Mehmet Fatih Göğüş

Machine Learning Logistic Regression

My accomplishments from the Lab

In this lab, I applied logistic regression to a dataset to classify individuals based on specific characteristics such as height, weight, and age. Logistic regression is a statistical model commonly used for binary classification problems where the goal is to predict one of two possible outcomes, 0 and 1. The main goal is to preprocess the dataset by manipulating categorical variables and scaling the numerical data, then split it into training and test sets for model evaluation. I trained a logistic regression model using Scikit-learn and analyzed its performance using various metrics including accuracy, confusion matrix, precision, recall, F1 score, and AUC-ROC. I also visualized key results such as confusion matrix and decision boundary to better understand the effectiveness of the model. At the end of this experiment, I aimed to evaluate how well logistic regression performed in distinguishing between different classes in the dataset.

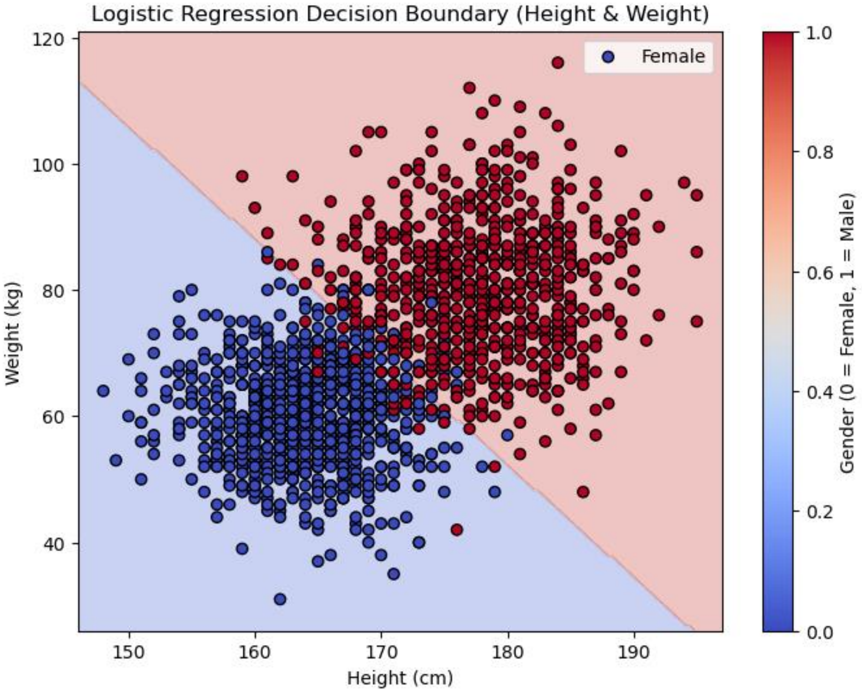


Figure 1 – Classification Graph

The decision boundary in the graph clearly separates two classes, Female (blue) and Male (red), based on Height (cm) and Weight (kg). The shaded regions indicate the model’s classification regions; blue for females, red for males. While most points were correctly classified, some misclassified points near the boundary indicate areas where the model struggled, likely due to overlapping height and weight distributions. Only “Female” is shown in the legend, but both classes are shown in different colors. Overall, the model appears to perform well, but there may be room for improvement in cases where male and female data points are closely clustered.

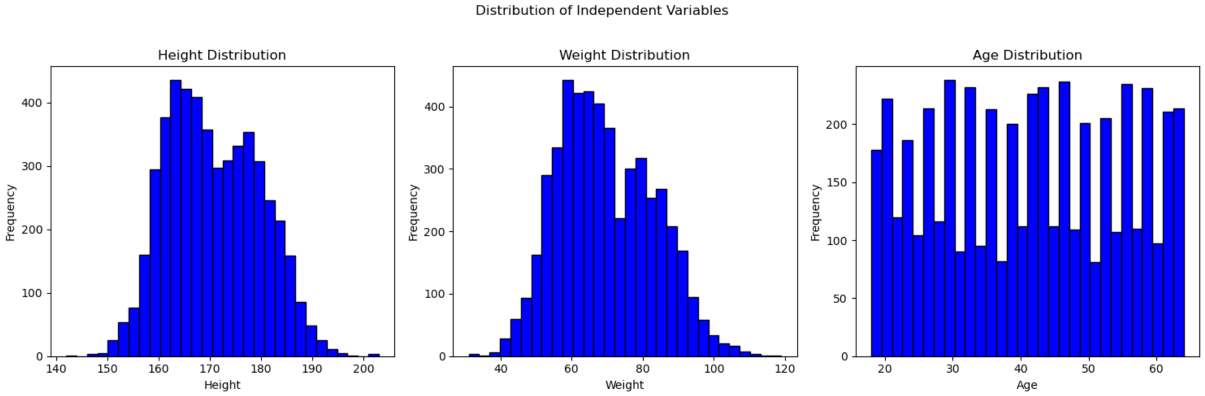


Figure 2 – Distribution of Independent Variables

The Distribution of Independent Variables plot shows that both height and weight follow a normal distribution, indicating that the dataset represents a typical population with a central tendency and few outliers. In contrast, the age distribution appears more uniform, indicating that the data may have been sampled evenly across different age groups rather than reflecting a naturally occurring distribution.

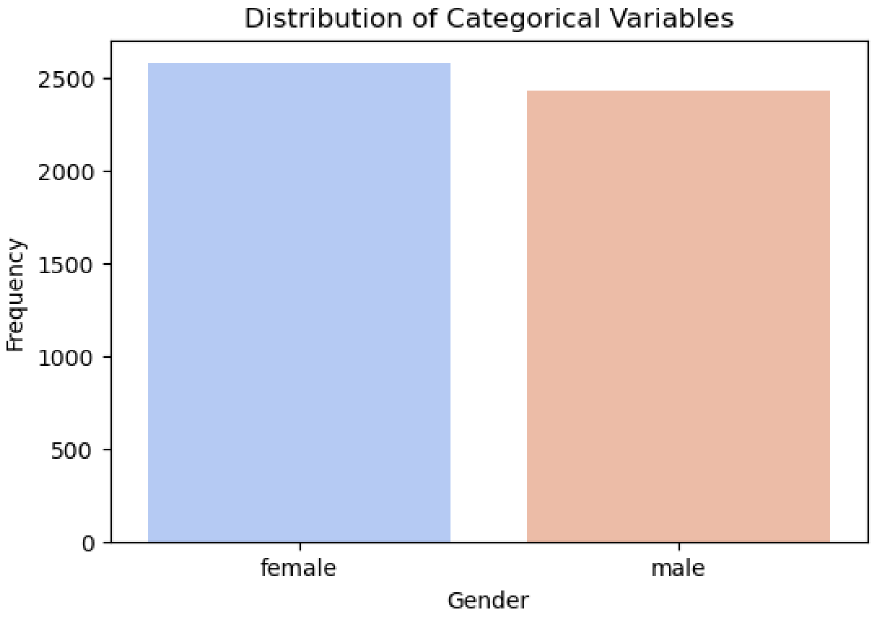


Figure 3 – Distribution of Categorical Variables

The Distribution of Categorical Variables plot shows that the dataset has an almost balanced gender distribution, with slightly more females than males. This balance is crucial for the logistic regression model because it helps prevent classification bias and ensures that predictions are not skewed toward one gender. The “test_size” parameter in the data pane (set to 0.33) ensures that 33% of the dataset is reserved for testing, allowing a fair assessment of the model’s performance while preserving enough data for effective training.

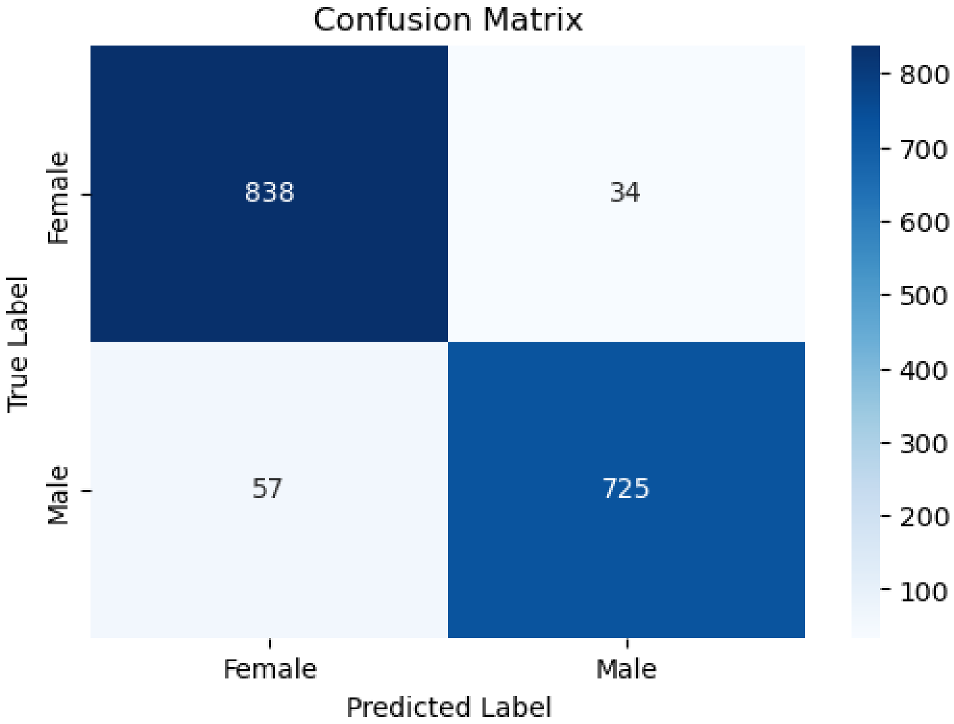
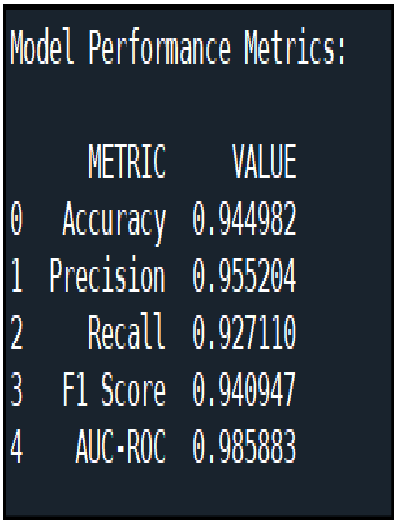


Figure 4 - Confusion Matrix

The complexity matrix shows the model's classification performance for the categories Female and Male. The true positives (correctly classified Females) are 838 and the true negatives (correctly classified Males) are 725, indicating strong accuracy overall. There are 34 false positives (Males misclassified Females) and 57 false negatives (Women misclassified Males). The model performs well but has slightly more false negatives, which can affect recall. The darker shades on the diagonal confirm that most of the predictions are correct, but some misclassifications still occur.



```
Model Performance Metrics:

      METRIC     VALUE
0   Accuracy  0.944982
1  Precision  0.955204
2     Recall  0.927110
3   F1 Score  0.940947
4    AUC-ROC  0.985883
```

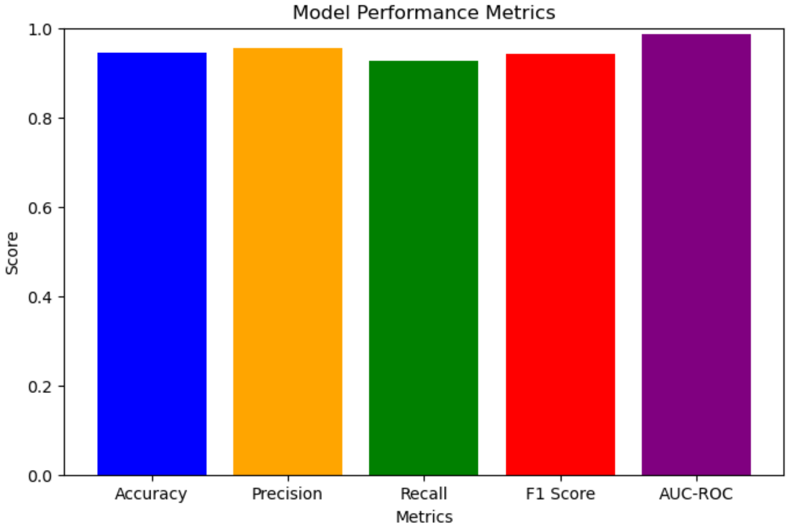


Figure 5 – Performance Metrices

Accuracy (0.9449): The model correctly classifies 94.49% of the data, which indicates a strong overall performance. However, while high accuracy is good, it is important to check precision and recall ensuring the model is not biased toward one class.

Precision (0.9552): Out of all instances predicted as Male, 95.52% were actually Male. This means that when the model predicts "Male," it is usually correct. High precision is useful when false positives need to be minimized, such as in medical or fraud detection applications.

Recall (0.9271): The model correctly identifies 92.71% of actual Male cases, meaning that it performs well in capturing most of the Male instances. However, a lower recall compared to precision suggests that some Males are misclassified as Females. If recall were too low, it could indicate an issue with false negatives.

F1 Score (0.9409): Since F1-score is the harmonic mean of precision and recall, a value of 0.9409 suggests a balanced performance. This means the model is effective at making positive predictions while also capturing most of the actual positive cases. A high F1-score is desirable when both false positives and false negatives are important.

AUC-ROC (0.9859): The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the model's ability to distinguish between classes. A score of 0.9859 is nearly perfect, meaning the model is excellent at differentiating between Male and Female cases. This suggests that the decision boundary is well-defined with minimal overlap.

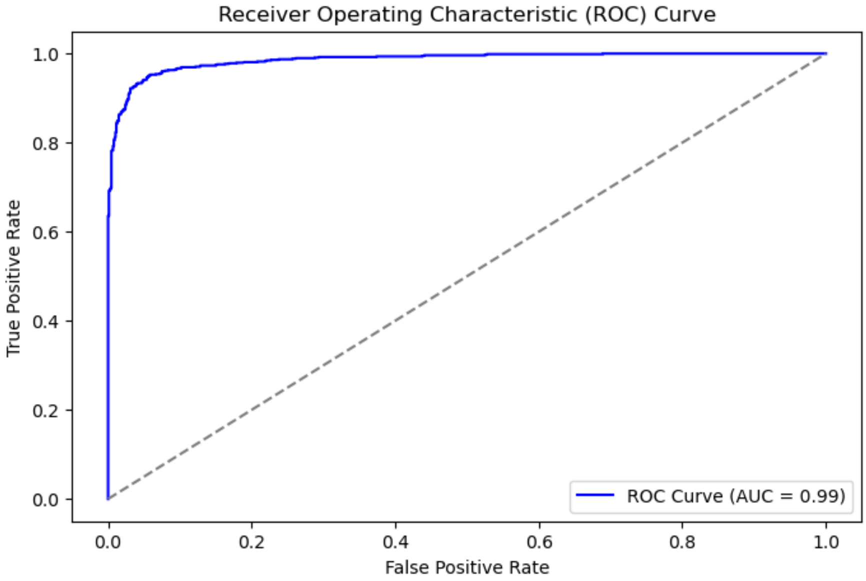
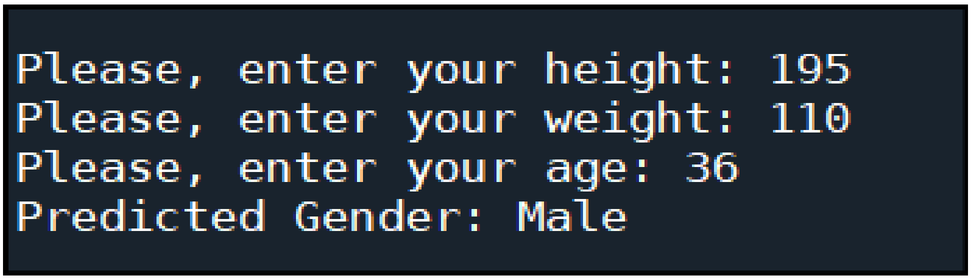


Figure 6 - ROC Curve

The ROC curve in the image shows a strong classifier performance, with the curve closely hugging the top-left corner and an AUC score of 0.99. This indicates that the logistic regression model is highly effective in distinguishing between the two classes (male and female) with minimal false positives and high true positive rates.



```
Please, enter your height: 195
Please, enter your weight: 110
Please, enter your age: 36
Predicted Gender: Male
```

Figure 7 - Realtime Prediction

The user input section in the code allows for real-time predictions based on height, weight, and age. Once the user enters their values, the model standardizes the input using the same scaling applied to the training data before making a prediction. The predicted gender is then displayed. Given the result, the model predicts “Male” for a person who is 195 cm tall, weighs 110 kg, and is 36 years old, which is consistent with the typical male characteristics in the dataset. This feature demonstrates the practical use of logistic regression in classification tasks, but its accuracy depends on how well the input matches the patterns learned during training.

Conclusion

In this lab, I successfully implemented logistic regression to classify individuals based on height, weight, and age. By preprocessing the dataset, training the model, and evaluating its performance using various metrics, I gained insights into how well logistic regression performs in binary classification. The model achieved high accuracy, precision, recall, and AUC-ROC scores, indicating strong predictive capabilities. However, some misclassifications were observed, suggesting that factors beyond these three features might influence gender classification. The decision boundary and confusion matrix provided valuable visualizations of the model’s strengths and weaknesses. Overall, this experiment reinforced my understanding of logistic regression, performance evaluation, and real-world classification challenges.

My Code:

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar  7 17:13:05 2025

@author: fatihgogus
"""


# Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score, roc_auc_score, roc_curve

# Import Data
data = pd.read_csv('GenderData.txt')
#pd.read_csv("data.csv")
print(data)

x = data.iloc[:,1:4].values # independent variables
y = data.iloc[:,4:].values # dependent variable

le = LabelEncoder()
y = le.fit_transform(data['gender'])  # Adjust the column name as needed

# To divide data for training and testing
x_train, x_test,y_train,y_test = train_test_split(x,y,test_size=0.33, random_state=0)

# Scale data
sc=StandardScaler()
X_train = sc.fit_transform(x_train)
X_test = sc.transform(x_test)

logr = LogisticRegression(random_state=0)
logr.fit(X_train,y_train)

y_pred = logr.predict(X_test)
print(y_pred)
print(y_test)

# Accuracy, Confusion matrix, Recall, Precision, F1-score, and AUC-ROC

# Model Accuracy
accuracy = accuracy_score(y_test, y_pred)

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Precision
precision = precision_score(y_test, y_pred)

# Recall (Sensitivity)
recall = recall_score(y_test, y_pred)

# F1 Score
f1 = f1_score(y_test, y_pred)

# AUC-ROC Score
auc_roc = roc_auc_score(y_test, logr.predict_proba(X_test)[:, 1])

# Create a DataFrame for displaying metrics in a table
metrics_df = pd.DataFrame({
  "Metric".upper(): ["Accuracy", "Precision", "Recall", "F1 Score", "AUC-ROC"],
  "Value".upper(): [accuracy, precision, recall, f1, auc_roc]
})

# Display the metrics table
print("\nModel Performance Metrics:\n")
print(metrics_df)

# Plot Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=["Female", "Male"], yticklabels=["Female", "Male"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.legend(["Female", "Male"], loc="upper right")
plt.show()

# Plot bar chart for model performance metrics
plt.figure(figsize=(8, 5))
plt.bar(metrics_df["METRIC"], metrics_df["VALUE"], color=['blue', 'orange', 'green', 'red', 'purple'])
plt.xlabel("Metrics")
plt.ylabel("Score")
plt.title("Model Performance Metrics")
plt.ylim(0, 1)
plt.show()

# Plot ROC Curve
fpr, tpr, _ = roc_curve(y_test, logr.predict_proba(X_test)[:, 1])
plt.figure(figsize=(8, 5))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC = {auc_roc:.2f})')
plt.plot([0, 1], [0, 1], linestyle='--', color='grey')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Receiver Operating Characteristic (ROC) Curve")
plt.legend()
plt.show()

# Categorical Variables Bar Chart
plt.figure(figsize=(6,4))
sns.countplot(x=data['gender'], palette='coolwarm')
plt.title("Distribution of Categorical Variables")
plt.xlabel("Gender")
plt.ylabel("Frequency")
plt.show()

# Independent Variables Distribution - Centered in 3 columns
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
fig.suptitle("Distribution of Independent Variables")

columns = ["Height", "Weight", "Age"]
for i, ax in enumerate(axes):
    ax.hist(data.iloc[:, i+1], bins=30, color='blue', edgecolor='black')
    ax.set_xlabel(columns[i])
    ax.set_ylabel("Frequency")
    ax.set_title(f"{columns[i]} Distribution")

plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()

# PLOT

# Retrain the logistic regression model using only height & weight (2 features)
X_2d = data.iloc[:, 1:3].values  # Selecting only height & weight (excluding age)
y = LabelEncoder().fit_transform(data.iloc[:, 4].values)  # Encoding gender

# Split data into training and testing sets
X_train_2d, X_test_2d, y_train, y_test = train_test_split(X_2d, y, test_size=0.33, random_state=0)

# Train logistic regression model
logr_2d = LogisticRegression(random_state=0)
logr_2d.fit(X_train_2d, y_train)

# Create mesh grid for decision boundary
x_min, x_max = X_test_2d[:, 0].min() - 2, X_test_2d[:, 0].max() + 2
y_min, y_max = X_test_2d[:, 1].min() - 5, X_test_2d[:, 1].max() + 5
xx, yy = np.meshgrid(np.linspace(x_min, x_max, 200), np.linspace(y_min, y_max, 200))

# Predict values across the grid
Z = logr_2d.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

# Plot decision boundary
plt.figure(figsize=(8, 6))
plt.contourf(xx, yy, Z, alpha=0.3, cmap="coolwarm")
scatter = plt.scatter(X_test_2d[:, 0], X_test_2d[:, 1], c=y_test, cmap="coolwarm", edgecolors="k", label=["Female", "Male"])
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
plt.title("Logistic Regression Decision Boundary (Height & Weight)")
plt.colorbar(scatter, label="Gender (0 = Female, 1 = Male)")
plt.legend(["Female", "Male"], loc="upper right")
plt.show()
```