BRAC UNIVERSITY

Inspiring Excellence

CSE422: Artificial Intelligence

Project Report

Project Name: Diabetes Prediction

Group No: 14, CSE422 Lab Section: 01, Spring 2024	
ID	**Name**
21101255	Niloy Ahsan
21101191	Chaity Rani Ghosh

Table of Contents

Introduction

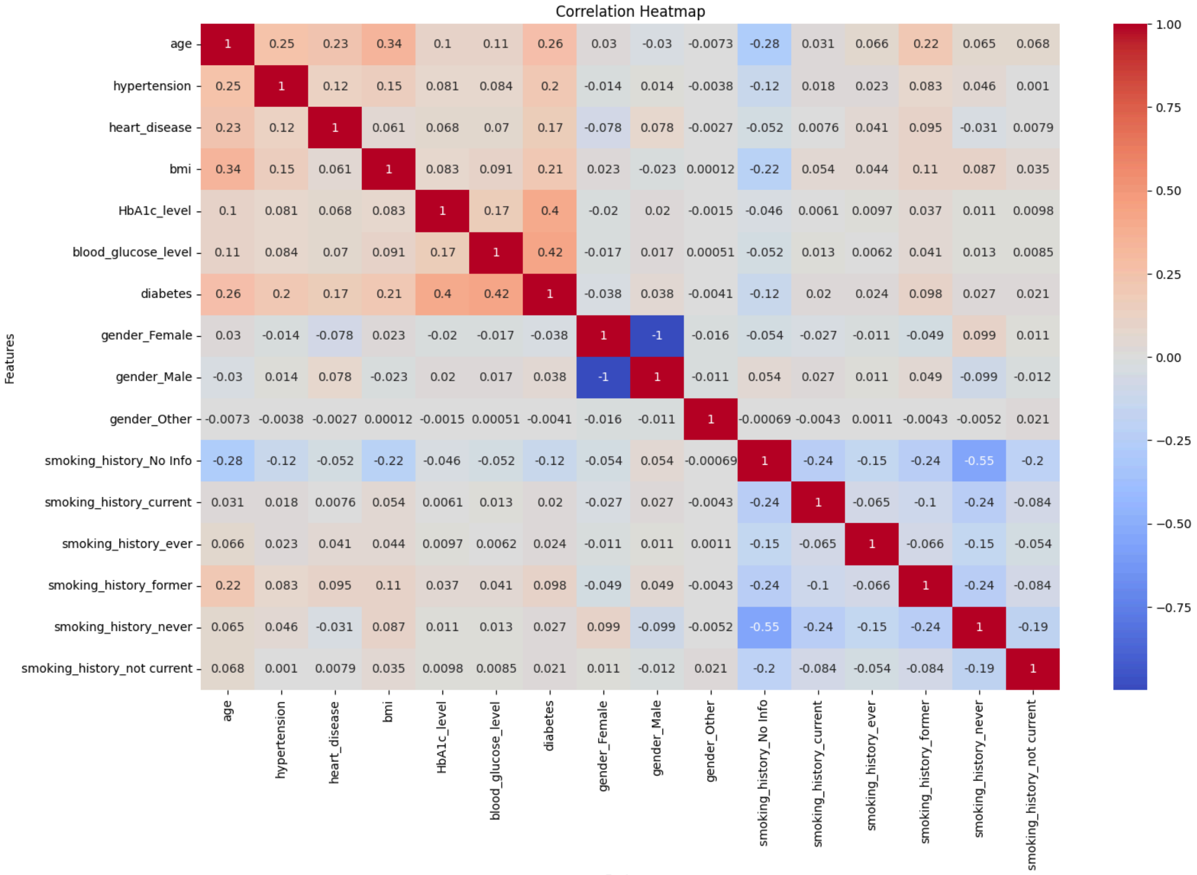
Diabetes is becoming a major public health concern around the world, necessitating the development of reliable prediction models to aid in early detection and intervention. The objective of our project is to create a prediction model for diabetes detection using Machine Learning methods. Diabetes is a chronic metabolic disease identified by high blood sugar levels which can lead to major complications such as cardiovascular disease, kidney failure, and blindness if it is not addressed earlier. Early detection of diabetes allows for appropriate intervention and lifestyle changes to prevent or postpone the onset of problems. Our project aims to tackle the growing problem of diabetes worldwide. Many people are diagnosed late which leads to serious health issues. We're using machine learning to create better tools for spotting diabetes early. This will help doctors make decisions more easily and improve patient outcomes. In this project, we'll use data about age, gender, BMI, and blood glucose levels to build a model that predicts the likelihood of diabetes. By analyzing this data with machine learning algorithms like KNN, SVM, Random Forest, and Decision Trees, we aim to improve the early detection and management of diabetes. This could lead to better patient outcomes and reduce the strain on healthcare systems.

Dataset Description

Source: Kaggle (https://www.kaggle.com/datasets)
Link: Diabetes prediction dataset

In the dataset, the total number of features is **15** and the total number of columns is **16**. This is a dataset with **100,000** observations related to diabetes diagnosis. The dataset consists of several attributes capturing various patient characteristics, including quantitative features such as Age, BMI, HbA1c_level, and Blood_glucose_level, represented by a wide range of numerical values. The dataset also contains categorical features like Gender, Hypertension, Heart_disease, and Smoking_history. These features are represented by discrete values such as "Male" or "Female" for gender, "0" or "1" for hypertension and heart_disease (indicating the absence or presence of the condition), and various categories such as "never", "No Info", "current", or "former" for smoking_history. For the target column, each entry in the dataset is labeled with binary values (0 or 1) representing the absence or presence of diabetes, respectively. This serves as the categorical target variable for classification tasks.

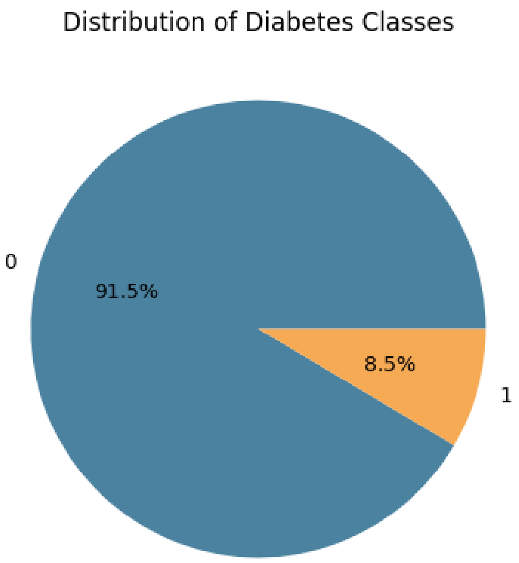
Correlation of the features along with the label



Here, the heatmap displays the correlation between all the features in the dataset, including both input features and the output label (diabetes or not diabetes). The correlation values range from -1 to 1, where a value closer to 1 indicates a strong positive correlation, a value closer to -1 indicates a strong negative correlation and a value around 0 indicates no correlation. This visual representation helps us understand the relationships between different features and the target variable.

Distribution of Unique Classes

In the dataset, label 0 (indicating "not diabetes") has 91,500 instances and label 1 (indicating "diabetes") has 8,500 instances. This indicates a significant class imbalance, where the majority class (label 0) has much larger instances than the minority class (label 1). To represent this imbalance we can use a pie chart:

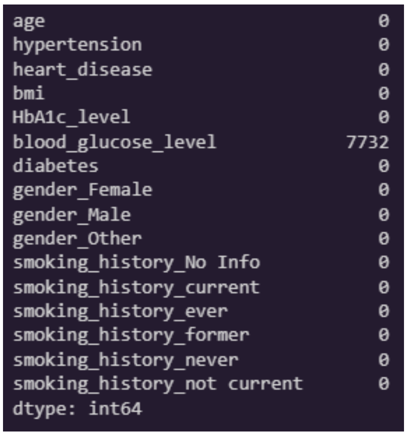


This pie chart visually represents the class distribution, highlighting the significant class imbalance in the dataset.

Data Pre-Processing

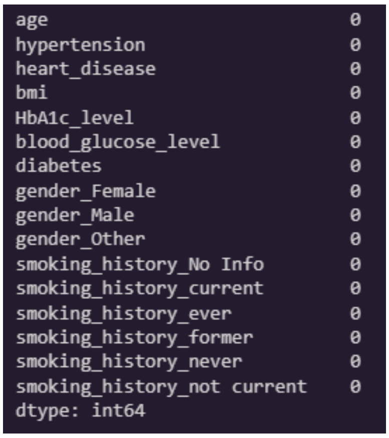
In the dataset, there were 7732 Null values in only one column named blood_glucose_level which is a huge number. Null values can disrupt the analysis and modeling process, leading to biased results or errors. So, we use the imputation technique to solve this problem. We calculated the **Median** value of that column which was 145.0 and then imputed it into the Null values.

Before Imputation:



```
age                              0
hypertension                     0
heart_disease                    0
bmi                              0
HbA1c_level                      0
blood_glucose_level           7732
diabetes                         0
gender_Female                    0
gender_Male                      0
gender_Other                     0
smoking_history_No Info          0
smoking_history_current          0
smoking_history_ever             0
smoking_history_former           0
smoking_history_never            0
smoking_history_not current      0
dtype: int64
```

After Imputation:

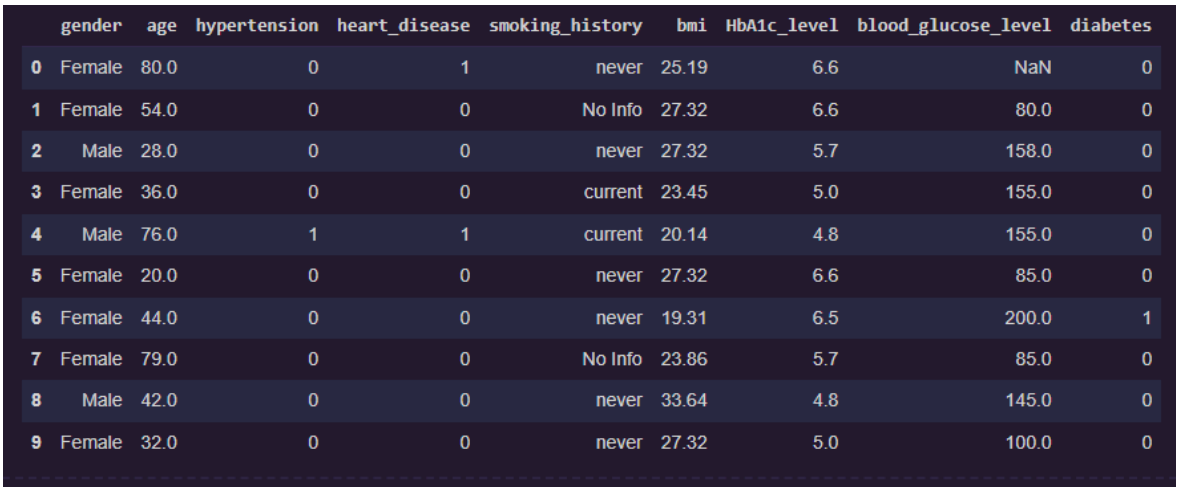


```
age                            0
hypertension                   0
heart_disease                  0
bmi                            0
HbA1c_level                    0
blood_glucose_level            0
diabetes                       0
gender_Female                  0
gender_Male                    0
gender_Other                   0
smoking_history_No Info        0
smoking_history_current        0
smoking_history_ever           0
smoking_history_former         0
smoking_history_never          0
smoking_history_not current    0
dtype: int64
```

Also, there were some categorical values in the dataset as mentioned before. The categorical values were present in the Gender, and Smoking_history column. So these columns have non-numeric values and we had to convert the categorical values into a numeric format because most of the Machine Learning algorithms are based on mathematical and statistical calculations which require numerical values for computation. Here we used One-Hot

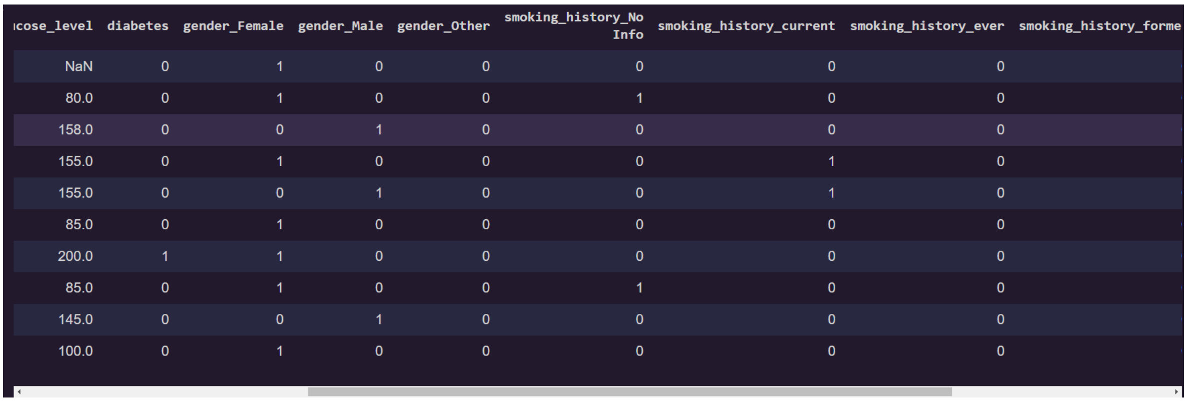
Encoding to convert the categorical data into numeric format. This ensures the accuracy of the model.

Before Encoding:



	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	NaN	0
1	Female	54.0	0	0	No Info	27.32	6.6	80.0	0
2	Male	28.0	0	0	never	27.32	5.7	158.0	0
3	Female	36.0	0	0	current	23.45	5.0	155.0	0
4	Male	76.0	1	1	current	20.14	4.8	155.0	0
5	Female	20.0	0	0	never	27.32	6.6	85.0	0
6	Female	44.0	0	0	never	19.31	6.5	200.0	1
7	Female	79.0	0	0	No Info	23.86	5.7	85.0	0
8	Male	42.0	0	0	never	33.64	4.8	145.0	0
9	Female	32.0	0	0	never	27.32	5.0	100.0	0

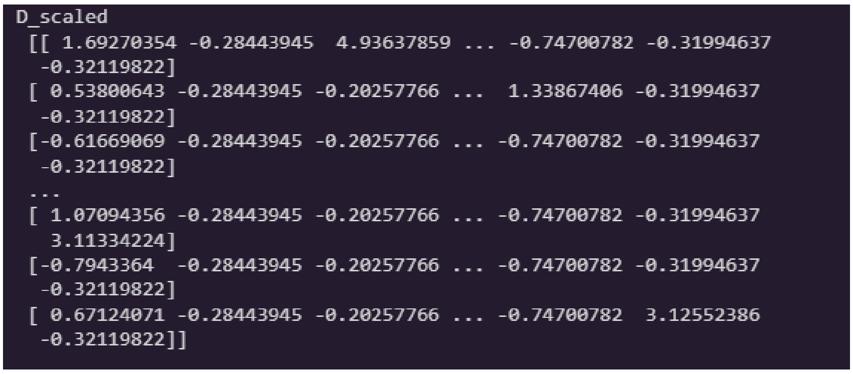
After Encoding:



ıcose_level	diabetes	gender_Female	gender_Male	gender_Other	smoking_history_No Info	smoking_history_current	smoking_history_ever	smoking_history_forme
NaN	0	1	0	0	0	0	0	
80.0	0	1	0	0	1	0	0	
158.0	0	0	1	0	0	0	0	
155.0	0	1	0	0	0	1	0	
155.0	0	0	1	0	0	1	0	
85.0	0	1	0	0	0	0	0	
200.0	1	1	0	0	0	0	0	
85.0	0	1	0	0	1	0	0	
145.0	0	0	1	0	0	0	0	
100.0	0	1	0	0	0	0	0	

Feature Scaling

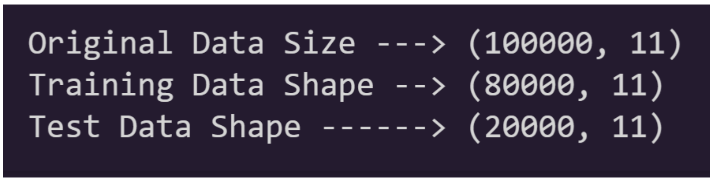
We have used the Standard Scaling technique to scale all the selected features to ensure equal importance of the features and stability. We tested the accuracy of some models before the Scaling and after Scaling. It performed better after Scaling the features.



```
D_scaled
 [[ 1.69270354 -0.28443945  4.93637859 ... -0.74700782 -0.31994637
  -0.32119822]
 [ 0.53800643 -0.28443945 -0.20257766 ...  1.33867406 -0.31994637
  -0.32119822]
 [-0.61669069 -0.28443945 -0.20257766 ... -0.74700782 -0.31994637
  -0.32119822]
 ...
 [ 1.07094356 -0.28443945 -0.20257766 ... -0.74700782 -0.31994637
   3.11334224]
 [-0.7943364  -0.28443945 -0.20257766 ... -0.74700782 -0.31994637
  -0.32119822]
 [ 0.67124071 -0.28443945 -0.20257766 ... -0.74700782  3.12552386
  -0.32119822]]
```

Dataset Splitting

In the dataset-splitting process, the dataset is split into two subsets. 80% of the data is allocated for training and the remaining 20% is allocated for testing. On this ratio, our Training Dataset is 80,000 and the Test Dataset is 20,000.

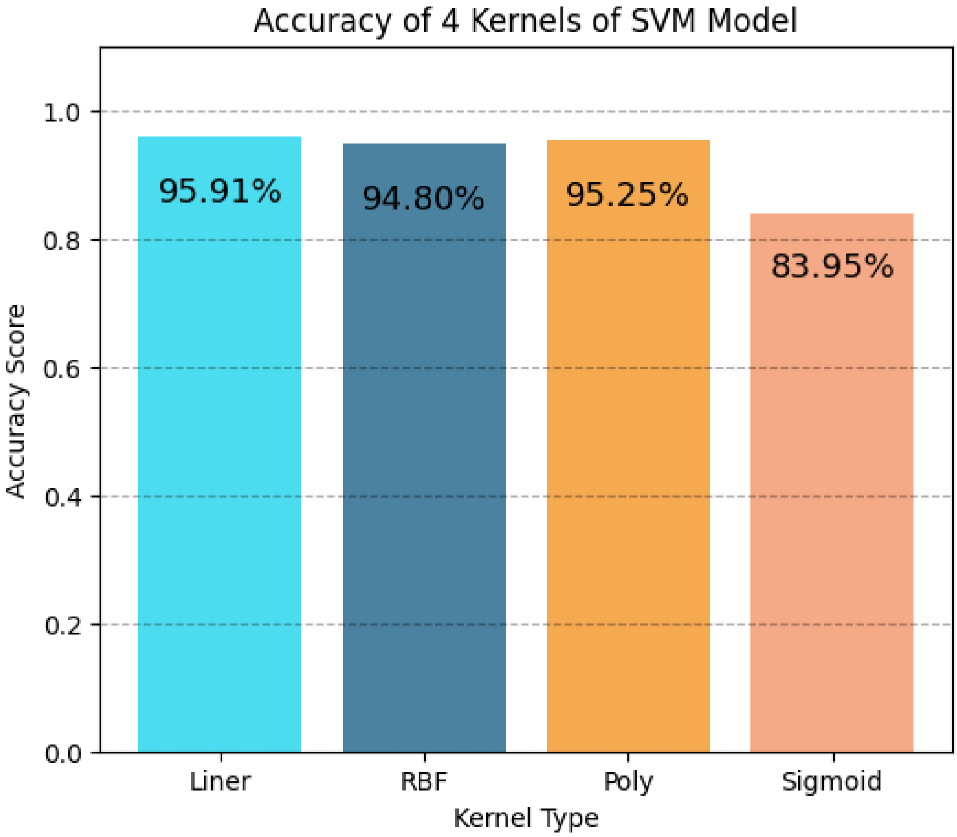


```
Original Data Size ---> (100000, 11)
Training Data Shape --> (80000, 11)
Test Data Shape ------> (20000, 11)
```

This 80:20 ratio will allow the model to learn from the majority of the data during training and will perform better during testing.

Model Training and Testing

We have used 4 ML Models to train our dataset: k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, and Random Forest Model. While assessing the SVM model, we experimented with various kernels including ***Linear, RBF, Poly,* and *Sigmoid*.**



Following this, we proceeded to select the kernel with the highest accuracy for further evaluation.

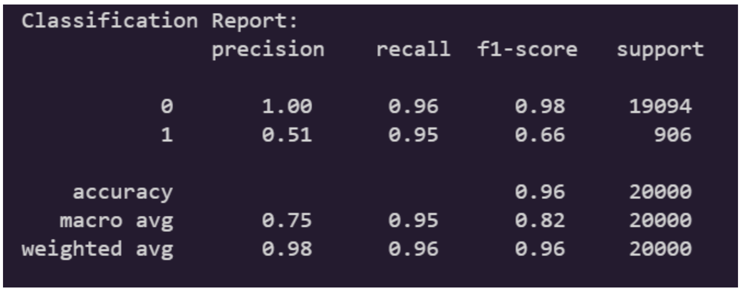
Model Selection/Comparison Analysis

We evaluated the accuracy scores for each model to measure their predictive performance. We did this in two cases, before and after Feature Scaling.

I. Before Feature Scaling:

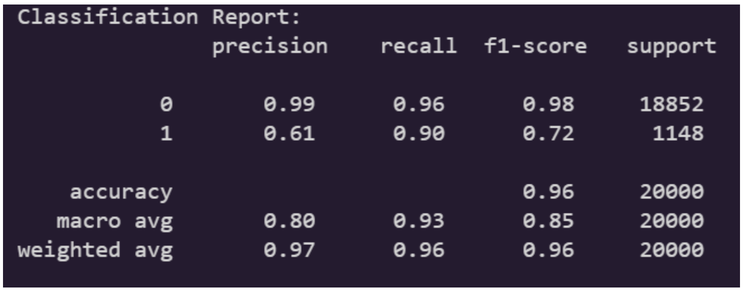
A. Classification Report:

1. k-Nearest Neighbors:



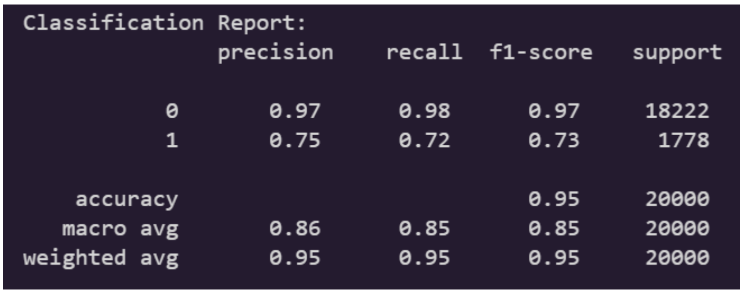
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     19094
           1       0.51      0.95      0.66       906

    accuracy                           0.96     20000
   macro avg       0.75      0.95      0.82     20000
weighted avg       0.98      0.96      0.96     20000
```

2. Support Vector Machine:



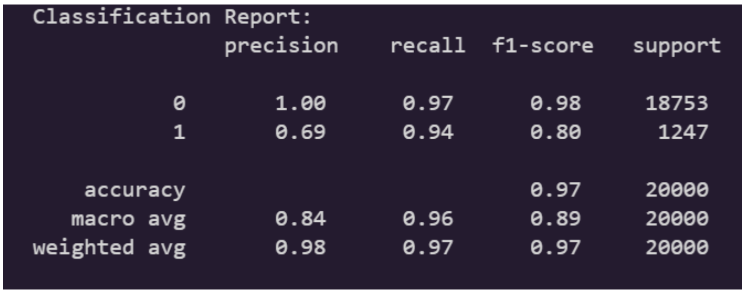
```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.96      0.98     18852
           1       0.61      0.90      0.72      1148

    accuracy                           0.96     20000
   macro avg       0.80      0.93      0.85     20000
weighted avg       0.97      0.96      0.96     20000
```

3. Decision Tree:



```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     18222
           1       0.75      0.72      0.73      1778

    accuracy                           0.95     20000
   macro avg       0.86      0.85      0.85     20000
weighted avg       0.95      0.95      0.95     20000
```

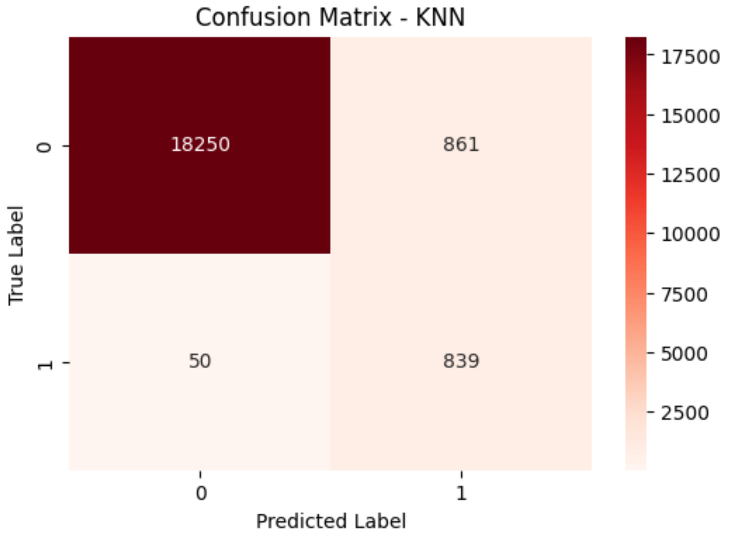
4. Random Forest:



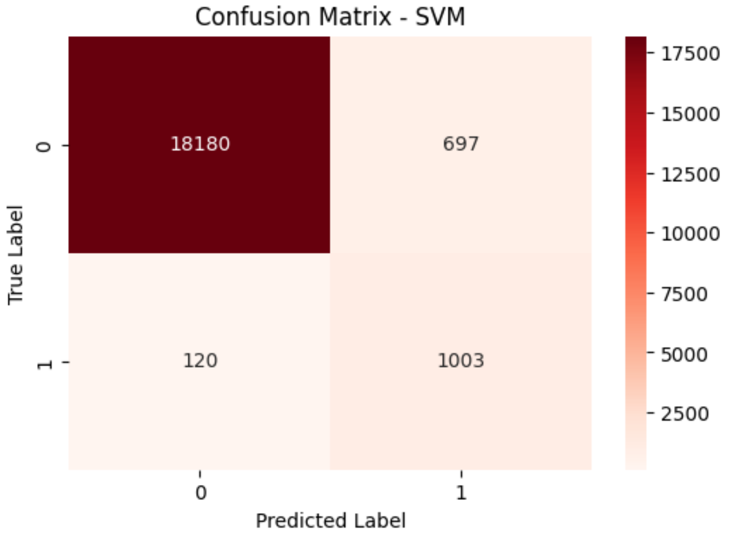
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     18753
           1       0.69      0.94      0.80      1247

    accuracy                           0.97     20000
   macro avg       0.84      0.96      0.89     20000
weighted avg       0.98      0.97      0.97     20000
```

B. Confusion Matrix:

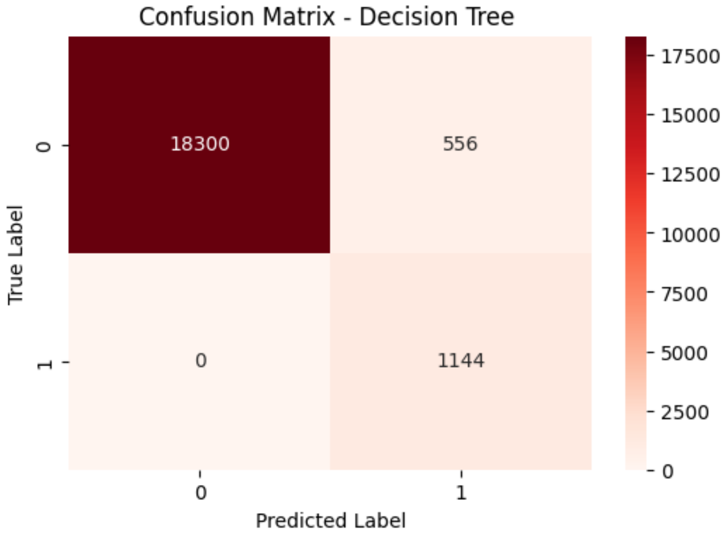
1. k-Nearest Neighbors:



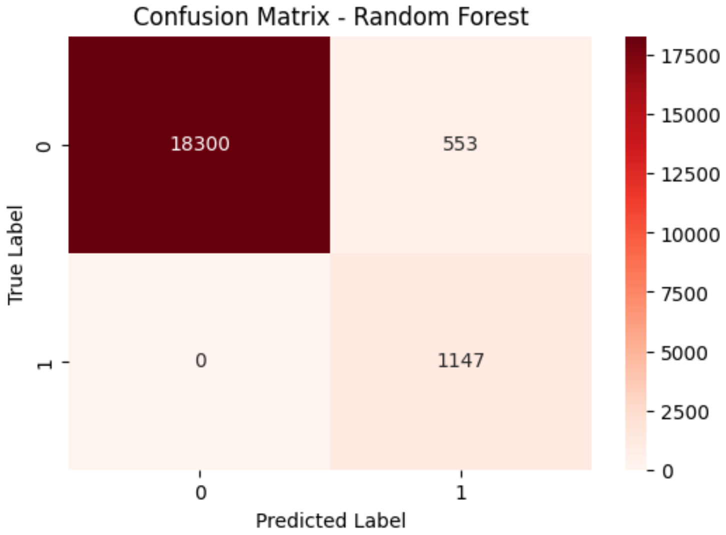
2. Support Vector Machine:



3. Decision Tree:

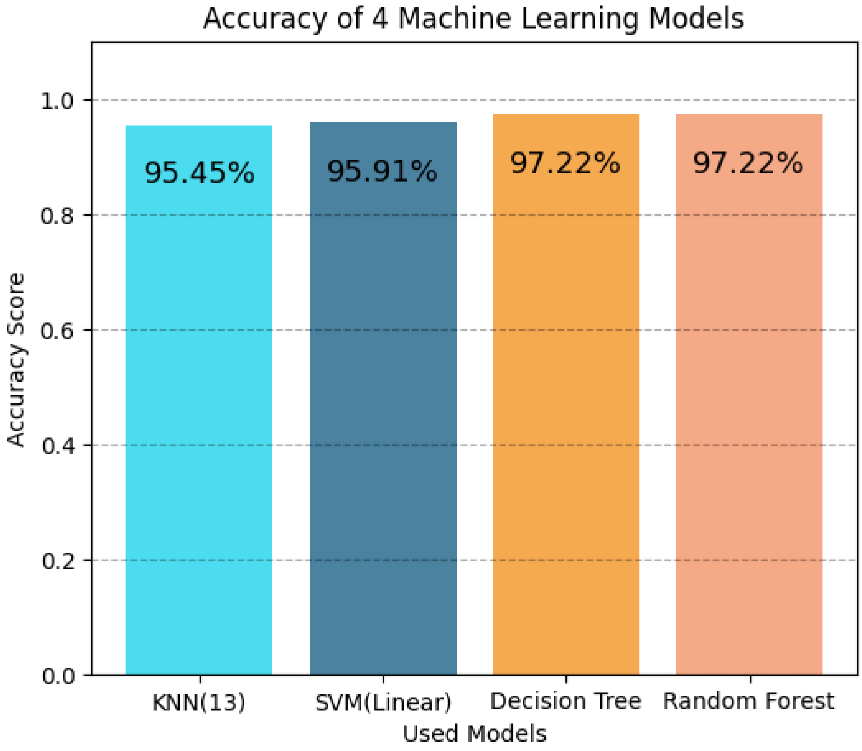


4. Random Forest:



C. Accuracy cores for each model:

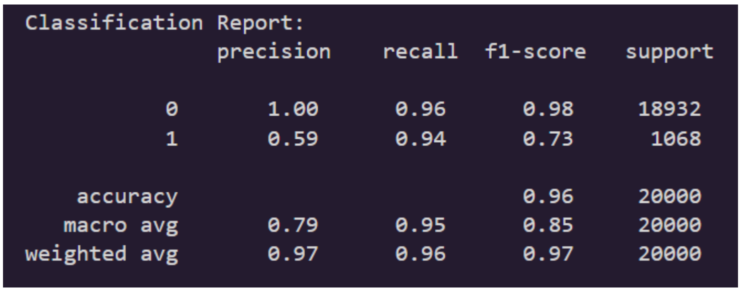
1. KNN: 95.45 %
2. SVM: 95.91 % *(with selected kernel: Linear)*
3. Decision Tree: 97.22 %
4. Random Forest: 97.22 %



II. After Feature Scaling:

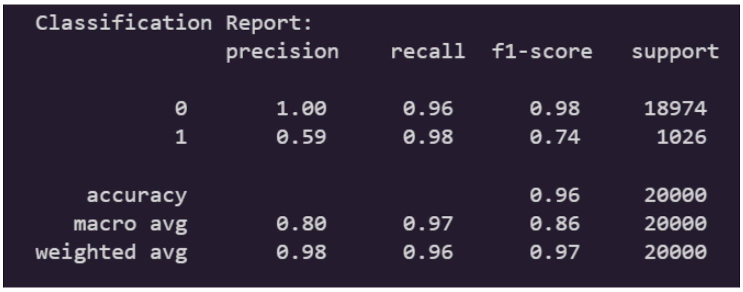
A. Classification Report:

1. KNN:



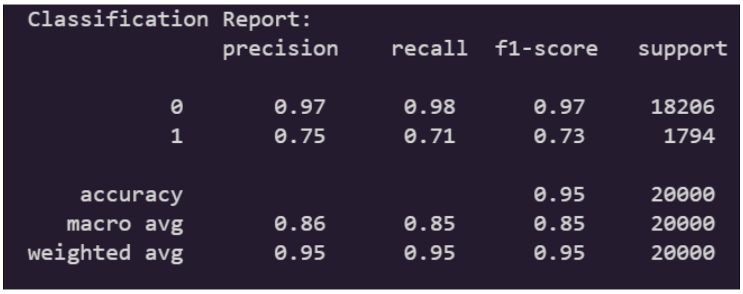
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     18932
           1       0.59      0.94      0.73      1068

    accuracy                           0.96     20000
   macro avg       0.79      0.95      0.85     20000
weighted avg       0.97      0.96      0.97     20000
```

2. SVM:



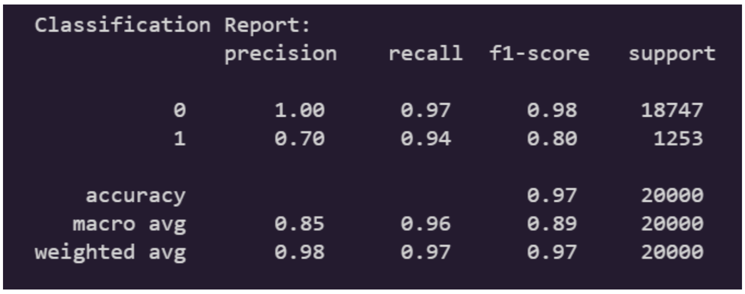
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     18974
           1       0.59      0.98      0.74      1026

    accuracy                           0.96     20000
   macro avg       0.80      0.97      0.86     20000
weighted avg       0.98      0.96      0.97     20000
```

3. Decision Tree:



```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     18206
           1       0.75      0.71      0.73      1794

    accuracy                           0.95     20000
   macro avg       0.86      0.85      0.85     20000
weighted avg       0.95      0.95      0.95     20000
```

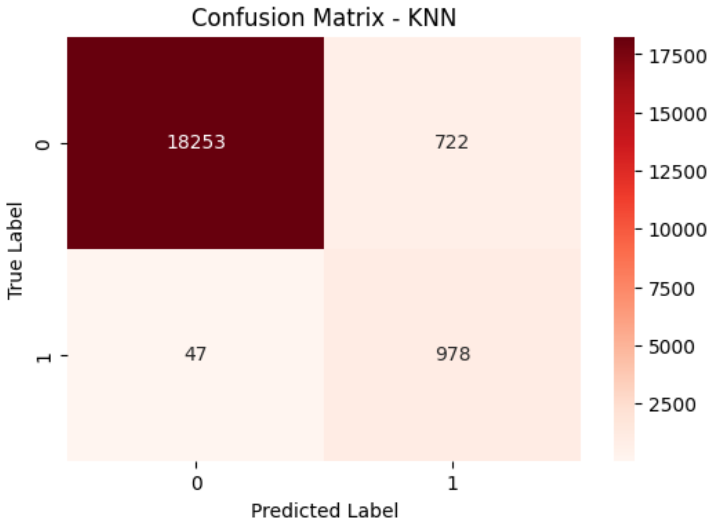
4. Random Forest:



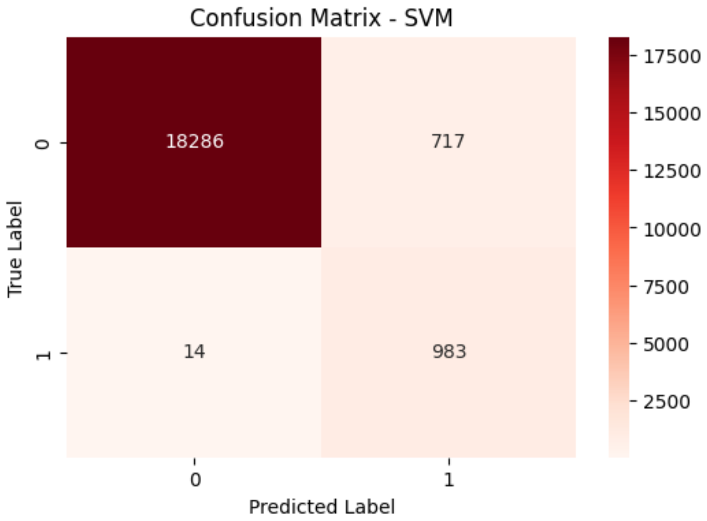
```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     18747
           1       0.70      0.94      0.80      1253

    accuracy                           0.97     20000
   macro avg       0.85      0.96      0.89     20000
weighted avg       0.98      0.97      0.97     20000
```

B. Confusion Matrix:

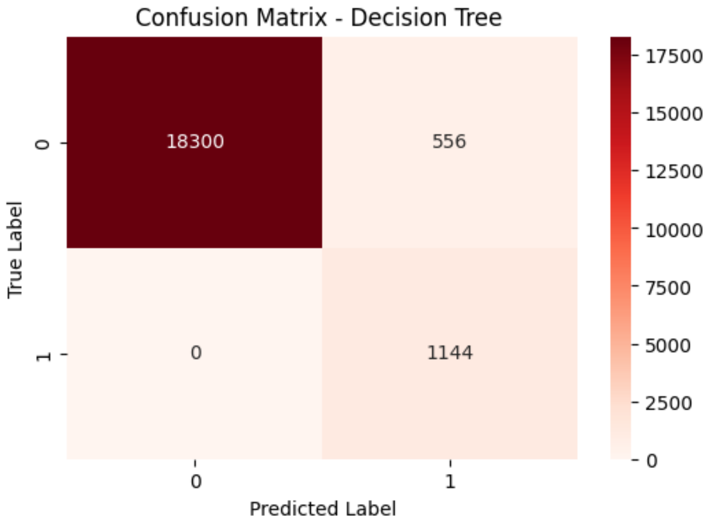
1. KNN:



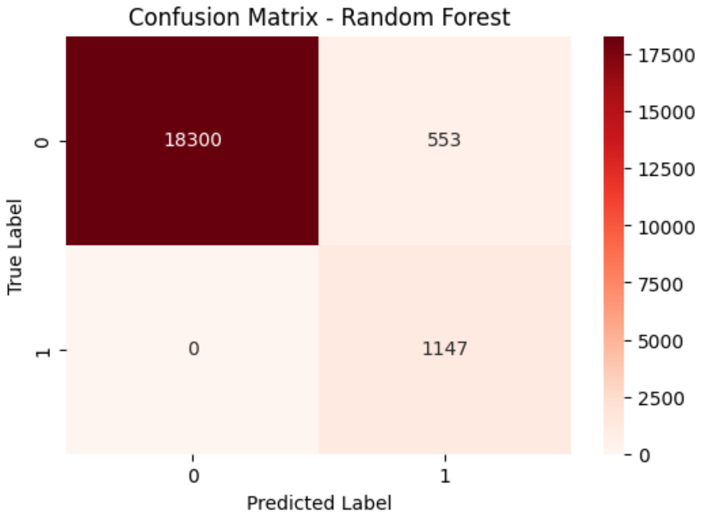
2. SVM:



3. Decision Tree:

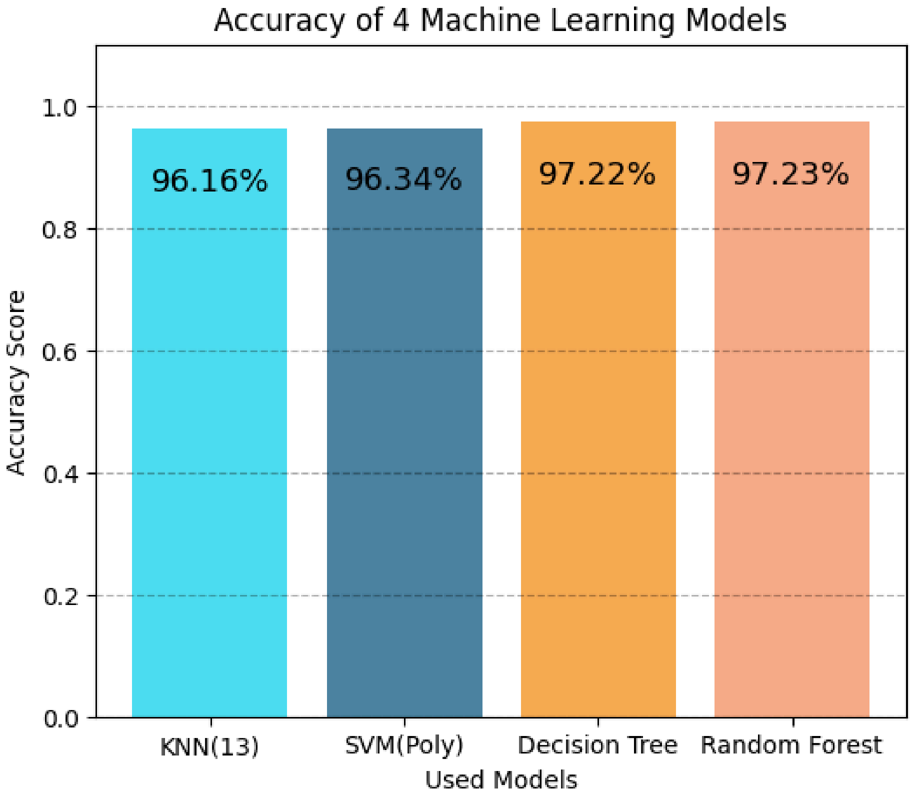


4. Random Forest:



C. Accuracy cores for each model:

1. KNN: 96.16 %
2. SVM: 96.34 % *(with selected kernel: Linear)*
3. Decision Tree: 97.22 %
4. Random Forest: 97.23 %



In both cases, the Decision Tree and Random Forest Models were more accurate and showed similar accuracy in predicting Diabetes among those 4 models.

Conclusion

In summary, the goal of our project amied to deploy machine learning techniques to formulate an efficient predictive model for diabetes identification. We implemented a variety of approaches to machine learning, which include k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, and Random Forest, to evaluate a dataset that included demographic and clinical information to create prediction models. We made sure the data was appropriate for training our models by carefully preparing it using techniques including resolving missing values, encoding category features, and scaling numerical characteristics.

After feature scaling, prediction performance improved significantly throughout model training and testing. With the best accuracy in both pre-scaled and scaled scenarios, the Decision Tree and Random Forest models have been demonstrated to be the most effective.

Furthermore, encouraging outcomes were also demonstrated by the SVM model with a poly kernel, especially following feature scaling.

Our findings from the results highlight how important machine learning is to enhancing diabetes early diagnosis and treatment, which improves patient outcomes and lessens the burden on healthcare systems. In the end, this project demonstrates the effectiveness of machine learning in tackling complex healthcare issues and emphasizes the critical role that data-driven initiatives play in improving public welfare and well-being.
