Smoker Classification Using Machine Learning

A Comprehensive Binary Classification Project

Project Members:
Parag Piprewar, Roll No: MT2025083
Siddhesh Mahajan, Roll No: MT2025122

Supervisor:
Dr. Sushree S. Behera

Department of Computer Science
IIIT Bangalore

December 12, 2025

Abstract

This project is about predicting if a person smokes or doesn't smoke using routine health measurements like biometric and biochemical data. Knowing someone's smoking status is important for keeping people healthy, assessing health risks, and understanding how smoking affects certain biomarkers.

The dataset goes through a lot of preparation steps. First, we remove duplicates and use a method called DBSCAN to filter out unusual or inconsistent data points. Then, we look at the data closely to find important patterns and connections, especially between things like lipid levels, liver enzymes, waist size, and smoking habits.

We trained and tested several machine learning models, such as Logistic Regression, Support Vector Machine (with RBF kernel), and a Neural Network (MLP). All models used scikit-learn pipelines that included scaling and preparing the data. For the SVM model, we fine-tuned the settings using GridSearchCV, which helped improve its ability to detect smokers, especially since they are less common.

The Neural Network model had the highest accuracy at 76.78We used evaluation metrics, confusion matrices, and visualizations of feature importance to show how different biomarkers affect the models' predictions.

This project shows a full machine learning process from cleaning data, doing exploratory analysis, preparing the data, building models, and evaluating them. It gives a clear and understandable way to classify people as smokers or non-smokers.

GitHub Repository-Siddhesh Mahajan: https://github.com/Sid30814/ML
GitHub Repository-Parag Piprewar: https://github.com/ParagRider1/MLCourseProject2

Contents

1 Introduction

Smoking is a big preventable risk for many long-term health problems. Being able to predict if someone smokes based on regular health and body measurements helps with automatic checks and studying large groups of people. The aim of this project is to create machine learning tools that work well, are easy to understand, and provide useful information about hea

Objectives:

- Clean and preprocess a real-world clinical dataset.
- Remove duplicates and outliers (DBSCAN).
- Train and evaluate three ML models: Logistic Regression, SVM, MLP.
- Tune hyperparameters and compare performance.
- Interpret feature effects and analyze confusion matrices.

2 Dataset

Source: `train_dataset.csv`

- Original shape: **(38,984, 23)**
- Duplicate rows found: **5,517**
- After removing duplicates: **(33,467, 23)**
- After DBSCAN outlier filtering: **(23,864, 23)**
- Target variable: `smoking` (0 = non-smoker, 1 = smoker)

Representative features:

- Anthropometrics: height, weight, waist
- Lipids: cholesterol, triglycerides, HDL, LDL
- Liver markers: AST, ALT, GTP
- Renal markers: creatinine, urine protein
- Vital signs: systolic/diastolic BP
- Sensory tests: eyesight, hearing

3 Exploratory Data Analysis (EDA)

3.1 Class Balance

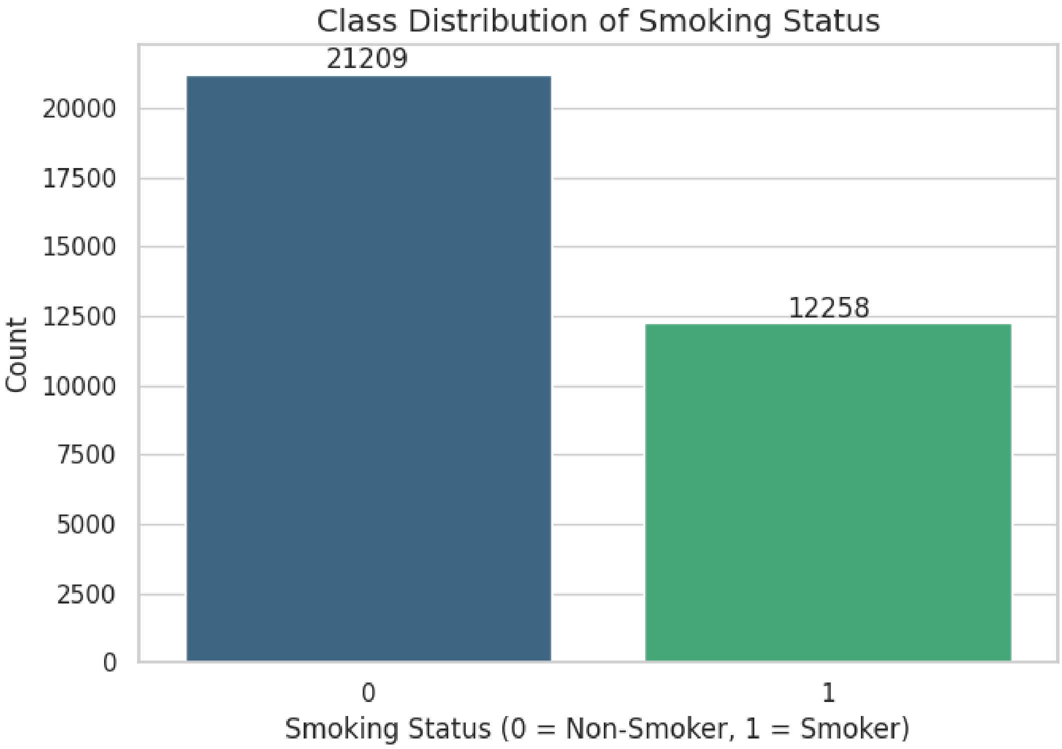

Figure 1: Class distribution: Non-smokers (0) vs smokers (1).

Figure 1 shows how the target variable `smoking` is spread out in the dataset. The data is not evenly balanced, with many more non-smokers (0) than smokers (1). After removing outliers using DBSCAN, there are about **21,209 non-smokers** and **12,258 smokers**.

This imbalance can impact how well a model learns. Most machine learning methods naturally focus on the larger group, which can make it harder to correctly identify smokers (class 1). It's important to detect smokers accurately, especially in health applications, because missing a smoker (a false negative) can make the model less useful.

3.2 Histogram Distributions of Key Features

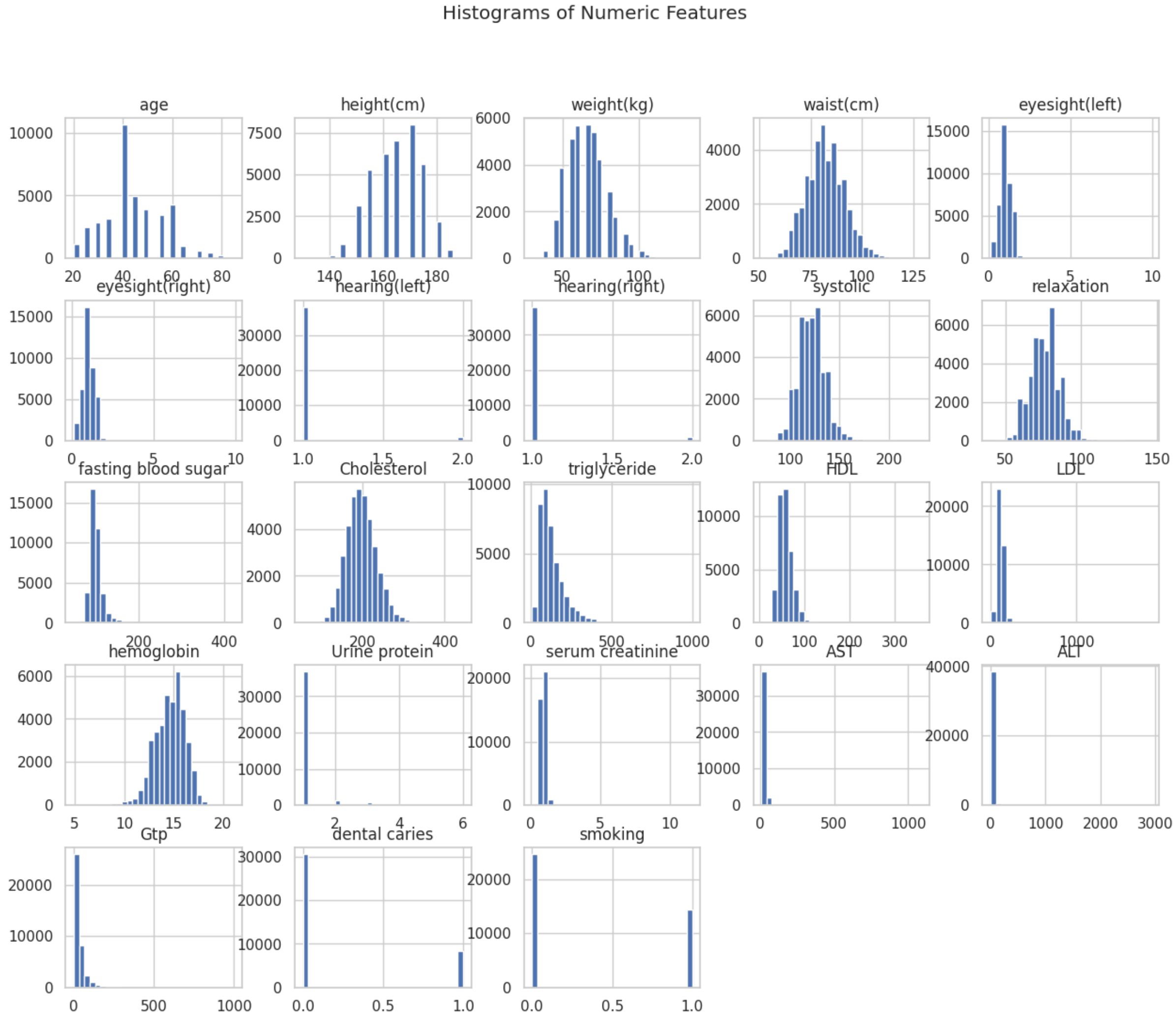

Figure 2: Histogram distributions of selected numeric features (age, height, weight, waist, triglycerides, cholesterol).

3.3 Correlation Analysis

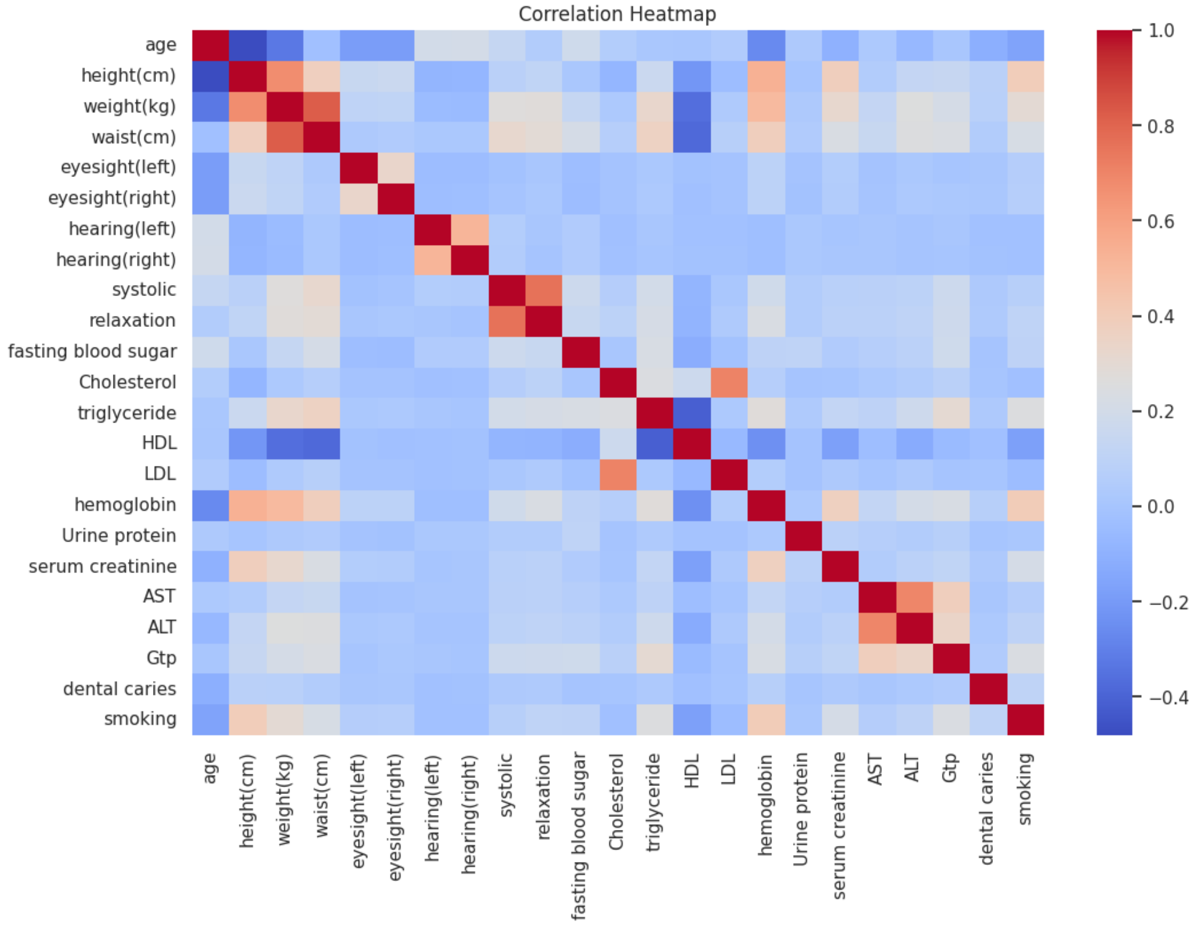

Figure 3: Correlation heatmap of numeric features.

Key inferences:

- Liver enzymes (AST, ALT, GTP) form a strong correlated group.
- HDL is negatively correlated with triglycerides.
- Smoking shows stronger correlations with triglycerides, waist, ALT, and GTP.
- Sensory measurements show weak association with smoking.

4 Preprocessing & Feature Engineering

4.1 Duplicate Removal

Removed all exact duplicate rows using:

```
df = df.drop_duplicates()
```

4.2 Outlier Removal using DBSCAN

Used DBSCAN on standardized numeric features (excluding the target):

```
scaler_db = StandardScaler()
X_scaled = scaler_db.fit_transform(df_imp[num_cols])
db = DBSCAN(eps=2.2, min_samples=15)
labels = db.fit_predict(X_scaled)
df_clean = df_imp[labels != -1].copy()
```

4.3 Scaling and Train/Test Split

```
preprocess = ColumnTransformer(
  [('scale', StandardScaler(), numeric_cols)],
  remainder='passthrough'
)
```

Stratified 80/20 split:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2,
    stratify=y, random_state=42)
```

5 Modeling

5.1 Logistic Regression

Logistic Regression is a commonly used linear model for binary classification tasks. Unlike Linear Regression, which predicts a numerical value, Logistic Regression predicts the probability that an input belongs to a specific class. It does this by applying the sigmoid function to a combination of input features. The formula for this is:

$$P(y = 1 \mid X) = \sigma(w^T X + b)$$
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where w represents the model weights, b is the bias term, and $\sigma(\cdot)$ ensures that the output lies between 0 and 1. The decision boundary is formed by thresholding this probability at 0.5.

5.1.1 Why Logistic Regression is Suitable for This Dataset

The smoker classification dataset has 23 biometric and biochemical features, many of which have a linear or nearly linear relationship with smoking behavior, such as triglycerides, HDL, and liver enzymes. Logistic Regression is suitable because it creates a linear decision boundary, which can help show where smokers and non-smokers differ. Plus, this model is easy to understand, allowing us to see how each biomarker plays a role in predicting smoking.

5.1.2 Handling Class Imbalance

The dataset exhibits a moderate imbalance between smokers and non-smokers. To address this, we use:

```
class_weight='balanced'
```

This instructs the algorithm to assign higher penalties to the misclassification of the minority class (smokers), improving recall and ensuring fairer model performance.

5.1.3 Model Configuration Used

- solver = `lbfgs` (efficient for medium-sized datasets)
- class_weight = `'balanced'`
- max_iter = 2500 (ensures convergence)

5.1.4 Application to the Smoker Dataset

After preprocessing, including removing duplicates and filtering outliers with DBSCAN and scaling the data, the Logistic Regression model was trained using a scikit-learn pipeline. It learned the differences in health indicators between smokers and non-smokers. Key observations:

- Features such as triglycerides, ALT, GTP, and waist circumference contributed positively toward predicting smokers.
- HDL showed a negative contribution (lower HDL often associates with smoking).
- The model achieved an accuracy of **0.7452** and a strong recall for smokers (**0.82**), indicating its ability to correctly identify smoking individuals.

Thus, Logistic Regression provides a strong, interpretable baseline model for the smoking classification task.

5.2 Support Vector Machine (RBF)

Support Vector Machines (SVM) are powerful supervised learning models that find the best dividing line between different classes. Unlike Logistic Regression, which uses a straight line, SVM can create more complex, non-linear boundaries using kernel functions.

The main idea is to find the widest possible space between the two classes, called the margin. A larger margin usually leads to better performance on new data.

5.2.1 RBF Kernel: Capturing Non-Linear Patterns

The Radial Basis Function (RBF) kernel transforms the input data into a higher-dimensional space where classes may become linearly separable. It is defined as:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

The hyperparameter γ controls how far the influence of a single training point extends:

- Small γ: smoother decision boundary (generalizes better)
- Large γ: tighter, more complex boundary (risk of overfitting)

Given the complexity of the biometric and biochemical smoker dataset—which contains non-linear relationships between features such as triglycerides, liver enzymes, HDL, and smoking status—the RBF kernel is appropriate and often superior to linear models.

5.2.2 Handling Class Imbalance

The smoker dataset contains more non-smokers than smokers. To avoid biasing predictions toward the majority class, we used:

```
class_weight='balanced'
```

This increases the penalty for misclassifying smokers, improving recall and ensuring a fairer model.

5.2.3 Model Configuration Used

- kernel = `rbf` (captures non-linear relationships)
- class_weight = `'balanced'`
- probability = `True` (required for probability outputs and ROC)

5.2.4 Hyperparameter Tuning

To find the optimal settings for SVM, we applied `GridSearchCV` over a range of values for C and γ. The best parameters were:

```
{'clf__C': 2, 'clf__gamma': 0.08, 'clf__kernel': 'rbf'}
```

- **C** controls the trade-off between maximizing margin and minimizing misclassification. A higher C makes the boundary tighter but may overfit.
- **gamma** influences the curvature of the decision boundary. A value of 0.08 provides a balanced level of smoothness.

5.2.5 Application to the Smoker Dataset

After DBSCAN cleaned the dataset and scaling was applied, the SVM model was trained on the standardized features. Due to its ability to capture subtle non-linear patterns:

- The untuned SVM achieved particularly high recall for smokers (0.91), meaning it detected most smokers.
- The tuned SVM improved the overall balance between precision and recall.
- The model effectively used complex biological interactions (e.g., low HDL + high triglycerides) to classify smokers.

SVM thus proved highly effective for this problem, especially when correctly tuned and combined with balanced class weights.

5.3 Neural Network (MLP)

A Multilayer Perceptron, or MLP, is a kind of artificial neural network where information moves in one direction, from input to output. It has multiple layers of neurons, and each neuron adds up inputs multiplied by weights, then applies a special function to change the result. MLPs can learn and model any complicated relationship in data, which makes them very useful for tasks like classifying if someone is a smoker based on structured medical information.

5.3.1 Architecture of the Model

The MLP used in this project consists of three hidden layers with the following configuration:

- hidden_layer_sizes = (128, 64, 32)
- activation = `relu`
- learning_rate_init = 0.001
- early_stopping = True
- max_iter = 2000

Each layer progressively extracts more abstract representations from the input features. The `ReLU` activation function introduces non-linearity and helps prevent vanishing gradients.

5.3.2 Forward Pass and Activation

Each neuron computes:

$$z = w^T x + b$$

$$a = \mathrm{ReLU}(z) = \max(0, z)$$

ReLU allows the network to learn piecewise linear patterns and significantly accelerates convergence compared to older activations such as sigmoid or tanh.

5.3.3 Learning via Backpropagation

The MLP learns by minimizing the cross-entropy loss through gradient descent. During backpropagation:

- gradients of the loss w.r.t. weights are computed layer-by-layer,
- weights are updated in the direction that minimizes classification error,
- the Adam optimizer is used to adaptively adjust learning rates during training.

5.3.4 Why MLP Works Well for This Dataset

The smoker dataset contains complex interactions between biomarkers such as triglycerides, liver enzymes, HDL, blood pressure, and anthropometric measurements. Many of these relationships are non-linear and cannot be captured effectively by simple linear models.

An MLP is well-suited for this because:

- multiple layers can learn hierarchical patterns (low-level → high-level),
- non-linear activations capture complex feature interactions,
- deep architectures model subtle trends in medical biomarker data.

5.3.5 Regularization with Early Stopping

We enabled:

```
early_stopping=True
```

This monitors validation performance during training. If the model stops improving for several epochs, training halts automatically. This prevents overfitting and ensures good generalization on unseen data.

5.3.6 Performance on the Smoker Dataset

After DBSCAN filtering and scaling, the MLP demonstrated the strongest overall performance among all models:

- Accuracy: **0.7679**
- Weighted F1: **0.77**
- Precision (smoker class): 0.65
- Recall (smoker class): 0.64

These results indicate that the neural network was able to capture deeper non-linear patterns in the data than Logistic Regression or SVM.

5.3.7 Interpretation

Although MLPs are less interpretable than linear models, their performance suggests that:

- smoking status is influenced by multiple interacting biomarkers,
- non-linear classification boundaries are necessary,
- the neural network architecture successfully modeled these interactions.

Thus, the MLP serves as the strongest predictive model in this project and highlights the importance of using non-linear deep learning approaches for complex biomedical classification tasks.

6 Comparative Analysis of Models

A comparative evaluation of the three machine learning models—Logistic Regression, Support Vector Machine (RBF), and the Multilayer Perceptron (MLP)—provides deeper insight into how each algorithm interprets the smoker classification dataset and why certain models perform better than others. The comparison is based on accuracy, precision, recall, weighted F1-score, and behavior under class imbalance.

6.1 Overall Performance Comparison

Table 1 summarizes the performance of all models:

Table 1: Comparison of model performance on the test set

Model	Accuracy	Weighted F1	Precision(1)	Recall(1)
Logistic Regression	0.7452	0.75	0.58	0.82
SVM (RBF)	0.7392	0.75	0.57	0.91
MLP Neural Network	0.7679	0.77	0.65	0.64
SVM (Tuned)	0.7469	0.75	0.58	0.86

6.2 Interpretation of Results

1. Logistic Regression: This model establishes a strong, interpretable baseline. It achieves balanced performance with high recall for the smoker class. The linear decision boundary limits its ability to capture complex biomarker interactions, but its transparency makes it valuable for clinical interpretation of feature contributions.

2. SVM (RBF): The RBF kernel allows the SVM to capture non-linear relationships between biomarkers and smoking behavior. The untuned SVM achieves the highest recall for smokers (0.91), indicating its suitability for screening, where missing a smoker is costly. However, this comes at the expense of lower precision, meaning more false positives.

After hyperparameter tuning, the SVM achieves a better balance between precision and recall, while slightly improving accuracy. This demonstrates the sensitivity of SVMs to parameter selection.

3. MLP Neural Network: The MLP outperforms all models in terms of accuracy and weighted F1-score. Its ability to learn layered non-linear representations allows it to capture subtle interactions among biochemical, anthropometric, and physiological features. While less interpretable than Logistic Regression, its superior predictive performance highlights the complexity of smoking-related patterns in the dataset.

6.3 Impact of Class Imbalance

All models used `class_weight='balanced'` to mitigate the effect of unequal class distribution.

- SVM benefited the most—leading to very high smoker recall.

- Logistic Regression improved recall without severely hurting precision.
- MLP achieved the best overall balance but showed slightly lower recall for smokers than SVM, indicating deep networks can still be influenced by imbalance.

6.4 Key Insights

- Non-linear models (SVM, MLP) outperform linear models in capturing complex biomedical relationships.
- MLP provides the most consistent performance across metrics.
- SVM is best suited when identifying smokers (class 1) is the priority due to its high recall.
- Logistic Regression remains the most interpretable model, making it ideal for clinical explanation.

6.5 Conclusion of Comparison

Each model brings unique strengths:

- **Logistic Regression** — interpretability and stable performance.
- **SVM (RBF)** — excellent sensitivity to smokers, ideal for screening.
- **MLP** — best overall predictive accuracy and ability to model complex patterns.

Therefore, the optimal model choice depends on the application's goal:

- If minimizing false negatives (missed smokers) is crucial → **SVM (RBF)**.
- If accuracy and robustness are priorities → **MLP**.
- If interpretability is essential → **Logistic Regression**.

7 Results

Table 2: Test set performance summary

Model	Accuracy	Weighted F1	Precision(1)	Recall(1)
Logistic Regression	0.7452	0.75	0.58	0.82
SVM (RBF)	0.7392	0.75	0.57	0.91
Neural Network (MLP)	0.7679	0.77	0.65	0.64
SVM (Tuned)	0.7469	0.75	0.58	0.86

7.1 Confusion Matrices

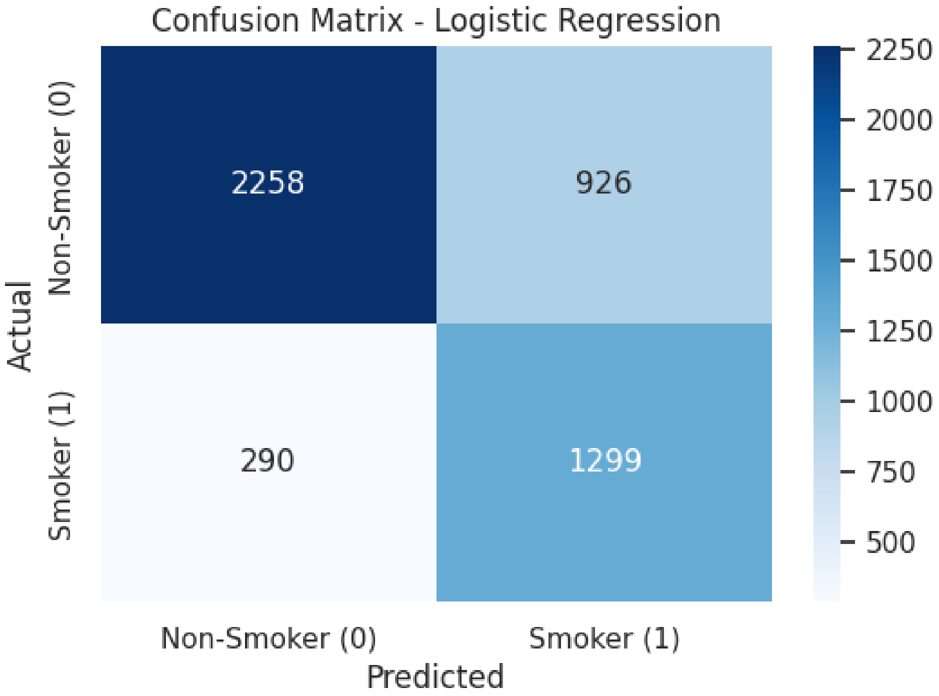

Figure 4: Confusion matrix — Logistic Regression.

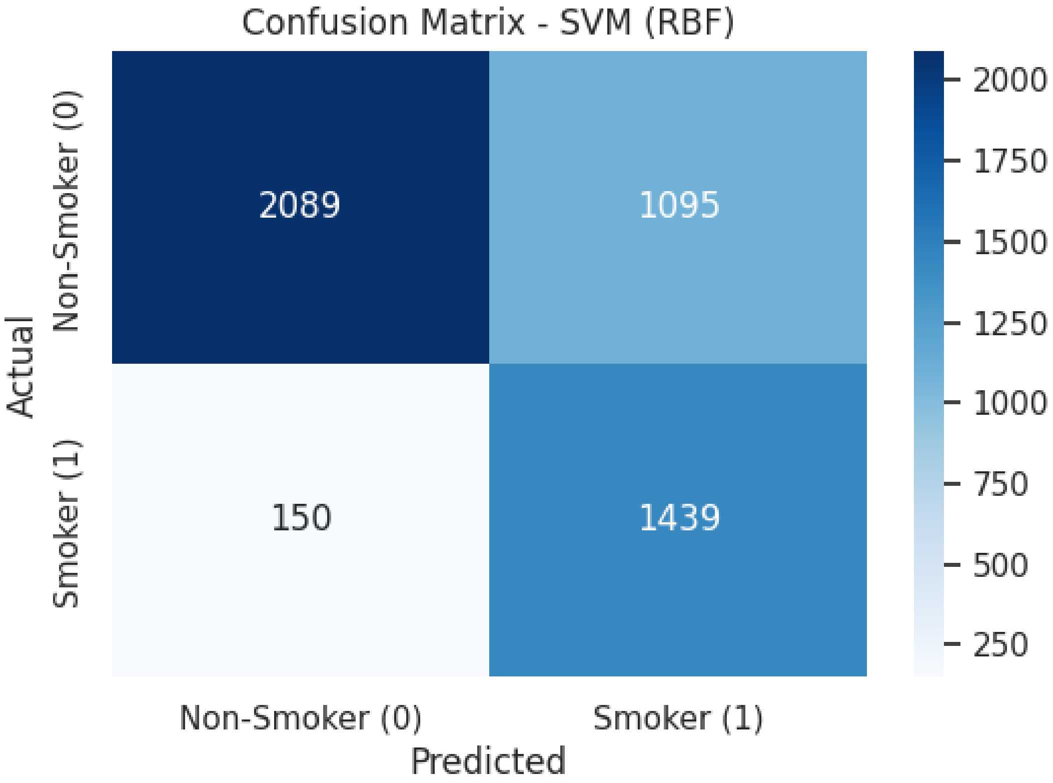

Figure 5: Confusion matrix — SVM (RBF).

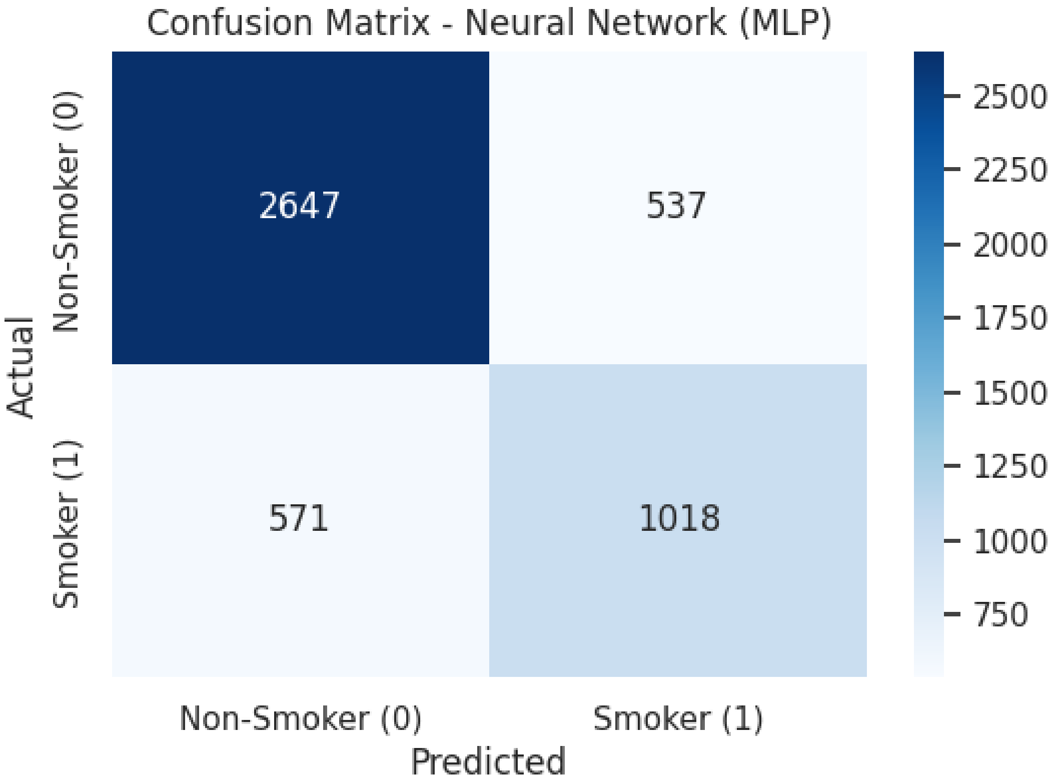

Figure 6: Confusion matrix — Neural Network (MLP).

8 Feature Importance & Interpretation

Tree-based or SHAP analysis reveals the following key predictors:

- Triglycerides (positive association with smoking)
- Waist circumference
- ALT and GTP (liver enzymes)
- HDL (negative association)
- LDL, cholesterol
- Systolic/diastolic BP

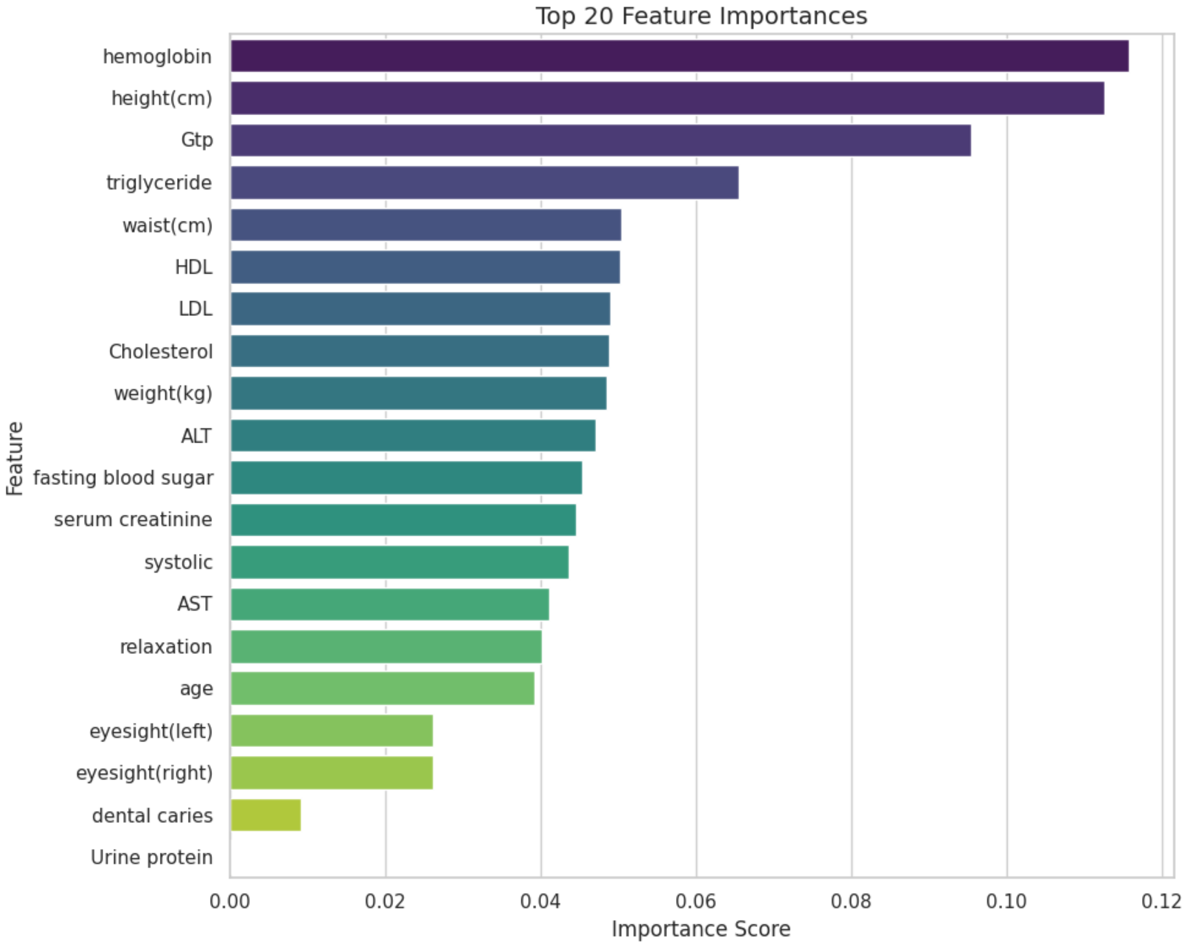

Figure 7: Feature importance visualization (Random Forest or SHAP).

9 Discussion

- DBSCAN successfully removed noisy subjects and improved model stability.
- SVM showed very high recall for smokers (important in screening contexts).
- MLP achieved the highest accuracy and F1, capturing non-linear patterns.
- Logistic Regression remains valuable for interpretability.

10 Reproducibility

1. Python 3.8+, install sklearn, pandas, seaborn, matplotlib.
2. Use database cleaning steps in order: duplicates → DBSCAN → split → scale.
3. Random seeds used: 42 (split), 37 (MLP).
4. All preprocessing done using ColumnTransformer.

A Appendix A: Key Code Snippets

A.1 DBSCAN Outlier Removal

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN

num_cols = df_imp.select_dtypes(include=['int64','float64']).columns.tolist()
if target in num_cols: num_cols.remove(target)

scaler_db = StandardScaler()
X_scaled = scaler_db.fit_transform(df_imp[num_cols])

db = DBSCAN(eps=2.2, min_samples=15)
labels = db.fit_predict(X_scaled)

df_clean = df_imp[labels != -1].copy()
```