Machine Learning Approaches for Diabetes Prediction

A Comparative Study of Class Balancing Techniques and Model Performance

Alexandros Angelakis
University of Crete

Introduction

- Machine learning is widely applied in healthcare.
- This study focuses on diabetes prediction using different models.
- Evaluates class balancing techniques and hyperparameter tuning.

Problem Definition

- Dataset: 100,000 patient records (large sample size).

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0

- Class Imbalance: 91,544 non-diabetic vs. 8,456 diabetic samples.
  - Imbalance Ratio: ≈10.82 (heavily imbalance)
- Goal: Develop models for early diabetes detection.

Data Processing and Preprocessing

- No missing values in the dataset.
- Categorical features one-hot encoded (except smoking history - frequency encoded).
- Numerical features standardized (zero mean, unit variance).
- Addressing class imbalance using:
  - Random undersampling
  - Class weighting
  - SMOTE (computationally intensive, not tested)

Machine Learning Pipeline

- Feature selection:
  - LASSO Regression
  - Backward Elimination
- Evaluated models:
  - Logistic Regression, SVM, Random Forest, Decision Tree, XGBoost, KNN, GNB.
  - Each model was implemented using scikit-learn or XGBoost. The dataset was split into 90-10.
- Hyperparameter tuning with nested cross-validation (outer loop 10-fold stratified cross-validation, inner loop 5-fold stratified cross-validation).
- Best overall model selected based on its average AUC score across all outer folds. Final model trained on the entire training dataset.
- Total number of trained models: 6801.

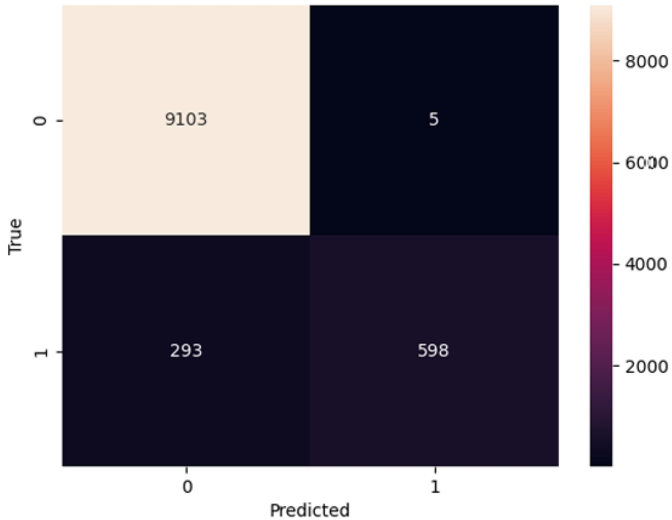
- The configurations tested using nested cross-validation

Model	**Hyperparameters**
Logistic Regression	C: [0.001, 0.01, 0.1, 1, 10, 100] Penalty: ['l1', 'l2'] Solver: ['liblinear', 'saga']
Support Vector Classifier (SVC)	C: [0.1, 1, 10, 100] Kernel: ['linear', 'rbf', 'poly', 'sigmoid'] Gamma: ['scale', 'auto']
Random Forest Classifier	n_estimators: [50, 100, 200] Max_depth: [None, 5, 10, 20]
XGBoost Classifier	Learning Rate: [0.01, 0.1, 0.2] n_estimators: [50, 100, 200] Max_depth: [3, 5, 7]
Decision Tree Classifier	Max_depth: [None, 5, 10, 20] Min_samples_split: [2, 5, 10]
K-Nearest Neighbors (KNN)	n_neighbors: [3, 5, 7, 9] Weights: ['uniform', 'distance'] Metric: ['euclidean', 'manhattan', 'minkowski']
Gaussian Naïve Bayes (GNB)	Var_smoothing: [1e-9, 1e-8, 1e-7, 1e-6, 1e-5]

Model Performance

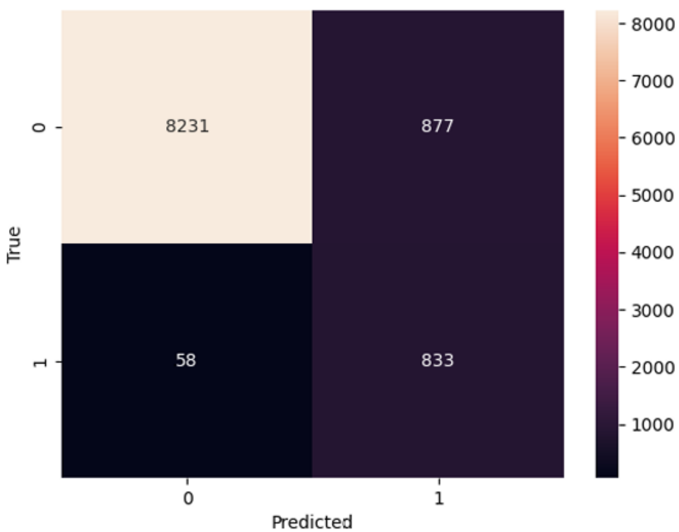
- Without balancing:
  - Best Model: XGBoost (AUC = 0.9805)
  - High precision but low recall (0.6712)
- In medical applications, recall is often prioritized to avoid missing critical cases. Need for less Type II errors.
- Low recall due to class imbalance (model more sensitive to majority class)
- With balancing:
  - Best model: XGBoost (AUC = 0.9831)
  - Recall improved (0.9349) but lower precision (0.4871)
- Best efficiency: Random undersampling (faster execution).

- Confusion matrix and performance metrics without using class balancing:



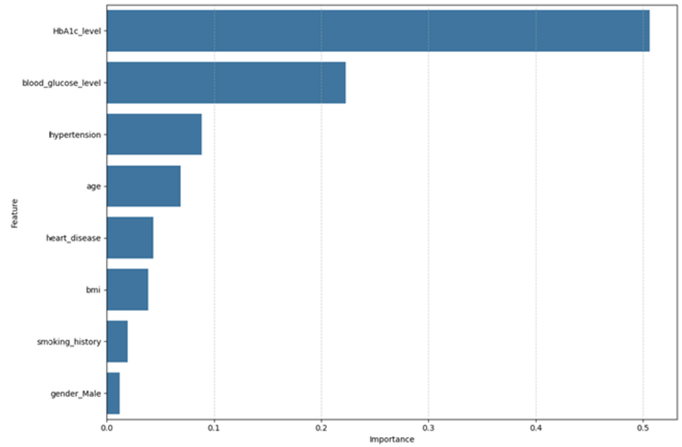
Metric	Value
ROC AUC	0.9806
Average Precision	0.8915
F1 Score	0.8005
F1 Macro	0.8922
Precision	0.9917
Recall	0.6712
Balanced Accuracy	0.8353
Accuracy	0.9702
Matthews Correlation Coefficient	0.8026

- Confusion matrix and performance metrics using class balancing:

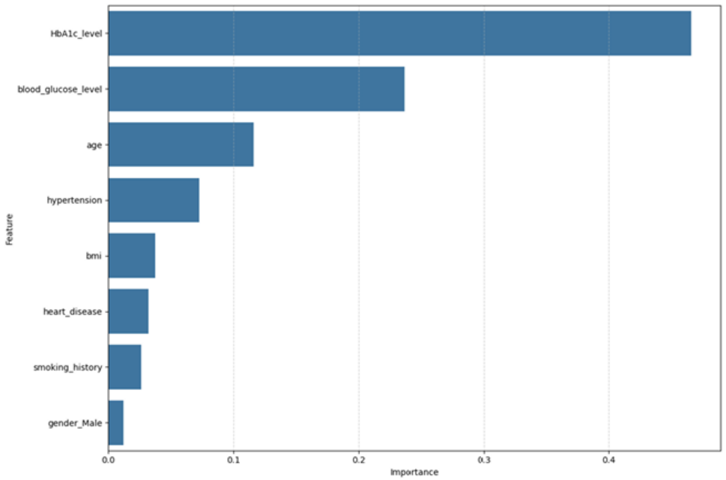


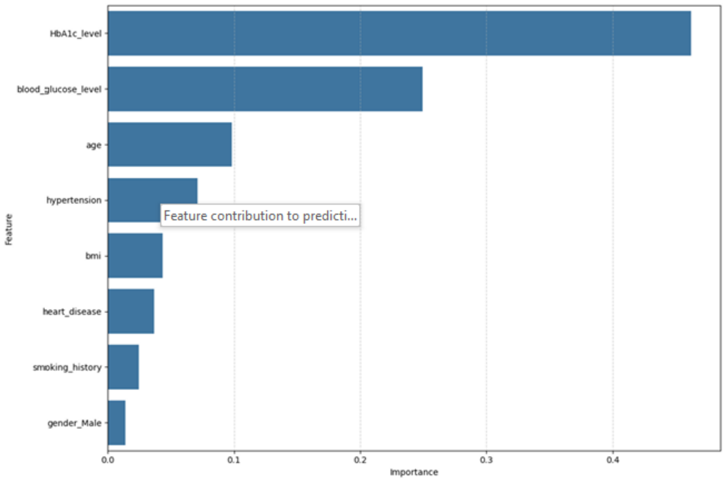
Metric	Random Undersampling	Weight Balancing
ROC AUC	0.9805	0.9805
Average Precision	0.8916	0.8913
F1 Score	0.6354	0.6405
F1 Macro	0.7904	0.7934
Precision	0.4831	0.4871
Recall	0.9282	0.9349
Balanced Accuracy	0.9155	0.9193
Accuracy	0.9051	0.9065
Matthews Correlation Coefficient	0.6285	0.6345

- Feature contribution to prediction without using class balancing:



- Feature contribution to prediction using class balancing:





AutoML - JADBio

- Used AutoML for comparison.
- Configurations using 90% - 10% hold-out method. Total number of models trained: 171.
- Best model configuration:

Step	Method/Algorithm	Hyper-Parameters
Preprocessing	Mean Imputation, Mode Imputation, Constant Removal, Standardization	N/A
Feature Selection	Epilogi Algorithm	equivAlpha = 0.01, stopping criterion = 0.01
Predictive Algorithm	Classification Decision Tree	Minimum leaf size = 4, pruning alpha = 0.05

- JADBio selected all the features for classification as well.

- Performance metrics of JADBio with confidence intervals:

Metric	Mean Estimate	CI
ROC AUC	0.971	[0.963, 0.978]
Mean Average Precision	0.928	[0.915, 0.941]
F1 Score	0.983	[0.981, 0.986]
F2 Score	0.991	[0.989, 0.992]
F0.5 Score	0.975	[0.972, 0.979]
Accuracy	0.970	[0.966, 0.975]
Balanced Accuracy	0.825	[0.805, 0.846]
Matthews Correlation Coefficient	0.780	[0.752, 0.809]
Precision	0.970	[0.966, 0.975]
True Positive Rate (Sensitivity, Recall, Hit Rate)	0.996	[0.994, 0.998]
Specificity	0.673	[0.635, 0.715]
True Positive Ratio	0.911	[0.904, 0.919]
True Negative Ratio	0.057	[0.052, 0.063]
False Positive Ratio	0.028	[0.023, 0.032]
False Negative Ratio	0.004	[0.002, 0.005]

- JADBio achieved high precision along with high recall, but did not manage to attain a high balanced accuracy.

Conclusion

- Class imbalance affects diabetes prediction.
- Random undersampling is efficient and effective.
- Tree based models, and especially XGBoost, performed best overall.
- Decreasing false negatives (Type II errors) in medical applications is crucial.
- Future work:
  - Improve balance between precision and recall.
  - Explore advanced resampling techniques (SMOTE or GANs)