# Diabetes Classifier

Vittorio Haardt[a], Luca Porcelli[b]

[a]*Univerità degli Studi di Milano-Bicocca, 853268, v.haardt@campus.unimib.it,*
[b]*Univerità degli Studi di Milano-Bicocca, 853189, l.porcelli@campus.unimib.it,*

**Abstract**

In this study, held by KNIME and the University of Milano-Bicocca as part of a machine learning challenge, a diabetes dataset was analyzed using the KNIME platform to develop a predictive model for determining the risk of developing diabetes. The aim of the project was to leverage the power of machine learning algorithms to improve early diagnosis and reduce the negative impacts of this chronic disease. The results of the analysis revealed insights into the risk factors for developing diabetes and the model developed with KNIME had promising accuracy in predicting the risk. This study highlights the potential for machine learning in healthcare and the versatility of the KNIME platform in data analysis and model development. The findings of this project will serve as a useful tool for healthcare professionals in their efforts to mitigate the harm caused by this disease.

**Contents**



## 1. Introduction

Diabetes is a widespread chronic disease affecting millions of Americans, leading to reduced quality of life and life expectancy, as well as a significant financial burden on the economy. It's caused by the body's inability to regulate glucose levels in the blood and can result in serious complications such as heart disease, vision loss, lower-limb amputation, and kidney disease. While there is no cure, early diagnosis and lifestyle changes can improve outcomes. Predictive models for diabetes risk are crucial tools for public health officials and those at risk. As of 2018, 34.2 million Americans have diabetes, with 88 million having prediabetes. Many are unaware of their risk and the disease disproportionately affects lower socioeconomic groups. The cost of diabetes is estimated at $327 billion for diagnosed cases and $400 billion for undiagnosed and prediabetic cases.

The scope of this project is to analyze a diabetes dataset and develop a machine learning model to predict the risk of developing diabetes. The aim is to utilize the insights gained from the data analysis and the predictive power of machine learning algorithms to improve early diagnosis and ultimately help reduce the negative impacts of this debilitating disease.

## 2. Dataset and Data exploration

The used dataset consists of 18 attributes, 17 of which represent various health factors, and the final attribute, labeled as "Diabetes", is the target or class attribute that needs to be predicted. In other words, the 17 attributes serve as input features, and the goal is to use these features to accurately predict whether an individual has diabetes or not, which is represented by the class attribute.

- `age`: 3-level age category (_AGEG5YR see codebook) 1 = 18-24, 9 = 60-64, 13 = 80 or older
- `sex`: 0 = female, 1 = male
- `HighChol`: 0 = no high cholesterol, 1 = high cholesterol
- `CholCheck`: 0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years
- `BMI`: Body Mass Index
- `Smoker`: Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no, 1 = yes
- `HeartDiseaseorAttack`: Coronary heart disease (CHD) or myocardial infarction (MI) 0 = no, 1 = yes
- `PhysActivity`: Physical activity in past 30 days - not including job 0 = no, 1 = yes
- `Fruits`: Consume Fruit one or more times per day 0 = no, 1 = yes

- `Veggies`: Consume Vegetables one or more times per day 0 = no, 1 = yes
- `HyAlcoholConsump`: Adult male: more than 14 drinks per week. Adult female: more than 7 drinks per week. 0 = no, 1 = yes
- `GenHlth`: Would you say that in general your health is: (scale 1-5) 1 = excellent, 2 = very good, 3 = good, 4 = fair, 5 = poor
- `MentHlth`: Days of poor mental health scale 1-30 days
- `PhysHlth`: Physical illness or injury days in past 30 days scale 1-30
- `DiffWalk`: Do you have serious difficulty walking or climbing stairs? 0 = no, 1 = yes
- `Hypertension`: 0 = no hypertension, 1 = hypertension
- `Stroke`: 0 = no, 1 = yes
- `Diabetes`: 0 = no diabetes, 1 = diabetes (Target variable)

As can be seen from the description, most of the variables in the dataset are binary or Boolean variables. These variables represent the presence or absence of certain health factors or conditions. The exception to this is the age variable, which is ordinal in nature and the variable `GenHlth` is also ordinal. `BMI`, on the other hand, is the only continuous variable in the dataset.

The target or class attribute, labeled as `Diabetes`, is a binary variable, this indicates that the problem we are trying to solve is a binary classification problem.

The fact that the dataset is perfectly balanced between the two classes of the target variable is considered ideal for machine learning. This is because the model will have an equal number of examples from both classes to learn from, reducing the chance of biased learning. However, this assumption of a perfectly balanced dataset may cause problems if the training set used for evaluation is different from the original dataset. In this case, true priors of the total population are unknown, so it is assumed that the training set that will be used to evaluate the model will be equally distributed like the training set used for learning.

## 3. Preprocessing

The dataset was already very clear and well-prepared, suggesting that it has undergone a previous cleaning process, which has removed any errors and inconsistencies that may have been present in the raw data.

The preprocessing that was performed on the dataset was to check for missing values, outliers, near-zero variance, and highly correlated variables, and to apply feature selection to determine the best subset of variables to use. By performing these preprocessing steps, the dataset is made ready for modeling.

The check for missing values in the dataset revealed that all variables are complete and have no missing data in any of the records. Additionally, the check for near-zero variance and highly correlated variables showed that none of the variables had near-zero variance and none of them were highly correlated with one another (Figure 1). This means that none of the variables in the dataset is redundant.

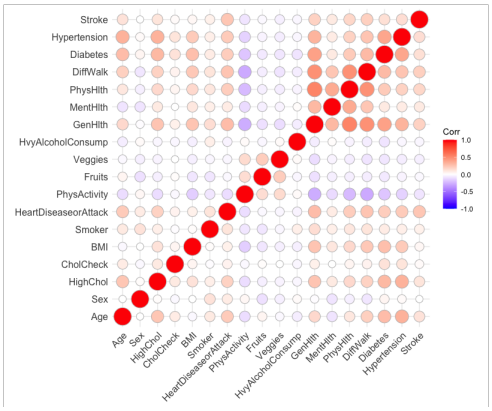


Figure 1: Correlation between attributes.

After this, the next step was to analyze the outliers present in the data. This was done by creating box plots for each attribute. The box plots provided a graphical representation of the distribution of the data and helped in identifying any outliers. It was observed that the variable *BMI* had many outliers. This information was crucial in understanding the distribution of the data and the potential impact of these outliers on models performance. In particular, it was observed that some values of BMI were over 80, which seemed unusual given the definition of BMI as:

$$BMI = \frac{weight}{height^2}$$

To understand this strange phenomenon, we consulted with some domain experts. The experts found these values to be highly unusual and suggested removing values with a BMI greater than 50 and converting the type of the variable. Following the expert advice, the observation with a value of $BMI < 15$ or a value of $BMI > 50$ was removed, and the variable was transformed into an ordinal one based to the following rules:

- $BMI \leqslant 16 \rightarrow$ "INANITION"
- $16 < BMI \leqslant 17.50 \rightarrow$ "UNDERWEIGHT"
- $17.50 < BMI \leqslant 18.50 \rightarrow$ "SLIGHTLY UNDERWEIGHT"
- $18.50 < BMI \leqslant 25 \rightarrow$ "NORMAL"
- $25 < BMI \leqslant 30 \rightarrow$ "OVERWEIGHT"
- $30 < BMI \leqslant 35 \rightarrow$ "CLASS I OBESE"
- $35 < BMI \leqslant 40 \rightarrow$ "CLASS II OBESE"
- $BMI > 40 \rightarrow$ "CLASS III OBESE"

The transformation of the BMI attribute was suggested because it is an imbalanced index and doesn't provide much information (in medical terms). It has been known to wrongly identify subjects who are very short or tall, or those who are muscular. In recent times, new calculations of BMI, like the "new BMI", are preferred in the medical field. By transforming the BMI

attribute into an categorical one, more information can be obtained and the variability of the index is reduced. This provides a more informative and useful representation of the data.

The ordinal variable was transformed into numerical ones in order to preserve the ordinal information. The nature of ordinal variables is often lost when they are transformed into either nominal or numerical variables. Transforming them into nominal variables discards the ordinal information, while transforming them into numerical variables gives them a numeric nature that is not inherent to their intrinsic nature. However, for the purpose of classification, the numerical transformation approach is sufficient and will provide a viable solution. A more in-depth and accurate approach would involve the partial order set theory, but, as just said, for the current purpose, the numerical transformation suffices.

It is now time to search for the optimal subset of attribute. In order to find it, a feature selection with wrapper method was applied to the dataset. In particular was used the *Boruta* method that use random forest algorithm. The method works by creating "shadow attributes", which are random copies of the original features, and then comparing the importance of the original features to their corresponding shadow attributes. If a feature is found to be less important than its corresponding shadow attribute, it is removed from the dataset. This process is repeated until all features have been evaluated. The final subset of features is considered to be the optimal set of variables for modeling.

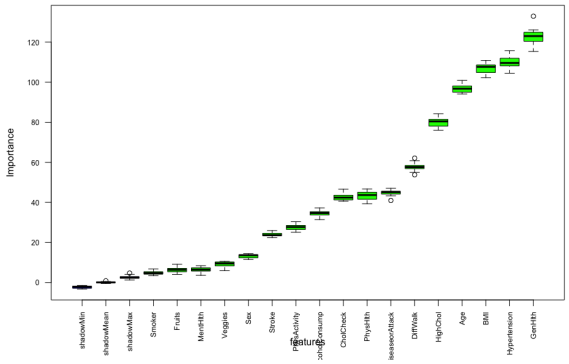


Figure 2: Attributes' importance.

As a result of the feature selection process using the Boruta method, no attributes were found to be less important than the "shadow attributes" (Figure 2). This means that the feature selection process did not identify an optimal subset of attributes, and all of the attributes was considered important and so they will be used for the classification task.

## 4. Performance Measure and Models

### 4.1. Logloss

The evaluation of the models built for the challenge will be conducted using a separate test set and the *logloss* metric. Logloss, also known as *cross-entropy loss*, is a widely used performance metric in machine learning for binary classification problems.

The logloss metric measures the accuracy of a classifier's predicted probabilities by calculating the likelihood of these predictions being correct. In other words, it evaluates how well the predicted probabilities match the actual class labels. A lower values indicating better performance.
The logloss function is give by:

$$LogLoss_i = -[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

The average of the loss function is given by:

$$LogLoss = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

In this sense, the best parameters for each trained model will be determined, and the best model will be selected based on the *logloss* metric. Although logloss is used as the primary metric in evaluating models, other metrics such as *accuracy* and the *AUC* (area under the ROC curve) are also used to provide a more comprehensive view of the binary classification problem.

The calculation of logloss for each individual instance can result in a problematic situation when the predicted probabilities are either 0 or 1. This is because of the logarithmic function in the logloss formula, which results in an infinite value when taking the logarithm of 0. To address this issue, a small positive value is selected, close to 0 but still within the range of the system's handling capabilities, and is used in place of 0. This ensures that the logloss remains finite, avoiding any potential system errors. This approach is commonly used to deal with the issue of having logloss values that are not computationally manageable.

The substitute value chosen for 0 is $10^{-15}$, and so, $1 - 10^{-15}$ is used for 1. This substitution results in a logloss of more than 34, which is a relatively high value. It is a manageable and acceptable value for the calculation, ensuring that the logloss remains finite and avoiding any potential system errors, but impacting the overall value still.

### 4.2. Models

In this section, we will briefly present the various classification techniques that were compared to determine the best model. The models that were evaluated and their tuning parameters will be discussed, with the goal of finding the best model. However, the detailed analysis of the performance and the comparison of the models will be discussed in a later section. This section serves as an introduction to the different classification techniques that were considered and the optimization process that was carried out to obtain the best models. The models will be presented in terms of the techniques used, their properties, and the tuning parameters that were adjusted to optimize their performance and how they were searched.

In order to find the best parameters for each of the models, different search strategies were employed. The parameters for some models were searched using a grid search (brute force approach), where every possible combination of parameters was

tested. For other models, a random search was preferred, where a specified number of random combinations of parameters were tested. The number of iterations was determined based on the computational requirements of the models.

***Decision Tree***. A C4.5 tree was used and the "minimum number of records per node" parameter was tuned using a grid search. Despite being highly unstable, decision trees can still bring many benefits such as revealing highly correlated variables with the target, and they are easily readable, allowing for understanding of the decisions made by the model.

***Naive Bayes***. Naive Bayes does not require parameter tuning as the model makes the assumption that all features are independent from each other. Despite its simplicity, Naive Bayes can still provide fast and accurate prediction, it is also highly scalable, allowing for efficient training on large datasets. One of the main benefits of Naive Bayes is its interpretability, as it provides an understanding of the important features for the classification.

***Random Forest***. Random forest is a powerful machine learning model that is capable of overcoming the limitations of individual decision trees and bagging trees. The parameters for this model were tuned using a random search of 1000 iterations, starting from a grid of possible values for "number of models" [100, 500], "maximum depth" [1, 30], and "minimum child size" [1, 31]. Additionally, the fraction of attributes that are used in each tree, which is $\sqrt{n\_attribute}$ by default, was also searched with values ranging from half, three halves, and double the default value. This model usually turn out to be a valid classifier.

***Gradient Boosting***. A gradient boosting model was selected for the binary classification problem because of it typically perform well. The parameters of the model were tuned using a random search with 1000 iterations, starting from a grid of possible values for the "number of models" [50, 150] , "learning rate" [0.05, 2], "maximum depth" [1, 10], "minimum child size" [50, 200], and "data fraction" [0.1, 1]. The gradient boosting model was selected because it is commonly considered to be the best model for this type of problem, due to its ability to effectively model complex relationships between the features and the target.

***XGBoost***. Even though the XGBoost model is built to perform optimally with numerical variables, it has been used and performed well in similar problems, so we decided to test it as well. The speed of this model is compromised by the type of input variables, so fewer iterations were used for the random search, around 100, in hopes of finding a local minimum. The random search started form a grid of "learning rate" [0.03, 2], "minimum child size" [1, 11]. "data fraction" [0.6,1], "gamma" [0.5, 2.5], "column sample by tree" [0.6, 1], "maximum depth" [3, 5]. The XGBoost model was selected because it is widely recognized as one of the top machine learning algorithms for solving problems in various domains.

***Staked Model***. The stacked model used for the binary classification problem is a type of blending, where multiple models are combined to improve the performance of the final model. The inspiration for using a stacked model came from the Netflix challenge, where it was shown that combining multiple models could lead to better results. However, this approach also poses a high risk of overfitting, so measures were taken to minimize this risk. The stacked model consisted of the XGBoost, the Decision Tree and the Gradient Boosting, with the best parameters. Each of this models trained a sample of the test set and tested on a validation set (from the holdout) to generate predictions, which were then used to train a meta-classifier. The selected meta-classifier is a logistic regression. The validation set was then entered into each of the inducers to compute probabilities, which were then tested in the meta-classifier. This approach allowed for the creation of a robust and accurate model that effectively leverages the strengths of multiple models.

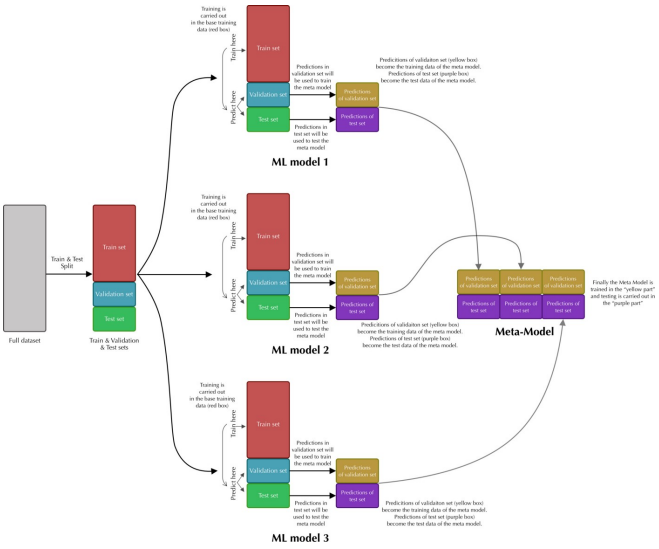


Figure 3: Stacked model schema.

## 5. Workflow

The research was focused on finding the most effective model for our binary classification, and an ad hoc work flow was designed to achieve this goal. The first step in the process was to divide the available dataset into two portions, with 70% being used for training the models and the remaining 30% being set aside for testing.

Once the data was divided, the different models were trained on the training data and their performance was validated using 10-fold cross-validation. This helped to determine the best parameters for each model. The tuning parameter search was performed by minimizing the logloss, as it was considered to be a key metric in evaluating model performance.

The models with the best parameters were then trained on the entire training set and tested on the test set in order to identify the best one. In addition to logloss, other performance metrics

were also considered in the final evaluation phase. These included the area under the ROC curve and accuracy, which provided a more comprehensive view of the model performance. The final selection was based on a combination of these metrics, taking into account the overall performance of the models. The goal was to find the model that performed the best on all metrics (logloss was considered the most important one anyway), providing the most effective solution for the task at hand.

## 6. Results

The performance of the models on the test set is the most critical aspect of the project. In general, most classifiers perform well on the training set as they have already learned the patterns and structures present in the data. However, the real evaluation of the model has to be done on data that it has not seen before, i.e., the test set. The evaluation of the model on the test set will provide a more accurate representation of its ability to generalize to new data. Hence, it is crucial to evaluate the different models on the test set, looking for the best one. Performing tests on the classification metric on both the train and test sets can provide valuable insights into the performance of the model. However, it is crucial to follow the best practice of using the metric achieved on the training set only to set the optimal parameters. Then, evaluating the performance on the test set will help select the model with the best performance. This approach helps to prevent overfitting on the test set and achieve a higher level of generalizability on the data that the model has not seen before. When testing the classification metric on the train and test sets, it is essential to keep in mind that the model may perform well on the training set due to its familiarity with the data. However, this does not necessarily mean that it will perform well on unseen data. Therefore, looking at the metric achieved on the test set is crucial in selecting the best model. In the evaluation part of the process, the values achieved on the test set will provide a more accurate representation of the model's performance. It is vital to ensure that the model has not overfit on the test set, which can lead to a false sense of performance.

| Model | Logloss |
|---|---|
| Decision Tree | 0.542 |
| Naive Bayes | 0.792 |
| Random Forest | 0.548 |
| Gradient Boosting | 0.511 |
| XGBoost | 0.513 |

Table 1: Table to test captions and labels.

Table 1 provides information on the performance of the different classifiers in terms of log loss. From the table, we can see that XGBoost and Gradient Boosting have the best performances in log loss. By performing a statistical test we can see that even tho Gradient Boosting has better performance on the test set it is not enough to achieve statistically significant better performance. Because of this, it is essential to note that the evaluation of the models should not be based on a single metric, but a combination of metrics. In this case, as said before, it is also crucial to consider the Receiver Operating Characteristic (ROC) curve. The ROC curve provides a visual representation of the trade-off between TPR and FPR for different classification thresholds. It shows how well the classifier can separate the positive and negative classes. A perfect classifier will have an ROC curve that goes straight up the left-hand side and then straight across the top. The area under the curve (AUC) is a measure of how well the classifier is able to separate the classes. Looking at the ROC curve, we can see that all the classifiers had

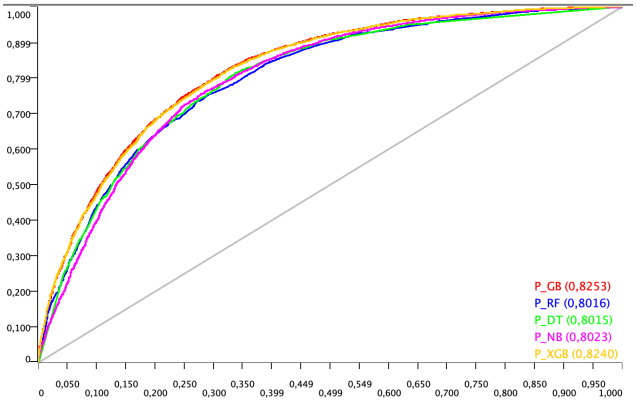


Figure 4: ROC curve of various models.

good performances, but the XGBoost and the Gradient Boosting outperform all the model. The two ROC curve intersect in some points, this implies that the classifiers have similar ability to separate the positive and negative classes, and there is no big difference in their performances. This is confirmed also by the AUC value in the figure where Gradient Boosting is better only for few points.

Figure 5 provides a graphical representation of the ROC curves for the different models. From the figure, we can see that the ROC curves for the two models intersect. The ROC curve for the best model should be up all the other for all the time. However, in the case of the models in Figure 5, we can see that

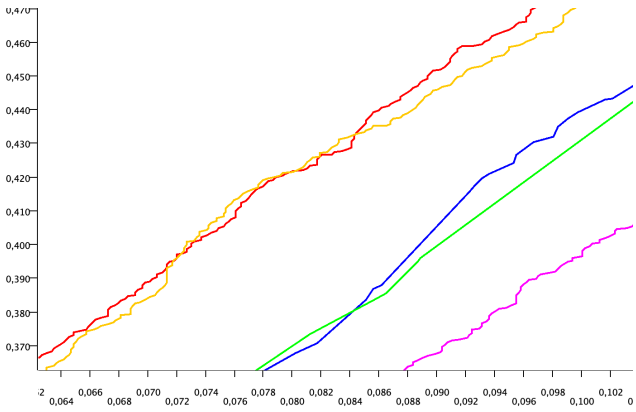

Figure 5: ROC curves intersecting.

there is no optimal ROC curve for the entire interval. This implies that the models have different strengths and weaknesses, and there is no single model that is optimal for all scenarios.

Note that however, we tend to prefer the Gradient Boosting because it has a better logloss, a better AUC and his ROC is up for most of the interval, although the differences are small.

Now that we have evaluated the different classifiers and identified the XGBoost and Gradient Boosting as the two best models, it is time to introduce the stacked model. The stacked model, combines the outputs of multiple base models to improve performance. The idea behind it is to leverage the strengths of the individual models and minimize their weaknesses, leading to an overall improvement in performance. In this case, we can combine the strengths of XGBoost and Gradient Boosting to create a stacked model. As said in the previous section the stacked model would use the outputs of the individual models as input features and train a new model on top of these features. This new model would learn to combine the strengths of the individual models and provide a better overall performance.

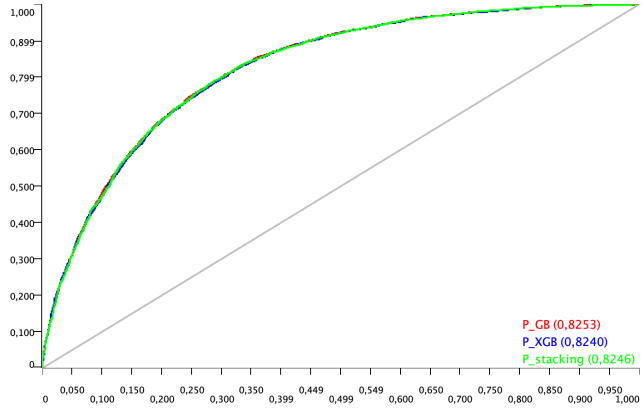


Figure 6: ROC curve for the two best models and the staked model.

After creating the stacked model by combining the two best models, we evaluated its performance using the same evaluation metrics, namely logloss, ROC curve, and AUC. Looking at Figure 6, which shows the ROC curves for the stacked model and the individual models, it is visible how the staked model doesn't seems having better performance than the Gradient Boosting. Furthermore, when we look at the AUC values for the different models, we see that the stacked model is only the second one with the highest AUC value. It seems that it take the straight from Gradient Boosting, but not from XGBoost, resulting in a disappointing middle way between the two approaches. Additionally we found that the staked model wasn't able to reduce the logloss of the individual models, resulting in a value of 0.515. For this reasons we can say that the staked model did not result in the aimed improvement in performance, so we prefer the individual models.

In conclusion, after evaluating different classifiers, we have found that XGBoost and Gradient Boosting perform the best in terms of accuracy (0.75), logloss, ROC curve, and AUC. While the stacked model combining these two models was an interesting approach, it did not result in the intended improvement in performance. Based on our findings, we recommend using Gradient Boosting as it outperforms XGBoost and the stacked model in all evaluation metrics.

## 7. Conclusion

During the course of the project, we conducted extensive testing and evaluation of various models to determine the most effective approach for tackling the binary classification problem of early diabetes diagnosis. Our primary goal was to identify the model that could accurately predict the presence or absence of diabetes in patients as early as possible.

To achieve this objective, we employed a meticulous and methodological approach, which involved carefully managing the data, selecting the most appropriate models, and carrying out a thorough evaluation of the chosen models to ensure accuracy and avoid any errors. We paid particular attention to the logloss metric and the overall performance of the classifiers we tested. Through this process, we identified the classifier that had exceptional performance in mimicking the logloss and exhibited the most promising results. The selected model demonstrated exceptional performance, particularly in terms of its ability to accurately predict the risk of developing diabetes. We evaluated its performance on a range of test data, and we are confident that the chosen model will perform well on unseen data. This means that our model can be relied upon to provide accurate and reliable predictions, which is essential for developing an effective diabetes prevention tool. Our team's ultimate

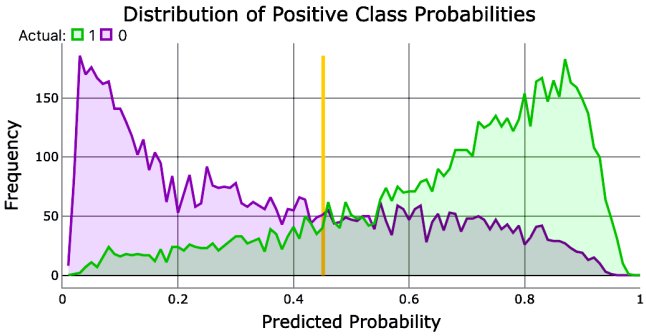


Figure 7: Optimal threshold for Gradient Boosting.

objective was to develop a web application that could help people assess their risk of developing diabetes. We achieved this goal by utilizing the chosen model to make predictions based on user-provided data. We set an optimal threshold for the diagnosis, which was determined to be 0.451, resulting in an accuracy of 0.75, as illustrated in Figure 7.

Our project has been instrumental in developing a reliable diabetes prevention tool that people can use to take proactive steps to manage their health. By providing an accurate diagnosis and recommending precautionary measures, our web application can help people take the necessary steps to reduce their risk of developing diabetes. Overall, we believe that our project has the potential to positively impact people's health and well-being, but there is room for further improvements. As with any research or development project, there is always the possibility of refining the model to increase its accuracy and re-

liability. To achieve this goal, we may consider exploring alternative modeling techniques or incorporating additional data sources to enhance the accuracy of our predictions. We could also conduct further testing and validation to ensure that the model's performance is consistent across different populations and datasets. Additionally, we could improve the user experience of our web application by implementing user feedback mechanisms, streamlining the data collection process, and providing personalized recommendations to users. This could help to increase user engagement and adoption of our tool. Overall, we are optimistic about the potential impact of our project on people's health and wellbeing, but we acknowledge that there is always room for improvement. We remain committed to exploring ways to enhance the accuracy, reliability, and usability of our tool to help people make informed decisions about their health.