CSE 422 LAB REPORT

MEMBER 01

Name: Nusrat Zahin Chowdhury
ID : 23301553

MEMBER 02

Name: Rafi Ali
ID : 23301483

Table of Contents

CSE422 Lab Project Report

Introduction
The main motive of this project is to inspect a medical dataset which is based upon diabetes. Further in this report we talked about how to develop a machine learning model which helps to predict if a person is diabetic or not depending on various health related features. As we know diabetes is a serious and ongoing health issue which can be seen all over the world. Thus early detection can greatly help in the long run.

The main motive for this project is to realize how various algorithms based on machine learning can perform on real-life healthcare data. Moreover we try to compare the effectiveness of using different evaluation methods. By utilizing these sorts of supervised and unsupervised learning methodologies these projects can be very helpful in gaining on hand knowledge and real life experience in the aspect of data pre-processing as well as model training , evaluation and finally result interpretation.

Dataset description
The dataset which is used in this project is a Diabetes Prediction Dataset, which contains medical and life related attributes of patients. The main objective of this dataset is to predict whether a person has diabetes or not based on several health issues provided in the dataset such as glucose level, BMI, blood pressure, age, etc. This dataset is used in machine learning models to study classification problems in the healthcare domain.

How many features?

The dataset contains total 9 features, out of which:

1. 8 are input features
2. 1 is the output / target feature

```
df.shape
```

```
(100000, 9)
```

```
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               90165 non-null  object 
 1   age                  89935 non-null  float64
 2   hypertension         89852 non-null  float64
 3   heart_disease        90081 non-null  float64
 4   smoking_history      90012 non-null  object 
 5   bmi                  90056 non-null  float64
 6   HbA1c_level          89861 non-null  float64
 7   blood_glucose_level  90120 non-null  float64
 8   diabetes             89918 non-null  float64
dtypes: float64(7), object(2)
memory usage: 6.9+ MB
```

Classification or Regression Problem? Why?

This is a classification problem because the target variable diabetes has discrete values which is 0 or 1 where 0 represents a non-diabetic person and 1 represents a diabetic person.

Since the output belongs to categorical values, this problem is treated as a classification problem.

How many data points?

The dataset contains 100,000 data points, which makes it suitable for applying different machine learning models and comparing their performance.

What kind of features are present?

The dataset contains both numerical and categorical features.

Numerical Features:

1. Age
2. Bmi
3. HbA1c_level
4. blood_glucose_level
5. hypertension
6. heart_disease

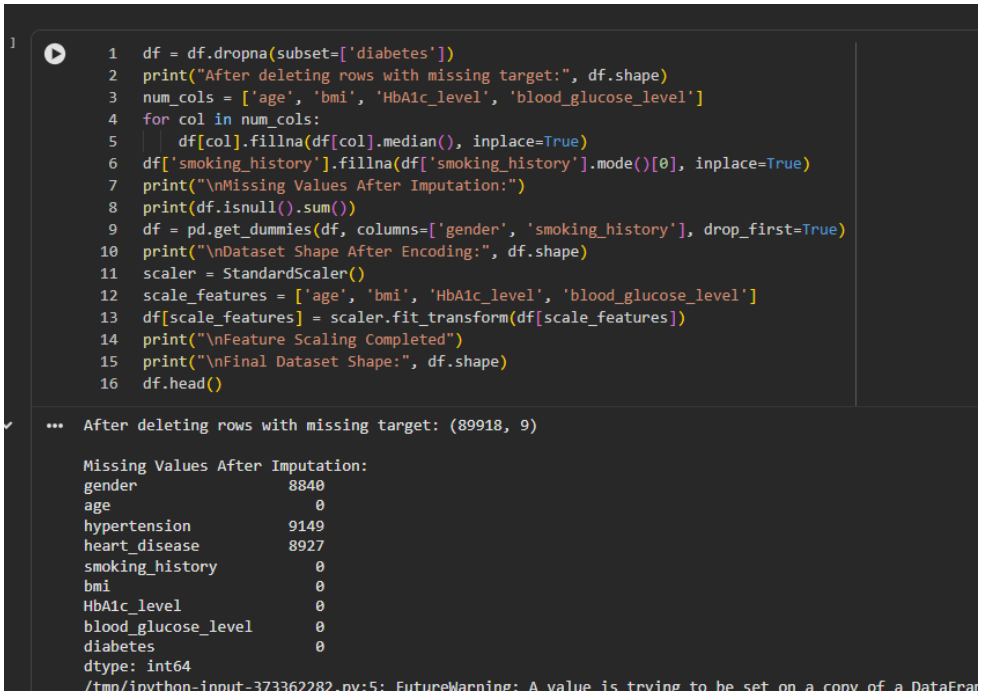
Categorical Features:

1. Gender
2. smoking_history

The target feature (diabetes) is also categorical.

Do we need to encode categorical variables? Why or why not?

Yes, categorical variables need to be encoded. Most machine learning algorithms cannot work directly with text or string values. For that reason, categorical features such as gender and smoking_history need to be converted into numerical form. In order to convert them we need to use encoding techniques like One-Hot Encoding which help the models to process the data correctly.



```
df = df.dropna(subset=['diabetes'])
print("After deleting rows with missing target:", df.shape)
num_cols = ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level']
for col in num_cols:
    df[col].fillna(df[col].median(), inplace=True)
df['smoking_history'].fillna(df['smoking_history'].mode()[0], inplace=True)
print("\nMissing Values After Imputation:")
print(df.isnull().sum())
df = pd.get_dummies(df, columns=['gender', 'smoking_history'], drop_first=True)
print("\nDataset Shape After Encoding:", df.shape)
scaler = StandardScaler()
scale_features = ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level']
df[scale_features] = scaler.fit_transform(df[scale_features])
print("\nFeature Scaling Completed")
print("\nFinal Dataset Shape:", df.shape)
df.head()
```

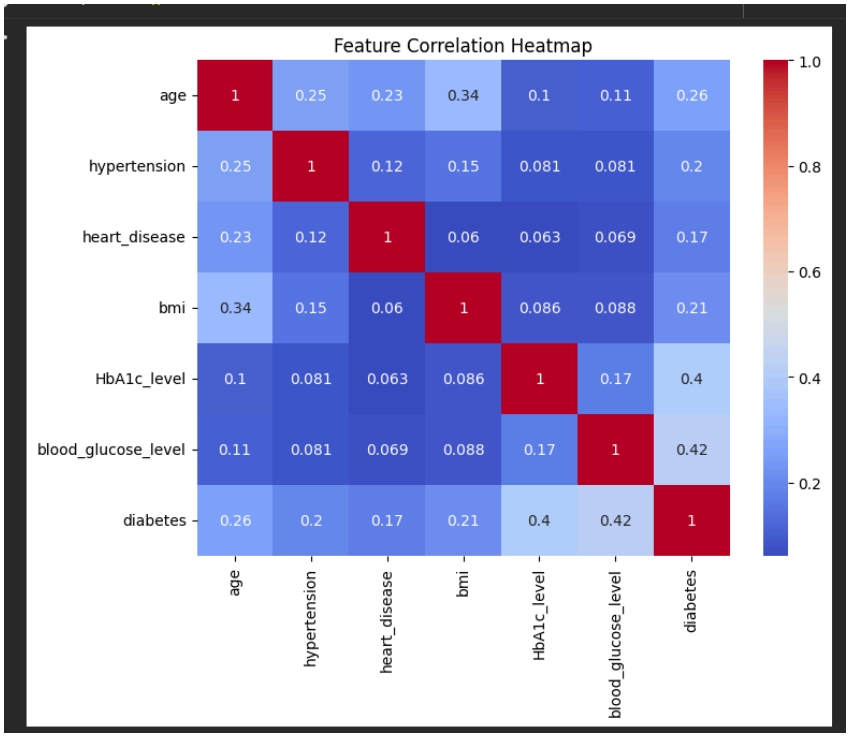
```
After deleting rows with missing target: (89918, 9)

Missing Values After Imputation:
gender                8840
age                      0
hypertension          9149
heart_disease         8927
smoking_history          0
bmi                      0
HbA1c_level              0
blood_glucose_level      0
diabetes                 0
dtype: int64
/tmp/ipython-input-373362282.py:5: FutureWarning: A value is trying to be set on a copy of a DataFra
```

Correlation of Features

A correlation heatmap was used to analyze the relationship between input features and the target variable. Observing the correlation test we can see how HbA1c_level and blood_glucose_level show a strong positive correlation with diabetes. Whereas, age has a moderate correlation with diabetes. Moreover, bmi shows a weaker but noticeable correlation where hypertension and heart_disease have low correlation

values.



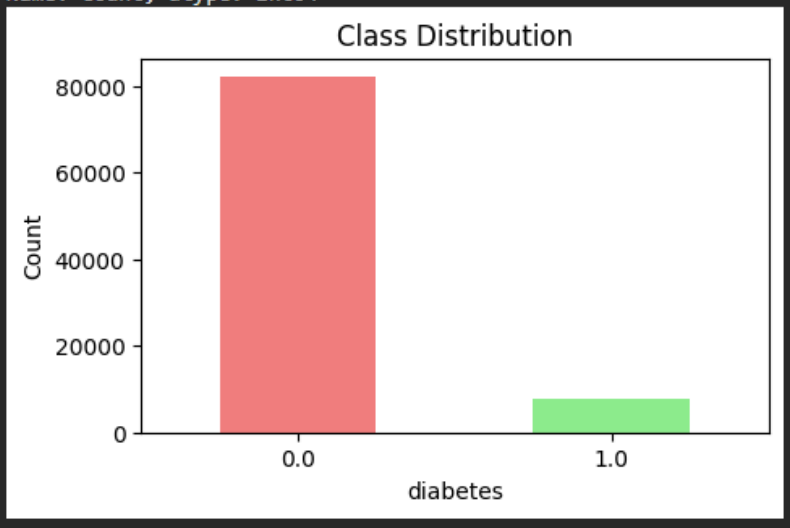
Understanding from Correlation: Blood sugar is the main feature in order to predict diabetes which makes sense and shows data is reliable.

Imbalanced Dataset

Are all classes equally distributed?

No, the dataset is imbalanced. The number of non-diabetic values is much higher than diabetic ones. This means the model might become biased toward predicting the majority class which is non-diabetic one.

```
diabetes
0.0    82284
1.0     7634
Name: count, dtype: int64
```



A bar chart of the target variable clearly shows that Class 0 which is Non-diabetic dominates the dataset whereas class 1 which is Diabetic has less samples. All of this confirms the presence of class imbalance.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand the structure and patterns in the dataset.

EDA Observations:

1. Diabetic patients generally have higher HbA1c and blood glucose levels.
2. Older individuals have a higher probability of being diabetic.
3. BMI shows some influence but it is not a strong predictor.
4. Smoking history shows mixed behavior and less directly related.

EDA helped identify important features which helped in preprocessing and model selection.

Dataset Pre-processing

Problem 1: Null / Missing Values

The dataset contains missing values both in input and output features.

Solution:

1. Rows with missing target values were deleted because target values cannot be guessed.
2. Missing numerical values were replaced using the median.
3. Missing categorical values were replaced using the mode.

Problem 2: Categorical Values

We faced challenges to handle text-based categorical data as Machine learning models faces difficulties in these sectors.

Solution:

We encoded the categorical features into numerical form by using One-Hot Encoding in order to avoid wrong ordinal relationships.

Problem 3: Feature Scaling

Features had different value ranges, which could bias distance-based models.

Solution:

Standard scaling was applied to normalize numerical features so that all features contribute equally.

Dataset Splitting

The dataset was split using stratified sampling due to class imbalance.

1. **Training set:** 80%

2. **Validation set:** 10% (from training data)

3. **Test set:** 20%

Stratification ensured that class proportions remained consistent across all subsets.

```python
from sklearn.model_selection import train_test_split
X = df.drop('diabetes', axis=1)
y = df['diabetes']
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.20, train_size=0.80, stratify=y, random_state=42)
print("Training Set Shape:", X_train.shape)
print("Testing Set Shape:", X_test.shape)
X_train, X_val, y_train, y_val = train_test_split( X_train, y_train, test_size=0.125, stratify=y_train, random_state=42)
print("Final Training Set Shape:", X_train.shape)
print("Validation Set Shape:", X_val.shape)
print("Test Set Shape:", X_test.shape)
```

```
Training Set Shape: (71934, 13)
Testing Set Shape: (17984, 13)
Final Training Set Shape: (62942, 13)
Validation Set Shape: (8992, 13)
Test Set Shape: (17984, 13)
```

Model Training & Testing

Supervised Learning Models Used:

1. Logistic Regression
2. Decision Tree
3. Neural Network (MLP)

The Neural Network was mandatory, and at least two other supervised models were applied as required.

Unsupervised Learning:

1. We applied K-Means clustering by ignoring the target labels.

2.The clusters revealed natural groupings based on health indicators which were found to be loosely aligned with diabetic as well as non-diabetic patterns.

Model Selection / Comparison Analysis

Metrics Used:

1. Accuracy
2. Precision
3. Recall
4. Confusion Matrix
5. ROC Curve and AUC score

Observations:

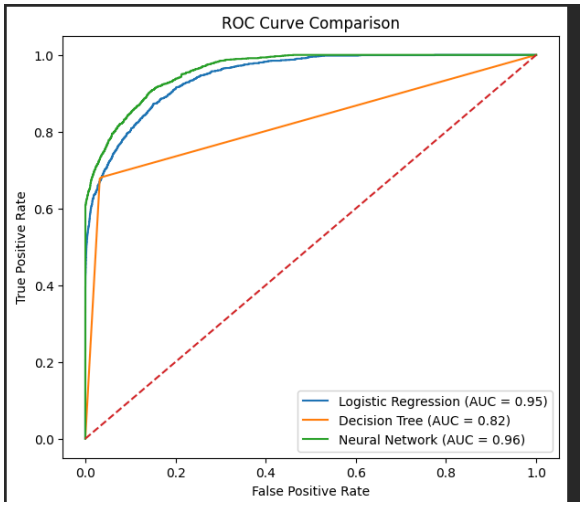
1.As a baseline model Logistic Regression performed well.

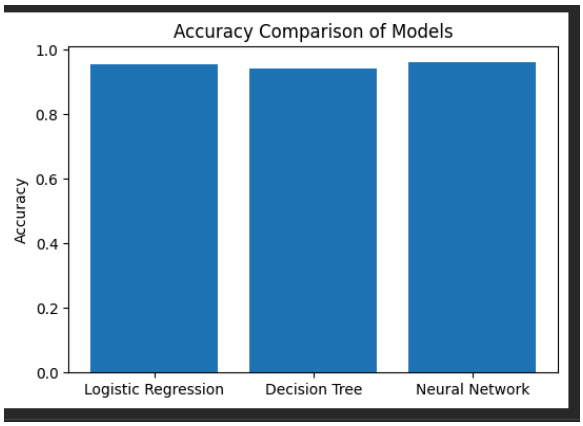
2.The non-linear patterns were captured by the Decision tree which resulted in improved recall.

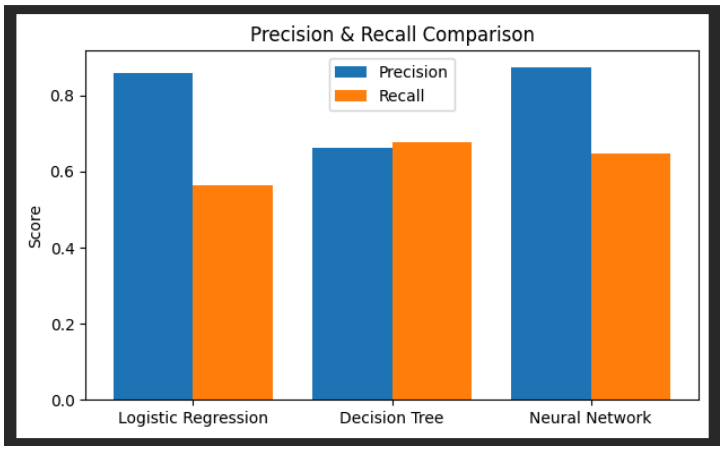
3.Overall the best performance was seen from Neural Network as it performs best across most of the metrics .

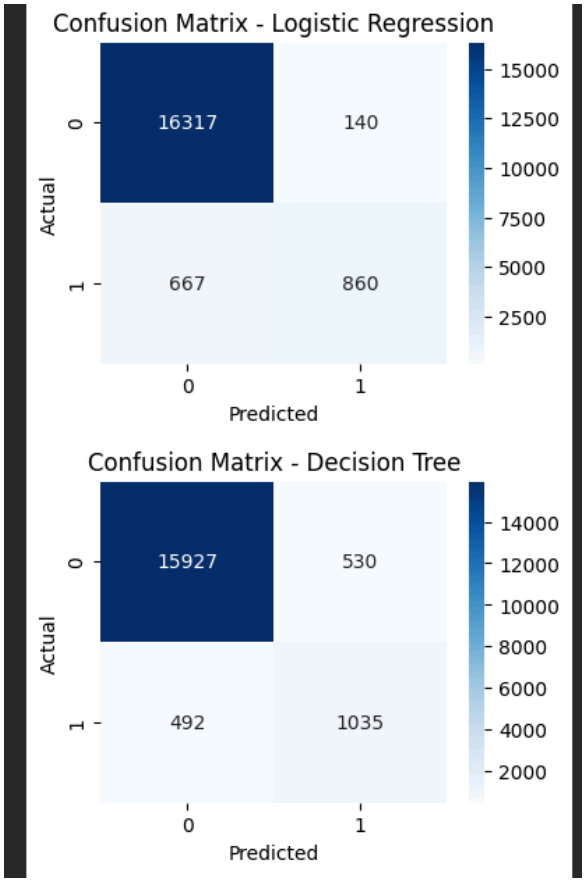
Note on Regression Metrics:

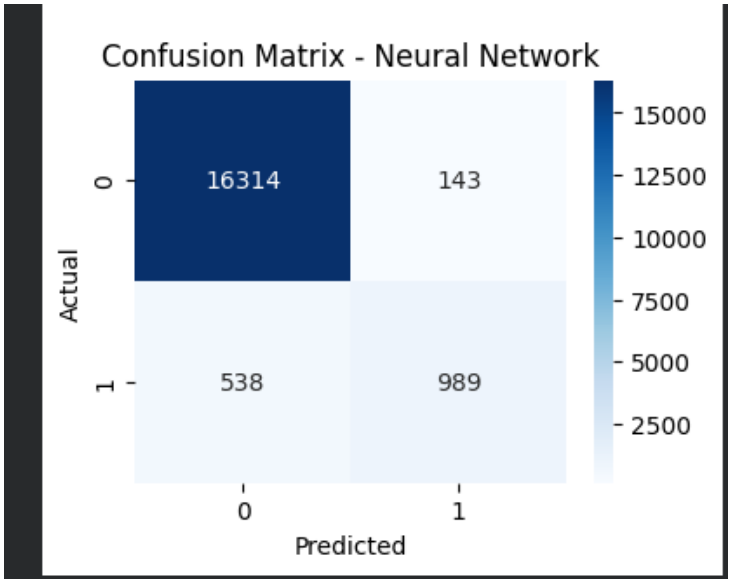
R^2 score and loss are applicable only for regression problems. Since this project focuses on classification, these metrics were not used.


ROC Curve Comparison
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (AUC = 0.95)
Decision Tree (AUC = 0.82)
Neural Network (AUC = 0.96)


Accuracy Comparison of Models
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression
Decision Tree
Neural Network


Precision & Recall Comparison
Precision
Recall
Score
0.0
0.2
0.4
0.6
0.8
Logistic Regression
Decision Tree
Neural Network


Confusion Matrix - Logistic Regression
16317
140
667
860
Actual
Predicted
0
1
15000
12500
10000
7500
5000
2500
Confusion Matrix - Decision Tree
15927
530
492
1035
Actual
Predicted
0
1
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - Neural Network
16314
143
538
989
Actual
Predicted
0
1
15000
12500
10000
7500
5000
2500

Conclusion

What do you understand from the results?

According to the results we can say that the Neural Network model outperformed or achieved more than the other models in different aspects such as accuracy , recall and also AUC score. Thus we can say it indicates its ability to capture complex relationships in the data.

Comments on Model Performance:

1.It is easier to interpret Simple models like Logistic Regression

2.Tree-based and neural models are more capable of handling non-linearity better.

3.But Some Models are affected by recall due to class imbalance.

Why do you think these results occurred?

The Neural Network performed better because:

1.It can model complex feature interactions

2.It works great with larger datasets.

3. We can have improved learning efficiency through preprocessing.

Challenges Faced:

1.We face great difficulties in working with the missing values without losing the data

2.Moreover dealing with the class imbalance also put a great toll on us

3.In selecting proper or appropriate evaluation metrics

4.Selecting the correct and accurate model was also a challenge we faced.