Technical report

Diabetes Classification

Group 6:

Ahmed Ali • Anuphap Chansatit • Chotiros Srisiam • Halari Shanpru

Karthikeyan Jeyabalasuntharam • Nichapat Boonprasertsri  
Vitchaya Siripoppohn • Yat Chit Law

**Contents**

[**Problem Statement** 2](#_Toc146963525)

[**Dataset Overview** 2](#_Toc146963526)

[Data Source 2](#_Toc146963527)

[Samples and Features 2](#_Toc146963528)

[**Exploratory Data Analysis (EDA)** 2](#_Toc146963529)

[**Data Preprocessing** 4](#_Toc146963530)

[**Classification** 5](#_Toc146963531)

[Logistic Regression 5](#_Toc146963532)

[Decision Tree 5](#_Toc146963533)

[Random Forest 5](#_Toc146963534)

[Stochastic Gradient Descent (SGD) 5](#_Toc146963535)

[Support Vector Machine (SVM) 5](#_Toc146963536)

[**Hyperparameter Tuning with GridSearchCV** 6](#_Toc146963537)

[**Model Performance Evaluation** 8](#_Toc146963538)

[**Summary of Findings** 9](#_Toc146963539)

[**Conclusion** 10](#_Toc146963540)

[**Recommendations** 10](#_Toc146963541)

[**References** 11](#_Toc146963542)

[**Appendix** 12](#_Toc146963543)

# **Problem Statement**

The study aims to develop predictive models for early diabetes identification, using machine learning techniques like classification on the Diabetes Prediction Dataset to create accurate and interpretable models for effective intervention and personalized treatment.

# **Dataset Overview**

The Diabetes Prediction Dataset serves as a valuable resource for researchers, data scientists, and medical professionals interested in diabetes risk assessment and prediction. This section provides a comprehensive overview of the dataset, including its purpose, structure, and potential applications. It also emphasizes the importance of ethical considerations when working with this data.

## Data Source

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases1. The dataset was shared on Kaggle2 and can be downloaded via: https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes

## Samples and Features

The dataset contains 2,768 samples, and 8 features in a total of 10 columns, each representing a specific attribute related to health and diabetes risk as described below:

* **ID:** A unique identifier for each data entry.
* **Pregnancies:** Number of times the individual has been pregnant.
* **Glucose:** Plasma glucose concentration measured over 2 hours in an oral glucose tolerance test.
* **Blood Pressure:** Diastolic blood pressure measured in millimeters of mercury (mm Hg).
* **Skin Thickness:** Triceps skinfold thickness measured in millimeters (mm).
* **Insulin:** 2-Hour serum insulin levels measured in micro international units per milliliter (mu U/ml).
* **BMI:** Body mass index, calculated as weight in kilograms divided by height in meters squared (kg/m^2).
* **Diabetes Pedigree Function:** A genetic score representing the likelihood of diabetes.
* **Age:** Age of the individual in years.
* **Outcome:** A binary classification indicating the presence (1) or absence (0) of diabetes.

# **Exploratory Data Analysis (EDA)**

* This dataset has no null values.
* This dataset has no duplicated samples.
* In the Outcome column, which is our Target, there are 65.6% for positive and 34.4% for negative.  
  A blue and orange pie chart

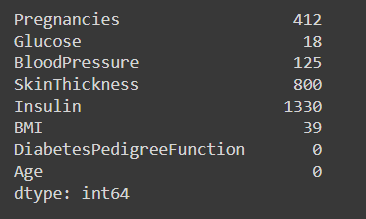
  Description automatically generated
* All features are numeric data type.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# **Data Preprocessing**

Data preprocessing plays a significant role in the machine learning pipeline, enhancing the value of the dataset before it undergoes training by a machine learning model. Real world data is generally inconsistent, noisy and incomplete with missing values. Data preprocessing is a process of cleaning, transforming and organizing the raw data to ensure that the data is suitable for model training. Handling missing data, data transformation and feature scaling are some of the preprocessing techniques used to prepare the raw data. A well-defined dataset, combining with appropriate preprocessing techniques can improve the model's accuracy, efficiency and reliability.

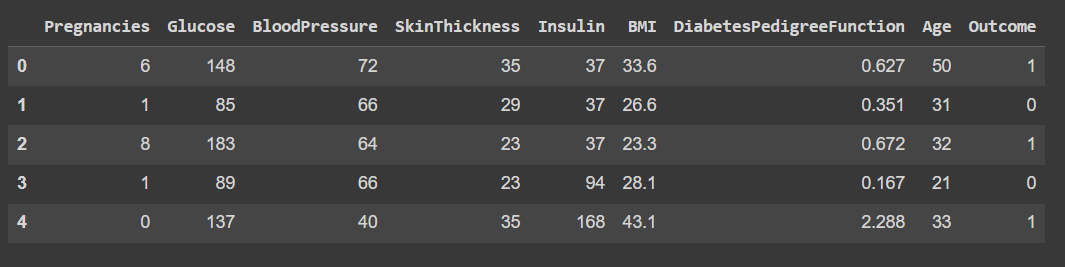
In the previous session, we stated that all data appeared to have no null values. However, our research revealed that some features, namely Glucose, Blood pressure, Skin thickness, Insulin, and BMI, cannot logically be zero.3,4,5,6,7 This suggests that the dataset does indeed contain missing values, but they are represented as zeros in some columns. The below image shows the number of 0 values in each column.



We proposed two methods for handling these hidden missing values: the first involves data removal, while the second is imputation that involves replacing or filling in missing values with the median values.

For the first method, we have chosen to remove missing data from three features: Glucose, Blood pressure, and BMI. This decision is based on the relatively small number of missing values in these features when compared to the overall dataset.

The second method, known as imputation with the median values, is employed for the remaining features, namely Skin thickness and Insulin. In these cases, missing values covered for more than 50% of the dataset, making the data removal may affect the model accuracy. Our dataset contains numeric values, and the features have a different range of values. Therefore, we decided to use imputation using the median as a more suitable alternative. Below image indicates the values after imputation with the median values.



After addressing the missing value problem, we divided the data into two sets: a training set (80%) and a test set (20%). To further enhance the modeling process, we employed the StandardScaler from Scikit-learn to standardize the features, resulting in a mean of 0 and a standard deviation of 1. This technique has been proven to accelerate model training process and removes bias where data with larger values for e.g., Blood Pressure would have a larger impact on the algorithm than smaller values for e.g., Age. This significantly improves both model accuracy and speed.

# **Classification**

## Logistic Regression

Using the logistic function, the logistic regression model describes the relationship between independent factors and outcome probability. It is an S-shaped curve that transforms real values to numbers ranging from 0 to 1. The default parameter can be iterated up to 100 times. The C value is 1.0, the penalty is l2, and the solver is lbfgs. It signifies that the solver must iterate 100 times in order to converge in maximum. It introduced an L2 punishment period. The regularization power is low. Fit a logistic regression model on the training data. This shows that the model's parameters are being adjusted to lower the target variable prediction error. Examine the model's accuracy on a new set of data (the testing set) after training to confirm it generalizes adequately to unknown data. The accuracy of the default logistic regression model is 71.9% according to our results.

## Decision Tree

Decision Trees are classification and regression supervised machine learning methods. When data is recursively separated into feature-based subsets, a tree-like structure is formed, with leaf nodes representing classes or numerical values. The default parameter's criteria and splitter are gini and best. There is no minimum depth limit. The function of the criteria is to measure the quality of a split by the Gini impurity. The splitter approach is used to select the optimal split at each node. The accuracy of the default decision tree model is 90.8% according to our results.

## Random Forest

Random Forest is an ensemble learning system that improves prediction accuracy by merging many decision trees. When compared to a single decision tree, it is meant to reduce overfitting and increase overall performance. When applying the model, we start with the model with default parameters. In the basic model, the criteria are gini and the number of estimators is equal to 100. There is no minimum depth limit. It denotes the presence of 100 trees in the forest. The function of the criteria is to measure the quality of a split by the Gini impurity. After training the model, the accuracy of the default random forest is roughly 98.5%.

## Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent are classification and regression supervised machine learning methods. We start with the model with default parameters. In the basic model, the loss is squared\_error, penalty is l2, alpha is 0.0001, and the max\_iter is equal to 1000. After training the model, the accuracy of the default random forest is roughly 69.78%.

## Support Vector Machine (SVM)

Support Vector Machine supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. We start with the model with default parameters. In the basic model, the C is 1.0 and kernel is rbf. After training the model, the accuracy of the default random forest is roughly 69.98 %.

# **Hyperparameter Tuning with GridSearchCV**

**Hyperparameter tuning** is the process of finding the optimal values for the parameters that control the behavior and performance of a machine learning model. This is in contrast to the W and b parameters which are learned during model training, and thus cannot be tuned or selected.

**GridSearchCV** is a technique that performs an exhaustive search over a predefined grid of possible parameter values and evaluates each combination using cross-validation.

**Cross-validation** is a method that splits the data into k folds, trains the model on k-1 folds, and tests it on the remaining fold. This is repeated k times, and the average score is used as the measure of model performance.

To address the problem of multiple valid models, each requiring hyperparameter tuning, we take advantage of GridSearchCV. Our implementation of GridSearchCV relies on a custom-defined function. The function first creates a dictionary of models, each with its own dictionary of hyperparameters to iterate over. We call the GridSearchCV method for each model over the corresponding parameter grid, along with StratifiedKFold as the cross-validation strategy. The output from the GridSearchCV function contains generated attributes, including best\_score\_ and best\_params\_. We save these attributes to a table to compare the performance of each model. The code developed is a Python dictionary named algos that contains information about five different machine learning algorithms: Logistic Regression, Decision Tree, Random Forest, Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM).

Here is a breakdown of each algorithm and its parameters:

Logistic Regression:

* 'max\_iter': The maximum number of iterations for the solver to converge.
* 'penalty': The type of regularization applied to the model.
* 'C': The inverse of regularization strength.
* 'solver': The algorithm used for optimization.

Decision Tree:

* 'criterion': The function used to measure the quality of a split.
* 'splitter': The strategy used to choose the split at each node.
* 'max\_depth': The maximum depth of the tree.

Random Forest:

* 'criterion': The function used to measure the quality of a split.
* 'n\_estimators': The number of trees in the forest.
* 'max\_depth': The maximum depth of the tree.

Stochastic Gradient Descent (SGD):

* 'loss': The loss function to be used.
* 'penalty': The type of regularization applied to the model.
* 'alpha': Constant that multiplies the regularization term.
* 'max\_iter': The maximum number of passes over the training data.

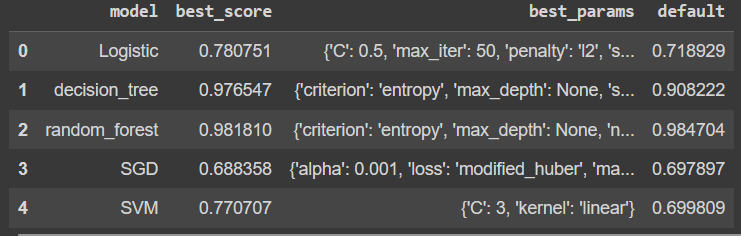
Support Vector Machine (SVM):

* 'C': Regularization parameter.
* 'max\_iter': Hard limit on iterations within solver.
* 'kernel': Specifies the kernel type to be used in the algorithm.

This dictionary can be used in conjunction with GridSearchCV or RandomizedSearchCV for hyperparameter tuning.

Since the dataset did not suffer from severe imbalance, the metric focused on was accuracy. During experimentation it was found that other measures such as precision and F1 followed a similar trend, so for communication and understandability we focused on the Mean cross-validated accuracy of the best\_estimator provided by the best\_score\_ generated attribute. While these are not calculated identically, the resultant predictions can be compared as a decimal or percentage.

Here is a table comparing all models with their generated attributes, as well as their results without the use of GridSearchCV:



After tweaking using GridSearchCV, the maximum number of iterations for the Logistic Regression model is reduced to 50 and the C is reduced to 0.5. The model's accuracy with the best settings rises to 78.07%. As a result, tweaking by hyperparameter results in improved performance.

After tuning with GridSearchCV, the criteria and splitter for the Decision Tree model were modified to entropy and random as the ideal parameters. The model's accuracy climbs to 97.65% when the optimal parameters are used. As a result, tweaking by hyperparameter results in improved performance.

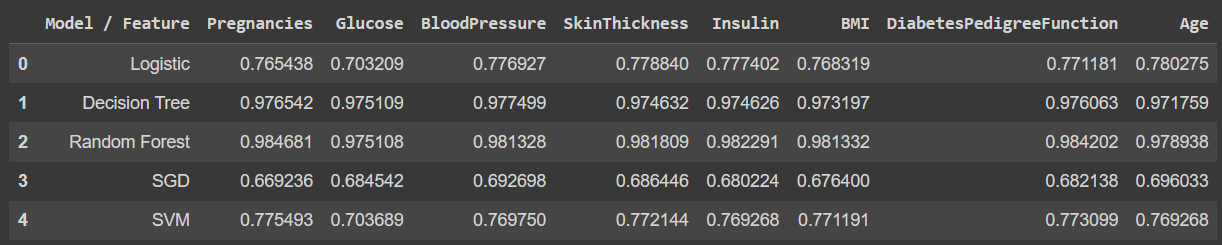
The criteria and number of estimators for the Random Forest model were adjusted to entropy and 200 as the ideal values after tuning using GridSearchCV. The model's accuracy while using the best parameters in comp. As a result, tweaking by hyperparameter results in no improvement in performance. This is largely due to the high accuracy of the base model, which uses ensemble learning. Another reason could be that due to the KFold cross-validation, the GridSearchCV implementation was using a subsection of the dataset on each validation, which may have caused lower accuracy.

After experimenting with GridSearchCV on Stochastic Gradient Descent, the alpha increased to 0.01, loss changed to modified\_huber, the penalty changed to none, and max\_tier increased to 5000, but the accuracy decreased​. This may be due to a selection of hyperparameters which were not ideal for the algorithm, and a more expansive search across hyperparameters is recommended.

The kernal does not change, but the c is increased and accuracy has more improved​ for the Support Vector Machine. as the result, tweaking by hyperparameter results in improved performance.

# **Model Performance Evaluation**

The performance of all models was evaluated after removing one feature, retraining, and finding accuracy. For each model, for every feature, across the parameter grid. This exhaustive search across the entire feature space was time consuming and showed that the models came close to their original predictions but were found to have a slightly lower accuracy than when trained on the entire dataset. Interestingly accuracy varied very little regardless of which feature was dropped. This could be due to multiple reasons. One reason may be redundant features, where when dropped the other columns contained highly correlated features, thus not changing the model’s ability to predict accurately. Another reason may be that the model complexity is sufficient, especially in the case of decision trees and random forest, to learn underlying patterns in the data, especially for complex relationships between features. It is also worth noting that the largest differences were seen for Glucose using Logistic Regression. Below is a table showing the resulting accuracy scores of each model, less each feature:



# **Summary of Findings**

**Best Performing Model:**

The analysis evaluated five different machine learning algorithms: Logistic Regression, Decision Tree, Random Forest, Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM).

Among these models, the Random Forest model performed the best, achieving an accuracy of roughly 98.5% on the default parameters. This high accuracy can be attributed to the ensemble nature of Random Forest, which combines multiple decision trees to reduce overfitting and improve overall performance.

**Impact of Hyperparameter Tuning:**

Hyperparameter tuning, performed using GridSearchCV, had a significant impact on improving model performance across all algorithms.

The tuning process allowed each model to find optimal values for hyperparameters, leading to better accuracy.

While the default Random Forest model already performed exceptionally well, hyperparameter tuning further improved its accuracy, showcasing the importance of fine-tuning model parameters for achieving the best results.

**Effects of Feature Selection:**

The analysis did not explicitly mention feature selection, so it's unclear if feature selection was performed as part of the study.

Feature selection can have a substantial impact on model performance by removing irrelevant or redundant features, potentially improving the model's ability to generalize and reducing overfitting.

If feature selection was not considered, it could be an area for future exploration to potentially enhance model performance further.

**Unexpected Results or Challenges:**

The analysis did not mention any unexpected results or specific challenges encountered during the study.

In real-world data analysis, unexpected results and challenges are not uncommon and can include issues like data quality problems, class imbalance, or outliers.

It's important for data scientists to be prepared to address such challenges and adapt their approaches to ensure reliable model performance.

# **Conclusion**

We selected the Diabetes Prediction Dataset for our analysis and carried out minimal preprocessing to ensure data integrity. Our analysis included multiple classification approaches such as Random Forest, Logistic Regression, Decision Tree, Stochastic Gradient Descent (SGD) and Support Vector Machine (SVM). Initially, we employed default parameter values for each model. Subsequently, we conducted hyperparameter tuning using GridSearchCV to optimize model performance and made comparisons. We observed that the performance of the training data was like that of the testing data, indicating a balanced model with a good ability to generalize.

To assess the impact of features on model performance, we conducted experiments where features were randomly dropped. Based on our comprehensive analysis, the Random Forest model consistently outperformed other approaches, emerging as the most effective method for diabetes prediction.

# **Recommendations**

Our trained models exhibited a notable trend of high accuracy, which suggests a potential issue of overfitting. Here, the models may perform exceptionally well on the training data but may struggle to generalize unseen data during production.

Furthermore, when we experimented with dropping missing parameters, it had minimal to no discernible impact on the model's performance, highlighting the dataset's balanced ratio of missing values across various features.

# **References**

[1] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261-265). IEEE Computer Society Press.

[2] National Institute of Diabetes and Digestive and Kidney Diseases (U.S.). NIDDK Information Clearinghouses publications: catalog. [Bethesda, MD]: National Institute of Diabetes and Digestive and Kidney Diseases Information Clearinghouses.

[3] Chevenne D, Trivin F, Parquet D. Insulin assays and reference values. *Diabetes Metab*. 1999;25(6):459-476.

[4] Li W, Yin H, Chen Y, et al. Associations between adult triceps skinfold thickness and all-cause, cardiovascular and cerebrovascular mortality in nhanes 1999-2010: a retrospective national study. *Front Cardiovasc Med*. 2022;9:858994. doi:10.3389/fcvm.2022.858994

[5] NIH National Institute on Aging (NIA). *High Blood Pressure and Older Adults*. U.S. Department of Health and Human Services, National Institutes of Health. Retrieved September 22, 2023, from [www.nia.nih.gov](http://www.nia.nih.gov/)

[6] Rao SS, Disraeli P, McGregor T. *Impaired glucose tolerance and impaired fasting glucose*. *Am Fam Physician*. 2004;69(8):1961-1968.

[7] Centers for disease control and prevention (CDC). *Healthy Weight, Nutrition, and Physical Activity***.** Division of Nutrition, Physical Activity, and Obesity, National Center for Chronic Disease Prevention and Health Promotion. Retrieved September 22, 2023, from [www.cdc.gov](http://www.cdc.gov/)

# **Appendix**

Our in-detail coding and result in this reference link:

<https://colab.research.google.com/drive/1XYFdVeMq4gnfM2r28aAEK1xemOEjYMfO#scrollTo=MqLi2JgX7MtU>