# A REPORT OF A COMPARATIVE STUDY OF EFFECTIVE MACHINE LEARNING MODELS FOR DAIGNOSIS OF DIABETES

**UNIVERSITY OF ESSEX, COLCHESTER**

**SCHOOL OF COMPUTER SCIENCE AND ELECTRONIC ENGINEERING**

**YUNUS ALADE**

**WORD COUNT: 1049 WORDS**

**Abstract**

The purpose of this research is to identify machine learning techniques that are best suitable to make an early prediction of diabetes in patients and how much the average glucose level of patients exceeds the recommended threshold.

**Methodology**

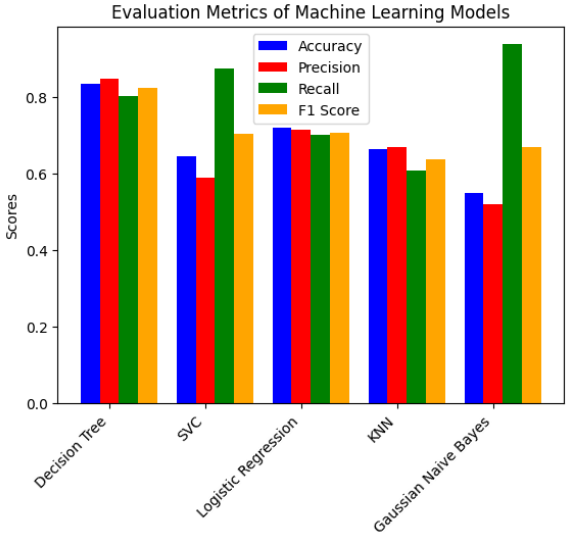
For this section will be divided into two parts, the first part will describe the methodology used in to tackle the process of early prediction of diabetes in patients while the other half will discuss the procedures used to determine how much patients exceed the recommended glucose level threshold.

2.1. **Predicting early risk of diabetes in patients**

**Exploring the data:**
Firstly, a historical data extracted from electronic medical records was provided by the by the client, these data came labeled and organized, except for some missing values in the column labeled "F20" of the data file tagged "CE802_P2_Data.csv". The model was trained on a Jupyter Notebook Workspace, using the "CE802_P2_Notebook.ipynb" file provided. The data file was mounted on google drive due to the size of the dataset. The provided dataset contained 1000 rows and 21 columns, with the 21st columns being the "class" columns.

**Splitting the data:**
Before splitting the data, we had to first fill in the missing values in "F20" with the K Nearest Neighbor Imputer function, this help to fill in the gap based on the values of the 5 nearest neighbors to the missing values. Now that we no longer have missing values, we then proceeded to split the data by first dropping the "class" column and then splitting the into Training (80%) and Test (20%).

**Training The Model(s):**
Per the recommendation of the client, the decision tree classifier was trained alongside four other models, namely, SVM, Logistic Regression, K Nearest Neighbor, and Gaussian Naive Bayes model. These models where chosen due to their ability to handle large data and give a binary classification based on the data that is fed into the model. In this case, the two classifications are labeled "True" and "False". The performance of each model was evaluated using Accuracy, Precision, Recall, and F1 score. A grouped bar chart was plotted to visualize the results of each model as shown in the diagram below:




**Findings:**
Based on the diagram above, we can see that the decision tree model, has the best accuracy score, although there were other models which had better recall such as SVC and Gaussian Naïve Bayes model, Decision Tree model had better accuracy, precision and f1 scores compared to the other models tested. We can tell from the diagram that the Gaussian Naïve Bayes model had the lowest accuracy, and the Decision Tree model had the best scores of the group, hence being the best model for the prediction.

**Prediction for test set:**
We have chosen Decision Tree as the best model for this class prediction, now we have to put our model to test using the data provided in the CE802_P2_Test.csv. Note that the original data file also has missing values on the "F20" column and this time the "class" column is empty. The task for this stage is to first fill in the missing data using the KNN Imputer as done on the "CE802_P2_Data.csv" file, then adding the Decision Tree model to the prediction snippet code provided in Part B of the "CE802_P2_Notebook.ipynb" file provided. The result of the prediction can be found in the "class" column of the "CE802_P2_Test_Predictions.csv" file.

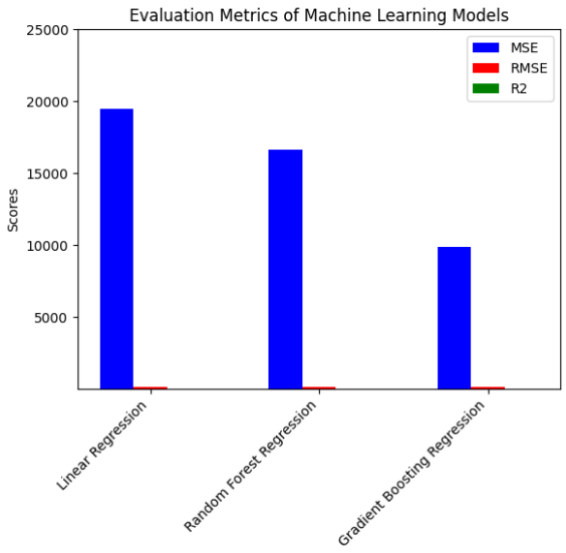
## 2.2. **Predicting patient's average blood glucose level**

**Exploring the data:**
A training set of historical data was provided by the client, it wan was mounted from google drive into the Jupyter Notebook file "CE802_P3_Notebook.ipynb" on Jupyter Notebook Workspace, unlike the data in the previous task, this one had no missing value, the challenge was that it had some values that were not encoded in column "F4" and "F9", considering that the machine learning models that will be used process only numerical inputs. To solve this, the ordinal encoder function was deployed to convert the NAN into numeric input. The provided dataset contained 1400 rows and 35 columns, with the $35^{ft}$ columns being the "Target" columns. To visualize the correlation between all the columns in the data frame, a correlation matrix was created using the "import seaborn as sns" function.

**Splitting the data:**
we split the data by first dropping the "Target" column and then splitting the into Training (80%) and Test (20%).

**Training The Model(s):**
Based on the recommendation of the client, the Linear Regression model was trained alongside two other models, namely, Random Forest Regression, and Gradient Boosting Regression Model.
The Random Forest Regression model was chosen due to its ability to handle complex data. Same goes for Gradient Boosting Regression Model, it can interpret complex relationship between features and the target variable.

**Findings:**
A group bar chart was plotted to visualize the relationship effectiveness of all the models, the image of the chart is included below:




Although it might not ne visually obvious, the Gradient Boosting Regression model had the lowest root mean squared error of 99.25, in this case it has performed better that Linear Regression and Random Forest Models, that scored 139.60, 128.9 respectively.

**Prediction for test set:**
We have chosen Gradient Boosting Regression model as the best model for to predict how far extent by which the patient's blood glucose level exceeds the diagnostic threshold if untreated. The

Gradient Boosting Regression model was fitted into the prediction code snippet that was provided in the "CE802_P3_Notebook.ipynb" file. After the test data was loaded, the code prompted an error that was preventing the model from processing the data, after series of investigation we found that some values in the data set were not numeric and need to be encoded, the label encoder was applied to "F4" and "F9", which solved the problem. The result of the prediction can be found on the "Target" column of the "CE802_P3_Test_Predictions.csv" file.

**Conclusion**

The Logistic Regression model proved to be very efficient in predicting the risk of diabetes in patents based on sufficient data provided on the other hand Gradient Boosting Regression perfumed better at predict how far extent by which the patient's blood glucose level exceeds the diagnostic threshold if untreated.

There is still a lot to explore in this field because the use of machine learning in the medical field is limitless if applied properly.