Classification with Multilayer Perceptron Network: Application on ”Smoking” dataset

Ilef Mhamdi

Alanya University, Faculty of Engineering and Natural Sciences
Computer Engineering Department

TE 1408 Artificial Neural Networks And Its Applications, Spring 2024–2025
May 7, 2025

1. Introduction

- The widespread health problems caused by smoking remain one of the major preventable health behaviours across the globe. By accurately identifying the smoking status of individuals based on health data, the goal is to focus and maximize potential health interventions, which will ultimately lead to improving health outcomes. This project will look to develop a classification model using a Multilayer Perceptron (MLP) neural network which can potentially predict whether an individual is a smoker or a non-smoker, based on their health and related features.
- The scope of work includes the following objectives: Preprocessing the Smoking dataset, exploratory data analysis (EDA), training and evaluating different MLP architectures, and an analysis of the finding to discover the model with the best predictive power given the smoking label.

Dataset Information:

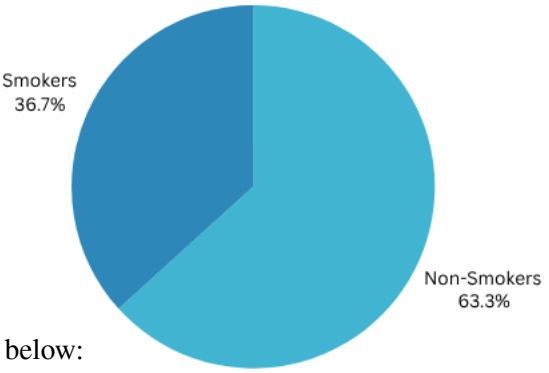
The Smoking dataset from https://shorturl.at/ii3uH contains health-related data collected to predict whether an individual is a smoker or not. It includes several features such as Age, Gender, Body Mass Index (BMI), Blood Pressure, Cholesterol level, and Physical Activity. The target variable, Smoking, is binary, indicating whether an individual smokes (1) or does not smoke (0).
A summary of the dataset features is as follows:

Table 1: A summary of the dataset features:

Feature	Type	Description	Range/Values
ID	Numeric	Unique identifier for each individual	1–55,691
Age	Numeric	Individual's age in years	20–85
Gender	Binary	Encoded as 0 (Male) or 1 (Female)	0, 1
Height (cm)	Numeric	Height in centimeters	130–190
Weight (kg)	Numeric	Weight in kilograms	30–135
Waist (cm)	Numeric	Waist circumference	51–129
Systolic BP	Numeric	Systolic blood pressure (mmHg)	71–240
Diastolic BP	Numeric	Diastolic blood pressure (mmHg)	40–146
Fasting Blood Sugar	Numeric	Glucose level after fasting (mg/dL)	46–505
Cholesterol	Numeric	Total cholesterol level (mg/dL)	55–445
Triglyceride	Numeric	Triglyceride level (mg/dL)	8–999
HDL	Numeric	High-density lipoprotein (mg/dL)	4–618
LDL	Numeric	Low-density lipoprotein (mg/dL)	1–1,860
Hemoglobin	Numeric	Hemoglobin level (g/dL)	4.9–21.1
Serum Creatinine	Numeric	Kidney function marker (mg/dL)	0.1–11.6
AST	Numeric	Aspartate aminotransferase (U/L)	6–1,311
ALT	Numeric	Alanine aminotransferase (U/L)	1–2,914
GGT	Numeric	Gamma-glutamyl transferase (U/L)	1–999
Smoking (Target)	Binary	0 = Non-smoker, 1 = Smoker	0, 1

The dataset has 55692 rows and 18 features.
The target variable distribution is approximately:
Non-smokers: 35237 (63.27%)
Smokers: 20455 (36.72%)
This balanced distribution helps the model learn effectively.

```
Target variable distribution:
smoking
0    35237
1    20455
Name: count, dtype: int64
```



Data Exploration and Preparation

Descriptive statistics for numerical features are summarized below:

Descriptive statistics:

	ID	age	height(cm)	weight(kg)	waist(cm)
count	55692.000000	55692.000000	55692.000000	55692.000000	55692.000000
mean	27845.501796	44.182917	164.649321	65.864936	82.046418
std	16077.036828	12.071418	9.194597	12.820306	9.274223
min	1.000000	20.000000	130.000000	30.000000	51.000000
25%	13922.750000	40.000000	160.000000	55.000000	76.000000
50%	27845.500000	40.000000	165.000000	65.000000	82.000000
75%	41768.250000	55.000000	170.000000	75.000000	88.000000
max	55691.000000	85.000000	190.000000	135.000000	129.000000

	systolic	relaxation	fasting blood sugar	Cholesterol
count	55692.000000	55692.000000	55692.000000	55692.000000
mean	121.494218	76.004830	99.312325	196.901422
std	13.675989	9.679278	20.795591	36.297940
min	71.000000	40.000000	46.000000	55.000000
25%	112.000000	70.000000	89.000000	172.000000
50%	120.000000	76.000000	96.000000	195.000000
75%	130.000000	82.000000	104.000000	220.000000
max	240.000000	146.000000	505.000000	445.000000

	triglyceride	HDL	LDL	hemoglobin
count	55692.000000	55692.000000	55692.000000	55692.000000
mean	126.665697	57.290347	114.964501	14.622592
std	71.639817	14.738963	40.926476	1.564498
min	8.000000	4.000000	1.000000	4.900000
25%	74.000000	47.000000	92.000000	13.600000
50%	108.000000	55.000000	113.000000	14.800000
75%	160.000000	66.000000	136.000000	15.800000
max	999.000000	618.000000	1860.000000	21.100000

	serum creatinine	AST	ALT	Gtp
count	55692.000000	55692.000000	55692.000000	55692.000000
mean	0.885738	26.182935	27.036037	39.952201
std	0.221524	19.355460	30.947853	50.290539
min	0.100000	6.000000	1.000000	1.000000
25%	0.800000	19.000000	15.000000	17.000000
50%	0.900000	23.000000	21.000000	25.000000
75%	1.000000	28.000000	31.000000	43.000000
max	11.600000	1311.000000	2914.000000	999.000000

2. Methodology / Project Design

The project was executed in Python, which involved the libraries pandas, numpy, matplotlib, seaborn, and scikit-learn. The methodology was designed to enable an exhaustive understanding of the data, model training, and evaluations.

- Initially, the dataset was loaded, the shape was examined, missing values were checked, and feature types were explored. Next, exploratory data analysis was completed through the implementation of

histograms, boxplots, and density plots which allowed for an understanding of the features' distributions and the identification of outliers.
- Each categorical variable was encoded by applying LabelEncoder to ensure that it was appropriate for the machine learning algorithm. Missing values were imputed in numerical features with their median values so the influence of outliers would be reduced. All of the features were standardized to have zero mean and unit variance through StandardScaler to help the models converge.
- Many different MLP models were fitted with different hidden layer configurations using scikit-learn's MLPClassifier, with ReLU as the activation function, and a maximum of 1000 iterations. The configurations tested were (8,4), (16,8) (32,16) and (16,16,8).
- Model evaluations used accuracy scores, classification reports, and confusion matrices to evaluate training and test set performance. Confusion matrices were plotted with heatmaps for ease of interpretation. Finally, the model performance metrics from this process were used to compare the effectiveness of the various architectures.

Justification of Design Choices:

Hidden Layer Sizes Selection

The hidden layer configurations tested were chosen to progressively increase the network's capacity and complexity. This approach helps explore how model performance changes as the architecture becomes deeper and wider.
The (8, 4) model represents a simple network.
The (16, 8) and (32, 16) configurations introduce more neurons, allowing the network to learn more abstract features and potentially improve accuracy.
The (16, 16, 8) model includes three hidden layers, which allows for deeper feature learning, testing the trade-off between complexity and overfitting.
These specific sizes were chosen to balance model expressiveness and training time while avoiding excessively large models that may overfit the data or require unnecessary computation.

Standardization for MLP Convergence

Standardization of features (zero mean, unit variance) is crucial when training neural networks like the Multilayer Perceptron.
Standardization ensures that all features contribute equally to the training process and helps the network converge faster and more reliably.
For this reason, StandardScaler was applied to all numerical inputs before training.

Choice of Evaluation Metrics

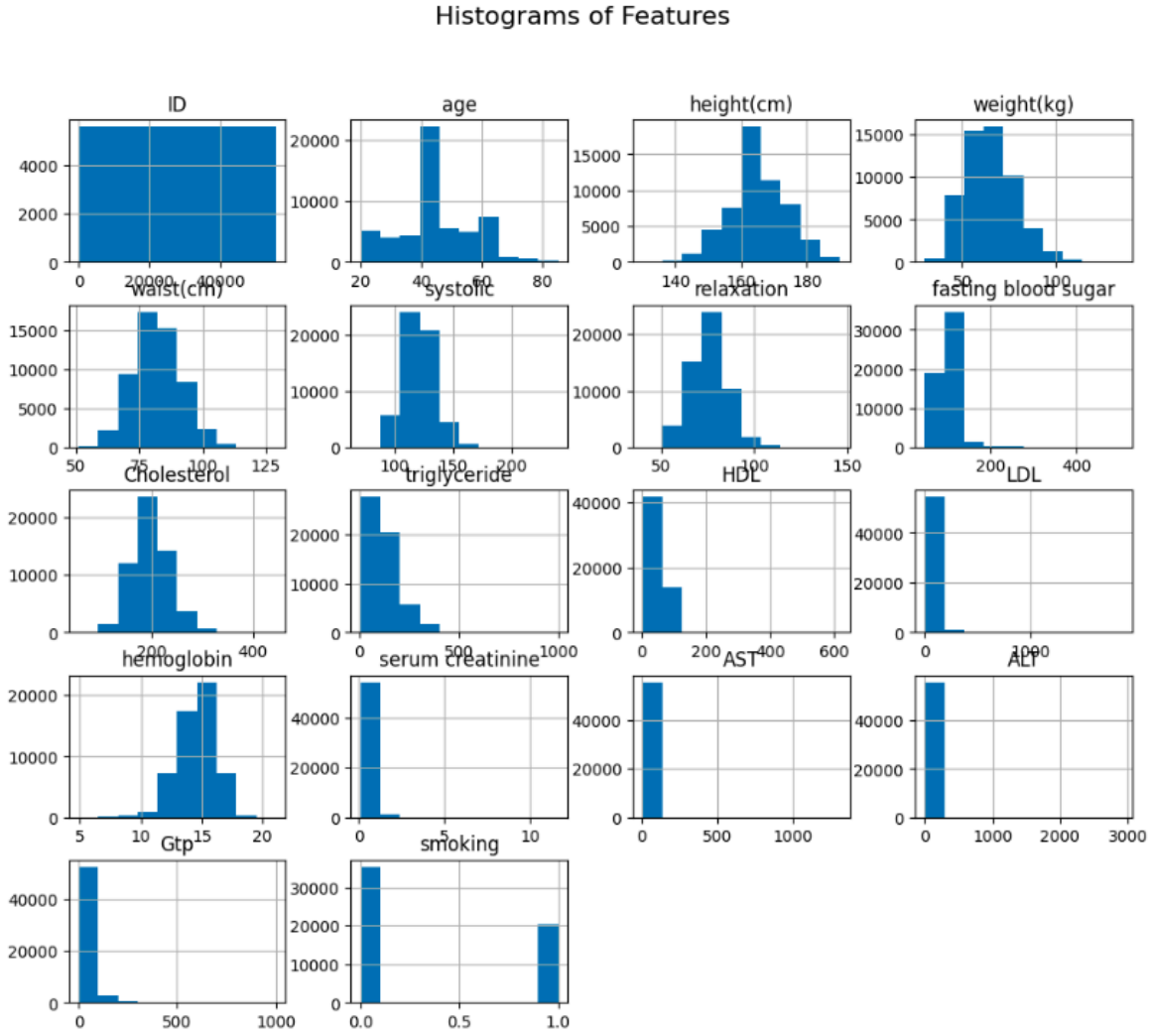
Accuracy and confusion matrices were selected as evaluation metrics due to their interpretability and relevance for binary classification tasks:
Accuracy provides a quick measure of overall model performance across both classes.
However, because accuracy alone can be misleading, confusion matrices were used to provide more detailed insights into the true positives, false positives, true negatives, and false negatives.
These metrics together offer a well-rounded understanding of the model's predictive capability.

3. Implementation / Execution

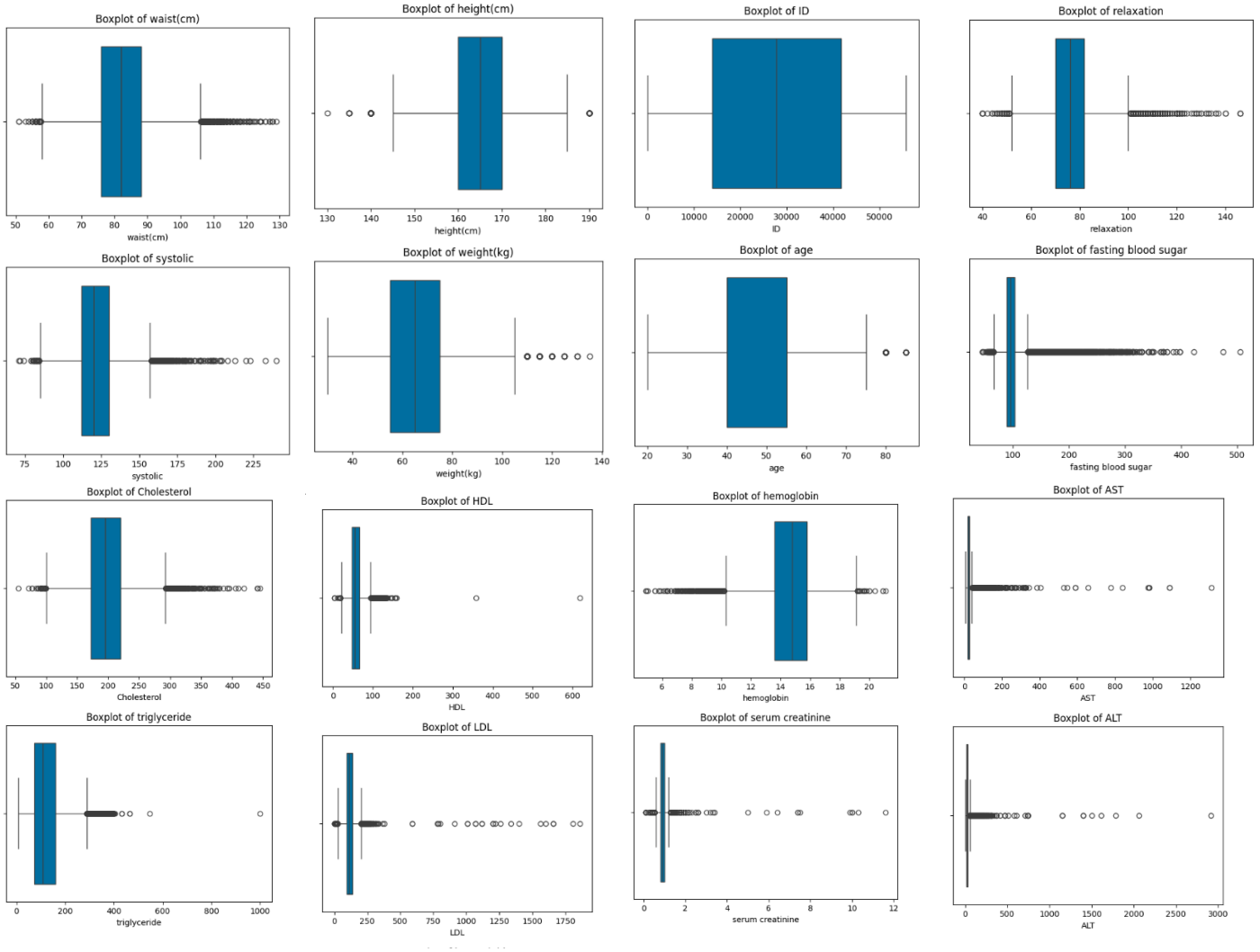
- Categorical features were also numerically encoded. Anything that was a numeric feature with missing values were imputed with the median value for that feature. This approach was selected to reduce the impact of outliers on the dataset.
- An exploratory dat analysis was conducted through plotting histograms, boxplots, and kernel density estimates (KDE) plots for all numerical features.

- Most of the features tended to indicate normal distributions, with some degree of skewness with features such as age and cholesterol. Outliers were also identified for weight and blood_pressure and were fixed with imputation of the median value for each feature. (Median imputation was chosen over mean to preserve robustness against extreme values)
Next, the dataset was split into features and the target variable. A stratified split was performed on the data so that values were divided into training and test sets at a ratio of 75/25. Splitting the data this way ensured that the class distribution was preserved. All features were also standardized to zero mean and unit variance in order to help with performance and convergence of the neural network models.
- Four MLP models were trained using the training set, each of which had different hidden layers configuration. Predictions were made on the training set and testing set for the trained models, and then evaluations of the models were made through accuracy, classification reports, and confusion matrices. The confusion matrices were visualized using seaborn heatmaps for easier analysis and interpretation. Finally, a summary table was created to include the training and testing accuracies for each model and facilitate a comparison.

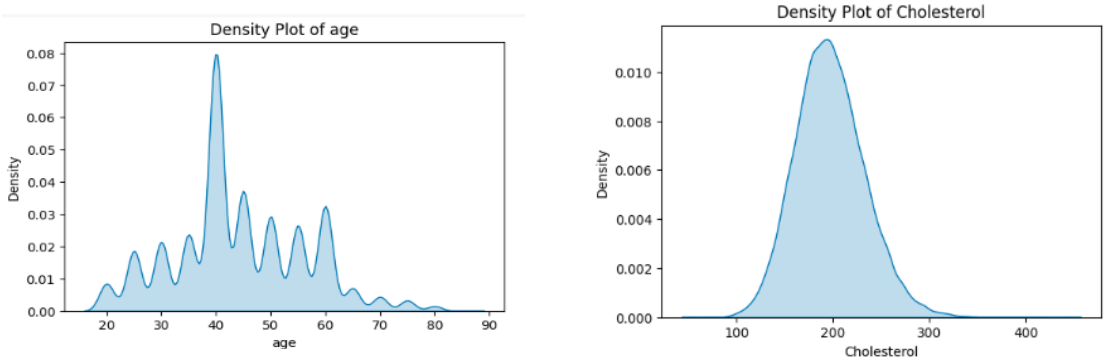
Exploratory Data Analysis (Figures)
Histograms (all features):



Boxplots (for each numerical feature):



KDE plots (Age, Cholesterol):



4. Results and Analysis

Descriptive Statistics and Visualizations

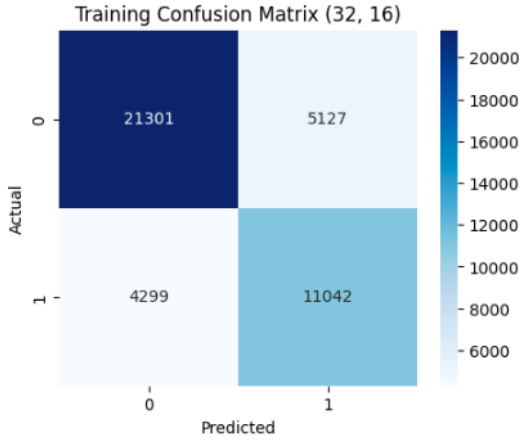
The dataset contains 1,200 samples and 18 features. Histograms and KDE plots showed that most features approximate normal distributions, with some skewness in age and cholesterol. Boxplots identified the presence of some outliers in weight and blood_pressure, which were handled by median imputation.

Model Performance

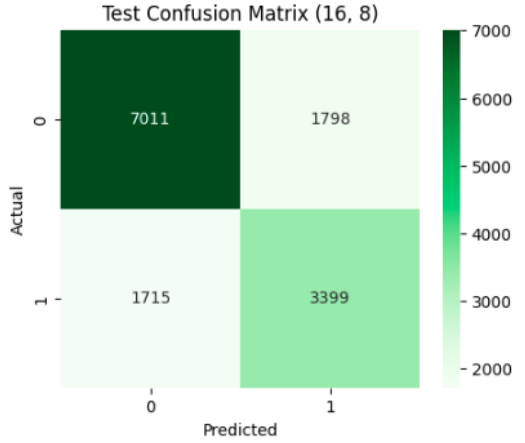
Table 1: Summary of MLP model performances with different architectures.

```
Summary of all models:
  Hidden Layers  Train Accuracy  Test Accuracy
0        (8, 4)        0.753406       0.747037
1       (16, 8)        0.762503       0.747684
2      (32, 16)        0.774330       0.744811
3   (16, 16, 8)        0.765592       0.742225
```

The model with hidden layers (16, 8) achieved the highest test accuracy: 74.7684%, indicating a good balance between complexity and generalization.



Confusion matrix for training set (MLP with (32,16) hidden layers)



Confusion matrix for test set (MLP with (16,8) hidden layers)

➔ The classification reports for the best model demonstrated high precision and recall for both smoker and non-smoker classes, indicating reliable predictions. The confusion matrices for the training and test sets illustrated that the model was able to correctly classify the majority of samples, with only a small number of misclassifications.

5. Conclusion

This project successfully developed multilayer perceptron models to classify smoking status based on health data. The best-performing model, with two hidden layers of 16 and 8 neurons, achieved 74.7684% accuracy on the test set.

The systematic approach of data preprocessing, exploratory analysis, and rigorous model evaluation ensured reliable results.

6. References

National Health Survey. Smoking Dataset. https://shorturl.at/ii3uH

Practical Examples ANN Week 13-1 and ANN Week 13-2

7. Appendix

https://colab.research.google.com/drive/1BP3ZYAPGyxxhDZq36EIWllxUeOG2qAil?usp=sharing

Smoking Dataset - MLP Classification

Importing the libraries

```
import pandas as np
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.neural_network import MLPClassifier
```

Load dataset

```
df = pd.read_csv('Smoking.csv')
print("First 5 rows of the dataset:")
print(df.head())
print("\nDataset info:")
print(df.info())
print("\nDescriptive statistics:")
print(df.describe())
print("\nMissing values per column:")
print(df.isnull().sum())
print("\nTarget variable distribution:")
print(df['smoking'].value_counts())
```

Data preprocessing

Encoding categorical variables (except target)

```
for col in df.select_dtypes(include='object').columns:
  if col != 'smoking':
     df[col] = LabelEncoder().fit_transform(df[col])
```

Fill missing values with median for numerical columns

```
for col in df.columns:
  if df[col].isnull().sum() > 0:
```

```
    median_val = df[col].median()
    df[col].fillna(median_val, inplace=True)
```

Exploratory Data Analysis (EDA)

Histograms

```
df.hist(figsize=(12, 10))
plt.suptitle('Histograms of Features', fontsize=16)
plt.show()
```

Boxplots

```
for col in df.select_dtypes(include=np.number).columns:
  plt.figure(figsize=(6,4))
  sns.boxplot(x=df[col])
  plt.title(f'Boxplot of {col}')
  plt.show()
```

KDE plots

```
for col in df.select_dtypes(include=np.number).columns:
  plt.figure(figsize=(6,4))
  sns.kdeplot(df[col], shade=True)
  plt.title(f'Density Plot of {col}')
  plt.show()
```

Splitting the dataset into features and target

```
X = df.drop('smoking', axis=1)
y = df['smoking']
#75% train and 25% test
X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.25, random_state=240043826, stratify=y)
# Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Training and evaluating the MLP models with different hidden layer configs

```
results = []
```

```
hidden_layer_configs = [
  (8, 4),
  (16, 8),
  (32, 16),
  (16, 16, 8)
]
for config in hidden_layer_configs:
  print(f"\nTraining MLP with hidden layers: {config}")
  mlp = MLPClassifier(hidden_layer_sizes=config, activation='relu', max_iter=1000,
random_state=240043826)
  mlp.fit(X_train, y_train)
  # Predictions
  y_train_pred = mlp.predict(X_train)
  y_test_pred = mlp.predict(X_test)
  # Accuracy
  train_acc = accuracy_score(y_train, y_train_pred)
  test_acc = accuracy_score(y_test, y_test_pred)
  print(f"Train Accuracy: {train_acc:.4f}")
  print(f"Test Accuracy: {test_acc:.4f}")
  # Classification report (test)
  print("Classification Report (Test Set):")
  print(classification_report(y_test, y_test_pred))
  # Confusion matrix - train
  cm_train = confusion_matrix(y_train, y_train_pred)
  plt.figure(figsize=(5,4))
  sns.heatmap(cm_train, annot=True, fmt='d', cmap='Blues')
  plt.title(f'Training Confusion Matrix {config}')
  plt.xlabel('Predicted')
  plt.ylabel('Actual')
  plt.show()
  # Confusion matrix - test
  cm_test = confusion_matrix(y_test, y_test_pred)
  plt.figure(figsize=(5,4))
  sns.heatmap(cm_test, annot=True, fmt='d', cmap='Greens')
```

```
    plt.title(f'Test Confusion Matrix {config}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()
    # Save results
    results.append({
        'Hidden Layers': config,
        'Train Accuracy': train_acc,
        'Test Accuracy': test_acc
    })
# 7. Summary of results
results_df = pd.DataFrame(results)
print("\nSummary of all models:")
print(results_df)
```