Diabetes Prediction Using Machine Learning Models

Objective of the Analysis

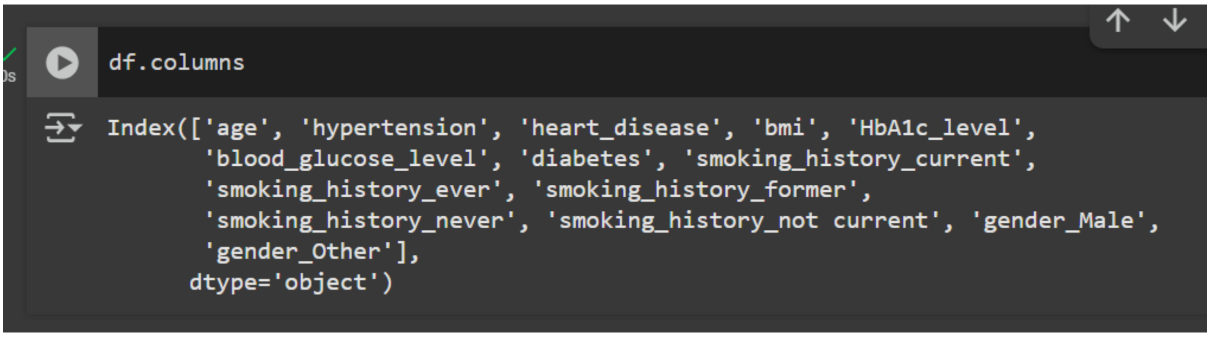
My project aims to predict whether a person has diabetes based on medical attributes. The primary goal is to evaluate different machine learning models to determine the most accurate and interpretable model for this prediction task.

Business Benefits

- Helps healthcare professionals detect diabetes early.
- Assists in preventive healthcare measures by identifying key risk factors.
- Provides insights into the most influential medical attributes affecting diabetes.

Dataset Description

The dataset was downloaded from **Kaggle** and contains multiple health-related features.



```
df.columns
```

```
Index(['age', 'hypertension', 'heart_disease', 'bmi', 'HbA1c_level',
       'blood_glucose_level', 'diabetes', 'smoking_history_current',
       'smoking_history_ever', 'smoking_history_former',
       'smoking_history_never', 'smoking_history_not current', 'gender_Male',
       'gender_Other'],
      dtype='object')
```

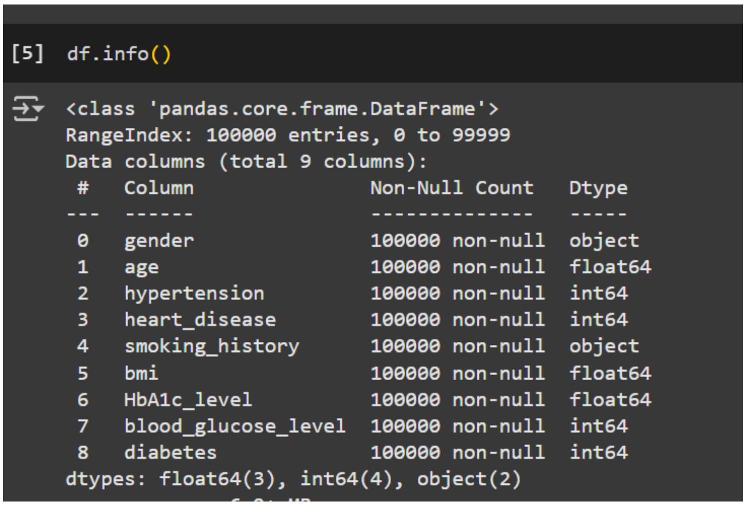
Data Preprocessing & Feature Engineering

Data Exploration Findings

- **Missing Values**: Checked and filled using median values.

  Since two features were objects, they were encoded while the numeric features were standard scaled.

- **Categorical Features**:
  - gender and smoking_history were converted to numerical values using One-Hot Encoding.
- **Feature Scaling**:
  - StandardScaler was applied to all numeric features for better model performance



```
[5]  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   gender               100000 non-null  object 
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64  
 3   heart_disease        100000 non-null  int64  
 4   smoking_history      100000 non-null  object 
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64  
 8   diabetes             100000 non-null  int64  
dtypes: float64(3), int64(4), object(2)
```

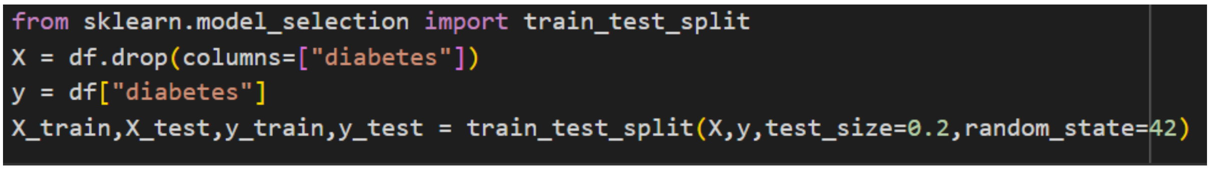
Model Training & Evaluation

Models Used

1. Logistic Regression
2. Random Forest
3. Support Vector Machine (SVM)

Training Setup

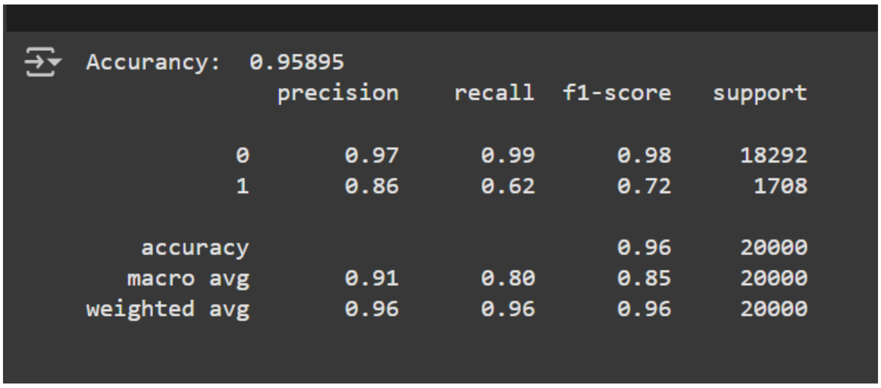
- **80% training, 20% testing split using train_test_split().**



```
from sklearn.model_selection import train_test_split
X = df.drop(columns=["diabetes"])
y = df["diabetes"]
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

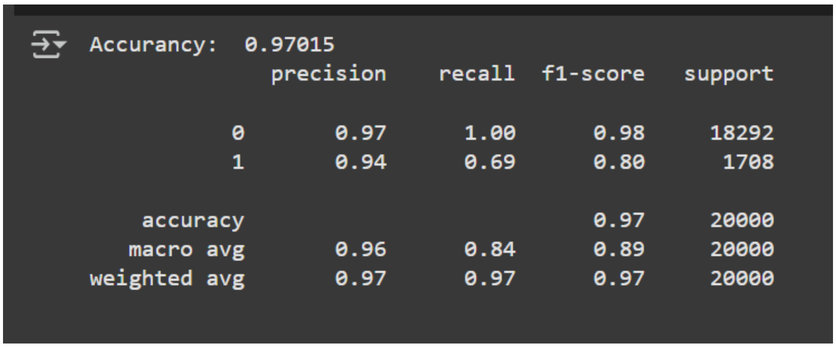
- Used accuracy, precision, recall, and F1-score for evaluation.

Logistic Regression Results:



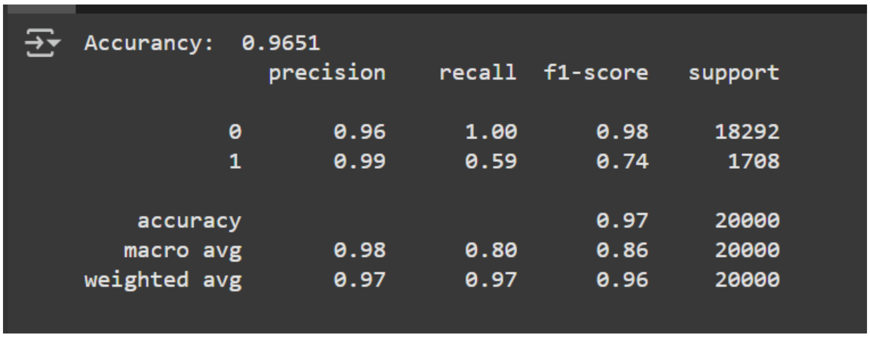
```
Accurancy:  0.95895
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     18292
           1       0.86      0.62      0.72      1708

    accuracy                           0.96     20000
   macro avg       0.91      0.80      0.85     20000
weighted avg       0.96      0.96      0.96     20000
```

Random Forest Results:



```
Accurancy:  0.97015
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     18292
           1       0.94      0.69      0.80      1708

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

SVM Results:



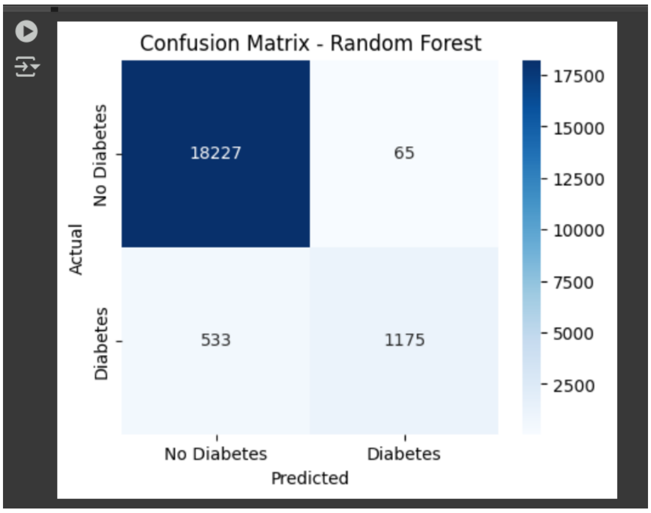
```
Accurancy:  0.9651
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     18292
           1       0.99      0.59      0.74      1708

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

Best Model Recommendation

The Random Forest Classifier was selected as the best model because:

- Random Forest had the highest accuracy with 97 percent .
- It provided feature importance, making it easier to understand key health factors affecting diabetes.
- It handled non-linearity well.

The confusion matrix analysis for Random Forest



True Positives (TP) = 1,175 → Correctly predicted **diabetes cases**

True Negatives (TN) = 18,227 → Correctly predicted **non-diabetes cases**

False Positives (FP) = 65 → Incorrectly predicted **diabetes when it's not**

False Negatives (FN) = 533 → Incorrectly predicted **non-diabetes when it is diabetes**

Conclusion:

Among the three models I used, Random Forest achieved the highest accuracy. However, its performance can be further improved by Hyperparameter tuning with GridSearchCV to optimize model parameters.