Project Report

Data Mining Project - Diabetes Risk Prediction

Team 11 - Support Vector Superstars
Matthias Fast, Philipp Gänz, Salome Heckenthaler,
Patricia Paskuda, Benedikt Prisett

December 8, 2024

Submitted to
Data and Web Science Group
Prof. Dr. Sven Hertling
University of Mannheim

1 Introduction

Diabetes is one of the most prevalent chronic diseases in the United States, affecting millions and costing over $400 billion per year (CDC 2024). The condition arises when the body is unable to produce sufficient quantities of insulin or to utilize it effectively, resulting in high blood sugar levels due to the inability to absorb glucose from the bloodstream. Given that glucose is the primary energy source for the brain and muscles, its proper regulation is essential. Prolonged high blood sugar levels can lead to severe complications, including cardiovascular disease, kidney failure, and vision loss. While diabetes can have a genetic basis, it can also develop in individuals without a family history. In 2021, the Centers for Disease Control and Prevention (CDC) reported that over 38 million Americans had been diagnosed with diabetes, with an additional 97 million at risk due to prediabetes (CDC 2024). These alarming figures are expected to increase, underscoring the urgency of early detection and intervention to mitigate disease progression and related complications.

The objective of this project is to develop an accurate predictive model capable of identifying individuals at risk of diabetes using the *Diabetes Health Indicators* (Teboul 2024) dataset from Kaggle. Following an initial exploratory analysis of the dataset, preprocessing techniques were applied to ensure cleaner and more reliable data for model training. Subsequently, several binary classification models were trained and fine-tuned, employing hyperparameter tuning and cross-validation to optimize performance. Finally, the performance of the models was evaluated using different performance metrics and precision-recall curves.

2 Dataset

In this project we use a dataset, originally provided by the CDC, a U.S. government agency, which contains data of a telephone survey from 2015 called the ”Behavioral Risk Factor Surveillance System” (BRFSS) (CDC 2022). This dataset consists of over 440,000 data points and 330 distinct features, representing the answers of U.S. participants. However, for the purpose of this project, we leverage an already preprocessed dataset by Alex Teboul which only contains diabetes-relevant data (Teboul 2024). Thus, the preprocessed version only contains 21 features and has no missing values. Furthermore, the number of instances was reduced to 253,680. The target variable consists of three classes: 0 for individuals without diabetes or only during pregnancy, 1 for individuals with prediabetes, and 2 for individuals with diabetes. Our following explorative data analysis makes use of a version of this dataset, where the prediabetes and diabetes classes have been merged to create a binary target variable, since the aim of this project is to train and evaluate binary classifiers. This also enhances plot simplicity and interpretability compared to using three classes.

Our exploratory analysis showed that the dataset is highly imbalanced with respect to the target variable, *Diabetes_binary*. Specifically, 86.07% of the individuals have “No Diabetes”, while only 13.93% have the disease. Furthermore, the dataset contains mostly binary features, while only three are numerical, namely *BMI*, *MentHlth* (Mental Health), and *PhysHlth* (Physical Health). During our data exploration we discovered that each distribution those three features is highly right skewed. The four ordinal features are as follows: *GenHlth* (General Health, rated on a scale from 1, “excellent”, to 5, “poor”), *Age* (binned into 13 categories ranging from 18 to above 80), *Education* (on a scale from 1 to 6), and *Income* (binned into 8 categories starting with “below $10,000” and ending with “above $75,000”). While there is a distinct skew in the distributions of *Education* and *Income* towards the upper categories, the other ordinal features display a more balanced distribution.

As previously mentioned, the majority of features - 14 in total - are binary. These include mainly features related to the medical condition, e.g., *HighBP* (High Blood Pressure), *HighChol* (High Cholesterol), *CholCheck* (Cholesterol Check), *Smoker*, *Stroke*, *HeartDiseaseorAttack*, *PhysActivity* (Physical Activity), or *DiffWalk* (Difficulty Walking). Features with a higher count in their negative class (indicating the absence of the condition) are: *HighBP*, *HighChol*, *Smoker*, *Stroke*, *HeartDiseaseorAttack*, *HvyAlcoholConsump* (Heavy Alcohol Consumption), *NoDocbcCost* (could not afford to see a doctor), and *DiffWalk*. The features with a higher count in their positive class (indicating the presence of the condition) are the following: *CholCheck*, *PhysActivity*, *Fruits* (eat fruit everyday), *Veggies* (eat vegetables everyday), and *AnyHealthcare* (any health care coverage). Additionally, we observed that the dataset contains more female participants.

An analysis of the correlation matrix shows that the feature most strongly correlated with the target variable, *Diabetes_binary*, is *GenHlth*, with a Spearman correlation coefficient of 0.29. The next highest correlation is observed for *HighBP* (0.26), followed by *BMI* and *DiffWalk*, both at 0.22. *HighChol* exhibits a positive correlation with the target variable, with a coefficient of 0.20, while *Age* and *HeartDiseaseorAttack* are correlated at 0.18. To assess the presence of multi-collinearity, the highest correlation was found between *PhysHlth* and *GenHlth* (0.52), followed by the pairs *DiffWalk* & *PhysHlth* (0.48), *DiffWalk* & *GenHlth* (0.46), and *Income* & *Education* (0.45). The respective plots we discussed above, along with additional insights, can be found in our *data_exploration.ipynb* notebook.

3 Preprocessing

We apply systematic preprocessing steps to prepare the dataset for analysis. As described in section 2, the original dataset initially contained 330 features but underwent prior preprocessing and feature selection by Alex Teboul, resulting in a clean dataset with 21 features relevant to diabetes prediction (Teboul 2024). This prior preprocessing by Alex Teboul involved the removal of outliers and missing values, therefore requiring only a few supplementary steps on our side.

Possible Inconsistencies and Outliers. During our data exploration we identified 11,638 duplicate rows, constituting approximately 4.5% of the dataset. This is not unusual given the large dataset size and the prevalence of binary and categorical features (17 out of 21). For instance, features such as *CholCheck* and *Stroke* exhibit imbalances of 96.3% and 95.9%, respectively, increasing the likelihood of duplicates. Based on these observations, we decided to retain the duplicate rows without further treatment.

The *BMI* feature exhibits an unequal distribution. Here, the values between 19 and 44 form the 95% confidence interval, yet there are extreme values up to 98. Outlier detection methods like interquartile range (IQR) and median absolute deviation (MAD) identified 9,847 and 25,516 instances as outliers, respectively. However, a recent study by CDC (2023) indicates that over 20% of Americans are obese, thereby supporting the representativeness of the data at hand. Nonetheless, we bin *BMI* into four medically established categories: *Underweight* (0–18.5), *Normal Weight* (18.5–25), *Overweight* (25–30), and *Obesity* (30+) (CDC 2023). This balances the *BMI* distribution, improving its interpretability and utility, e.g., for algorithms like decision trees and kNN.

Furthermore, we conducted additional consistency checks; however, as these do not significantly influence the dataset, we will not discuss them further. Detailed procedures are documented in the *preprocessing.ipynb* notebook.

Feature Engineering and Transformation. In order to enhance feature interpretability as well as the performance of machine learning models, we perform a series of feature transformations. First, we convert the original target variable (*Diabetes_012*) into a binary feature (*Diabetes*) by merging the prediabetes and diabetes categories. The rationale for this merger is twofold. Firstly, we argue that prediabetes is already a form of diabetes, necessitating special attention. Secondly, it breaks the model prediction down into two classes and slightly reduces the target imbalance from 86.1% to 84.24%.

Additionally, we apply z-score normalization on the numerical features *MentHlth* and *PhysHlth*, which initially range from 0 to 30, using the `StandardScaler` from the scikit-learn library. This ensures that the features are on a comparable scale, which is critical for machine learning algorithms sensitive to feature magnitudes, such as kNN.

As outlined in section 2, we observed that no single feature exhibits an overly high correlation with the target variable or with any other feature. Consequently, the issue of false predictors or multi-collinearity is not a concern which is why we retained all features in the final dataset.

Dataset Splits and Sampling Techniques. Before performing the above mentioned transformations, we split our dataset into three distinct subsets: training (70%), validation (10%), and test (20%). Given the high imbalance of the target variable, we here use a stratified split in order to maintain consistent class distributions across all subsets. To facilitate the subsequent analysis, each split was stored in a .csv file. Likewise, we implemented a custom *DataLoader* class to enable simple loading and usage of these datasets in downstream tasks.

4 Data Mining Approaches

Baselines. To evaluate and compare the performance of our trained machine learning models, we implemented four simple baselines. The **first baseline**, *majority class*, always predicts the most common class of the target variable, i.e. “no diabetes”. The **second baseline**, *stratified*, generates random predictions based on the class distribution of the target variable. Specifically, it predicts “no diabetes” with a probability of 0.86

and "diabetes" with a probability of 0.14. The **third baseline**, *highest correlation*, leverages the feature *GenHlth* as a threshold for predicting diabetes, where individuals with bad general health are predicted to have diabetes. Therefore, we calculated the likelihood of a person having diabetes in each of the five categories. We identified the largest probability difference between categories 3 and 4, which serves as our decision boundary. Thus, the disease is predicted for all individuals with a "fair" (4) or "poor" (5) general health. The **fourth baseline**, *Principal Component Analysis (PCA)*, uses the the first principal component, which explains 47.2% of the variance in our dataset, to fit a K-Nearest-Neighbor (KNN) classifier for predictions. Although this method is slightly more complex, it still serves as a relatively simple baseline for comparison. The results of each baseline can be seen in table 1.

Resampling Techniques and PCA. Given our dataset's high imbalance, as outlined in section 2, we tested various resampling techniques to balance the class distribution, enhance generalization, and reduce bias towards the majority class. The resampling techniques employed include *Random Undersampling*, which reduces the size of the majority class by randomly removing samples, and *Random Oversampling*, which increases the size of the minority class by duplicating samples. Additionally, we utilized *SMOTE* (Synthetic Minority Oversampling Technique), which generates synthetic samples for the minority class based on feature-space similarities, and *SMOTE Tomek*, a hybrid approach that combines SMOTE with Tomek links to enhance class balance by simultaneously oversampling the minority class and removing ambiguous or overlapping samples of the majority class.

Additionally, we perform Principal Component Analysis and asses the effect of only using 5, 10 or all the computed components. PCA reduces the dimensionality of a dataset by transforming the original variables with linear combinations into a new set of uncorrelated variables. The resulting components are ordered from most important to least important, such that the first components capture the most variance.

Model Training via Cross-Validation. To identify the best combination of model-specific hyperparameters, resampling methods (including none), and PCA configurations (including none), we used stratified k-fold cross-validation, optimizing for the F1-score. While we tested different cross-validation methods, *halving grid search* was found to be most efficient, due to its low computational effort, compared to grid and random search. When optimizing on accuracy or recall, we observed that these metrics are not suitable for our case. In particular, accuracy-tuned models failed to identify positive instances, whereas recall-tuned models overlooked precision, leading to an increase in false positives. Consequently, the F1-score proved to be a more balanced and meaningful metric for optimization, as it simultaneously accounts for both recall and precision.

5 Machine Learning Models

In the following, we briefly introduce each of the different machine learning models considered for our diabetes risk prediction task which were all trained using the previously described approach and utilize implementation from scikit-learn (Pedregosa et al. 2011).

Distance-Based Models. The *k-Nearest Neighbors* (kNN) and *Nearest Centroid* classifiers are versatile algorithms often used for classification tasks, each leveraging distance-based methods for decision-making. The kNN algorithm assigns a class to a data point based on the majority class among its k nearest neighbors, determined by a specified distance metric, making it flexible for datasets with irregular decision boundaries. For our use case, we optimized kNN hyperparameters using scikit-learn's HalvingGridSearch, identifying the optimal configuration with Manhattan distance, $k = 100$, and SMOTE-Tomek resampling for class balance. PCA was unnecessary, as the original features provided sufficient discriminatory power.

Similarly, the Nearest Centroid classifier determines class membership by comparing the distance between a data point and the centroids of each class, with centroids representing the mean positions of class members. For our implementation, cross-validation and hyperparameter tuning revealed that Manhattan distance and a shrinkage threshold of 0.1 worked best, with SMOTE oversampling addressing class imbalance. Like kNN, PCA did not enhance performance, highlighting the adequacy of the original feature space for effective classification. Both methods provided robust solutions tailored to our dataset's characteristics.

Logistic Regression. The logistic regression model is a discriminative classifier which is effective on and frequently applied to binary classification tasks. It models the log odds of the probability that an observation belongs to a particular class as a linear combination of the independent variables. By applying maximum likelihood estimation, it fits a logistic (sigmoid) function to the data, mapping the linear predictor to a probability between 0 and 1. By establishing a decision boundary, often 0.5 in the binary case, the model can then be used to make predictions.

For our use case we applied the logistic regression from scikit-learn in combination with cross-validation for the hyperparameters of *regularization strength, penalty, optimization algorithm* and its *stopping criterion* as well as the previously outlined resampling techniques and PCA. It was found that the best results could be obtained with a regularization strength of 0.1, applying an l2 penalty and using the newton-cg solver with tolerance of 0.001. To handle the class imbalance, cross-validation found that the use of random oversampling and not applying PCA lead to the best results.

Tree-Based Models. Decision Trees are supervised learning classifiers that use decision rules derived from training data to make generalized predictions. However, trees with too many decision rules can get overly complex and are potentially prone to overfitting. To address this, various hyperparameters can be adjusted, such as *max_depth*, which prunes the tree at a specific depth to mitigate overfitting. Other useful hyperparemeters include *max_leaf_nodes*, *min_samples_split*, and *min_samples_leaf*, which further help to regulate the model's complexity. Beyond these, other parameters can be fine-tuned to adapt the model to specific datasets and problem types, such as regression or in our case classification.

In their standard form, decision tree models consist of a single tree. However, *Ensemble Methods* combine multiple decision trees or other base learner to enhance model robustness and generalization. Two common ensemble techniques are *Bagging* and *Boosting.*

Bagging uses bootstrapped datasets (sampling with replacement) to train multiple classifiers, combining their predictions through a voting or averaging mechanism. Here, we used *Random Forests*, which aggregate predictions from several learned decision trees. In contrast to training multiple trees in parallel, Boosting trains classifiers sequentially, where each model focuses on correcting the errors of its predecessor (Dey 2024). In our case, we chose Adaptive Boosting (AdaBoost) as our example boosting model.

To optimize model performance, we applied a halving grid search hyperparameter tuning approach with an extensive parameter grid for Decision Tree, Random Forest, and AdaBoost models. Among these, Random Forest emerged as the best-performing classifier with some of the following hyperparameters: *criterion=log_loss*, *n_estimators=500*, *min_samples_split=50*, *bootstrap: True.*

Support Vector Machines. The Support Vector Classifier (SVC) is a kernel-based binary classification model that identifies the optimal linear decision boundary (or hyperplane) to separate two classes, maximizing the margin from the nearest data points, called support vectors. These support vectors are selected from the training data and play a crucial role in defining the decision boundary. The regularization parameter C controls the trade-off between model complexity and the number of misclassifications. While a smaller C allows more support vectors, thereby simplifying the model, it potentially increases the number of misclassifications. Since our data is not linearly separable, we apply kernel functions to map the features into a higher-dimensional space. In this transformed space, the SVC can fit a linear hyperplane to separate the classes effectively.

After hyperparemter tuning, the SVC employed undersampling of the training data. This aligns with Zughrat et al. (2014), who found that undersampling outperforms oversampling on imbalanced datasets, improving generalization and reducing the computational complexity (due to less support vectors). Additionally, halving grid search selected a regularization of $C = 0.1$, and a polynomial kernel with degree 2. Due to computational constraints, we did not include PCA as a hyperparameter.

Naive Bayes. The probabilistic machine learning algorithm of Naive Bayes is based on Bayes' theorem. The "naive" assumption of this algorithm is that all features in a given dataset are independent of each other given the target variable. While this assumption allows a significant reduction in computational complexity, it is rarely accurate in real-world scenarios. Nevertheless, the Naive Bayes classifier has been observed to perform well in practice, even when compared to more sophisticated models (Rish 2001).

The scikit-learn library implements five distinct Naive Bayes algorithms. The *Gaussian Naive Bayes* assumes that the features are normally distributed, whereas the *Multinomial Naive Bayes* is suited for multinomially distributed data, i.e. for features representing frequencies like word counts in texts. The *Complement Naive Bayes* is an adaptation of the multinomial algorithm that is designed for imbalanced data. The *Bernoulli* and *Categorical Naive Bayes* are particularly effective for datasets with binary and categorical features, respectively. Following cross-validation in conjunction with hyperparameter tuning, the *CategoricalNB* showed the best performance when tuning on the F1-score with a smoothing parameter *alpha* of 0.5. Similarly, no dimensionality reduction via PCA was applied, but random oversampling was deemed to be beneficial.

Neural Network. An artificial neural network uses a collection of neurons with weighted links between them to process input data and make predictions. It applies backpropagation to calculate the gradient of the loss function for the network's weights and iteratively adjusts these weights through gradient descent to minimize prediction errors. While our project focused mainly on applying various traditional machine learning algorithms, we also tested the use of a neural network as they can model complex relationships in data. As the training is resource-intensive, we decided against any extensive hyperparameter optimization in this case, which would be beyond the scope of this project and only trained the model with SMOTE oversampling.

Our feed-forward neural network was set up with three fully connected layers, with the first transforming the input into 128 features, the second into 64 features, and the final into a single output using a sigmoid activation, allowing for binary classification. The first two layers use ReLU as their activation function, and each is followed by a 30% dropout for regularization. While the training was run for 100 epochs, the loss already dropped significantly in the first few epochs with some continued improvement on the training data, but only very limited progress on the validation data in later epochs. The final evaluation on the test set, outlined in section 6, demonstrates the promising potential of applying neural networks in diabetes risk prediction.

6 Evaluation

Having trained and identified the best classifier of each model class, they can be compared against each other and the baseline, based on their performance on the previously unseen test set. Table 1 gives an overview of the obtained scores and while all main metrics were considered and can be found in the *evaluation.ipynb* notebook, we decided to mainly focus on the positive class, in particular its recall. Correctly identifying patients with diabetes is critical in a medical setting, where missing true positives can have severe consequences. At the same time, precision is likewise of importance, as unnecessary interventions from misclassifying healthy individuals as diabetic should be minimized to avoid not needed treatments. Therefore, it is sensible to also consider the F1-score on the positive class as a balanced performance metric.

While the four implemented baselines demonstrate a high accuracy and perform well on the negative class, they show significant shortcomings as their performance on all metrics related to the positive class is notably very poor. Consequently, they fall short in both overall performance and real-world applicability and are ultimately outperformed by every implemented machine learning model.

Comparing the different traditional models, and therefore excluding the Neural Network, it becomes evident that SVMs and Nearest Centroid perform the worst across metrics and are therefore less suitable for the task of diabetes risk prediction. Logistic Regression demonstrates a moderate performance, achieving a good F1-score. The trained Decision Tree achieves the highest recall on the positive class, but its low precision diminishes its overall effectiveness. Random Forest, closely followed by AdaBoost Decision Trees, emerge as the most effective model, achieving the highest F1-score of 0.4770. Despite their strong overall performance, it is worth noting that their recall on

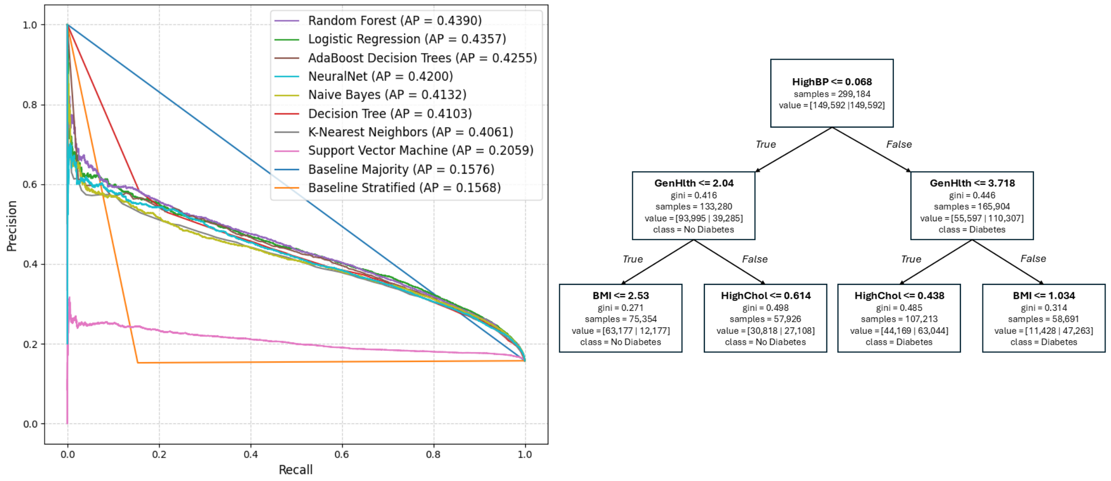
Model	Accuracy	Precision 1	Recall 1	F1-Score 1
Baseline Majority	**0.8424**	0.0	0.0	0.0
Baseline Stratified	0.7343	0.1547	0.1538	0.1542
Baseline Correlation	0.7940	**0.3620**	**0.4024**	**0.3811**
Baseline PCA	0.8154	0.2489	0.0849	0.1266
Logistic Regression	0.728	0.3395	0.7678	0.4708
Decision Tree	0.6988	0.3153	**0.7781**	0.4488
Random Forest	**0.7535**	**0.3583**	0.7131	**0.4770**
AdaBoost Tree	0.7467	0.3507	0.7132	0.4702
SVM	0.4792	0.1857	0.6809	0.2918
KNN	0.7433	0.3448	0.6988	0.4617
Nearest Centroid	0.6902	0.3051	0.7562	0.4348
Naive Bayes	0.7356	0.3411	0.7279	0.4645
Neural Network	**0.7775**	**0.3753**	0.6205	0.4677

Table 1: Performance Metrics for the Different Models

the negative class is only average when compared to other models. However, their higher precision enables these models to strike the best balance between performance metrics, making them the most reliable models for the problem at hand.

As commonly used for imbalanced datasets, the precision-recall (PR) curve of our classifiers in Figure 1a displays the inherent trade-off between precision and recall on the positive class. When decision thresholds are applied to the "majority" baseline, the resulting PR curve reveals a linear decline in precision as recall increases. Initially, the baseline predicts only the majority class, resulting in high precision but a recall of zero. As the decision threshold is adjusted to include progressively larger portions of the minority class, recall steadily increases and precision decreases. The curve converges at a recall of 1 and a precision of 0.16. For the "stratified" baseline, the curve decreases linearly early on and already at a recall of 0.2 a precision of 0.16 is reached and the curve stays constant from this point on. For the other implemented ML models, the PR curves generally fall between these two baselines and ultimately outperform the "majority" baseline for a recall higher than 0.8 as they are able to sustain better levels of precision. An exception from this is the SVM, which significantly underperforms compared to the other models. Comparing the average precision (AP), i.e., the largest area under the PR curve, Random Forests and Logistic Regression show the best scores, reflecting their previously observed good performance metrics.

After comparing the classifiers based on statistical and visual evaluations, we analyzed the structure of a symbolic model by examining the tuned decision tree and the first two trees from the random forest model as examples of this architecture. As shown in figure 1b, the tuned decision tree identifeies *HighBP* as its most important feature, positioned at the root node. This choice seems plausible, as *HighBP* is the feature with the second highest correlation to the target variable and is in contrast to *GenHlth* binary, enabling



(a) Precision Recall Curve

(b) First two Layers of tuned Decision Tree

Figure 1: Performance comparison of models and visualization of best decision tree.

a cleaner split. As the decision tree used random oversampling, the root node considered 299,184 samples which were equally distributed between the classes "no diabetes" and "diabetes". According to the first split, individuals with high blood pressure are nearly three times more likely to have diabetes than those without high blood pressure. The feature *GenHlth* is used for splitting the next layer on both sides of the decision tree, making it the second most important feature in this model. At the second layer, the features *BMI* and *HighChol* are used to further refine the purity of the data along each branch.

The random forest model, which was determined to be the best model, demonstrates greater variety in the feature selection compared to the tuned decision tree. For instance, in the first decision tree of the random forest, *BMI* serves as the root node, followed by the features *HeartDiseaseorAttack* and *Age* at the first layer. Although, the second decision tree uses the general health as its most decisive feature (root node), the feature *DiffWalk* is used for further splitting the samples in the first layer. However, the feature selection in the trees under consideration are not arbitrary, as all those features observed in the higher levels demonstrate a high correlation with the target variable. The differences in feature importance between the random forest and the standard decision tree are not contradictory but rather reflect the random forest's architecture. Each tree within the forest is trained on a random subset of data and features, a design for reducing overfitting and promoting generalization. Consequently, features like *BMI* or *GenHlth* may be root nodes in some trees while appearing in higher layers of the standard decision of the standard decision tree (see figure 1b). Ultimately, the features of *HighBP*, *GenHlth*, and *BMI* emerged as the three most important features across the observed example trees. This finding aligns with our observations from the data exploration phase, reinforcing the reliability of these features for predicting diabetes. Furthermore, we evaluated the accuracy of our assumption that prediabetes is associated with diabetes. For this, we examined the predictions of our models on the test data that corresponded to prediabetes before merging with diabetes. Our findings revealed that our models predicted in over 60% of cases prediabetes (diabetes) correctly. Consequently, we conclude that our decision to merge prediabetes with diabetes was appropriate and did not result in erroneous predictions.

In addition to the traditional machine learning models, a more advanced neural network was implemented. Despite limited training and hyperparameter tuning, this neural network demonstrates remarkable initial results, showing its potential to contend with the other trained models. It achieves the highest accuracy of 0.7775 and a precision of 0.3753, alongside a strong F1-score. However, its performance is notably hindered by a significant shortcoming in recall, which lags behind the other models. Nevertheless this preliminary result underscores the promising potential of neural networks in this context. With further tuning of the model architecture, adjustment of the decision boundary and hyperparameters, as well as more extensive training, it is likely that the neural network's performance could further improve.

Comparison with Kaggle Notebook. To further validate our results, we identified a "state-of-the-art" notebook on Kaggle that utilized the same dataset for diabetes prediction. To ensure a robust comparison, we selected the highest-"upvoted" notebook that met our criteria. The sixth-ranked notebook (Macherini 2023) by Gabriel Macherini satisfied all requirements, as the notebooks in the top five were either unsuitable due to using different datasets, or lack appropriate machine learning techniques.

Since the original notebook did not report precision and recall on the unbalanced test set, we replicated and reran the code to extract the required metrics for comparison. The best-performing model in the notebook was a *Random Forest* classifier (as in our evaluation), achieving a test accuracy of 0.78 using the *SMOTE* resampling technique. On the positive class, the model attained a recall of 0.57 and precision of 0.34, resulting in an F1-score of 0.43. These metrics are notably lower than its performance on a balanced test set, where it achieved a precision of 0.88, recall of 0.96, and an F1-score of 0.92. Compared to the imbalanced test set, our model demonstrated superior recall with the same precision, outperforming the referenced model.

Conclusion. In conclusion, this project successfully implemented a variety of machine learning models and conducted extensive hyperparameter optimization, including evaluations of the most effective resampling strategies. As a result, several well-performing models were developed, that are on par with models implemented by others, though their real-world applicability remains to be explored. The project demonstrates how a variety of data mining techniques can be leveraged to implement predictive models in the context of diabetes risk prediction. The code of this project can be found in the following GitHub repository: `https://github.com/benediktpri/ie500_data_mining_project`

Bibliography

CDC (2022, April). CDC - 2014 BRFSS Survey Data and Documentation.

CDC (2023, September). Adult Obesity Prevalence Maps.

CDC (2024, July). National Diabetes Statistics Report.

Dey, R. (2024, January). Bagging v/s Boosting.

Macherini, G. (2023). Diabetes EDA & Prediction.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Rish, I. (2001, 01). An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell 3*, 6.

Teboul, A. (2024). Diabetes Health Indicators Dataset.

Zughrat, A., M. Mahfouf, Y. Yang, and S. Thornton (2014). Support vector machines for class imbalance rail data classification with bootstrapping-based over-sampling and under-sampling. *IFAC Proceedings Volumes 47*(3), 8756–8761.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Translation and Rephrasing	Throughout	+
GitHub Copilot	Code generation	Throughout	++
ChatGPT	Code generation	Throughout	++

P. Gänz

Salome Heckenthaler

B. Prisett

Unterschrift
Mannheim, den 08. Dezember 2024