataset.

Initial Modeling

Even though this is not a trivial regression problem, we started off with a simple linear regression. After splitting the data in to train and validation sets, we ran a LinearRegression from the sklearn-package. This gave us the first fit, wit a R2 score of 0.708 for the validation set.

The R2 score was looking quite good already, but understanding the nature of the data, we decided to apply some other algorithms: Random Forest, Gradient Boost and XGBoost.

From sklearn package we found GridSearchCV, which can be used to find the parameters that maximize given target, in this case R2-score. We applied this method to all three main models, and found the optimal parameters for them.

Additionally, we applied simple Principal Components Analysis to the data to see if we can get some better results with better feature selection. This was done with sklearn packages methods PCA and StandardScaler. We selected to retain 95% of the variance and select components as such, which left us with 21 features.

Results

The resulting R2 scores for the various models are in the table:

Model	R2 Score
Linear Regression	0.707675
Random Forest	0.720054
Gradient Boosting	0.740156
Random Forest with PCA	0.707120
Gradient Boosting with PCA	0.718062
XGBoost	**0.745027**
XGBoost with PCA	0.724619

The competition of pretty close, but XGBoost with all features was the winner. It also got the best score with the actual test data on Kaggle.

Final (best) model

XGBoost won, with a small margin. The final public score for the competition was 0.67702

Discussion

At this stage, it’s hard to tell if whether the results were good or not. Our domain knowledge on the subject matter is lacking, to say generously. Therefore the project was done in the dark, with understanding only on the format of the data available, and methods to make the predictions. XGBoost turned out to be the winner, which is not a huge surprise given its power and popularity.