Prediction of B-cell Epitopes in Trypanosoma cruzi Peptides: A Data Mining Pipeline

Coursework Summary

Kazi Moshiur Rahaman
240378802

MSc in Data Science

1. Introduction

This report outlines a machine learning-based data mining pipeline designed to predict linear B-cell epitopes in *Trypanosoma cruzi*, the parasite responsible for Chagas disease. The analysis aimed to develop a robust predictive model using high-dimensional biological data with over 1,600 features across 45,000 peptide samples. Key challenges addressed include extreme class imbalance (only 0.74% positives), high-dimensional feature space, and the need to avoid data leakage due to protein-based sample grouping. Our final goal was to deploy a reproducible and interpretable pipeline capable of generalizing to unseen proteins, simulating real-world use in bioinformatics and immunology.

2. Exploratory Data Analysis (EDA)

The dataset comprises 45,000 peptide rows and 1,650 columns, including 14 Info columns and 1,635 features. Class distribution analysis revealed an extreme imbalance with only 332 positive samples (class = 1), yielding an approximate ratio of 1:134. Visualizations using bar and 2D plots highlighted the disparity. Group-wise analysis identified 94 unique proteins (Info_proteinID), with group sizes ranging from 6 to 1,268 peptides. Because peptides within a protein may share patterns, we used GroupKFold to ensure peptides from the same group remained in the same fold, reducing data leakage during validation.

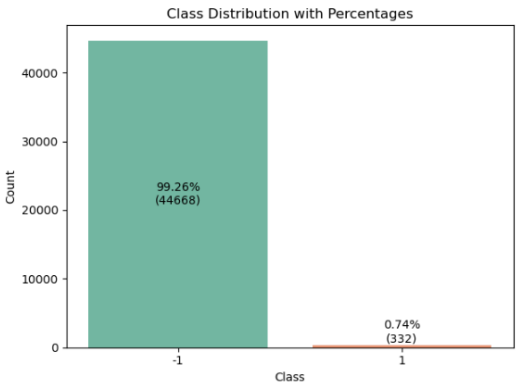

Fig: imbalance dataset

3. Data Preprocessing and Feature Reduction

All features were standardized using StandardScaler. To address class imbalance, we used SMOTE (Synthetic Minority Over-sampling Technique), resulting in a balanced dataset of 89,336 samples, verified through visual plots.

Feature reduction was conducted using three methods: Mutual Information, F-test, and Random Forest. We selected the top 200 features from each method and retained the 153 features that

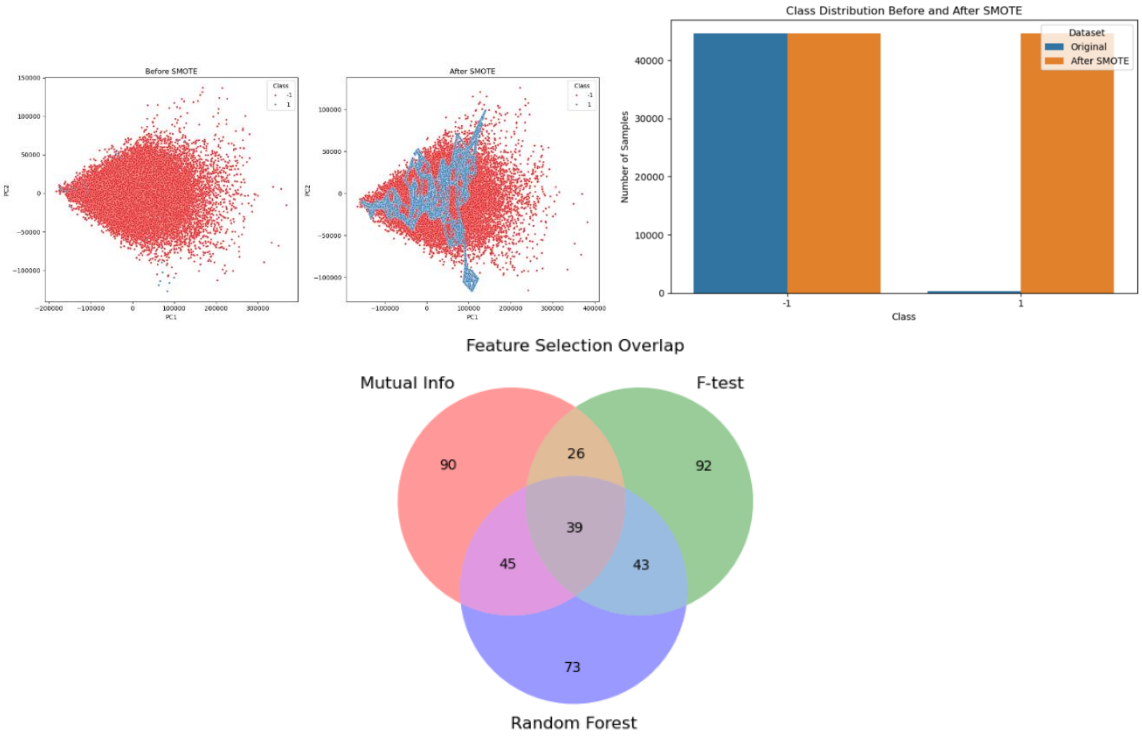

Fig : make balance the data and selection of important feature

appeared in at least two, forming a consensus set. (we can see from the Ven Diagram) PCA was also tested, reducing features to 979 components explaining 95% variance, but this was not chosen for the final model.

4. Model Development and Evaluation

We evaluated Logistic Regression, KNN, Random Forest, and XGBoost models using both the 153 consensus features and the 979 PCA features. Each was validated using GroupKFold cross-validation with balanced accuracy as the performance metric.

```
Consensus Feature Set Results:
                    Balanced Accuracy
RandomForest                   0.5000
LogisticRegression             0.6272
KNN                            0.5205
XGBoost                        0.5488
PCA Feature Set Results:
                    Balanced Accuracy
RandomForest                   0.5000
LogisticRegression             0.6110
KNN                            0.5908
XGBoost                        0.5283
```

Fig: different algorithm comparison

Logistic Regression on the consensus features yielded the best performance with a balanced accuracy of **0.6272**. Its simplicity, speed, and consistency across folds made it our chosen final model.

Other models like Random Forest and KNN showed signs of overfitting or sensitivity to high dimensionality.

5. Hyperparameter Tuning

We performed GridSearchCV tuning on Logistic Regression using parameters for C, solver, and penalty. The best-tuned configuration achieved a lower balanced accuracy of **0.5425**, indicating that hyperparameter tuning did not improve performance in this case. The discrepancy may have resulted from noise sensitivity or suboptimal interaction between the parameter space and the imbalanced data, despite the use of GroupKFold.

6. Final Pipeline and Predictions

The final pipeline involved:

- Selection of 153 consensus features
- Standardization with StandardScaler
- Oversampling using SMOTE
- Model training with untuned Logistic Regression
- GroupKFold validation for robust evaluation

This pipeline was then used to predict classes for the holdout set (df_holdout.csv). The submission file (Rahaman_Kazi_ AM41UDJ_Predictions.csv) includes Info_PepID, Info_pos, and Prediction columns as required.

7. Conclusion

This project successfully built a reproducible pipeline for epitope prediction in *T. cruzi*. We tackled severe class imbalance using SMOTE, reduced dimensionality through consensus feature selection, and evaluated models with leakage-free GroupKFold cross-validation. Logistic Regression stood out for its balance between accuracy, simplicity, and interpretability. The model was deployed to predict unseen data, demonstrating real-world readiness.

8. Limitations and Future Work

- Hyperparameter tuning was only performed on Logistic Regression due to computational constraints. Running it on all algorithms (e.g., KNN, XGBoost) was impractical.
- Slower execution times limited experiments, particularly for models sensitive to dimensionality.
- The fact that the untuned Logistic Regression performed better than the tuned one might be due to specific interactions between default parameters and the balanced dataset.
- Future work can benefit from high-performance computing to enable broader model tuning.
- Additional directions include integrating ensemble methods, deep learning, or domain-specific biological knowledge to enhance performance and interpretability.

Reference:

1.Maruf, A. (2024). Prediction of B-cell Epitopes using Data Mining Techniques [Unpublished coursework report]. Aston University.

2.Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). ***SMOTE: Synthetic Minority Over-sampling Technique****. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953*

3.Mahmoudian, M., Sechidis, K., and Brown, G., 2021. Stable Iterative Variable Selection. Bioinformatics, 37(24), pp.4810-4817