Structural Bioinformatics

Machine learning for RING contact classification

Andrea Auletta
Marco Bernardi
Niccolò Zenaro

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

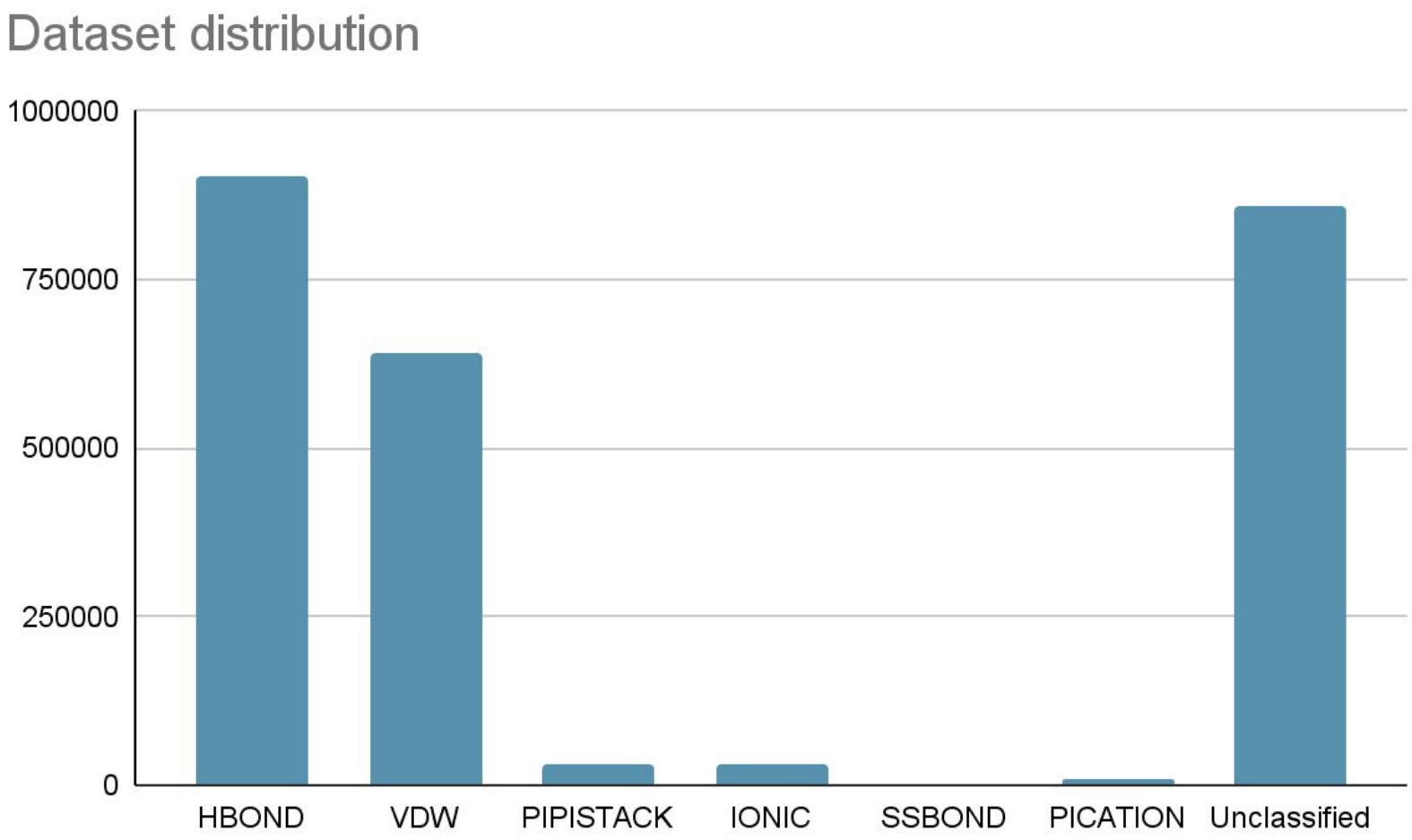
Academic Year 2023/2024

DIPARTIMENTO
MATEMATICA

Introduction

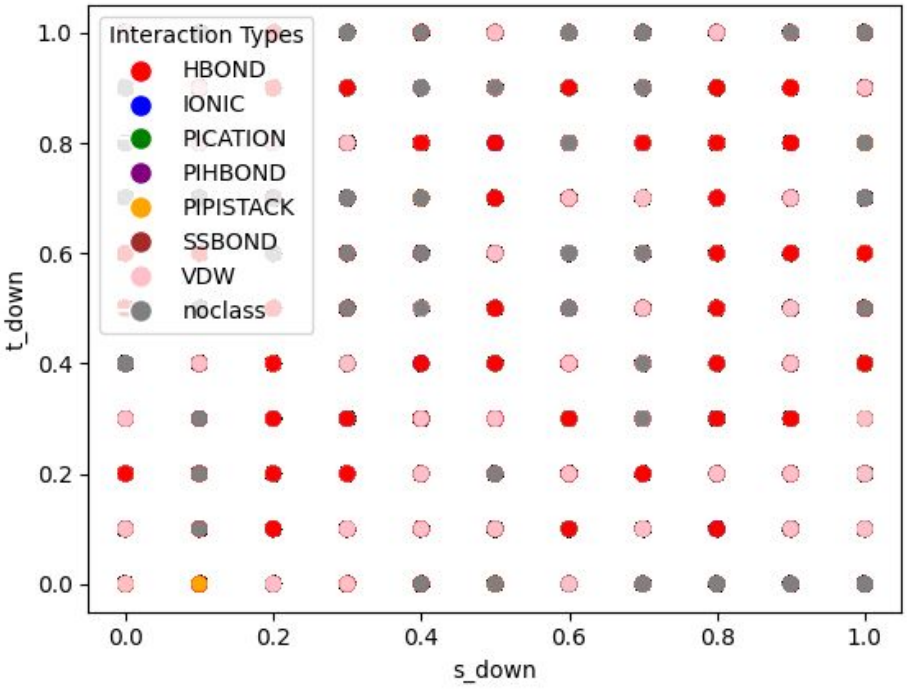
- Predictive model which classifies contacts between atoms
- The classification of the contacts refers to the RING classification
- Machine learning models were used instead of looking at geometrical and physical properties of atoms

Dataset

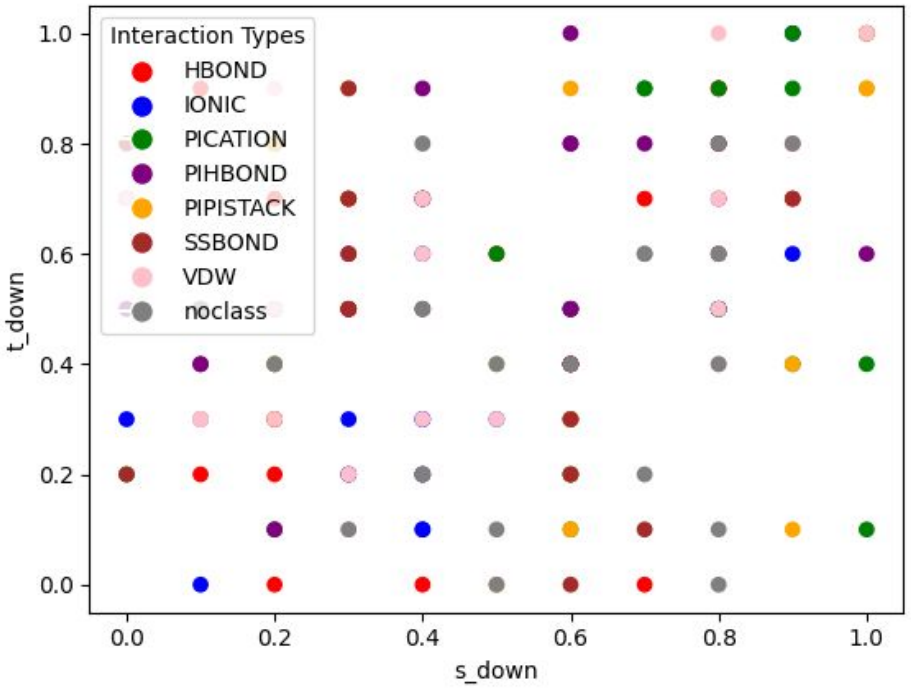

Dataset distribution
1000000
750000
500000
250000
0
HBOND
VDW
PIPISTACK
IONIC
SSBOND
PICATION
Unclassified

Challenges

- Imbalanced Dataset
- The data are not easily separable



Scatter plot of the original dataset



Scatter plot of the subsampled dataset

Pre-processing

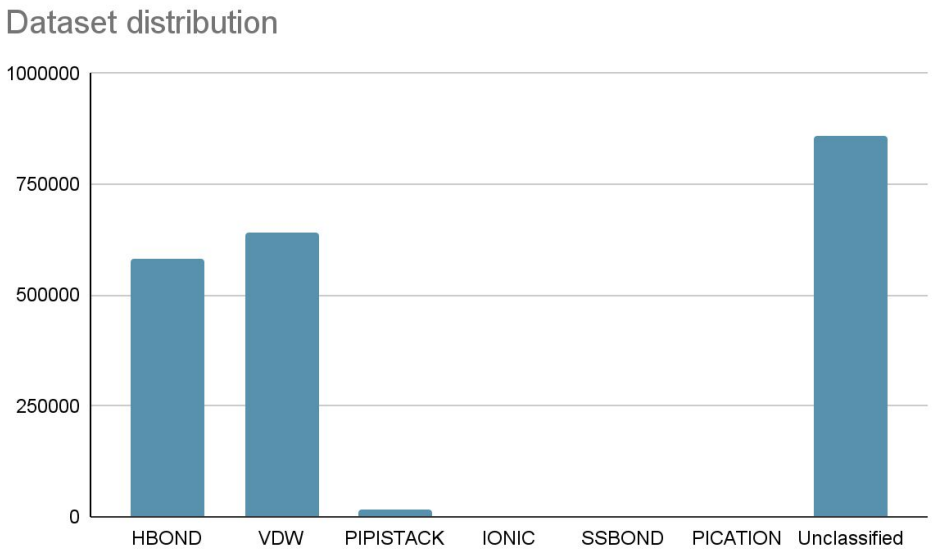
- Undersampling:
  - Fast but may result in the model's inability to fully learn the distinguishes features of the majority classes
- Class merging:
  - Fast but the sizes are still too unbalanced
- Oversampling with SMOTE:
  - Slower method but more effective

SMOTE - Synthetic Minority Over-sampling TEchinque

- Data augmentation technique
- Generates new samples interpolating between the samples of the minority class taken into account
- It uses the k-nn to choose a set of points from which the algorithm will interpolates for new sample

Pre-processing

- Maintaining one type of interaction for each contact
- Data can fit different definition of contacts
- RING can return all the types of contacts which the data fit



Histogram of the dataset maintaining one type of interaction for each sample

Neural Network

- The model consists of four fully connected layers
- The hidden layers use ReLU activation functions
- The final output layer uses a softmax activation function to produce probability distributions across the target classes, suitable for multi-class classification

Neural Network

- Dropout layers are applied after each hidden layer to prevent overfitting
- Mini-batch algorithm has been used for the training of the model

XGBoost

- Gradient Boosting algorithm
- Uses decision trees as weak learners - ensemble method
- The training algorithm uses a gradient descent with Cross-Entropy Loss
- Parameter selection:
  - max depth, learning rate, num_boost_round, early stopping

Feature selection

- XGBoost is capable of providing the importance of the features
- It has three different methods to calculate it:
  - Weight: total number of times a feature is used to split data across all trees
  - Gain: average loss reduction gained when using feature for splitting
  - Cover: the number of times a feature is used to split data across trees weighted by training data points

Evaluation and metrics

The following metric have been used to evaluate our models:

- Balanced Accuracy
- Matthews Correlation Coefficient
- ROC-AUC Score
- Average precision

The data set has been splitted in the following way:

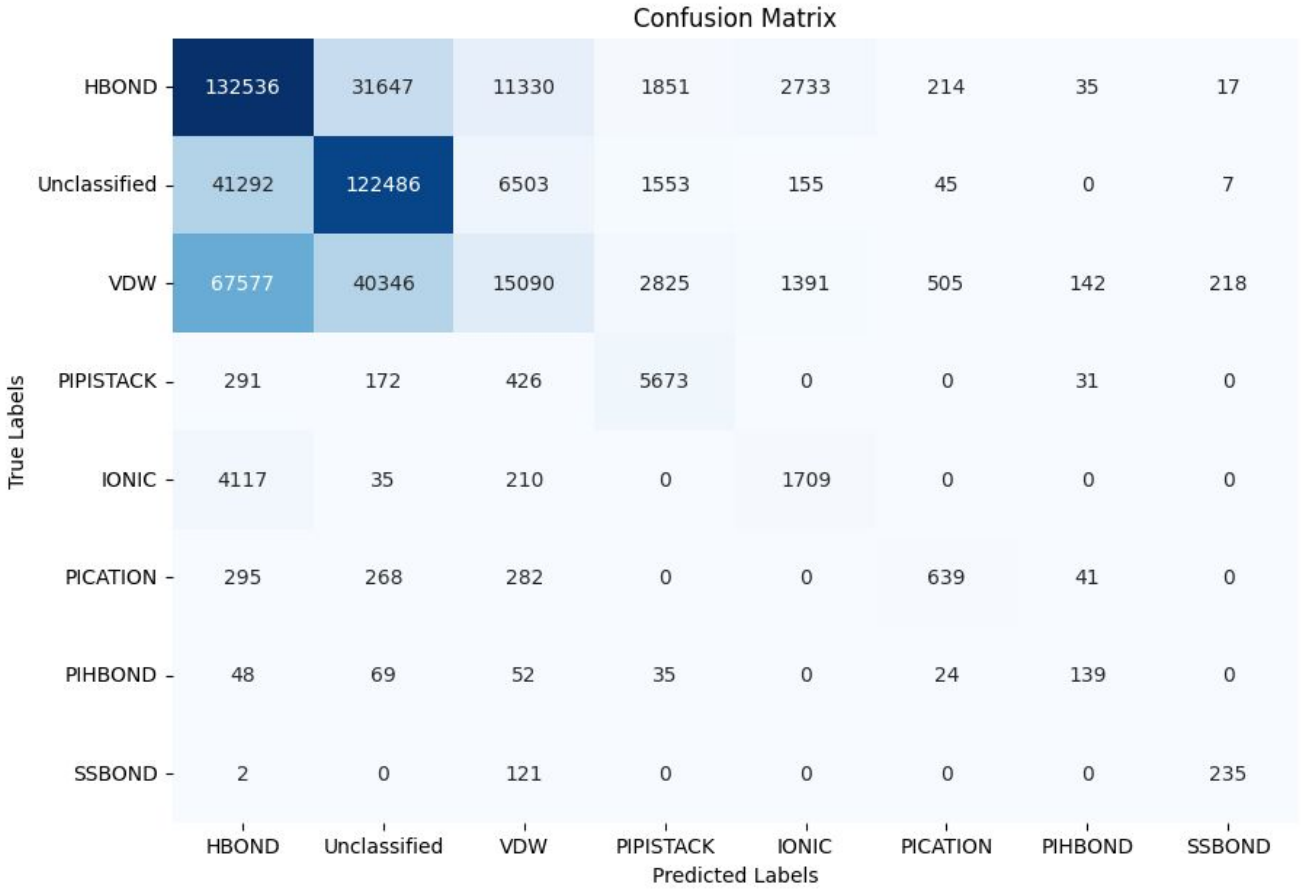
- 80% SMOTE Dataset as Training set
- 20% SMOTE Dataset as Validation set
- 20% Initial Dataset as Test set

Results

UNIVERSITÀ DEGLI STUDI DI PADOVA

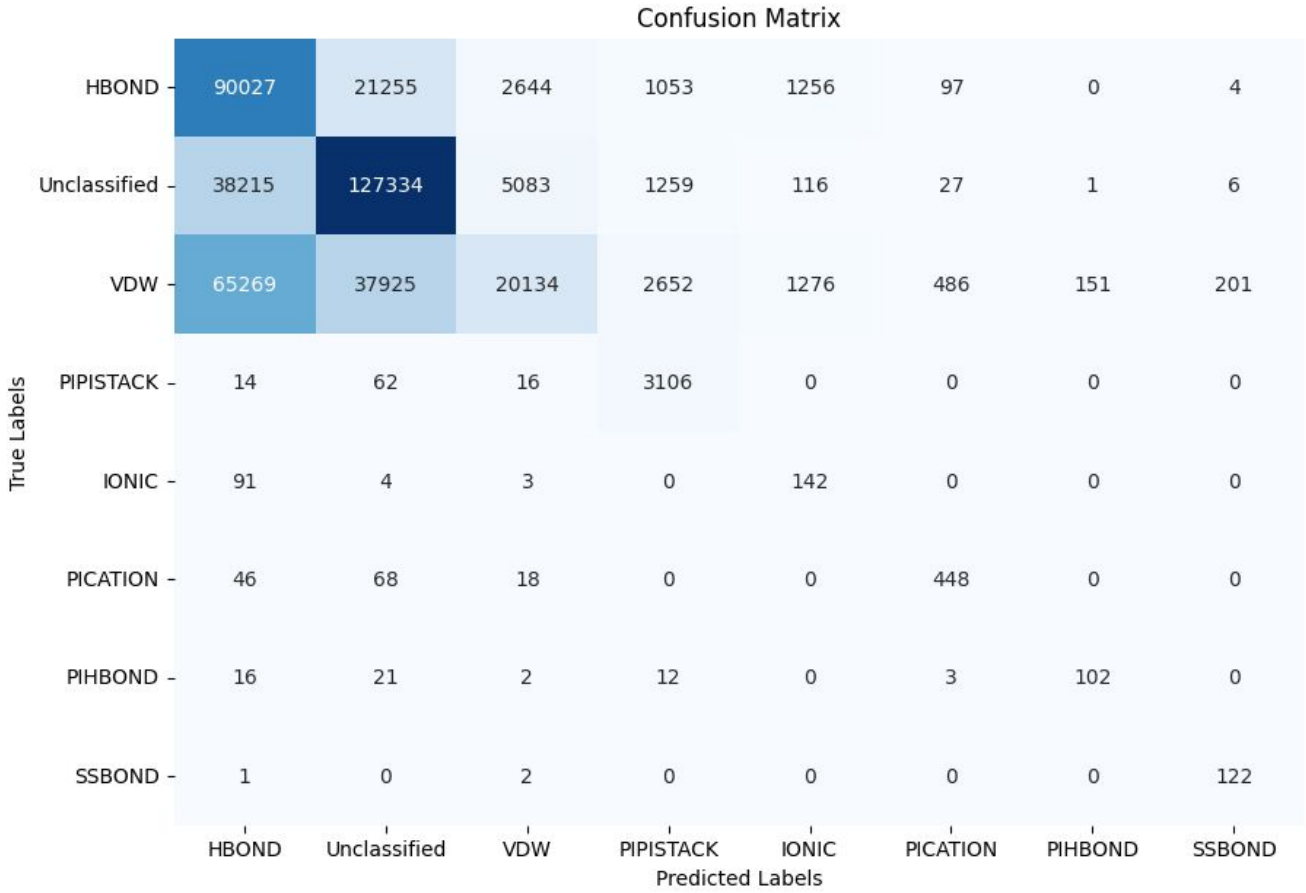
Model	Bal_Acc	MCC	AUC-ROC	AVG_Prec
XGBoost (Initial Dataset)	0.3669	0.2956	0.8741	0.5193
XGBoost (SMOTE)	**0.5610**	**0.4752**	**0.9228**	**0.6704**
XGBoost (SMOTE, No duplicates)	**0.7045**	**0.3923**	**0.9195**	**0.6651**
XGBoost (Feature selection)	0.5161	0.3554	0.8516	0.5080
SimpleNN (SMOTE)	0.4043	0.2321	0.7388	0.4432

Results



Confusion matrix of the XGBoost model with SMOTE

Results



Confusion matrix of the XGBoost model with SMOTE and no duplicated lines

Examples

- Correct prediction:
  - Real: 5mt2 A 238 K E 0.244 4 25 -1.423 2.492 H 1.831 -0.561 0.533 -0.277 1.648 A 267 D T 0.62 3 20 -1.169 2.155 H 1.05 0.302 -3.656 -0.259 -3.242 **HBOND**
  - Prediction: 5mt2 A 238 K E 0.244 4 25 -1.423 2.492 H 1.831 -0.561 0.533 -0.277 1.648 A 267 D T 0.62 3 20 -1.169 2.155 H 1.05 0.302 -3.656 -0.259 -3.242 **HBOND**
- Wrong prediction:
  - Real: 5mt2 A 116 K S 0.366 16 4 -2.01 2.937 H 1.831 -0.561 0.533 -0.277 1.648 A 186 D S 0.564 3 16 -1.573 0.215 H 1.05 0.302 -3.656 -0.259 -3.242 **IONIC**
  - Prediction: 5mt2 A 116 K S 0.366 16 4 -2.01 2.937 H 1.831 -0.561 0.533 -0.277 1.648 A 186 D S 0.564 3 16 -1.573 0.215 H 1.05 0.302 -3.656 -0.259 -3.242 ***Unclassified***

Conclusions

- The best approach tested is the one that utilizes XGBoost with SMOTE applied to the dataset
- The feature selection did not yield better results
- The main problem were the resources: time or computational power
- Idea for improving the project with more resources: try to project data into higher-dimensional space to achieve better separation between data

Thanks for your attention!