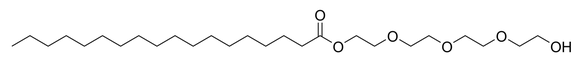
Prediction of melting points of organic molecules based on presence and number of functional groups



Group G1: Katharina Kirchmayr, Linus Krüer, Helena Windolf

Institute of Computer Science, University of Tartu

Introduction

Motivation

- Melting points are crucial in chemistry and chemical engineering for **material selection**, **process design**, compound identification and safety. Experimental measurements are reliable but **slow and expensive**, however, many predictive systems still lack sufficient accuracy.
- Two project members study **Chemistry**/**Biochemistry**, providing domain insight and motivating interest in thermophysical properties.
- The **Kaggle competition**[1] encouraged benchmarking, experimentation and competitive improvement.

Goals

- Develop a **machine-learning model** predicting melting points from functional-group compositions.
- Compare multiple algorithms, select the **best-performing model** and achieve a **low test error** (target: Mean Absolute Error (MAE) < 25%).
- Gain practical experience in **model tuning**, validation and evaluation.

Approach

- Distributed initial method exploration: **KNN, Random Forest, SVR, Linear Models, Neural Networks** tested individually.
- Compared baseline performance across methods to identify the **most promising candidates**.
- Applied systematic **hyperparameter tuning** (Randomized Search, Grid Search, Optuna) to refine selected models.
- Investigated **chemical structure patterns**, molecule shapes and functional-group distributions to better understand data characteristics.
- Visualized **correlations**, distributions and clustering patterns to guide further modelling decisions.
- Combined **parallel model exploration**, **RDKit-based feature engineering** and domain-informed analysis to iteratively improve prediction accuracy.

Data and Methods

Data

- Originates from Kaggle competition “Thermophysical property: melting point”.[1]
- Two files in CSV format: train set (2662 instances), test set (666 instances).
- Each instance corresponds to an **organic molecule** with a specific ID-number, its **name in SMILES notation** and the number of occurrences of 424 different **functional groups** (denoted by numbers, not chemical identity of the groups).
- Instances of train set additionally contain the melting points of the molecules.
- An **additional Bradley Thermophysical Datasets**[2,3] was used to augment training data and to allow the model to learn more robust melting point relationships from a broader distribution of molecular structures.

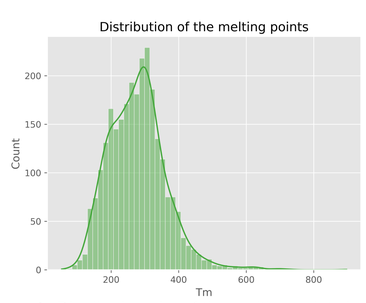


Figure 1: Distribution plot of the melting points in the test data.

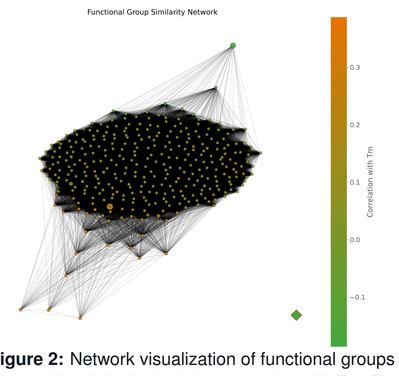


Figure 2: Network visualization of functional groups clustered by similarity in their correlation with Tm. Two groups are connected if they occur in molecules with very similar Tm properties (cos-similarity > 0.85).

Methods

- **Machine learning models** (Scikit-learn)
  - **K-Nearest Neighbours Algorithm (KNN)**
  - **Random Forest (RF)**
  - **Support Vector Regression (SVR)**
  - **Linear Regression**
  - **Neural Networks**
  - **Gradient Boosted Trees** (XGBoost)
- Optimization and **hyperparameter tuning** (Scikit-learn, Optuna)
  - **Randomized search**: used when a large parameter space had to be tested and as a first-pass to reach a suitable range.
  - **Grid search**: mainly used method for hyperparameter tuning, exhaustive search of all possible combination of defined parameter grid.
  - **Optuna**: used to obtain better optimization results beyond grid search.
  - **Cross-validation**: done in combination with the hyperparameter tuning methods, to prevent overfitting and ensure good generalization on the test data.
- **Feature engineering** (RDKit):
  The **RDKit** module was used to compute five **physicochemical descriptors** (molecular weight, lipophilicity, polarity, number of hydrogen bond donors, number of hydrogen bond acceptors) and the **Morgan fingerprint** (represents the structure of a molecule as a binary vector) from the SMILES names. Together, these made up the new feature space, which was also used to train the different models.
- **Data visualization** (Seaborn, Matplotlib):
  Visualization methods were employed for exploration of the data and to find the correlation of individual groups with the melting point.

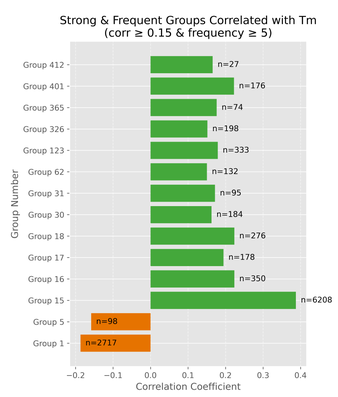


Figure 3: Correlation bar plot of frequent groups with the highest correlation with Tm.

Results

The **results** of the different approaches with the used settings are shown in **Table 1**. **Figure 4** shows the results in a bar plot for better **comparison**.

Table 1: Used methods with the used settings and their best Mean Absolute Error (MAE) (*marker for feature engineering).

Method	MAE	Settings
Random Forest	33.39	n_estimators = 698, max_depth = 48, min_samples_split = 10, max_features = 0.7, criterion = "absolute_error"
	28.05*	n_estimators = 600, max_depth = 40, max_features = 0.2
KNN	40.29	n_neighbors = 9, weights = "distance", p = "manhattan"
	39.79*	n_neighbors = 11, weights = "distance", p = "manhattan"
Linear Regression: Lasso	24.25	5-fold CV (shuffle=True, random_state=42), alpha = 0.001
Ridge	24.56	5-fold CV (shuffle=True, random_state=42), alpha = 0.1
SVR	29.16	kernel = "rbf", gamma = "scale", epsilon = 0.5, C = 1000
	32.40*	kernel = "rbf", gamma = "auto", epsilon = 1, C = 1000
Neural Networks	35.25	Architecture: Input(n_features) → Dense(512, relu) → Dense(256, relu) → Dense(128, relu) → Dense(1) Batch size = 128, Optimizer = Adam(learning_rate = 1e-3) Loss = MSE, Epochs = 300 Early stopping: patience = 30, restore_best_weights = True Dataset: shuffle(5000), batch(128), prefetch(AUTOTUNE)
	34.95*	Architecture: Dense(512, relu) → Dense(256, relu) → Dense(128, relu) → Dense(1) Batch size = 128, Epochs = 300 Early stopping: patience = 30, restore_best_weights = True Dataset: shuffle(5000), prefetch(AUTOTUNE)
Gradient-Boosted Trees	7.73*	n_estimators = 1500, max_depth = 8, learning_rate = 0.03, subsample = 0.8, colsample_bytree = 0.8, reg_alpha = 1.0, reg_lambda = 2.0, objective = "reg:absoluteerror", tree_method = "hist"

The Random Forest, SVR and Neural Network models all had a similar performance with MAEs of around 30%. KNN had a higher MAE, while Linear Regression provided a lower one. Using the engineered features instead of the original ones improved scores in the case of RF, KNN and Neural Networks; for SVR it was the other way around. In all cases, however, the MAEs from models trained with the original vs. the engineered features were still in a similar size range. By far, the best predictions were achieved by the Gradient-Boosted Trees trained on the engineered features (MAE of around 8%).

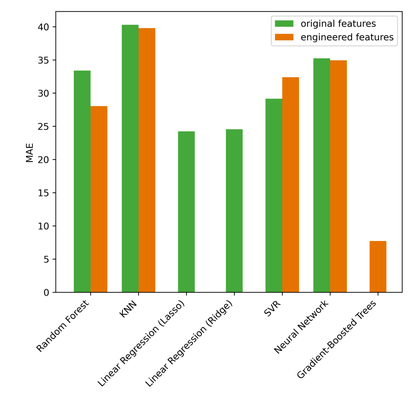
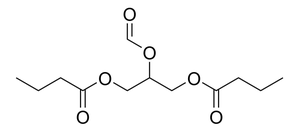


Figure 4: Comparison of the models by their best MAE.

With the use of the Gradient-Boosted Trees method, we achieved the **62$^{\text{th}}$ position** on the leaderboard at the time of completion (5$^{\text{th}}$ of December 2025; scan QR code for current ranking).



Scan me:

Conclusion

- The **best achieved MAEs** can give an indication about which models work best on this data, but **cannot be taken as absolute**, as further optimization might have been possible with more time.
- The models covered in the lecture (apart from Gradient-Boosted Trees) only were able to achieve MAEs between 24 and 35, which shows the general **complexity of predicting melting points**. The melting point is being influenced by numerous factors, some of which are hard to compute from the given data. Additionally, the spare data may have altered the outcomes as well (too few examples of certain functional groups).
- Using the original features (i.e. the raw counts of 424 functional groups) forces the models to act similarly to group-contribution approaches: the melting point is inferred only from the summed contributions of individual groups. This **ignores structural context** (connectivity, arrangement, intramolecular interactions), which explains why feature engineering based on RDKit descriptors and fingerprints often improves performance.
- While the Bradley datasets added a large number of **additional melting points** ($\sim$ 25 000) and improved model training, it **did not provide meaningful structural context**; further datasets and **domain-specific curation would be required** to enrich the structural information.

References

[1] Kaggle. *Melting Point Prediction Competition*. https://www.kaggle.com/competitions/melting-point/overview. Accessed: 2025-11-21. 2025.

[2] Jean-Claude Bradley, Antony Williams, and Andrew Lang. *Jean-Claude Bradley Open Melting Point Dataset*. http://dx.doi.org/10.6084/m9.figshare.1031638. Accessed: 2025-11-30. 2014.

[3] Jean-Claude Bradley, Antony Williams, and Andrew Lang. *Jean-Claude Bradley Double Plus Good (Highly Curated and Validated) Melting Point Dataset*. https://doi.org/10.6084/m9.figshare.1031637. Accessed: 2025-11-30. 2014.