Classification of Contacts in Protein Structures

Structural Bioinformatics
University of Padova
Academic Year 2022-2023

Bedin Veronica
Canel Alessandro
Riccò Lorenzo
Pase Emanuele

Aim of the project

Create a software in order to classify contacts inside a protein structure.

Input: PDB file

Output: a table in which every residue-residue interaction of the protein has a specific propensity of belonging to the different contact types considered, following RING guidelines.

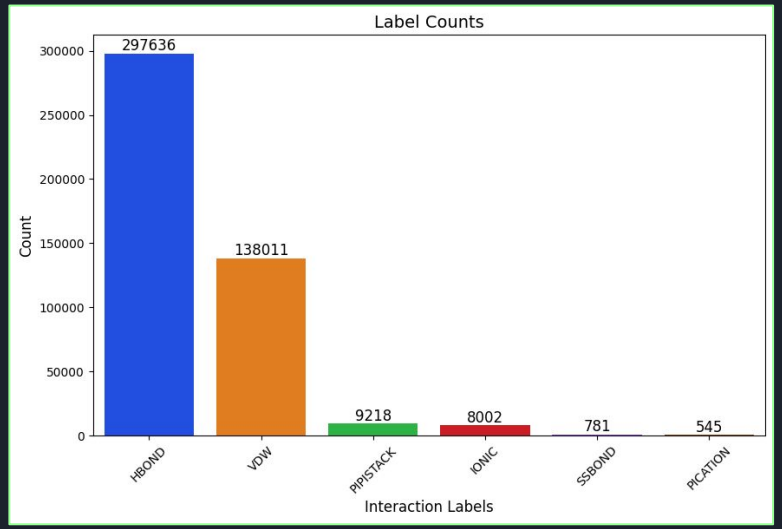
Outline

- **Veronica: Feature Analysis**
  Starting Point, Observations & Speculations
- **Lorenzo: Supervised Machine Learning Methods Deployment**
  Initial Thoughts, Pipeline, Breaking Points
- **Alessandro: MLP Development**
  Implementation, Step Done, Results
- **Emanuele: Software Assembly**
  Documentation & Reproducibility

Why? Different Backgrounds, Different Characteristics

Feature Analysis - Distribution of the interactions

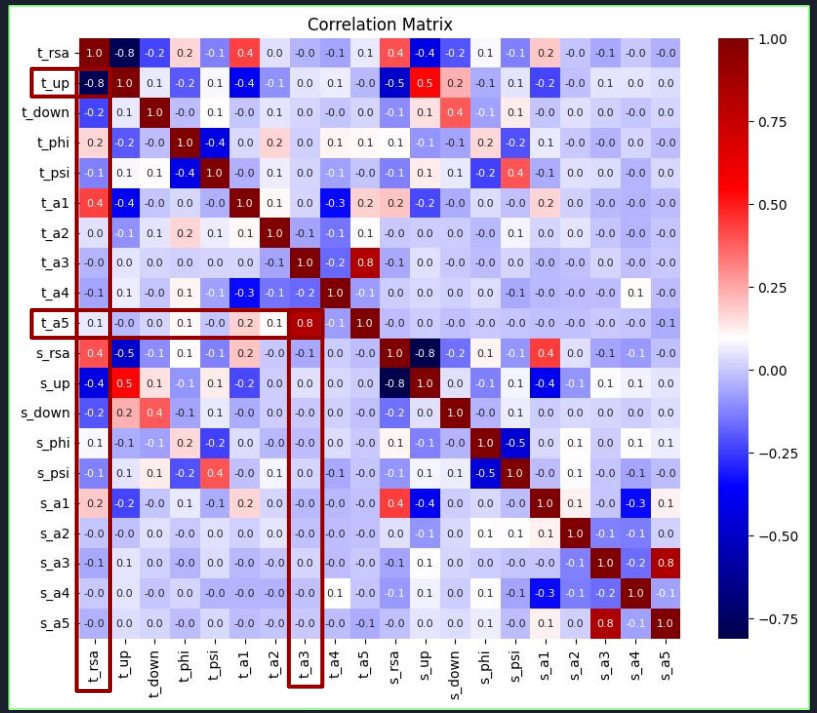
Input: list of 25 pre-calculated features.



High unbalance → over or/and under sampling solutions

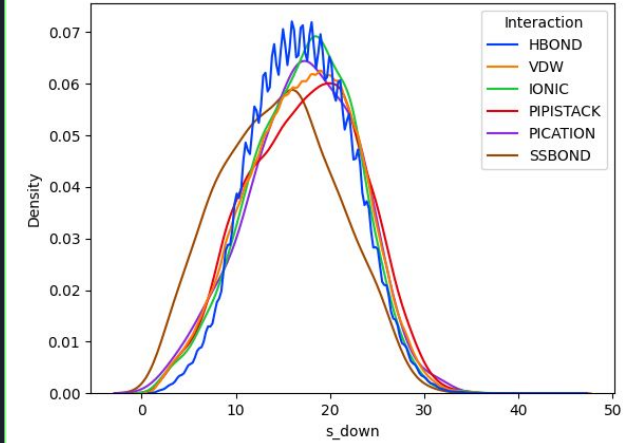
1. **oversampling technique: RandomOverSampler and SMOTE**
2. **undersampling manual techniques to manage the complexity of the Neural Network**

Feature Analysis - Correlation matrix

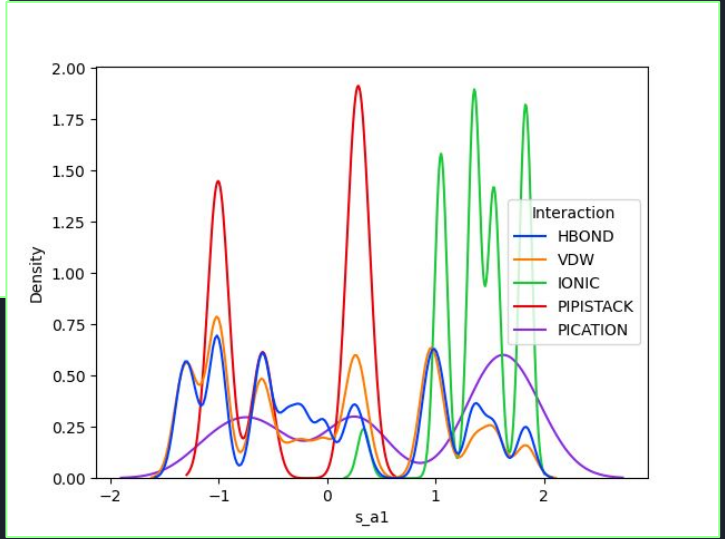


High correlation shared among some of the features → redundant information

Feature Analysis - Features distribution



The frequency of appearance of ***s_down*** feature.

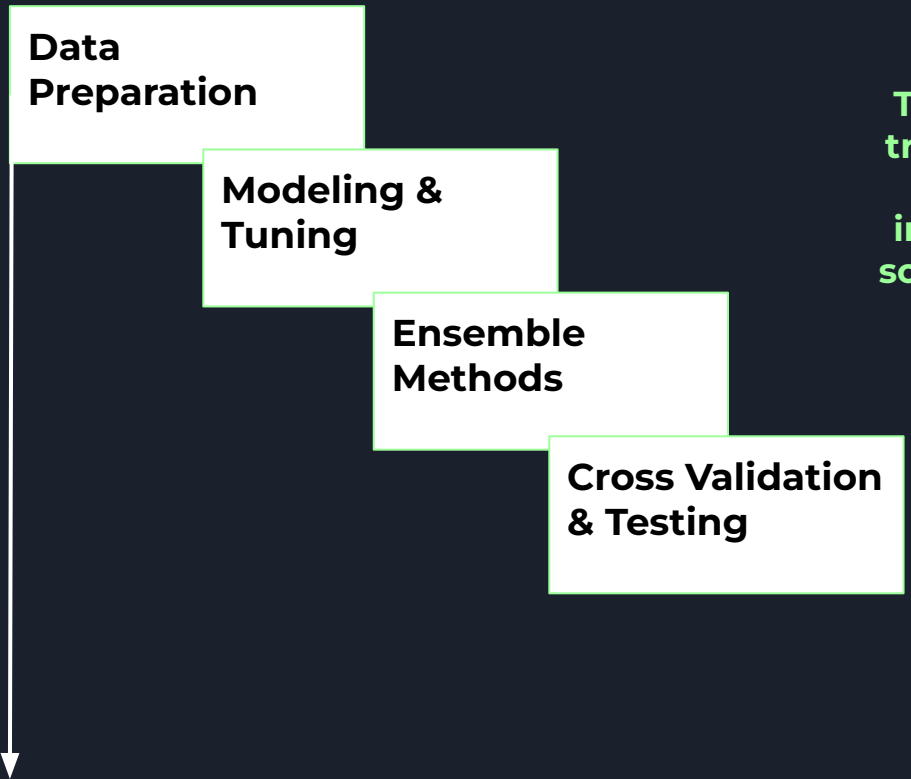


The frequency of appearance of ***s_a1*** feature.

Possible idea for future work....

- Enhance the model with additional features, e.g. the embeddings produced by ProtTrans.
- Implement the ss8 categorical feature

First Path Followed

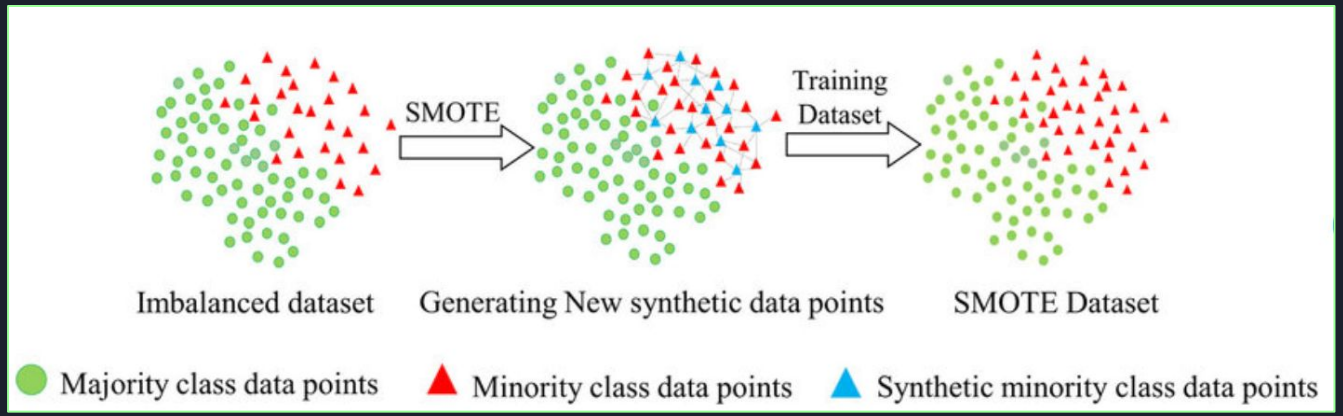


The approach followed has been to train different methods for different labels (of the target variable interaction), using for each one the so called ‘one-vs-all’ (or ‘one-vs-rest’) technique

Main Ingredients

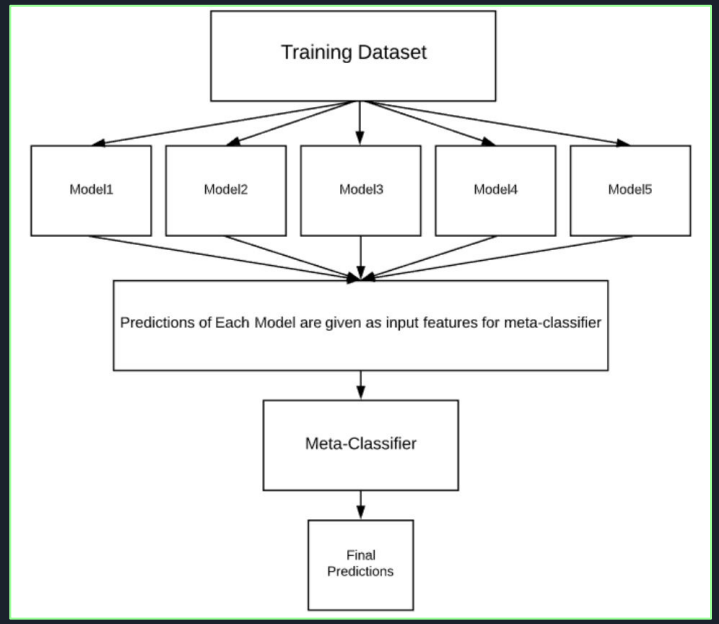
1. RandomOverSampler vs SMOTE technique

2. Logistic Softmax Regression, K-Nearest Neighbor and Decision Tree Definition with Grid Search



Ensemble Methods

Stacking Classifier vs AdaBoost vs Random Forest



From a theoretical point of view the most promising idea was the Stacking Classifier, from a practical perspective it was infeasible

Article
Residue–Residue Interaction Prediction via Stacked Meta-Learning

Results were not enough

Overall less than 0.50 balanced accuracy, less than 0.1 for F1 score and MCC was most of the times a negative value.

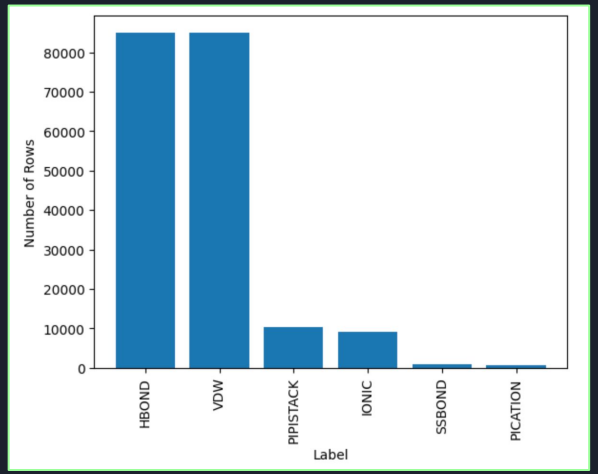
In order to maximise the performance on what we have developed in the second part we have used part of the results obtained here and through a comparison between us we have decided which procedure would have been the best to devote on our attention.

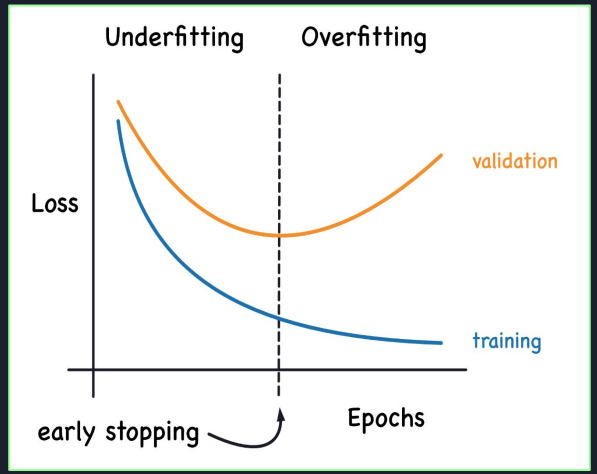
In our second strategy we have developed a Multi-Layer Perceptron

Second Path Followed

About the implementation...

- Custom Pythorc Dataset Class with Undersampling Operations
- Early Stopping Strategy to avoid Overfitting





Moreover...

- ***Tanh*** **as Activation Function**
- **RandomForestClassifier to define the best set of features for each type of interaction**

	rsa	up	down	phi	psi	a1	a2	a3	a4	a5
HBOND	s,t	s,t	s	s,t	s,t				s,t	
IONIC						s,t				t
PICATION	s,t	s,t	s,t	s,t	s,t					
PIPISTACK						t		s	s,t	s,t
SSBOND						s,t	s,t		s,t	
VDW	s,t	s,t	s,t	s,t	s,t					

Step by Step Results

```
Number of successful predictions: HBOND       556
IONIC         564
PICATION      561
PIPISTACK     567
SSBOND        557
VDW           555
dtype: int64
Total number of predictions: 4032
```

```
Number of successful predictions: HBOND       22565
IONIC         40787
PICATION      50778
PIPISTACK     46831
SSBOND        50705
VDW           22797
dtype: int64
Total number of predictions: 305700
```

Watching these improvements has led us to use the MLP as estimator for our software

Final Results & Output

```
Trained model for label: HBOND
Epoch [1/5], Loss: 2.9020, Accuracy: 45.63%, Precision: 0.6137, Recall: 0.4563, F1: 0.5197, MCC: 0.1332
Epoch [2/5], Loss: 2.7292, Accuracy: 61.63%, Precision: 0.6287, Recall: 0.6163, F1: 0.5800, MCC: 0.2138
Epoch [3/5], Loss: 2.6172, Accuracy: 61.49%, Precision: 0.6349, Recall: 0.6149, F1: 0.5703, MCC: 0.2144
Epoch [4/5], Loss: 2.5278, Accuracy: 61.30%, Precision: 0.6375, Recall: 0.6130, F1: 0.5636, MCC: 0.2124
Epoch [5/5], Loss: 2.4575, Accuracy: 61.18%, Precision: 0.6392, Recall: 0.6118, F1: 0.5592, MCC: 0.2111
```

```
['HBOND', 'IONIC', 'PICATION', 'PIPISTACK', 'SSBOND', 'VDW']
[[ 0.99971104, 0.8275111 ,-0.2660997 , 0.5423344 ,-0.5260288 , 0.9997435 ],
 [ 0.986195  , 0.6212192 ,-0.11434213, 0.6391837 ,-0.09014948, 0.98416084],
 [ 0.98831624, 0.27979788,-0.19604951, 0.39312083,-0.28991652, 0.9844556 ],
 [ 0.9933971 , 0.3192664 , 0.04922092, 0.60969687,-0.34486082, 0.9886575 ],
 [ 0.9844098 , 0.38797307,-0.14083804, 0.4531882 ,-0.27964717, 0.9572974 ],
 [ 0.9677642 , 0.47882152,-0.21274246, 0.45303816,-0.06792552, 0.9874763 ],
 [ 0.9749832 , 0.20228659,-0.04278616, 0.53187174,-0.05475084, 0.9765046 ],
 [ 0.9923894 , 0.43356708,-0.00028678, 0.42271408,-0.281109  , 0.9940485 ],
 [ 0.99008834, 0.3467796 ,-0.22094116, 0.37428227,-0.0686644 , 0.9889687 ],
 [ 0.9871265 , 0.44171667,-0.26570114, 0.5997406 ,-0.23158766, 0.9926982 ],
 [ 0.98401785, 0.5200946 ,-0.18619867, 0.60674894,-0.216435  , 0.99045086],
 [ 0.9883753 , 0.6060021 ,-0.28214097, 0.72025484,-0.34363052, 0.96610236],
```

About the Software Development

```
Enter the PDB ID:
```

```
Choose which model to use:
0: MCC
Select a number from the list:
```

```
Select a threshold:
```



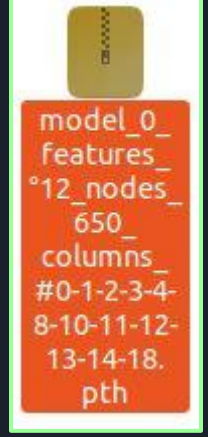
```
Enter the PDB ID: ImNotAProtein
Error: Unable to download PDB structure with ID 'ImNotAProtein': execution finished
```

Problem and Solution

Implementation of the MLP class and correct directory

```python
class MLP(nn.Module):
    def __init__(self, input_size, hidden_size):
        super(MLP, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.tanh = nn.Tanh()

    def forward(self, x):
        out = self.fc1(x)
        out = self.tanh(out)
        return out
```



Other possible Output defined by the User

Is possible to set the threshold at the preferite level

```
['HBOND', 'IONIC', 'PICATION', 'PIPISTACK', 'SSBOND', 'VDW']
Threshold: 0.99
[[1. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0.]
```