Molecular Similarity Prediction Model of FDA Approved Drugs

by John Hopkins

Made with Gamma

Application

- The model can be used by students for research and projects.
- Chemists and researchers seeking similarity data.

Made with Gamma

Data obtained from ZINC15

- "Resources." DISI, . 9 Feb 2017, 18:30 UTC. 3 Sep 2023, 08:39 **http://wiki.docking.org/index.php?title=Resources&oldid=9919**.

Made with Gamma

- The data consisted of 1615 molecules in SMILES format and ZINCID's
- The simplified molecular-input line-entry system (SMILES) is a specification in the form of a line notation for describing the structure of chemical species using short ASCII strings.
- Algorithms have been developed to generate the same SMILES string for a given molecule; of the many possible strings, these algorithms choose only one of them. This SMILES is unique for each structure, although dependent on the canonicalization algorithm used to generate it.
- ASCII - is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices.

> - **https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system**
> - **https://en.wikipedia.org/wiki/ASCII**

Made with Gamma

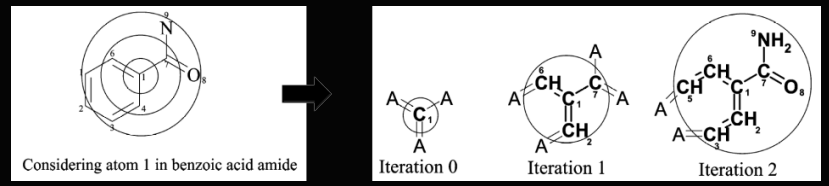
Data Cleaning and Pre-Processing

The RDKit package was used for data cleaning and pre-processing

- Validity Check - Ensures that all SMILES strings are valid according to the SMILES notation rules
- Canonicalization - A process of converting SMILES strings into a standardized form
- Sanitization - The process of removing non-chemical characters or artifacts from SMILES strings
- Fingerprinting
- Tanimoto Similarity Matrix

Made with Gamma

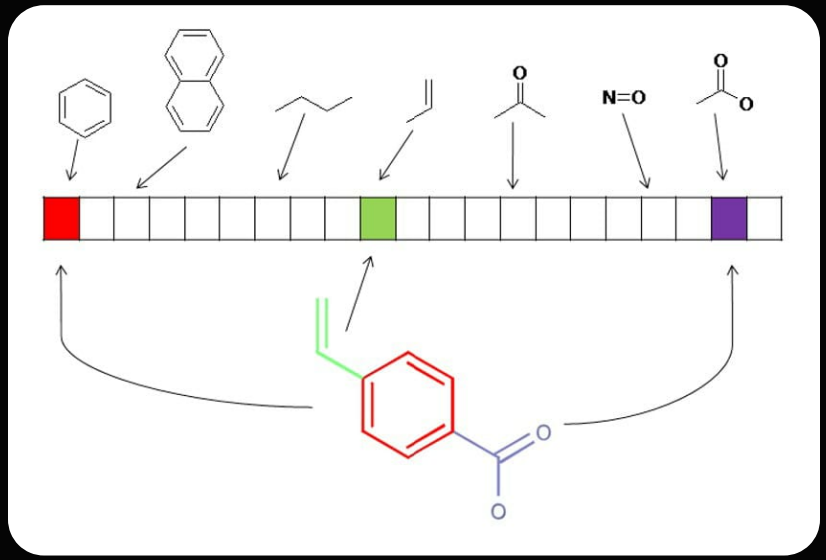
Molecule Fingerprinting



Fingerprints

Morgan/Circular Fingerprints

- Numbering invariant atom information into an initial atom identifier
- Identifiers are generated independently of previous identifiers and the intermediate results are discarded
- The iteration process is continued until every atom identifier is unique
  - **https://oi.readthedocs.io/en/latest/bioinfo/mol_fp/ecfp.html**



- The molecule above is represented by a series of binary bits showing the presence or absence of particular substructures.
- 1615-bit fingerprints were selected in order to encompass the entire similarity matrix and for ease of DataFrame and array handling. The common bit sizes used are 2048 or 1024.

Made with Gamma

Tanimoto Similarity

- Also known as the Tanimoto coefficient or Jaccard index.

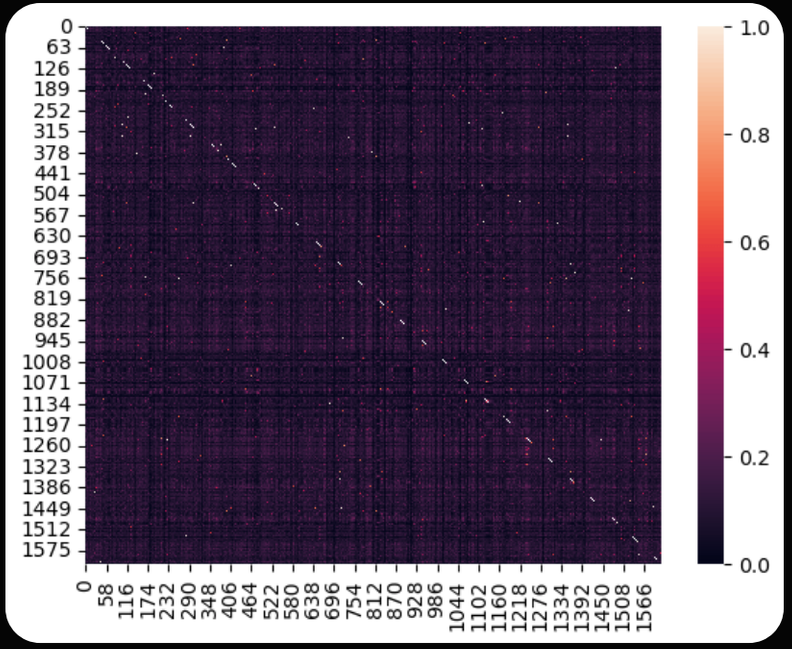
- The formula calculates the ratio of the number of common elements between two sets to the total number of elements in both sets. The resulting value ranges between 0 and 1, with 1 indicating that the two sets are identical and 0 indicating that they have no common elements.

- Tanimoto similarity = C / (A + B - C)

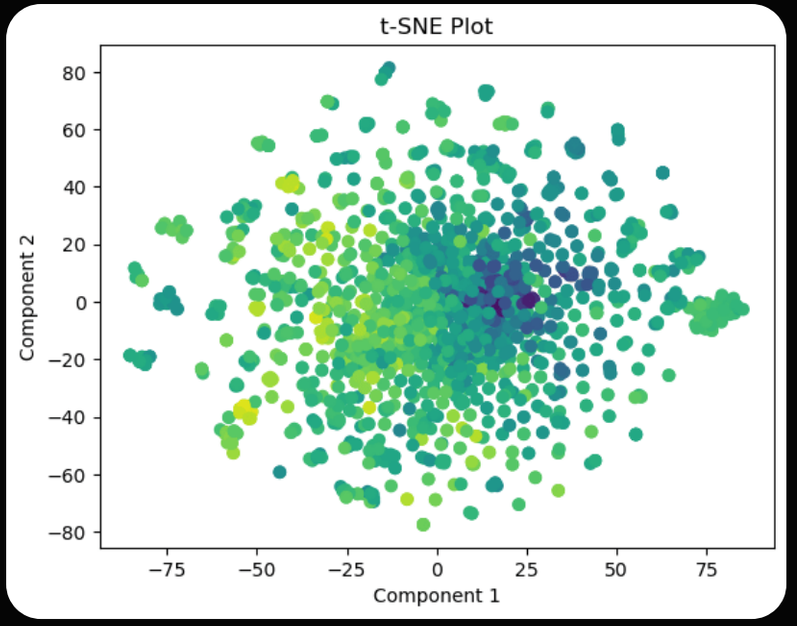
- Where A and B are the total number of bits in the two fingerprints being compared, and C is the number of bits that match between the two fingerprints.

Made with Gamma

Data Visualizations



- In the visualization of the similarity matrix, the dark purple with speckles of different hues suggests that there are multiple clusters or groups of molecules in the dataset that have different similarity profiles.
- This could indicate that there is some level of structural diversity in the dataset and that the molecules may have distinct chemical features or properties.



t-SNE Plot (Hamming Distance)

- The dense center, like an exploding star, could suggest that there is a common group of data points that are very similar to each other based on the Hamming distance. As you move outwards from the center, the data points become less similar to each other.

Made with Gamma

Model Analysis

The Pycaret package was used for model creation, tuning, and analysis

- Set-up comprised of setting imputation to drop rows containing missing values and pre-processing off (no transformations are applied).
- The mean similarity scores were selected as the target variable and the binary fingerprints were selected as the independent variables.
- The compare_model feature was used to run 10-fold cross-validation (cv) on numerous models and rank on R^2 to choose the best performer.
- Tuning of hyperparameters was performed using a 10-fold cv with both RandomGridSearch and Optuna (Bayesian) optimizations.

Made with Gamma

Criteria

The models were judged by the following criteria:

R^2, Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE)

- The value of R^2 is calculated by dividing the explained variance by the total variance. **R^2 values close to 1 indicate that the model is a good fit for the data,** while values close to 0 indicate that the model does not fit the data well.
- RMSE - calculates the square root of the average of the squared differences between predicted and actual values. It is expressed in the same units as the dependent variable, and **lower RMSE values indicate better predictive performance.**
- MAPE - is used to calculate the average absolute percentage difference between the forecasted value and the actual value. It's expressed as a percentage, and **lower MAPE values indicate higher accuracy.**

Made with Gamma

Results

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.0032	0.0000	0.0044	0.9515	0.0041	0.0373	3.8900
catboost	CatBoost Regressor	0.0033	0.0000	0.0045	0.9509	0.0041	0.0387	12.3030
huber	Huber Regressor	0.0029	0.0000	0.0046	0.9481	0.0042	0.0377	4.4040
xgboost	Extreme Gradient Boosting	0.0032	0.0000	0.0048	0.9427	0.0044	0.0377	4.3310
ridge	Ridge Regression	0.0031	0.0000	0.0049	0.9426	0.0044	0.0393	2.8270
gbr	Gradient Boosting Regressor	0.0044	0.0000	0.0056	0.9232	0.0051	0.0503	4.6110
omp	Orthogonal Matching Pursuit	0.0042	0.0000	0.0057	0.9227	0.0052	0.0525	2.8140
rf	Random Forest Regressor	0.0040	0.0000	0.0057	0.9198	0.0052	0.0481	3.6490
br	Bayesian Ridge	0.0039	0.0000	0.0063	0.9017	0.0057	0.0460	4.3550
et	Extra Trees Regressor	0.0045	0.0001	0.0074	0.8631	0.0068	0.0548	3.6780
dt	Decision Tree Regressor	0.0047	0.0001	0.0078	0.8502	0.0071	0.0574	2.8390
ada	AdaBoost Regressor	0.0069	0.0001	0.0084	0.8296	0.0076	0.0804	4.1990
lar	Least Angle Regression	0.0103	0.0003	0.0147	0.3068	0.0133	0.1172	3.3470
knn	K Neighbors Regressor	0.0130	0.0003	0.0177	0.2249	0.0163	0.1438	2.8300
lasso	Lasso Regression	0.0154	0.0004	0.0204	-0.0019	0.0188	0.2132	2.7570
en	Elastic Net	0.0154	0.0004	0.0204	-0.0019	0.0188	0.2132	2.7950
llar	Lasso Least Angle Regression	0.0154	0.0004	0.0204	-0.0019	0.0188	0.2132	2.7810
dummy	Dummy Regressor	0.0154	0.0004	0.0204	-0.0019	0.0188	0.2132	2.7480
par	Passive Aggressive Regressor	0.0683	0.0049	0.0697	-10.9370	0.0653	0.7056	2.7990
lr	Linear Regression	0.1248	0.0751	0.2045	-194.7046	0.1253	1.3797	4.1870

LightGBM (Gradient Boosting Machine)

- Of all the models tested using Pycaret's compare_models feature LightGBM was the clear winner in all metrics.
- After a tuning search to try and optimize the model; both RandomGridSearch and Optuna methods were performed, the original model remained the best performer.
- **The results were: R² = 0.95, RMSLE = 0.0041, MAPE = 0.0373**

Boosted Model

```
# ensemble model
boosted_model = ensemble_model(best_model, method = 'Boosting', choose_better = True)

type(boosted_model)
# >>> sklearn.ensemble._weight_boosting.AdaBoostRegressor

print(boosted_model)
```

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.0026	0.0000	0.0036	0.9663	0.0033	0.0298
1	0.0031	0.0000	0.0042	0.9713	0.0039	0.0509
2	0.0031	0.0000	0.0043	0.9588	0.0039	0.0354
3	0.0031	0.0000	0.0045	0.9421	0.0041	0.0344
4	0.0029	0.0000	0.0043	0.9531	0.0039	0.0304
5	0.0030	0.0000	0.0043	0.9637	0.0039	0.0330
6	0.0029	0.0000	0.0043	0.9456	0.0039	0.0325
7	0.0033	0.0000	0.0047	0.9409	0.0043	0.0364
8	0.0026	0.0000	0.0036	0.9624	0.0032	0.0284
9	0.0029	0.0000	0.0040	0.9645	0.0037	0.0325
Mean	0.0030	0.0000	0.0042	0.9569	0.0038	0.0344
Std	0.0002	0.0000	0.0003	0.0102	0.0003	0.0060

```
AdaBoostRegressor(estimator=LGBMRegressor(device='gpu', n_jobs=-1,
                                          random_state=158),
                  n_estimators=10, random_state=158)
```

AdaBoost Regressor

- Ensemble methods were attempted to increase model performance.
- The AdaBoostRegressor was able to give a slight improvement in all metrics.

R² = 0.96, RMSLE = 0.0038, MAPE = 0.0344

- Prediction based on hold-out data results

```
# Predict on hold-out
predict_model(boosted_model)
```

Python

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	AdaBoost Regressor	0.0028	0.0000	0.0039	0.9612	0.0036	0.0312

Made with Gamma

LightGBM (Light Gradient Boosting Machine)

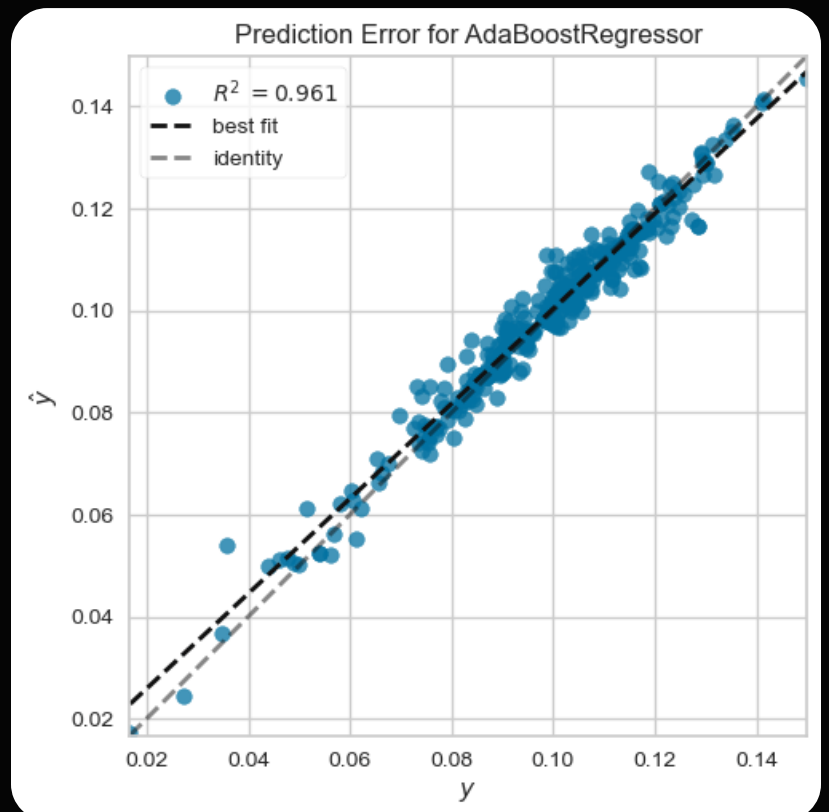
- A open-source, gradient-boosting framework that uses a tree-based learning algorithm. It uses the leaf-wise growth strategy that selects the leaf with the maximum delta loss to grow, which helps it to achieve higher accuracy.
- Each tree is trained on the error residuals of the previous trees, applying an iterative gradient descent process that progressively improves the model accuracy.

AdaBoost (Adaptive Boosting)

- An ensemble method that combines multiple weak learners to create a stronger and more accurate predictor.
- Each weak learner is trained using only a subset of the training data, and the weights of the training examples are adjusted to give more emphasis to the examples that are harder to predict.
- The final prediction is a weighted sum of all the weak learner predictions, where the weight of each weak learner reflects their predictive power.

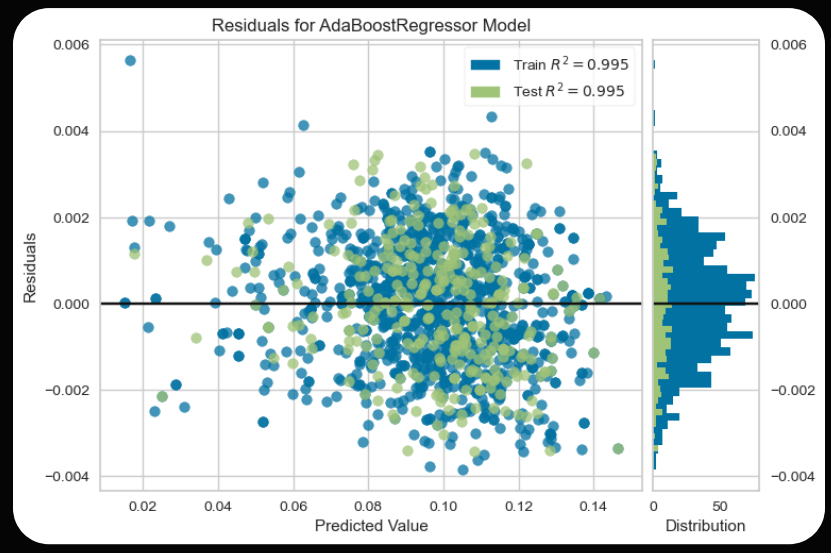
Made with Gamma

Results Visualizations



Prediction Error Plot

- A prediction plot is used to analyze the output of a model.
- In molecular similarity prediction, the plot displays the actual versus predicted values of molecular similarity scores.
- The plot shows some bunching in the upper right quadrant, this indicates that the model is slightly overestimating the molecular similarity scores.



Residuals Plot

- A residuals plot is used to visualize the goodness of fit of a regression model, which evaluates whether the model accurately captures the patterns in the data.
- The residuals plot shows the difference between the actual value and the predicted value of the target variable on the y-axis, and the predicted values of the target variable on the x-axis.
- In general, a plot showing no clear pattern and some center bunching suggests that the model's performance is relatively good at predicting the mean value of the target variable but may struggle to predict values that deviate significantly from the mean.

Made with Gamma

An Ensemble Method

- A process of combining multiple models to improve the accuracy and robustness of the predictions. They help in reducing overfitting, increasing stability, and improving generalization performance.

Boosting

- Is a type of ensemble method in which multiple weak models are combined to form a strong model. At each iteration, a new weak model is trained to correct the errors made by the previous models.

Ensembling LigthGBM and AdaBoost

- LightGBM is a popular gradient boosting algorithm known for its speed and scalability, AdaBoost is an adaptive boosting algorithm known for its ability to handle noisy data and reduce bias and variance.
- Combining the strengths of both algorithms and potentially improving the performance and stability of the overall model.
- Using LightGBM as the base model and employing AdaBoost was able improve the model based on all metrics chosen.

Made with Gamma

Made with Gamma