Pocket Seeker

Binding Site Predictor

Model Performance Report

Marc Arreaza

Marta Pérez

A report submitted in partial fulfilment of the requirements of
the University of Pompeu Fabra for the master's degree of
Bioinformatics in Health Sciences in *Structural Bioinformatics and Introduction to Python courses*

April 22, 2025

Contents

Chapter 1

Introduction

This report presents a comprehensive evaluation of the performance of a Random Forest classifier designed to predict protein binding site residues. The model was trained splitting randomly the protein dataset into two subsets: 80% training set and 20% test set.

At the beginning of the analysis, we identified a critical limitation in the dataset composition: there was a substantial class imbalance, with binding site residues being severely underrepresented in the training data. This issue arises from the biological reality that only a small fraction of residues in a typical protein are involved in ligand binding, while the vast majority are part of the protein's structural framework. Without addressing this imbalance, the classifier would likely learn to favor the majority class and fail to effectively recognize binding sites, leading to poor generalization and low sensitivity.

To mitigate this problem and improve the model's predictive performance, we implemented a two-step preprocessing strategy:

1. **Feature-based filtering**: As a first step, we applied a filtration procedure to exclude residues that are buried in the protein core and therefore unlikely to participate in binding interactions. This was achieved by leveraging structural features such as solvent-accessible surface area (SASA) and solvent exposure metrics. Residues with low SASA values were removed from the dataset prior to training. This step not only reduced the size of the dataset and computational cost, but also enriched the sample pool with surface residues that are more biologically relevant for binding site prediction.

2. **Class imbalance handling**: To further address the imbalance between binding and non-binding residues, we utilized the `class_weight='balanced'` parameter available in the scikit-learn implementation of the Random Forest classifier. This setting dynamically assigns higher weights to the minority class (i.e., binding sites) during model training, thus increasing their influence on the learning process. By doing so, the classifier becomes more sensitive to the presence of positive examples, which helps reduce the bias towards the negative class. The final classifier was trained using the following configuration:

   ```
   clf = RandomForestClassifier(n_estimators=100, random_state=42,
   class_weight='balanced')
   ```

   Additionally, to further enhance recall (true positive rate), we adjusted the decision threshold of the classifier to 0.2, which means that a residue is classified as a binding site if the predicted probability exceeds this value. This threshold was chosen to prioritize the identification of true binding residues, even at the cost of a higher false positive rate.

Chapter 2

Models

1 Model 1: Initial Training with Class Balance

The first model was trained using a balanced dataset without applying any specific oversampling or undersampling techniques. The goal was to establish a baseline for evaluating future improvements.

Metrics:

- **Accuracy:** 0.94 — The model correctly classified 94% of the residues overall.
- **Precision:** 0.52 — Of all the predicted binding sites, 52% were correct.
- **Recall:** 0.46 — The model identified 46% of the actual binding sites.
- **F1 Score:** 0.48 — The harmonic mean between precision and recall.

Classification Report:

	Precision	Recall	F1-Score	Support
Class 0	0.9659	0.9729	0.9694	12869
Class 1	0.5153	0.4563	0.4840	813
Macro Average	0.7406	0.7146	0.7267	13682
Weighted Avg	0.9391	0.9422	0.9405	13682

Although the model achieved high accuracy overall, its ability to detect true binding sites (recall) was limited. Nearly half of the actual binding sites were missed, which could severely limit the model's usefulness in downstream applications where identifying such sites is critical.

2 Model 2: Increasing Recall with SMOTE

To address the class imbalance issue and improve recall, we applied the Synthetic Minority Over-sampling Technique (SMOTE). This method synthetically generates new instances of the minority class, in this case, the binding sites, to increase their representation during training.

Metrics:

- **Accuracy:** 0.75 — Lower than the first model due to the trade-off with recall.

- **Precision:** 0.17 — A significant drop, indicating more false positives.
- **Recall:** 0.79 — A major improvement in identifying actual binding sites.
- **F1 Score:** 0.28 — Reflects the imbalance between precision and recall.

Classification Report:

	Precision	Recall	F1-Score	Support
Class 0	0.9827	0.7516	0.8517	12869
Class 1	0.1674	0.7909	0.2764	813
Macro Average	0.5751	0.7712	0.5641	13682
Weighted Avg	0.9343	0.7539	0.8176	13682

With the application of SMOTE, the recall increased dramatically, demonstrating that the model was now capable of identifying a much larger proportion of the true binding sites. However, this improvement came at the expense of precision and overall accuracy, which dropped significantly. This suggests that the model was over-predicting binding sites, leading to a high rate of false positives.

3 Model 3: Adjusting the Decision Threshold

In order to strike a better balance between precision and recall, we adjusted the decision threshold of the classifier trained with SMOTE. The threshold was changed from 0.2 to 0.3, with the intention of reducing the number of false positives while still capturing a reasonable number of true positives.

Metrics:

- **Accuracy:** 0.87 — Improved compared to Model 2.
- **Precision:** 0.25 — Better than Model 2, although still relatively low.
- **Recall:** 0.60 — A compromise that remains above 50%.
- **F1 Score:** 0.36 — A more balanced harmonic mean between precision and recall.

Classification Report:

	Precision	Recall	F1-Score	Support
Class 0	0.9724	0.8892	0.9289	12869
Class 1	0.2550	0.6002	0.3579	813
Macro Average	0.6137	0.7447	0.6434	13682
Weighted Avg	0.9298	0.8720	0.8950	13682

By increasing the decision threshold, the model became more conservative in predicting binding sites. This reduced the number of false positives and helped recover some of the lost accuracy. Most importantly, the recall remained at a satisfactory level, surpassing 60%, which makes this model more suitable for practical use.

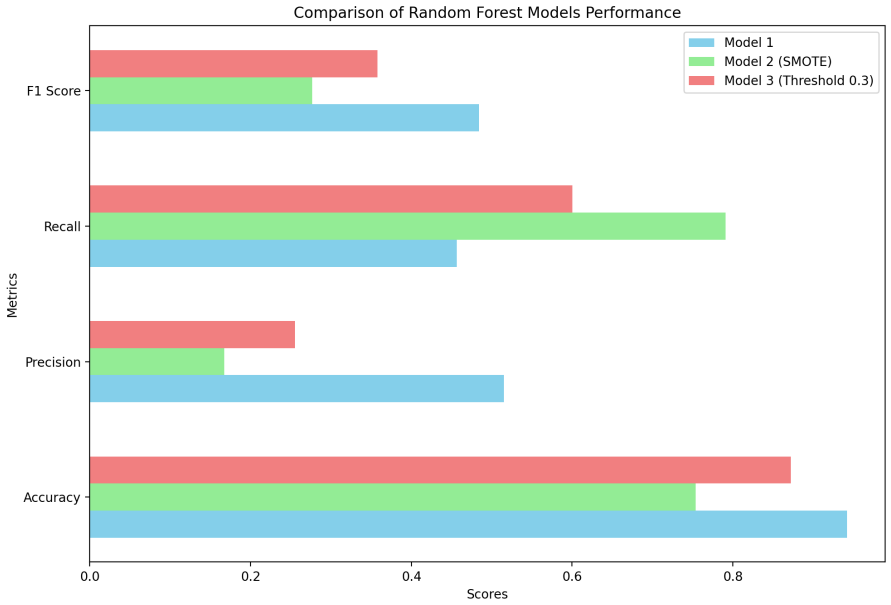

Figure 2.1: Performance comparison of Models 1–3 in terms of Precision, Recall, and Accuracy.

Conclusion

The comparative analysis of the three models demonstrates the classic trade-off between precision and recall in classification problems involving imbalanced datasets. While the initial model had high accuracy and precision, its recall was insufficient for detecting binding sites reliably.

The application of SMOTE significantly improved recall, but at the cost of precision and overall accuracy. Adjusting the decision threshold in the SMOTE-enhanced model helped recover accuracy and improve the balance between true and false positives.

Ultimately, Model 3 was selected as the final model due to its superior balance between performance metrics. By combining SMOTE and adjusting the decision threshold to 0.3, the model achieves a better balance between sensitivity and precision, improving recall from 46% to 60% while maintaining acceptable accuracy. This configuration is therefore more suitable for identifying potential binding sites, even if some false positives are tolerated.

Below are the confusion matrices and ROC curves for the final model, providing further insight into the classification behavior.

Chapter 3

Evaluation of the Final Model

1 Confusion Matrix

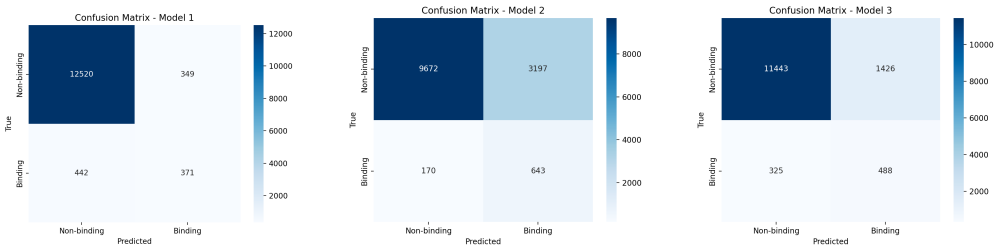

Figure 3.1: Confusion matrices for Models 1, 2, and 3

The confusion matrices for each model illustrate the classification performance with respect to true positives, false positives, true negatives, and false negatives. In Model 1, while the accuracy was high, a large number of false negatives indicated that many actual binding sites were not being detected. Model 2, after applying SMOTE, significantly reduced the number of false negatives—hence improving recall—but at the cost of a sharp increase in false positives.

Model 3 strikes a better balance: the number of true positives exceeds the number of false negatives, which is a huge improvement respect the Model 1; and the number of false positives is lower than in Model 2. This balance is crucial for our task, where predicting a binding site where there is none (false positive) can be more detrimental than not detecting one (false negative).

2 ROC Curve

The Receiver Operating Characteristic (ROC) curves illustrate the relationship between the true positive rate and the false positive rate across different classification thresholds. In our case, all three models exhibit fairly similar ROC curves, indicating that the underlying discriminatory power of the classifier remains relatively stable across different preprocessing and threshold adjustments.

Interestingly, Model 1, despite having lower recall, achieves the highest Area Under the Curve (AUC), suggesting that it performs better overall when all possible thresholds are considered. Models 2 and 3, while designed to increase recall through the use of SMOTE and threshold tuning, show a slight decrease in AUC, even though their recall improves significantly.

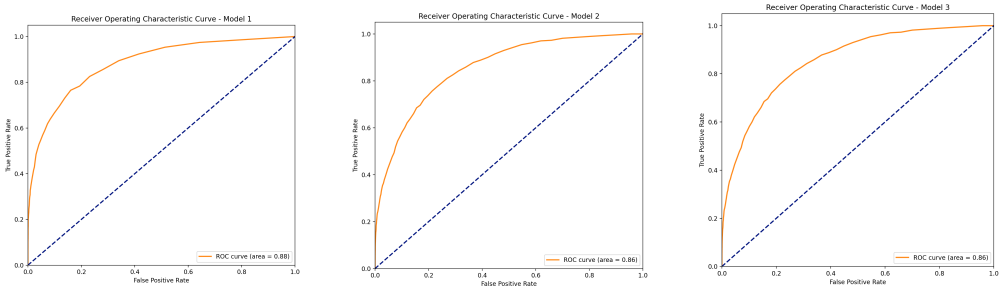

Figure 3.2: ROC curves for Models 1, 2, and 3

These results highlight that while AUC is a useful overall performance metric, it does not always reflect improvements in recall that are critical for our specific task. Therefore, the final model (Model 3) was selected not because it maximized AUC, but because it offered the best trade-off between recall and precision for detecting protein binding sites.

Chapter 4

Future Directions

While the model performs well in identifying non-binding residues, it struggles with binding site detection. This limitation suggests that there is room for improvement, especially in terms of recall for binding sites, which is critical for the practical use of the model. To enhance the performance, future improvements may include:

- **Resampling techniques (oversampling or undersampling)**: Addressing the class imbalance more effectively could improve model performance, particularly in detecting binding site residues. Oversampling the minority class or undersampling the majority class could provide a better balance between the two classes.
- **Hyperparameter tuning**: Further optimizing the model's hyperparameters could improve its ability to generalize. Techniques such as grid search or random search could help identify the best combination of parameters for the Random Forest classifier.
- **Incorporating more features or domain knowledge**: Integrating additional features that capture more specific biological information, such as residue-specific properties or structural features, could improve the model's sensitivity to binding sites. This could involve using sequence-based features or protein structure data.
- **Exploring alternative models**: While Random Forest is a robust model, exploring alternative machine learning techniques such as support vector machines, gradient boosting, or deep learning models might provide better results in terms of recall and precision.

Improving recall for binding site residues is essential to make the model more applicable for real-world applications, such as drug design or functional annotation of proteins. By addressing these challenges, we aim to create a more accurate and reliable tool for predicting protein binding sites, thereby contributing to advancements in bioinformatics and related fields.