Data Intensive Computing
Project Phase #2

Project Overview :

This study seeks to unravel the intricate relationship between health, lifestyle, and the rising prevalence of diabetes. By analyzing extensive demographic, health, and lifestyle data, it aims to uncover the primary factors driving diabetes risk and understand how lifestyle choices and existing health conditions intersect with diabetes rates. Given the global burden of diabetes, understanding these connections is imperative for developing targeted interventions and preventive strategies.

With diabetes mellitus posing significant challenges to both individuals and healthcare systems worldwide, this research aims to quantify the impact of demographic variables, lifestyle choices, pre-existing health conditions, obesity, and blood sugar control on diabetes incidence. Through rigorous analysis, the study aims to provide insights crucial for effective diabetes management and prevention efforts.

Data Sources:

The dataset analyzed in this study comprises patient health records, including demographic information, health conditions, and lifestyle choices. It was sourced from a comprehensive health database [Kaggle], ensuring a broad representation of the population. The dataset contains around 100,019 records and 10 columns, adhering to the requirement for a large enough dataset for significant data analysis. Diabetes prediction dataset (kaggle.com) We also augmented the existing dataset for achieving more cleaning/Analysis of data.

Dataset Overview:

The dataset includes the following columns, each offering unique insights into the patient's health and lifestyle:

“**gender**” : Contains the gender of a person which is categorical data.
“**age**” : Contains the age data of different people.
“**hypertension**” : tells us whether a person has hypertension or not using ones and zeros.
“**heart_disease**” : tells us whether a person has heart related disease or not using ones and zeros.
“**smoking_history**” : Contains the smoking history of a person which is categorical data.
“**bmi**” : patient's Body Mass Index, with some missing values containing continuous data.

“**HbA1c_level**”: Continuous variable representing the patient's Hemoglobin A1c level, a key indicator of diabetes management, with some missing values.
“**blood_glucose_level**” : Continuous variable indicating the patient's blood glucose level.
“**diabetes**” : has ones or zeros representing whether a patient has diabetes or not.
“**Smoking_history_unknown**” : Binary variable specifically encoding unknown smoking history as a distinct category. Using the above features together can be used to predict the risk of diabetes.

In the previous phase we applied various data munging methods on the raw data we took and got a final cleaned dataset.
After performing we dropped a few rows which are not needed for our output prediction which is “gender” and “smoking history” because we need numerical values.

```
merged_df =merged_df.drop(columns =['gender', 'smoking_history'], axis =1)
```

Model implementation steps

1. Data Pre-processing:
    This step involved dropping duplicate records from the dataset, handling null values, and encoding categorical data points using one encoding approach.
2. Train-test split:
    The dataset was split into training and testing sets using train_test_split from scikit-learn at an 80%-20% ratio at a random state of 42. The test data was used for model evaluation.

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=.2, random_state =42)
```

3. Training:
    The model implements a fit method which is used to train the model on the training set. The model found the ideal coefficients to minimize the logistic loss function during training.
4. Evaluation:
    The trained model was evaluated on the testing set to determine its accuracy i.e. how well it can predict unknown data features.

For the current project the following models we took into consideration

1. Logistic regression
2. Naives bayes
3. K Nearest Neighbors
4. SVM
5. Random tree classifier
6. Neural networks

1. Logistic Regression :

Logistic Regression was chosen as the algorithm for this problem due to its suitability for binary classification tasks, which aligns with our target variable having two possible outcomes. The decision was based on the observed linearity between features and the target variable within our dataset. Additionally, Logistic Regression offers several advantages for our analysis:

Interpretability:

The results of Logistic Regression are easily interpretable, due to the coefficients clearly show the kind and degree of the link between the independent and dependent variables. This facilitates understanding the factors influencing diabetes prevalence.

Efficiency:

Large datasets can be handled with comparatively little computer effort thanks to the computational efficiency of logistic regression., which is advantageous considering the volume of data involved in our study.

Regularization:

The algorithm supports regularization techniques like L1 (Lasso) and L2 (Ridge), enabling us to mitigate overfitting by penalizing large coefficient values.

To optimize the model's performance, extensive tuning and training were conducted. This involved adjusting regularization parameters, feature selection, and handling imbalanced data. The effectiveness of the algorithm was assessed using relevant metrics:

Accuracy:

The model achieved an accuracy of 96.03%, indicating that it correctly classified the majority of test samples, reflecting strong overall performance.

Precision:

With a precision of 87.55%, the model provides a high level of correctness when predicting the positive class, crucial for identifying individuals at risk of diabetes.

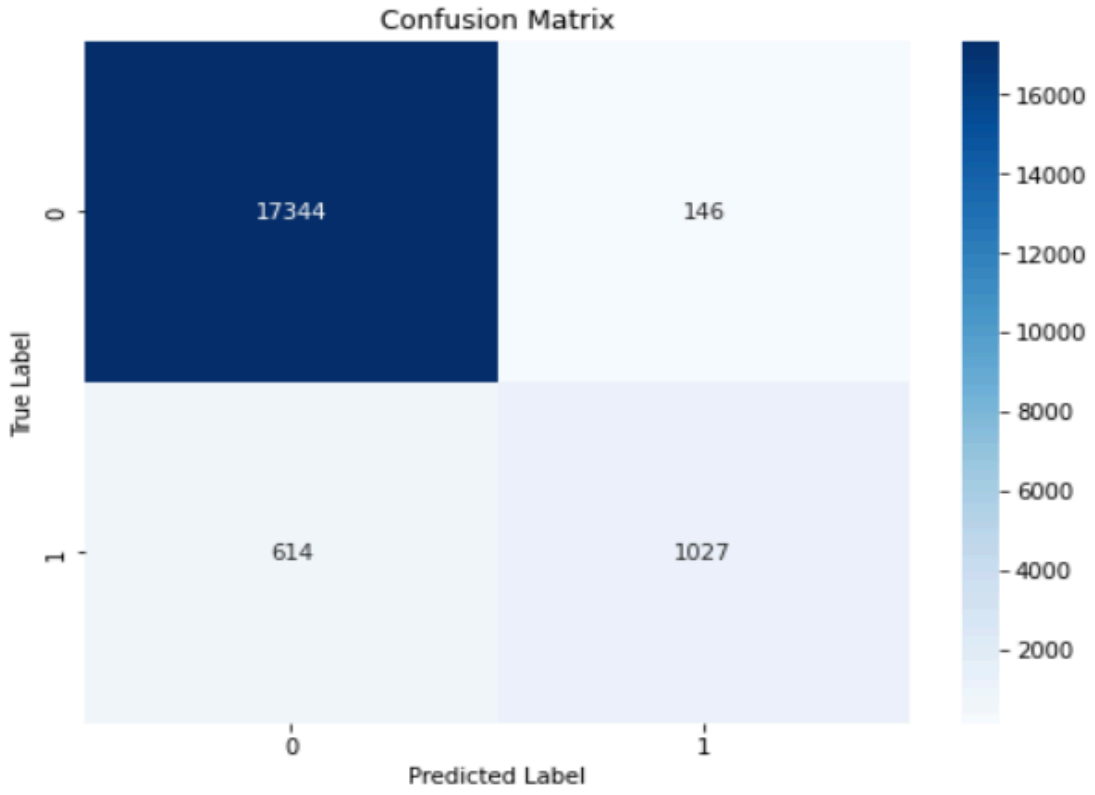
F1 Score:

The F1 score of 72.99% signifies a good balance between precision and recall, suggesting the model's effectiveness in capturing both true positives and minimizing false positives.

	Metric	Score
0	Accuracy	0.960274
1	Precision	0.875533
2	F1 Score	0.729922

Overall, the Logistic Regression model proved effective in addressing our problem statement, offering high accuracy and precision with a balanced F1 score. Insights gained from the application of this algorithm provide valuable understanding of the key factors influencing diabetes prevalence, aiding in the development of targeted interventions and preventive measures.

Following is the confusion matrix representation using heatmap.



2. Gaussian Naives Bayes:

It's a probabilistic classifier which models the distribution of each feature within a class as a Gaussian distribution and assumes features are independent of each other given the class.
Why I chose Gaussian Naïve Bayes

- Simplicity – The implementation of the Gaussian Naïve Bayes algorithm is simple and straightforward. It requires minimal regularization and fine tuning making it less prone to overfitting in relation to more complex algorithms.
- Efficiency – The simplicity of nature makes it more suitable for small or medium sized datasets and cases where computational power is limited.
- Robustness – GNB implicitly performs feature selection by estimating the parameters of the Gaussian distributions for each class independently. This capability makes it perform well even for irrelevant features.

Model effectiveness

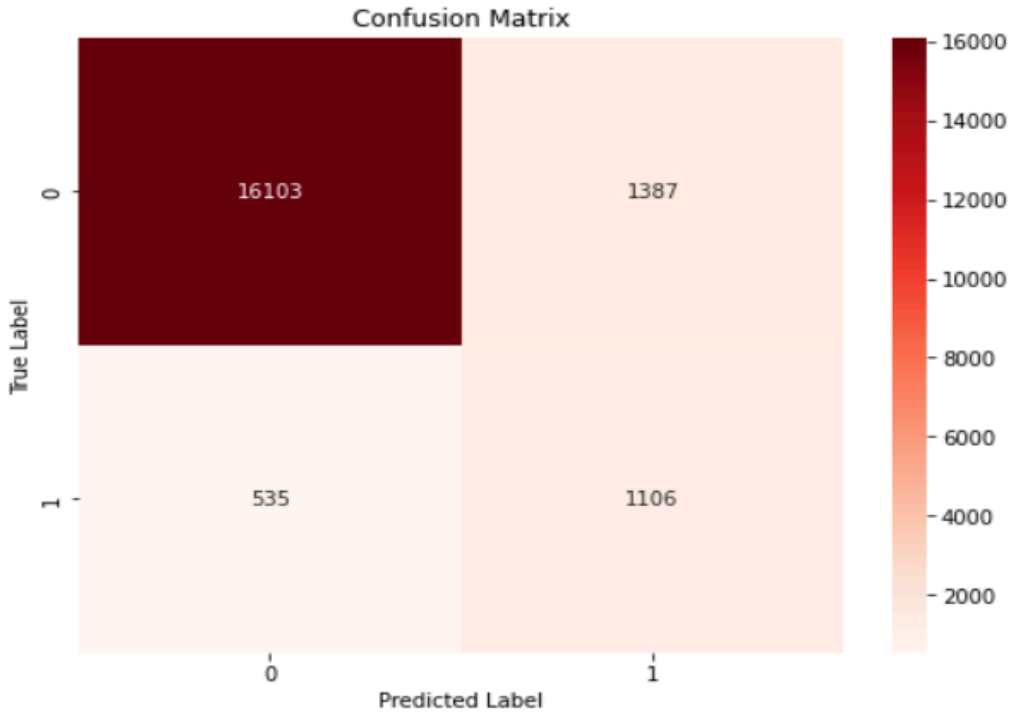
Accuracy - The model accuracy is 89.95%. In this case, the model correctly classified approximately 89.95% of the test samples, which indicates strong overall performance.

Precision: The precision of the model is 44.36%. A precision of 44.36% implies that in the event that the model forecasts a positive class, it is correct approximately 44.36% of the time.
F1 Score: The F1 score of the model is 53.51%. An F1 score of 53.51% indicates a reasonable balance between precision and recall.

	Metric	Score
0	Accuracy	0.899535
1	Precision	0.443642
2	F1 Score	0.535075

The model appears to have moderate effectiveness. While the accuracy is relatively high, the precision is lower, indicating that the model may have a higher rate of false positives.

Confusion matrix metric for the above model is as follows.



3.Support Vector Machines (SVM):
Support Vector Machines (SVM) was chosen as the algorithm for this problem due to its suitability for classification tasks and its ability to effectively position a decision boundary (hyperplane) to classify data into classes. This choice was influenced by several factors relevant to our dataset and problem statement:

Robustness: SVM's robustness is enhanced by proper regularization and fine-tuning of hyperparameters, such as the kernel parameters and regularization parameter, which enhance generalization ability and reduce overfitting..

Effectiveness in High-dimensional Spaces: SVM works effectively in high-dimensional spaces, which makes it appropriate for datasets containing a large number of characteristics. This capability is crucial, considering the dataset contains multiple features related to health and lifestyle factors that influence diabetes risk.

Flexibility: SVM allows the use of various kernel functions (e.g., linear, polynomial, RBF, sigmoid) to capture complex relationships in the data, enhancing its capability to model non-linear decision boundaries effectively. This flexibility is essential for accurately predicting diabetes risk based on diverse features.

To optimize the model's performance, extensive tuning and training were conducted, including regularization parameter tuning and kernel selection. Additionally,

preprocessing steps such as handling missing values and encoding categorical variables were performed. The effectiveness of the SVM algorithm was evaluated using relevant metrics:

```
svm = SVC(kernel='linear')
```

Accuracy: The model achieved an accuracy of 96.04%, indicating strong overall performance in correctly classifying test samples, which is crucial for accurately predicting diabetes risk.

Precision: With a precision of 92.66%, the model provides a high level of correctness when predicting the positive class (presence of diabetes). This indicates that when the model predicts a positive class, it is correct approximately 92.66% of the time, minimizing false positives.

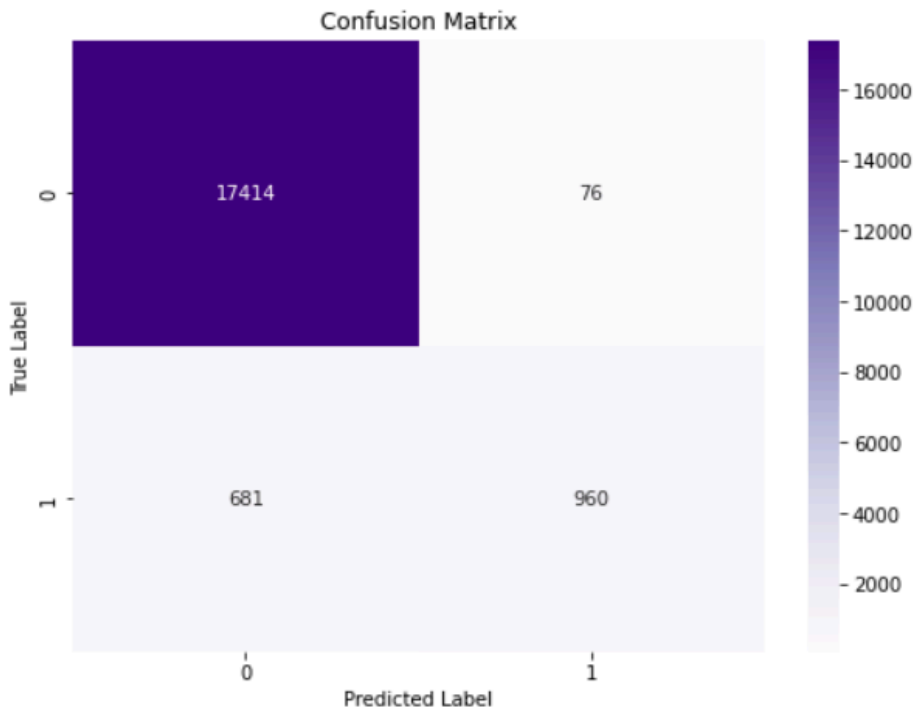
F1 Score: The F1 score of 71.72% reflects a reasonable balance between precision and recall, indicating the model's effectiveness in capturing both true positives and minimizing false positives, crucial for identifying individuals at risk of diabetes.

	Metric	Score
0	Accuracy	0.960431
1	Precision	0.926641
2	F1 Score	0.717221

By utilizing features such as gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, blood glucose level, and smoking history unknown, the SVM model effectively predicts the risk of diabetes. Insights gained from the application of this algorithm provide valuable understanding of the complex relationships between various health and lifestyle factors, aiding in the development of targeted interventions and preventive measures against diabetes.

The model appears to be quite effective, achieving high accuracy and precision with a reasonable F1 score.

Graphical representation of its evaluation metrics



4. KNeighbors Classifier:

It's a non-parametric, supervised learning algorithm to handle classification problems. It makes suggestions about how to arrange a single data point based on proximity.

Why I chose K-Neighbors Classifier

- Robustness - this algorithm relies on local information rather than global trends making it robust to noisy data and outliers, hence capable of handling datasets with irregular decision boundaries with minimal data pre-processing required.
- Simplicity – KNN is simple and easy to understand and implement. The algorithm does not make strong assumptions about data distribution, hence suitable for classification problems.
- Non-parametric: KNN makes predictions based on the local neighborhood of data points, and not by assumptions on the functional form of the underlying data distribution, making it flexible and adaptable to different types of data.

Model Effectiveness

Here we chose K = 10 so that it satisfies balanced bias-variance tradeoff and effective while evaluating the model.

```
knn = KNeighborsClassifier(n_neighbors=10)
```

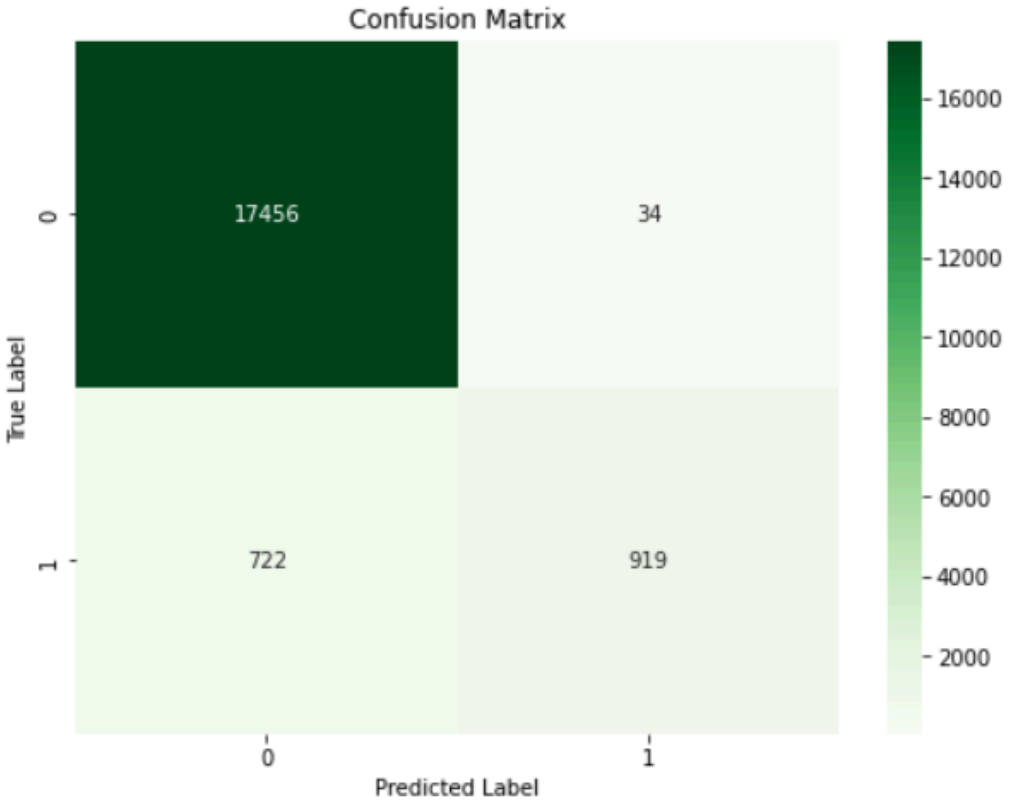
Accuracy - The model accuracy is 96.04%. In this case, the model correctly classified approximately 96.04% of the test samples, which indicates strong overall performance.

Precision: The precision of the model is 96.43%. A precision of 96.43% suggests that when the model predicts a positive class, it is correct approximately 96.43% of the time.
F1 Score: The F1 score of the model is 70.86%. An F1 score of 70.86% indicates a reasonable balance between precision and recall.

	Metric	Score
0	Accuracy	0.960483
1	Precision	0.964323
2	F1 Score	0.708558

The model appears to be quite effective, achieving high accuracy and precision with a reasonable F1 score.

Confusion matrix of KNN,



5. Random Tree Classifier:
This is a machine learning algorithm based on decision trees for classification problems. The algorithm is robust to overfitting and has the capability to handle high-dimensional data and capture complex data relationships.
Why I chose Random Tree Classifier

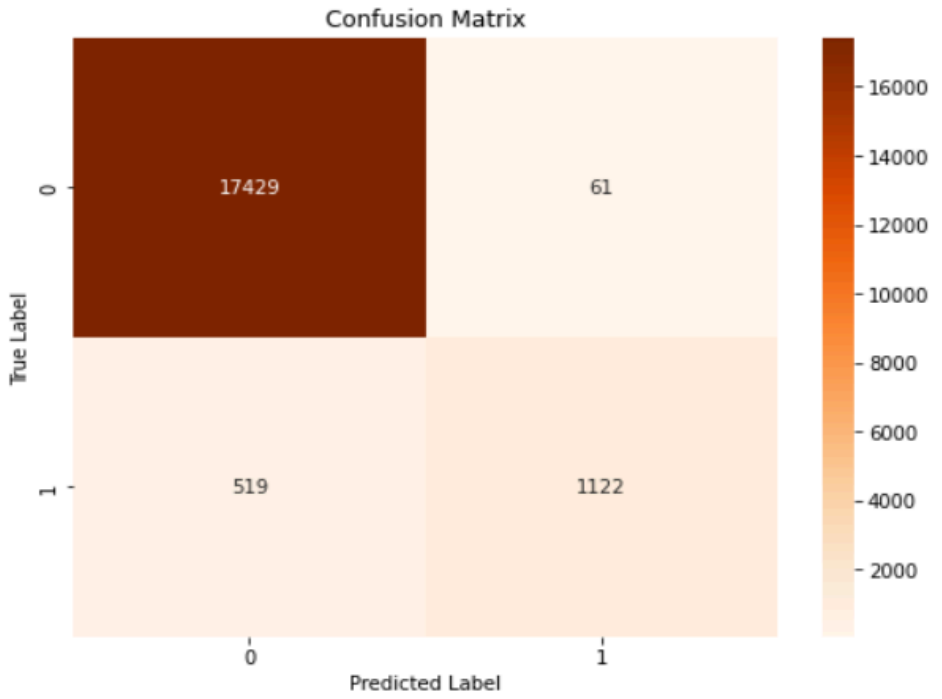
- Ensemble Learning –In order to produce a final forecast that is more reliable and less prone to overfitting, this algorithm constructs numerous decision trees and aggregates their predictions.
- Robustness - Random Forest Classifier implements an averaging mechanism making it more robust to noisy data. This makes it capable to handle missing values and categorical features with minimal pre-processing.
- Feature Importance - the classifier provides a measure of feature importance to help identify the most relevant features for classification. This capability allows for feature selection and understanding of the underlying data patterns.

Model Effectiveness

Accuracy - The model accuracy is 96.97%. In this case, the model correctly classified approximately 96.97% of the test samples, which indicates strong overall performance.
Precision: The precision of the model is 94.84%. A precision of 94.84% suggests that when the model predicts a positive class, it is correct approximately 94.84% of the time.
F1 Score: The F1 score of the model is 79.46%. An F1 score of 79.46% indicates a reasonable balance between precision and recall.

	Metric	Score
0	Accuracy	0.969683
1	Precision	0.948436
2	F1 Score	0.794618

The model appears to be highly effective, achieving high accuracy and precision with a good F1 score. This suggests that the model performs well in correctly classifying instances and maintaining a low rate of false positives.
Confusion Matrix representation,



6. Multi Layer Perceptron :

I chose the Multi-layer Perceptron (MLP) classifier for its flexibility, scalability, and generalization capabilities, making it suitable for our classification problem. The decision was influenced by several factors:

Flexibility: MLP is able to understand intricate, non-linear correlations between target variables and attributes, making it adept at approximating continuous functions. This flexibility enables it to handle a wide range of problems, including classification tasks.

Scalability: The algorithm is scalable and can efficiently handle large datasets with extensive samples and features. This scalability ensures that the model remains effective even when dealing with large amounts of data, such as in our classification task.

Generalization: MLP has the ability to generalize well to unseen data, especially when appropriately regularized. This capability ensures that the model performs reliably on new, unseen instances, enhancing its practical utility.

The model was instantiated with the following parameters:

```
nn = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000, random_state=42)
```

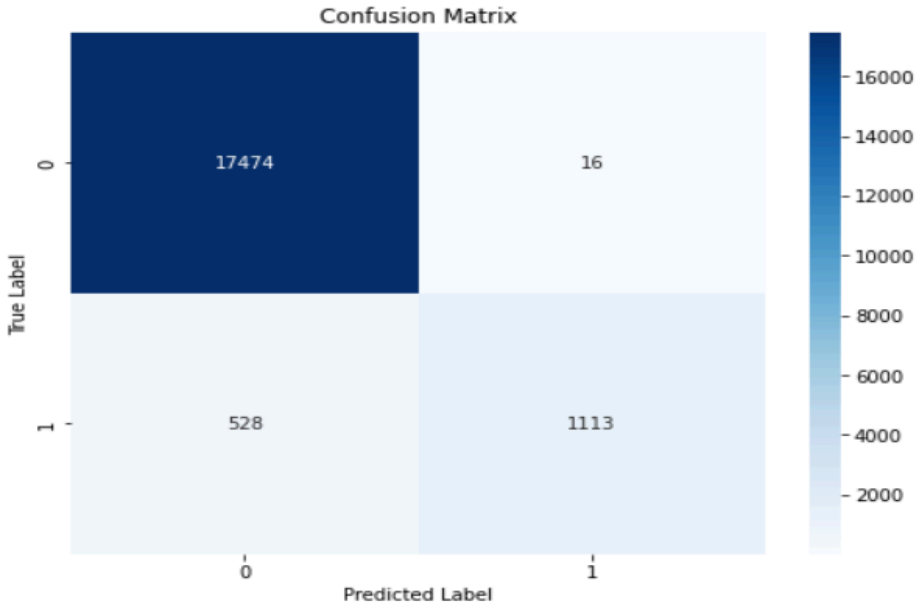
The MLP classifier underwent training with a maximum iteration limit of 1000 and utilized a single hidden layer with 100 neurons.

Model Effectiveness

Accuracy - The model accuracy is 97.15%. In this case, the model correctly classified approximately 97.15% of the test samples, which indicates strong overall performance. Precision: The precision of the model is 98.58%. A precision of 98.58% suggests that when the model predicts a positive class, it is correct approximately 98.58% of the time. F1 Score: The F1 score of the model is 80.36%. An F1 score of 80.36% indicates a reasonable balance between precision and recall.

	Metric	Score
0	Accuracy	0.971564
1	Precision	0.985828
2	F1 Score	0.803610

The model appears to be highly effective, achieving very high accuracy and precision with a good F1 score. This suggests that the model performs well in correctly classifying instances and maintaining a low rate of false positives.
Confusion matrix of MLP,

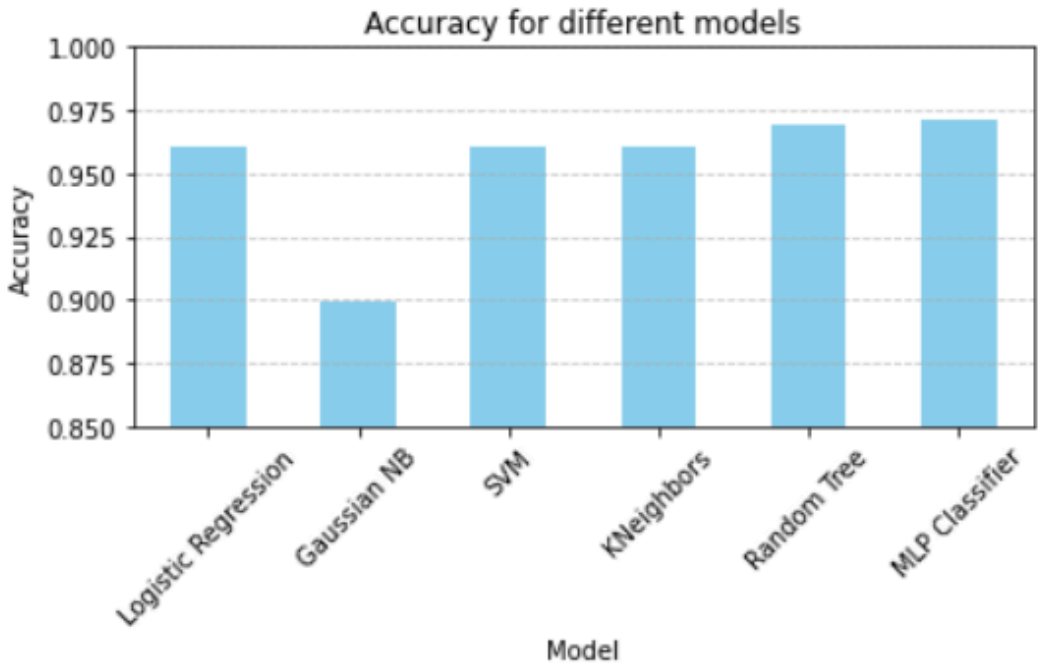


Final Comparison of accuracies and metrics for above models :

```
: Model_Accuracies ={"Model": [], "Accuracy": []}
```

We used empty lists for storing the model and its accuracy. As we implement each model we keep on appending to the lists.

At last we converted it into a dataframe. Later on, plotting a graph is as follows.



After thoroughly evaluating several machine learning models for predicting diabetes based on various health and lifestyle factors.Here we see Gaussian Naive Bayes has low accuracy as one of the reasons may be Hyperparamter tuning.
Overall, all models perform relatively well, with accuracies ranging from 89.95% to 97.16%. The optimal model to use depends on a number of variables, including the problem's particular requirements, interpretability, and computing complexity. The MLP Classifier is the clear superior in this instance in terms of accuracy.

In conclusion, by leveraging the insights gained from the application of various machine learning models, medicine practitioners and companies can develop targeted interventions and preventive measures to mitigate or decrease the risk of diabetes.

Peer Evaluation Form for Final Group Work

CSE 487/587A

- Please write the names of your group members.

Group member 1: Harshavardhan Reddy Nadedi

Group member 2: Likhith Kongara

Group member 3: Sai Sohan Kosaraju

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	Group member 1	Group member 2	Group member 3
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude.	5	5	5
Contributes significantly to the success of the project .	5	5	5
TOTAL	20	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%):

Group member 1: 33.3

Group member 2: 33.3

Group member 3: 33.4