Diabetes Classification

Nicholas Wong
University of North Carolina at Chapel Hill
nicwjh@ad.unc.edu

Hannes Brinklert
Lund University
ha7075br-s@student.lu.se

Abstract

Diabetes is a metabolic condition that can lead to severe health complications, even premature death, if not detected and treated early. In this paper, we propose a machine learning-based approach for the binary classification of diabetes based on a limited set of input features to test the viability of easy detection of the disease when access to a full set of input features is not possible. For diabetes classification, we deploy five different classifiers - logistic regression, Naive Bayes, nearest neighbors, decision trees, and Support Vector Machine (SVM). The motive of this study is to attempt to find a model with a limited set of input features that can prognosticate the likelihood of incidence of diabetes in a patient with maximum accuracy. Experiments are run on the Diabetes Health Indicators Dataset sourced from the Kaggle repository. For the performance evaluation of our classifiers, measures such as precision, recall, and f-score are used as measurement metrics. Results show that Naive Bayes and SVM are the best models to use with a limited input feature set. For the case of decision tree, it achieves better performance with fewer input features compared to the full feature set, indicating possible issues with overfitting large feature sets.

1 Introduction

Diabetes, where one's body is unable to produce the requisite amount of insulin to regular blood sugar, is one of the most pressing health problems facing Americans today. According to the Centers for Disease Control and Prevention (CDC), approximately 1 in 10 American have diabetes. Of which, 1 in 5 are unaware that they are afflicted. Moreover, the COVID-19 pandemic has culminated in widespread lockdowns and social-distancing measures in an attempt to contain the spread of the virus. In the advent of increased isolation, access to physicians, and consequently early diabetes detection, is becoming increasingly prohibitive. These factors in tandem may pose as a potential barrier preventing the early detection of diabetes. Early detection of diabetes is essential to prevent severe health complications and allow the patient time to take steps to reduce risk of diabetes, prolong the disease, and reduce cardiovascular morbidity and mortality[14]. Consequently, early identification is the only foolproof way to remedy these complications.

This project aims to explore whether convenient health screenings with fewer input features can provide a viable alternative to a full physician's model with more input features in the context of diabetes classification. Specifically, we aim to compare the performance of logistic regression, Naive Bayes, nearest neighbors, decision trees, and SVM in diabetes classification for models with more and fewer features by evaluating their performance through precision, recall, and f-score. With the identification of an important subset of input features, early detection of diabetes becomes easier as patients no longer have to seek out the full set of health data for accurate diabetes prediction. As a result, proper treatment can be provided at an earlier time period to mitigate risk, prolong the disease, and avoid complications associated with late detection and treatment of diabetes.

2 Related work

Prior to beginning, we surveyed other projects that used machine learning algorithms to explore a similar problem space. Several projects applying machine learning for classification included a data pre-processing step to clean and refine raw data for improved classification. Sharma et al., for instance, conducted pre-processing on image data for improved feature selection [22]. Common in the data cleaning step was to implement methods such as regression and clustering to remove noisy data. For this research work, we adapt a binning method for feature engineering our continuous input features.

Roy et al. in [21] normalized his data through standard scaling methods to prepare them as inputs for his Artificial Neural Network (ANN) machine learning model to optimize accuracy. Normalization makes sense for the implementation of models where continuous variables are used as input features as differing magnitudes of input features can skew the model in favor of one feature over another - normalization curbs this problem by rescaling each input feature. In this project, normalization is not used because all input features to our models are categorical variables after data preprocessing. While it is good practice to normalize features in different scales because these features are multiplied by model weights, it makes no sense in the case of categorical variables because encoded variables often contain values 0 and 1 which have little meaning in and of itself.

As a corollary to the above point on normalization, the utilization of one-hot encoding was also commonplace for normalization of non-numeric data. One-hot encoding is a normalization technique frequently used to quantify categorical data before feeding them into machine learning algorithms. Fiker et al. in [8] uses one-hot encoding to normalize categorical input features before feeding them into an ANN backpropagation algorithm. Ahamed et al. in [6] uses a similar procedure of one-hot encoding to transform categorical values into numeric data before feeding them into ML algorithms. For our project, we opt not to one-hot encode our variables due to the large number of categories in the dataset - high memory consumption may become a potential issue if we were to one-hot encode all of our categorical variables. Moreover, a large proportion of our categorical input features are ordinal. In the case of ordinal input features, multicollinearity may be a potential problem with one-hot encoding[11].

Across the different studies we surveyed, a multitude of different machine learning classification algorithms were used. However, one commonality many seemed to share was the utilization of logistic regression. Logistic regression was frequently used by prior studies - either directly in the study or as a baseline metric due to its performance against optimised machine learning models [17]. Notably, Lynam's study found that traditional regression modelling compares favorably to machine learning when using a small number of well understood, strong predictor variables[17]. Since our project is primarily concerned with comparing models with few input features to models with more input features, logistic regression serves as an appropriate baseline.

As such, we have determined that logistic regression will serve as the baseline classifier in our model. This decision was made not simply because logistic regression is frequently used, but because the panoply of past data on logistic regression available makes sense for it to serve as a baseline for comparison. With more prior results for comparison, it may also provide a valuable learning opportunity for us in applying our selected classification models.

Certain studies we surveyed used Artificial Neural Networks (ANN) for diabetes classification. Most common among these studies was the utilization of a multilayer feedforward neural network with no loops, feedback, or signal moves in the backwards direction (from output, hidden, to input layer) [7]. These studies also commonly used the Levenberg-Marquardt learning algorithm. For this project, we exclude the use of ANNs for classification for several reasons. Neural networks are more susceptible to overfitting due to the number of variables and hidden layers. Given the large number of parameters being estimated with our full model with all available input features, the decision was made that logistic regression would be a more appropriate choice since we would be concerned with the data to parameter ratio in the case of implementing an ANN. A ratio of data to parameters that skewed too small would be cause for concern in the case of a complex ANN. Hence, we exclude the implementation of neural networks in this research work for concision and to avoid the risk of overfitting.

3 Methods

As a high-level overview of our methodology, the data is first read and preprocessed. Data preprocessing occurs in several stages; continuous input features are transformed to categorical features and the data is partitioned into training and test sets. Feature engineering is implemented by bucketing BMI, thereby transforming the numeric data into categorical features before they are fed into our ML algorithms.

After data preprocessing, we conduct feature selection for our model with fewer features in a three-stage process through hand-picking, LASSO regression (L1 regularization with logistic regression), and forwards greedy selection. 10-fold cross-validation then is conducted for hyperparameter tuning where we find the optimal hyperparameter k to be used for our nearest neighbor method.

Finally, our models with more and less features are run against two different test sets to evaluate performance.

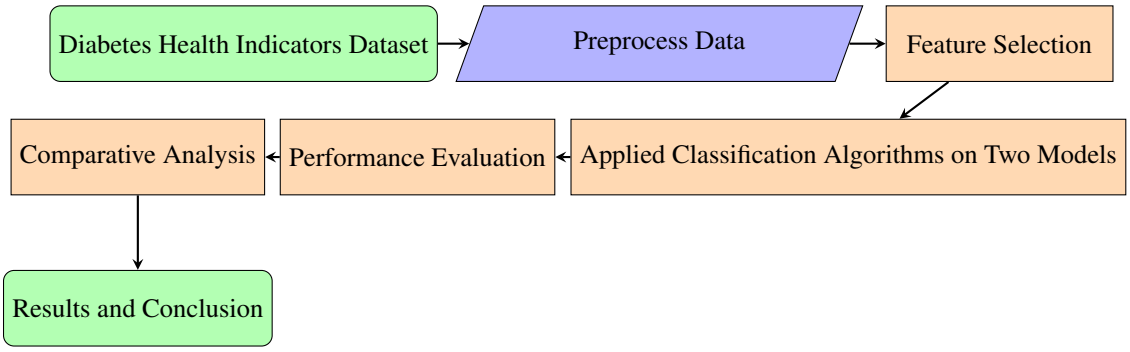

Table 1: Architecture of the Proposed System

These individual methods are expanded upon in greater detail within the subsections below.

3.1 Feature selection

In order to have an accurate assessment of how well different ways (logistic regression, Naive Bayes, decision tree, K-nearest neighbors, SVM) of fitting an estimator work with different subsets of features, we partition the data into a training and test set in an 80/20 ratio. The larger portion of this split is used to train our models and perform feature selection. The remaining 20% serves as held-out data that will be used for model evaluation.

A substantial portion of this project hinges on the appropriate selection of input features to be included in the model with fewer features. To finalize this list of input features to be included in the smaller model, we hand-picked a set of input features (7 of the 21 available input features) and run two different feature selection algorithms, L1 regularization and sequential feature selection, to get two more sets of 7 input features. After these algorithms are run, we compare the three sets of 7 input features and observe for overlapping features. Any input feature that occurs more than twice in the three different sets of features is then included in our finalized list of input features for the smaller model. We present a more detailed breakdown of how these three sets of input features are obtained below.

3.1.1 Hand-picked features

Our hand picked features are *high blood pressure, high cholesterol, BMI, physical activity, general health, mental health, sex* and *age*. These features were selected based on preliminary research by our team on the highest factors that correlate with incidence of diabetes. Diabetics are twice as likely to have high blood pressure [12]; diabetics are more prone to having high cholesterol [10]; prevalence of diabetes is multiple times higher for obese individuals [20]; men are twice as likely to develop type II diabetes as women [18]; diabetes holds a high correlation with depression and general mental well-being[9].

We present a table summarizing these input features below, with an additional table summarizing the rest of the input features in the appendix[6].

Table 2: Hand-picked Reduced Feature Set

Feature	Description
High Blood Pressure (HighBP)	0 = no high BP 1 = high BP
High Cholesterol (HighChol)	0 = no high cholesterol 1 = high cholesterol
Body Mass Index (BMI)	Continuous feature
Physical Activity	Physical activity in past 30 days - not including job 0 = no 1 = yes
General Health	Scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor
Mental Health	Ranking of mental health (1-30)
Sex	0 = female 1 = male

3.1.2 Feature Selection with L1 Regularization

We deploy a logistic regression model with L1 regularization, also known as LASSO regression, as an alternative method of performing feature selection. LASSO was chosen specifically because it has the ability to reduce some of the coefficients to zero - this allows for shrinkage and model selection[28]. When these coefficients go to zero, that feature can be safely removed from the model. We deploy LASSO regression, with loss defined as:

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_1^n |w_i|$$

The λ hyperparameter, controlling the regularization penalty, is particularly pertinent for LASSO because different values of λ will result in different features being selected. To accomplish this, we deploy 10-fold cross-validation on the 80% training data. We optimize with the $\lambda \sum_1^n |w_i|$ penalty, select the remaining features with nonzero coefficients, train an unpenalized model with those features, and compare performance via average F-score. We repeat this process for five different values of λ (λ = 1, 0.5, 0.1, 0.05, 0.001) on each of the 10 folds and choose the λ that gives the best performance by F-score. λ = 0.001 was chosen as the outcome of this process.

3.1.3 Sequential feature selection

LASSO regression, as implemented above, acts as a backwards greedy feature selection algorithm. We further implement sequential feature selection, a forwards greedy selection method[23]. At each stage, the best feature is chosen to be added based on the cross-validation score of logistic regression. A feature subset is formed through this iterative process where features are sequentially selected in a greedy fashion.

3.2 Models

Our deployed classification algorithms are the following: *logistic regression, categorical Naive Bayes, decision tree, 9-nearest neighbors and SVM*. Two models will be produced for each algorithm - one trained with our limited set of overlapping features and one trained with the full set of input features. Both models will be trained with the same amount of observations and tested on the same survey instances but with different amounts of input features.

The models are trained on the same training set - 80% of the balanced dataset. Subsequently, they will be tested on two different test sets. One of these test sets is the held out 20% of the balanced dataset. The other is true unseen, held-out data that we obtained through a process of data wrangling.

Sklearn's implemented models for the aforementioned algorithms will be adapted and customized for use in this project. The implementation of the project will be written in the Python programming language and developed in the Jupyter Notebook environment.

The aim of the project is to see if it is possible to find a model that is more parsimonious with respect to input features that can serve as a viable alternative and achieve comparable performance to a model with the full set of input features in the context of diabetes classification.

3.2.1 Logistic Regression

Logistic regression is a supervised learning method that predicts the outcome of a categorical dependent variable. We implement logistic regression with L2 regularization, otherwise known as ridge regression, to predict our single outcome label (diabetes/no diabetes)[24]. Ridge regression is implemented with an L2 norm penalty controlling variability, where the loss is defined as:

$$Loss = Error(Y - \widehat{Y}) + \lambda \sum_{1}^{n} w_i^2$$

During implementation, we had an issue with convergence that required adaptation of our logistic regression model. The input parameter *max_iter* for the Logistic Regression controls the ceiling of iterations allowed for the convergence of the solver. However, with Sklearn's default value of 100, the solver did not converge and *max_iter* was increased to 500 to remedy this issue.

3.2.2 Decision Tree

Decision trees are a supervised learning method that we deploy for classification. Decision rules are inferred from the data features to create a model that mimics a tree structure through a piecewise constant approximation. We deploy Sklearn's Decision Tree classifier with a binary output label for diabetes classifiation[1].

3.2.3 Nearest neighbors

K-nearest neighbors is a non-parametric supervised learning classifier that makes predictions based on the grouping of an individual data point. Based on a predefined number of training samples that are in closest distance to that datapoint, a label is predicted.

To choose the hyperparameter *k* for the nearest neighbors algorithm, we run 10-fold cross validation on the balanced dataset, iterating through values {3, 4, 5, 6, 7, 8, 9, 10}. Out of this set of values, *k* = 9 produced the best performance by f-score.

Hence, we deploy the nearest neighbors algorithm with hyperparameter *k* = 9, chosen after a process of cross-validation[3].

3.2.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning method that has multiple uses including classification, regression, and outlier detection.

Sklearn's standard support vector classifier (SVC) has a fit time that scales quadratically with sample size and is thus not practical for classifying a dataset as large as ours[27]. As a result, we opted to implement Sklearn's Linear Support Vector Classifer (LinearSVC) which has more flexibility in regards to penalties/loss functions that should scale better with sample size[26].

SVM is implemented with a linear kernel, L2 regularization, and squared hinge loss function (L2 loss) defined as:

$$max(0, 1 - y \cdot \hat{y})^2$$

The hinge loss function is a loss function used to train classifiers in SVM. The squared hinge loss function was chosen to smooth the surface of the error function and make it easier to deal with. While it may also be more sensitive to outliers, the better performance achieved on binary classification made it the more appropriate choice for this research work. We deploy L2-SVM with a linear kernel for the classification of our binary output label[2].

3.2.5 Naive Bayes

The Naive Bayes classifier is a supervised machine learning algorithm that is used for classification tasks. Unique to Naive Bayes is the "naive" assumption of conditional independence, where features are assumed to be independent given class[4].

For this project, since all input features are categorical after data preprocessing, we implement the categorical Naive Bayes (CategoricalNB) classifier which assumes that each feature has its own categorical distribution[4].

For categorical Naive Bayes, the probability of category t in feature i given class c is given by the likelihood:

$$P(x_i = t|y = c; \alpha) = \frac{N_{\text{tic}} + \alpha}{N_c + \alpha n_i}$$

Where $N_{\text{tic}} = |\{j \in J|x_{\text{ij}} = t, y_j = c\}|$ is the number of times category t appears in the samples x_i, which belong to class c; $N_c = |\{j \in J|y_j = c\}|$ is the number of samples with class c, α is a smoothing parameter and n_i is the number of categories of feature *i*.

4 Experiments/Results

4.1 Data

This project uses two datasets from a 2015 health survey, sourced from the Kaggle page *Diabetes Health Indicators Dataset*[29]. Three different data sets were posted on the page from the 2015 survey; a balanced dataset with equal class distribution; an unbalanced dataset with unequal class distribution; an unbalanced dataset, identical to the second one, but with three output labels instead of two. The third dataset will not be used because it is contains 3 output classes and is otherwise identical to the second dataset[29]. The three output labels in this unused dataset are: no diabetes, prediabetes, and diabetes with labels of 0, 1, and 2 respectively. Since this project is primarily concerned with implementing a binary classification of diabetes, the two datasets with binary output labels will be used where labels 1 and 2 (prediabetes and diabetes) are combined and denoted with label 1.

Both data sets used in this project contain 22 features, one of which is an output label indicating if the person has diabetes or not (1 if the person has diabetes or prediabetes and 0 for no diabetes)[29]. The primary difference between the two data sets is that one is balanced in its output classes (the same amount of diabetes and non-diabetes responses), with 70,692 total responses; the other dataset is unbalanced, with 253,680 total responses[29]. The first dataset was obtained by retaining all observations with output label 1 in the unbalanced dataset and matching that number with a random subset of observations with output label 0[13]. Effectively, this makes the balanced dataset a subset of the unbalanced dataset.

All of the input features are either binary or categorical, except for BMI. In data preprocessing, we perform feature engineering by bucketing BMI categorically in order to fit a categorical Naive Bayes model. We take the difference between the maximum and minimum values of BMI, 98.0 and 12.0, to obtain the range of BMI in the dataset. Observations are then fitted into categories, where each category represents an individual integer range (e.g. 12.0-12.99) for a BMI value. After bucketing, the BMI data has 87 categories from 12.0 to 98.0, where for instance, bucket 15 contains the values $15 \leq bucket_{15} < 16$. The balanced dataset is then split using sklearn's *model_selection.train_test_split* function with a training/test split in an 80/20 ratio.

Since the balanced dataset is a subset of the unbalanced dataset, the unbalanced dataset provides a bevy of true unseen, held-out data that can be used for testing our models. To take advantage of this and obtain more unseen data for testing, we take the set difference between the unbalanced and balanced dataset. The result of this set difference operation is a new dataset with 160,417 observations that is unseen with respect to our trained models, but with an output label of 0 for all observations. Observations with an output label of 1 represent the intersection between the initial balanced and unbalanced datasets, so all observations with diabetes (output label = 1) are removed as a result of this set difference operation.

4.2 Evaluation

The two models are evaluated using recall, precision, and f-score. Recall is also known as sensitivity, which can be explained as the share of found ones of all the ones in the data[19]. Precision can be explained as the share of predicted ones that are correctly predicted[19]. F-score, a measurement based on recall and precision, is a harmonic mean where 1.0 is the highest value and 0.0 is the lowest[15]. In the calculation of precision and recall, true positives *tp*, false positives *fp*, and false negatives *fn* are used. True positives(tp) are defined as the correctly predicted ones; false positives(fp) are predicted ones that are zeros in the correct data; false negatives(fn) are predicted zeros which are

ones in the correct data[19]. These metrics are calculated empirically as follows: $precision = \frac{tp}{tp+fp}$, $recall = \frac{tp}{tp+fn}$, $F = 2 \cdot \frac{recall \cdot precision}{recall+precision}$[15, 19].

To calculate the above metrics, Sklearn's *precision_score, recall_score* and *f1_score* classes will be used. However, since the unseen test set we obtained by taking the set difference does not contain any observations with an output label of 1, the above metrics cannot be used to evaluate performance on that test set since *tp* is 0. This would result in a division by zero in the calculation of the f-score. To circumvent this problem, Sklearn's *accuracy_score* will be used for performance evaluation[25]. Instead of relying on *tp*, *accuracy_score* calculates a fraction of correct predictions over n_{samples} as a measure of performance: $accuracy(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$[5].

4.3 Results

Table 3 displays the result from the two feature selection algorithms alongside our hand-picked features. As described in the methods section, features that have a frequency of two or higher between these three selection methods are included in our final list of overlapping features, which forms the feature set for our model with fewer features. We present the results of our feature selection process below.

Table 3: Selected Features for Smaller Model

L1 Regularization	SequentialFeatureSelector	Hand-picked	Overlapping Features
HighBP	HighBP	HighBP	HighBP
HighChol	HighChol	HighChol	HighChol
BMI	CholCheck	BMI	BMI
GenHlth	BMI	PhysActivity	GenHlth
MentHlth	HeartDiseaseorAttack	GenHlth	MentHlth
Age	GenHlth	MentHlth	Age
Income	Age	Sex	

Across all classifiers in our balanced dataset(Table 4), models with less features perform comparably well as models with the full feature set with the exception of decision tree, where the performance difference between models is more significant. Notably, when comparing models with less features, the Naive Bayes classifier achieves the best performance, beating out our baseline logistic regression. Comparing overall performance across both models with less and more features with our baseline of logistic regression, the Naive Bayes and SVM classifiers compare most favorably while decision tree the least.

Table 4: Balanced Test Set

Model	# of features	Precision	Recall	F-score
Logistic Regression	less	0.740322	0.760538	0.750294
	more	0.745007	0.762638	0.753719
9-nearest neighbor	less	0.710054	0.741773	0.725567
	more	0.707478	0.748495	0.727409
Naive Bayes	less	0.731243	0.773841	0.751939
	more	0.744883	0.723708	0.734143
Decision Tree	less	0.704923	0.683798	0.694200
	more	0.667198	0.645148	0.655987
SVM	less	0.717394	0.777039	0.746026
	more	0.727248	0.781087	0.753207

In the unbalanced test set(Table 5), a similar theme is repeated where the Naive Bayes and SVM classifiers achieve the best performance by accuracy when compared to our baseline of logistic regression. Interestingly, better performance was achieved in the model with fewer features for two classifiers - decision tree and nearest neighbors.

Moreover, between both the balanced and unbalanced data, we observe decision tree have the greatest discrepancy in performance between the model with fewer and more features. This may be suggestive of an overfitted/overly complex tree model being generated by the decision tree classifier when the full set of input features are provided.

Table 5: Unbalanced Test Set

Model	# of features	Accuracy
Logistic Regression	less	0.687633
	more	0.695101
9-nearest neighbor	less	0.656233
	more	0.650430
Naive Bayes	less	0.674586
	more	0.705175
Decision Tree	less	0.676655
	more	0.630893
SVM	less	0.675895
	more	0.685376

5 Conclusion

Diabetes is amongst the leading causes of death worldwide, being responsible for over 100,000 mortalities in the U.S in 2021[16]. The development of machine learning algorithms that can assist with early classification of diabetes with a limited input feature set could prove instrumental for curbing this chronic problem during times of increasing social isolation.

In accordance with our obtained results, we have the highest likelihood of obtaining better accuracy in classification of diabetes with a limited input feature set when it is applied to the SVM or Naive Bayes classifiers. The decision tree classifier, on the other hand, performs poorly relative to our baseline classifier of logistic regression. This can be attributed to a few potential factors - we suspect that the decision tree classifier may have created an overly complex tree that does not generalize our data well. Steps such as pruning (reducing the size of the decision tree), setting a depth threshold, or sample threshold for the leaf node could potentially help improve the performance of the decision tree classifier[1].

The conditional independence assumption of our Naive Bayesian network, however, could be limiting in certain scenarios when feature independence given class is not realistic. Consequently, a potential area for future expansion of this research work to circumvent this limitation, briefly touched on in the related works section, is the implementation of neural networks for diabetes classification with a limited set of input features. We decided to omit the implementation of neural networks in this research work but it is important to note that our exclusion of ANNs is not a commentary on its utility for diabetes classificaton. Alić et al. found that ANNs frequently outperform Naive Bayesian networks for health classification, achieving higher accuracy metrics in the classification of both diabetes and cardiovascular diseases[7]. For this research work, ANNs were omitted for two primary reasons - concerns with overfitting and to keep our recommended models limited to classifiers with relatively few hyperparameters for greater reproducibility. However, exploring the utility of ANNs for diabetes classification could be of interest to future researchers.

References

[1] *1.10. Decision Trees — scikit-learn.org*. https://scikit-learn.org/stable/modules/tree.html. [Accessed 20-Apr-2023].

[2] *1.4. Support Vector Machines — scikit-learn.org*. https://scikit-learn.org/stable/modules/svm.html. [Accessed 20-Apr-2023].

[3] *1.6. Nearest Neighbors — scikit-learn.org*. https://scikit-learn.org/stable/modules/neighbors.html. [Accessed 20-Apr-2023].

[4] *1.9. Naive Bayes — scikit-learn.org*. https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed 23-Apr-2023].

[5] *3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn.org*. https://scikit-learn.org/stable/modules/model_evaluation.html. [Accessed 25-Apr-2023].

[6] B. Shamreen Ahamed et al. "Diabetes Mellitus Disease Prediction and Type Classification Involving Predictive Modeling Using Machine Learning Techniques and Classifiers". In: *Applied Computational Intelligence and Soft Computing* 2022 (Dec. 2022). Ed. by Babek Erdebilli (B. D. Rouyendegh), pp. 1–11. DOI: 10.1155/2022/7899364. URL: https://doi.org/10.1155/2022/7899364.

[7] Berina Alić, Lejla Gurbeta, and Almir Badnjević. "Machine learning techniques for classification of diabetes and cardiovascular diseases". In: *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. 2017, pp. 1–4. DOI: 10.1109/MECO.2017.7977152.

[8] Fiker Aofa et al. "Early Detection System Of Diabetes Mellitus Disease Using Artificial Neural Network Backpropagation With Adaptive Learning Rate And Particle Swarm Optimization". In: *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*. 2018, pp. 1–5. DOI: 10.1109/ICICOS.2018.8621683.

[9] CDC. *Diabetes and Mental Health — cdc.gov*. https://www.cdc.gov/diabetes/managing/mental-health.html. [Accessed 13-Apr-2023].

[10] *Cholesterol and Diabetes — heart.org*. https://www.heart.org/en/health-topics/diabetes/diabetes-complications-and-risks/cholesterol-abnormalities--diabetes. [Accessed 13-Apr-2023].

[11] Victor Dey. *When to Use One-Hot Encoding in Deep Learning? — analyticsindiamag.com*. https://analyticsindiamag.com/when-to-use-one-hot-encoding-in-deep-learning/. [Accessed 16-Apr-2023].

[12] *Diabetes and High Blood Pressure — hopkinsmedicine.org*. https://www.hopkinsmedicine.org/health/conditions-and-diseases/diabetes/diabetes-and-high-blood-pressure. [Accessed 13-Apr-2023].

[13] *Diabetes Health Indicators Dataset Notebook — kaggle.com*. https://www.kaggle.com/code/alexteboul/diabetes-health-indicators-dataset-notebook/notebook. [Accessed 25-Apr-2023].

[14] *Early Detection and Treatment of Type 2 Diabetes Reduce Cardiovascular Morbidity and Mortality: A Simulation of the Results of the Anglo-Danish-Dutch Study of Intensive Treatment in People With Screen-Detected Diabetes in Primary Care (ADDITION-Europe) — ncbi.nlm.nih.gov*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4512138/. [Accessed 16-Apr-2023].

[15] *F-score*. Feb. 2023. URL: https://en.wikipedia.org/w/index.php?title=F-score&oldid=1139808388.

[16] *FastStats — cdc.gov*. https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm. [Accessed 20-Apr-2023]. 2021.

[17] Anita L. Lynam et al. "Logistic regression has similar performance to optimised machine learning algorithms in a clinical setting: application to the discrimination between type 1 and type 2 diabetes in young adults". In: *Diagnostic and Prognostic Research* 4.1 (June 2020). DOI: 10.1186/s41512-020-00075-2. URL: https://doi.org/10.1186/s41512-020-00075-2.

[18] PhD Morgan Meissner. *Type 2 diabetes in men vs. women — medicalnewstoday.com*. https://www.medicalnewstoday.com/articles/diabetes-affects-men-women. [Accessed 13-Apr-2023].

[19] *Precision and recall*. Feb. 2023. URL: https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=1142085528.

[20] *Prevalence of Diabetes and Its Relationship With Body Mass Index Among Elderly People in a Rural Area of Northeastern State of India — ncbi.nlm.nih.gov*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7886600. [Accessed 13-Apr-2023].

[21] Kumarmangal Roy et al. "An Enhanced Machine Learning Framework for Type 2 Diabetes Classification Using Imbalanced Data with Missing Values". In: *Complexity* 2021 (July 2021). Ed. by M. Irfan Uddin, pp. 1–21. DOI: 10.1155/2021/9953314. URL: https://doi.org/10.1155/2021/9953314.

[22] Ayushi Sharma et al. "Machine Learning Approach for Detection of Diabetic Retinopathy with Improved Pre-Processing". In: *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. 2021, pp. 517–522. DOI: 10.1109/ICCCIS51004.2021.9397115.

[23] *sklearn.feature*$_s$*election.SequentialFeatureSelector* $---$ *scikit* $-$ *learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html. [Accessed 13-Apr-2023].

[24] *Sklearn.linear_model.logisticregression*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[25] *sklearn.metrics.accuracy*$_s$*core* $---$ *scikit* $-$ *learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html. [Accessed 23-Apr-2023].

[26] *sklearn.svm.LinearSVC — scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html. [Accessed 23-Apr-2023].

[27] *sklearn.svm.SVC — scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. [Accessed 23-Apr-2023].

[28] Great Learning Team. *A Complete understanding of LASSO Regression — mygreatlearning.com*. https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/. [Accessed 13-Apr-2023].

[29] Alex Teboul. *Diabetes health indicators dataset*. Nov. 2021. URL: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_health_indicators_BRFSS2015.csv.

A Appendix

Table 6: Other Features in Full Feature Set

Feature	Description
Cholesterol Check (CholCheck)	0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 year
Smoker	Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes
Stroke	(Ever told) you had a stroke. 0 = no 1 = yes
HeartDiseaseorAttack	coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes
Fruits	Consume Fruit 1 or more times per day 0 = no 1 = yes
Veggies	Consume Vegetables 1 or more times per day 0 = no 1 = yes
Heavy Alcohol Consumption	Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week) 0 = no
Healthcare coverage (AnyHealthcare)	Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no 1 = yes
Doctor accessibility (NoDocbcCost)	Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no 1 = yes
Physical Health (PhysHlth)	Incidents of physical injury during the most recent 30 days
Difficulty Walking (DiffWalk)	Do you have serious difficulty walking or climbing stairs? 0 = no 1 = yes
Age	Age categories ranging 1-13 (18 to 80 years of age)
Education	Categorical (ordinal) feature on a 1-6 scale
Income	Categorical (ordinal) feature on a 1-8 scale