HEALTH ANALYSIS

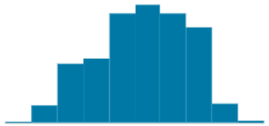
Adhyatma Chrysostomos Davu (1030123404 79)
Nazario Jose Valente Da Cruz (1030123505 52)
Anirudh Chaudhary (2024019796)
Mohammad Sahil (2024019804)

-

INTRODUCTION

In the healthcare domain, age plays a crucial role in determining a patient's risk factors, medical history, and treatment plans. Accurate age prediction based on simple health indicators can support early diagnosis and help tailor personalized healthcare strategies. With the rise of electronic health records, machine learning offers a promising approach to leverage available data for predictive insights

DATA SET

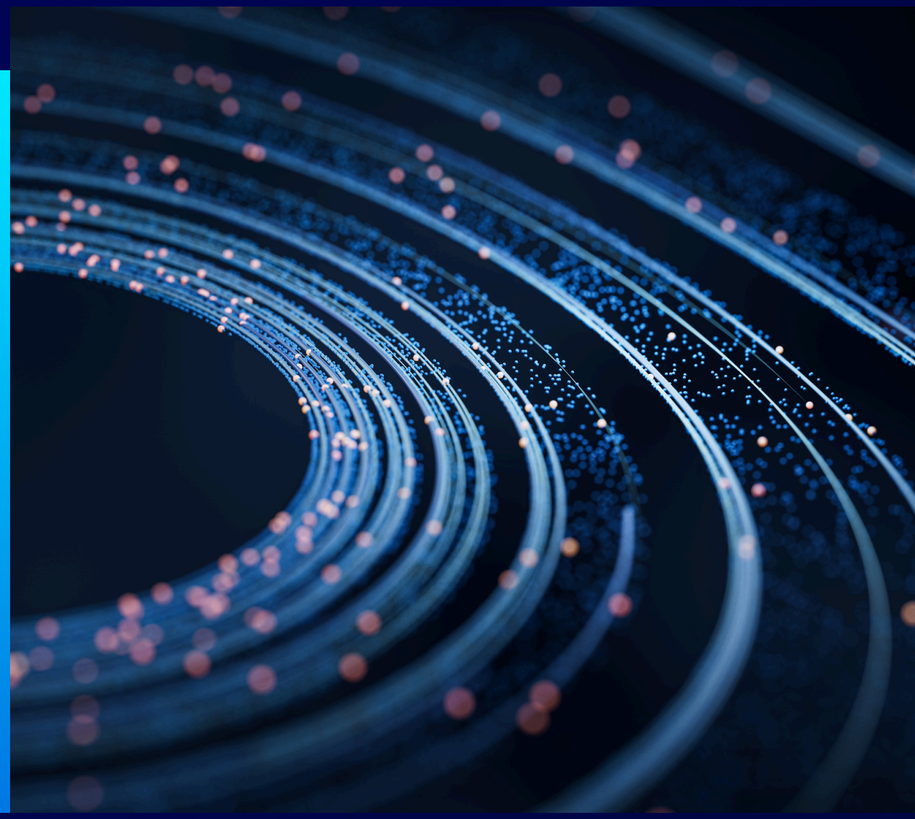
# age Integer	gender String	dementia String	dementia_all String	diabetes String	hypertension String
 39 90	female 52% male 48%	 true 81 4% false 1715 94% [null] 34 2%	 true 115 6% false 1715 94%	 true 224 12% false 1606 88%	 true 1232 67% false 598 33%
54	male	no	no	no	No
70	male	no	no	no	Yes
58	female	no	no	no	No
58	male	no	no	no	Yes

PROBLEM BACKGROUND

This study explores the potential of predicting a patient's age using basic health attributes, including:

- Gender
- Diabetes status
- Hypertension status

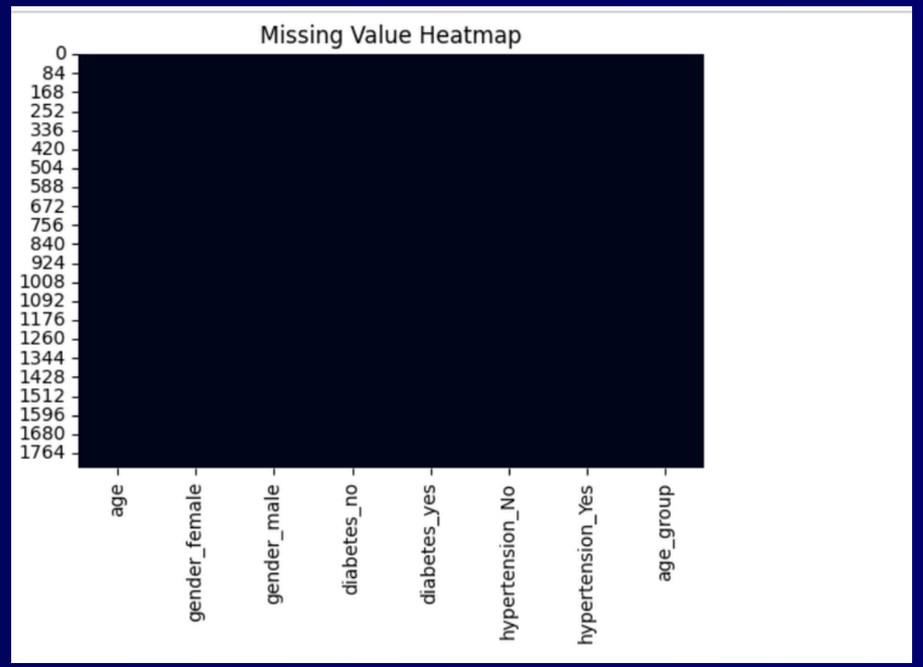
Although these features are commonly available and easy to collect, it remains a challenge to determine whether they carry enough predictive power for estimating age. We apply a linear regression model as a baseline and evaluate its performance using standard metrics and visual analysis.



DATA DATA PROCESSING & EVALUATION

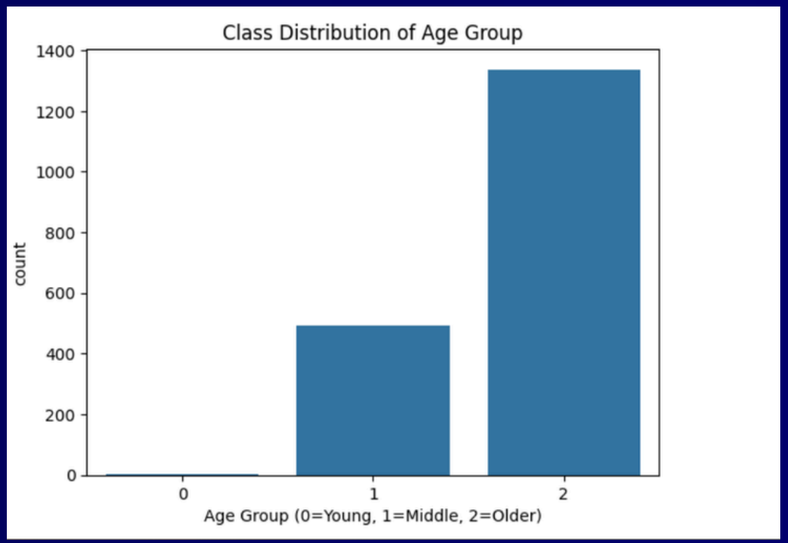
DATA LOADING & CLEANING

- Loaded health dataset from main.csv.
- Selected 4 core features: age, gender, diabetes, hypertension.
- Checked for missing values and visualized them using a heatmap.
- Removed rows with missing values to ensure clean data.

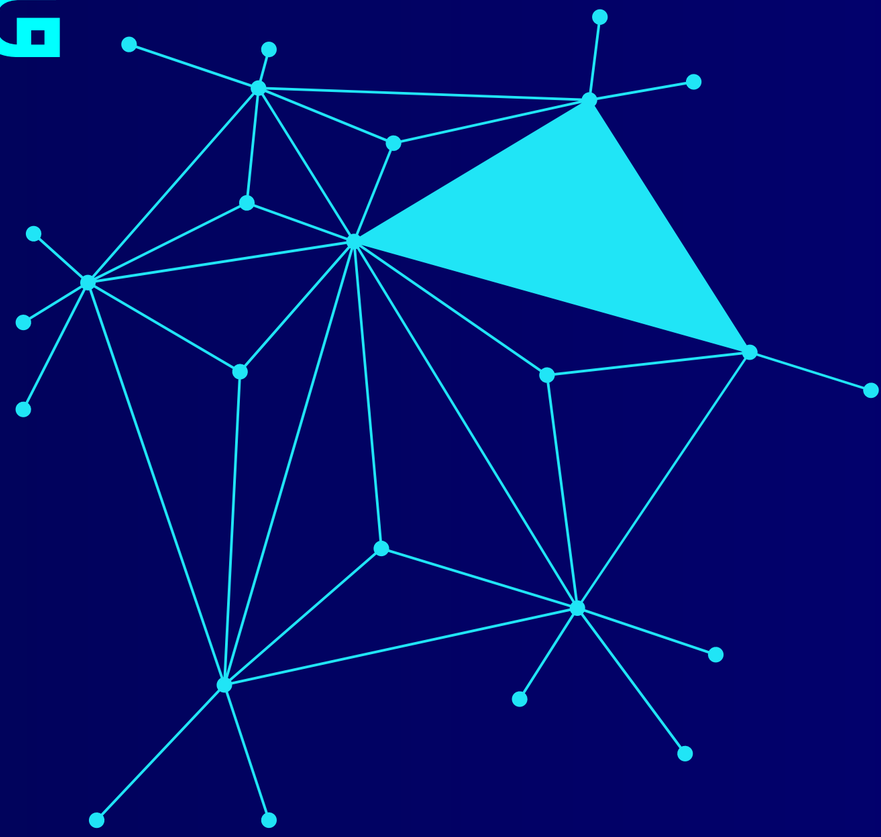


FEATURE ENCODING & TARGET TRANSFORMATION

- Applied One-Hot Encoding to categorical features (gender, diabetes, hypertension).
- Created a new target variable age_group:
  - 0 = age ≤ 40 (young)
  - 1 = age 41–60 (middle-aged)
  - 2 = age > 60 (older)
- Converted the task from regression to multiclass classification.



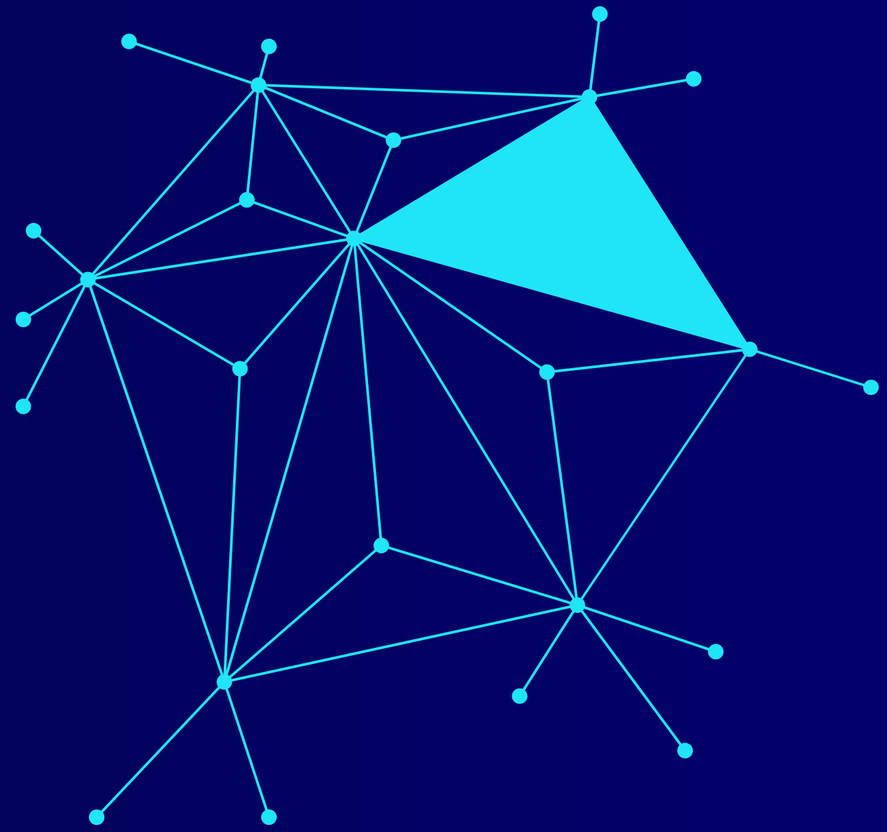
FEATURE SPLITTING & SCALING



- Defined:
  - X = input features (all except age and age_group)
  - y = target (age_group)
- Split data into training and test sets (80% training, 20% testing) with stratified sampling.
- Applied StandardScaler to normalize features.

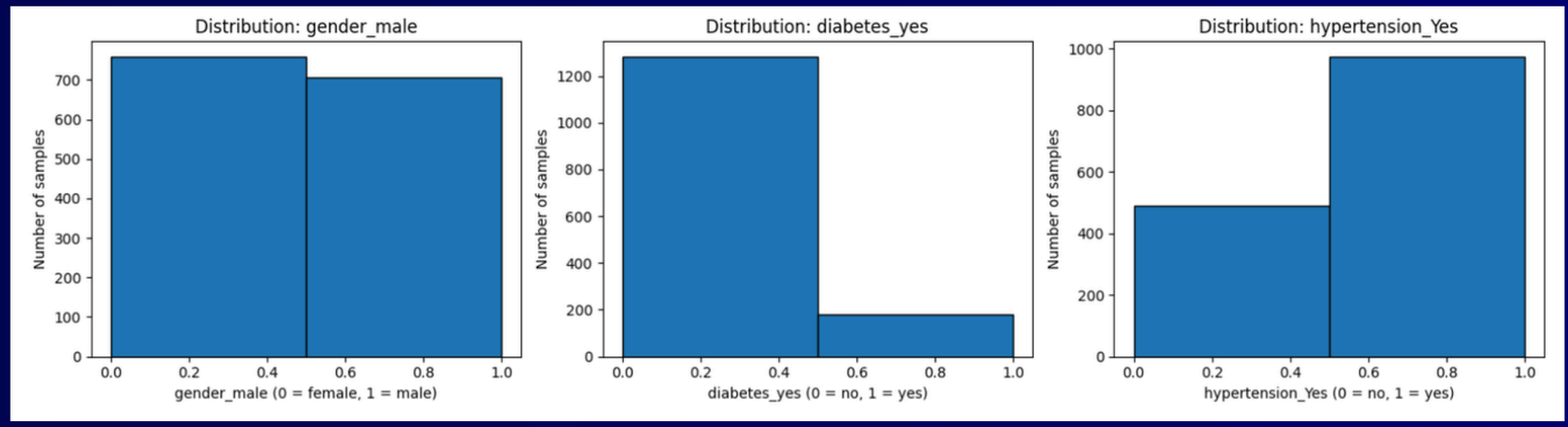
CLASS DISTRIBUTION ANALYSIS

- Visualized class distribution of age_group.
- Most samples fall into the older group (age > 60).
- Class imbalance detected may affect model performance on minority age groups.



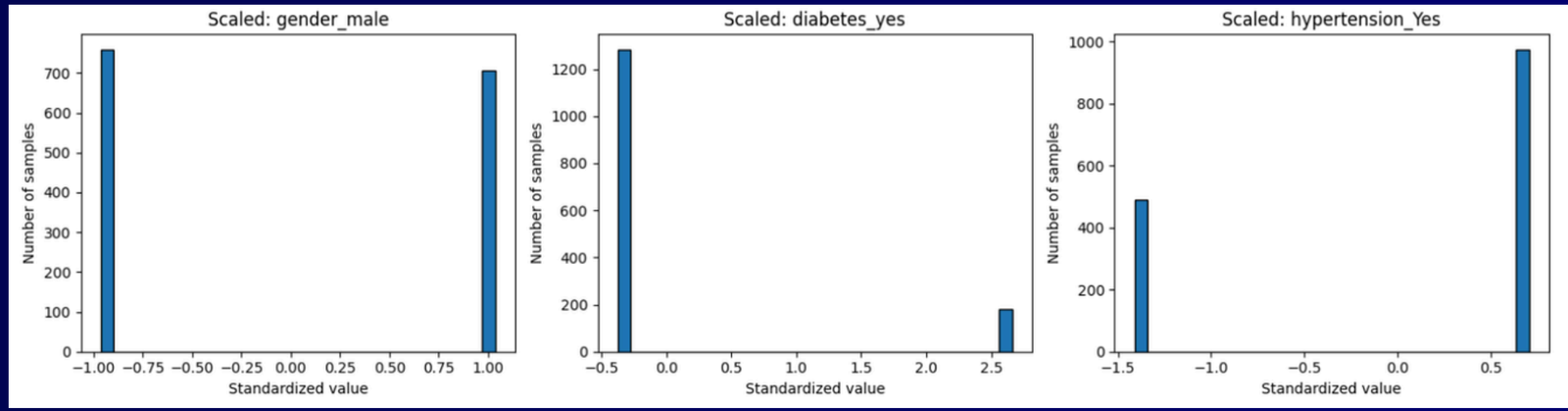
FEATURE DISTRIBUTION (BEFORE SCALING)

- Plotted histogram of raw binary features:
  - gender_male is well-balanced.
  - diabetes_yes is highly imbalanced (few diabetic cases).
  - hypertension_Yes shows mild imbalance toward hypertensive patients.



FEATURE DISTRIBUTION (AFTER SCALING)

- Plotted histograms of scaled features.
- All features centered around 0 with similar variance.
- Ensures fair contribution of each feature to machine learning models.

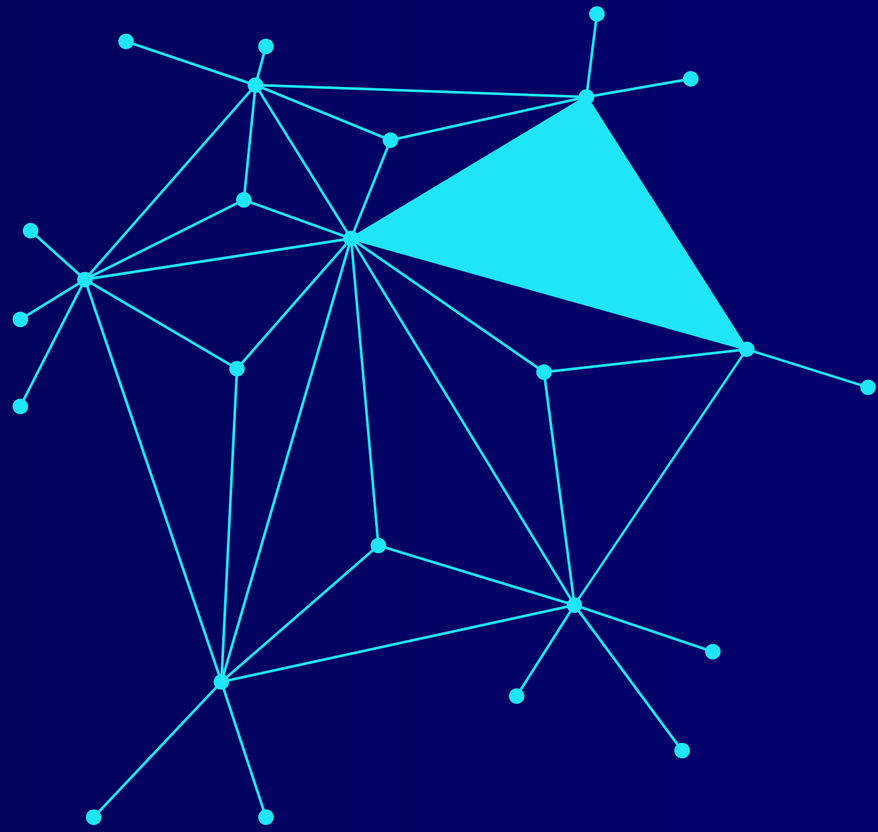


NAIVE BAYES CLASSIFICATION

MODEL TYPES EVALUATED

Two types of Naive Bayes classifiers were implemented:

- Gaussian Naive Bayes (GNB): Assumes features follow a normal distribution.
- Categorical Naive Bayes (CNB): Designed for discrete or binary features.



GAUSSIAN NAIVE BAYES (GNB)

- trained using scaled features (X_train_scaled).
- Made predictions on X_test_scaled.
- Evaluated using accuracy score and classification report.
- Confusion matrix was visualized to assess prediction errors.

```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(X_train_scaled, y_train)
y_pred_gnb = gnb.predict(X_test_scaled)
```

CATEGORICAL NAIVE BAYES (CNB)

- rained using raw (unscaled) binary features (X_train).
- Predictions were made on X_test.
- Evaluated using same metrics: accuracy and classification report.
- Confusion matrix was visualized for comparison.

```
from sklearn.naive_bayes import CategoricalNB

cnb = CategoricalNB()
cnb.fit(X_train, y_train)
y_pred_cnb = cnb.predict(X_test)
```

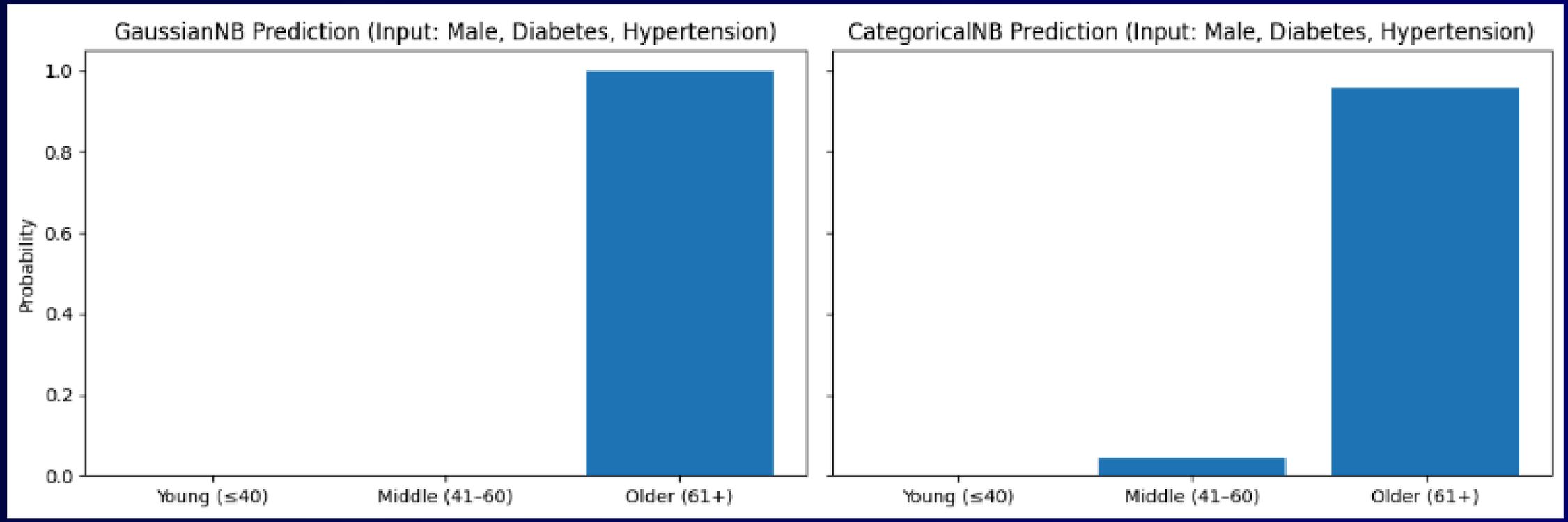
EVALUATION METRICS

- Accuracy scores were printed for both models.
- Classification reports included:
  - Precision
  - Recall
  - F1-score for each class (age group 0, 1, 2)
- Confusion matrices were plotted side-by-side using heatmaps.

```
GaussianNB Accuracy: 0.36065573770491804
GaussianNB Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.48      0.21      0.30        98
           2       0.79      0.41      0.54       268

    accuracy                           0.36       366
   macro avg       0.42      0.21      0.28       366
weighted avg       0.70      0.36      0.48       366

CategoricalNB Accuracy: 0.6939890710382514
CategoricalNB Classification Report:
              precision    recall  f1-score   support

           1       0.43      0.45      0.44        98
           2       0.80      0.78      0.79       268

    accuracy                           0.69       366
   macro avg       0.61      0.62      0.61       366
weighted avg       0.70      0.69      0.70       366
```

EXAMPLE PREDICTION

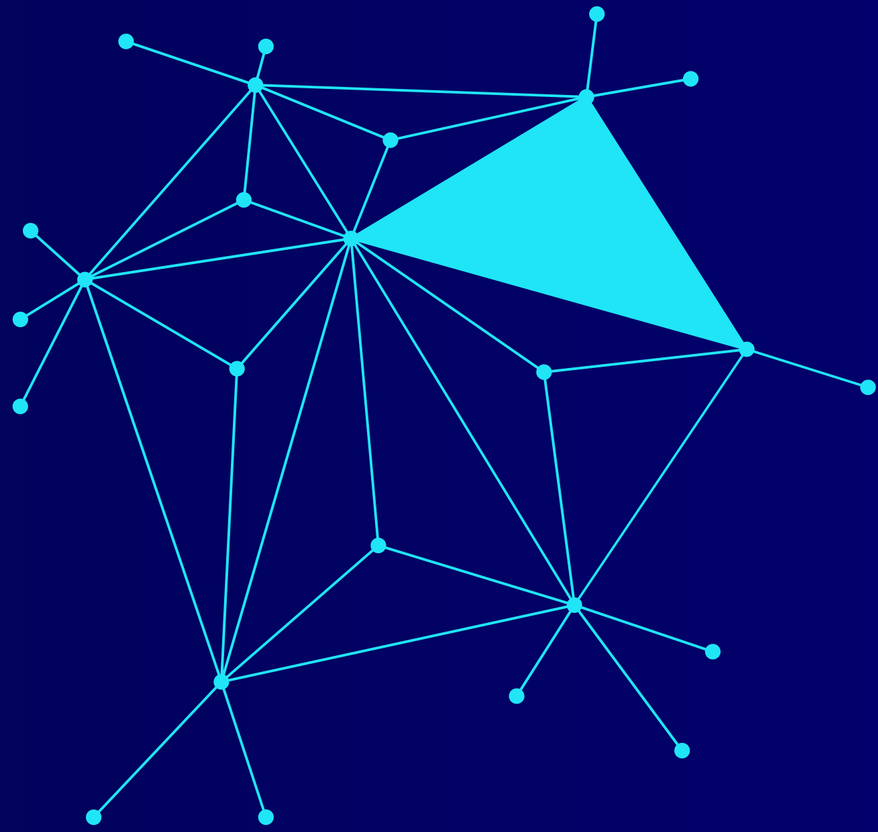
- Created a sample input: Male, Diabetic, Hypertensive.
- Predicted class probabilities using both models.
- Helps understand how each model handles real-world input.



DECISION TREE

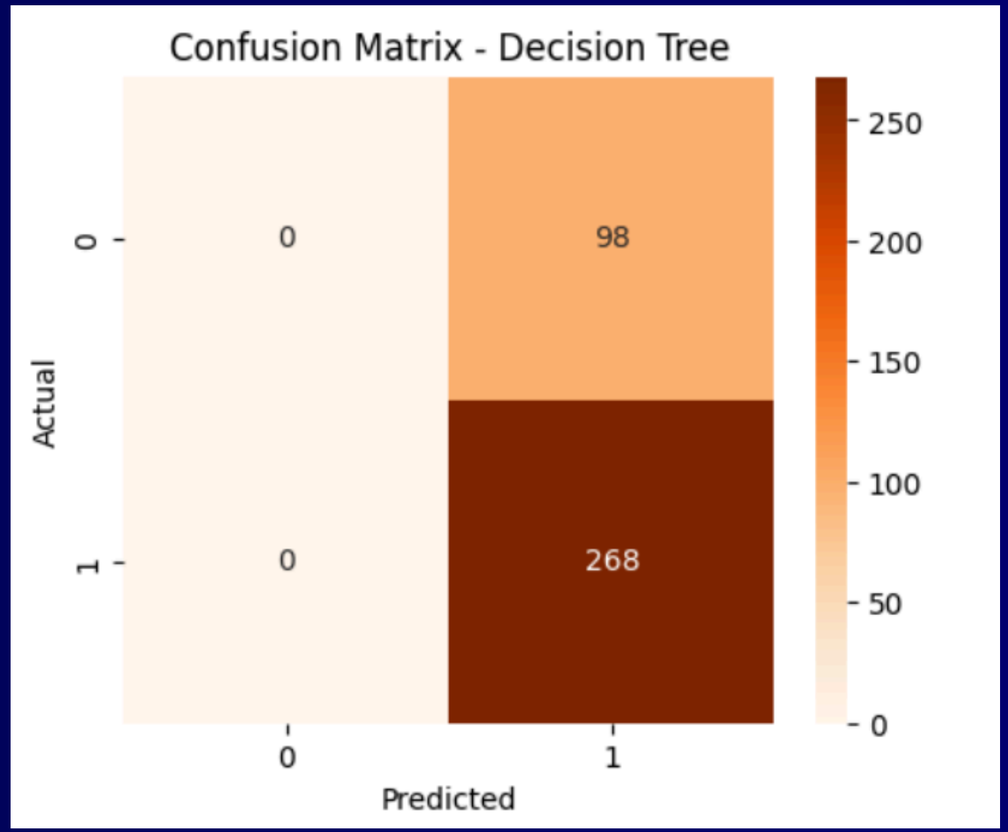
MODEL SETUP

- Used DecisionTreeClassifier with:
  - max_depth = 4 → limits the depth to avoid overfitting.
  - criterion = 'entropy' → uses information gain for splitting.
- Trained using patient features to predict age group (0 = Young, 1 = Middle, 2 = Older).



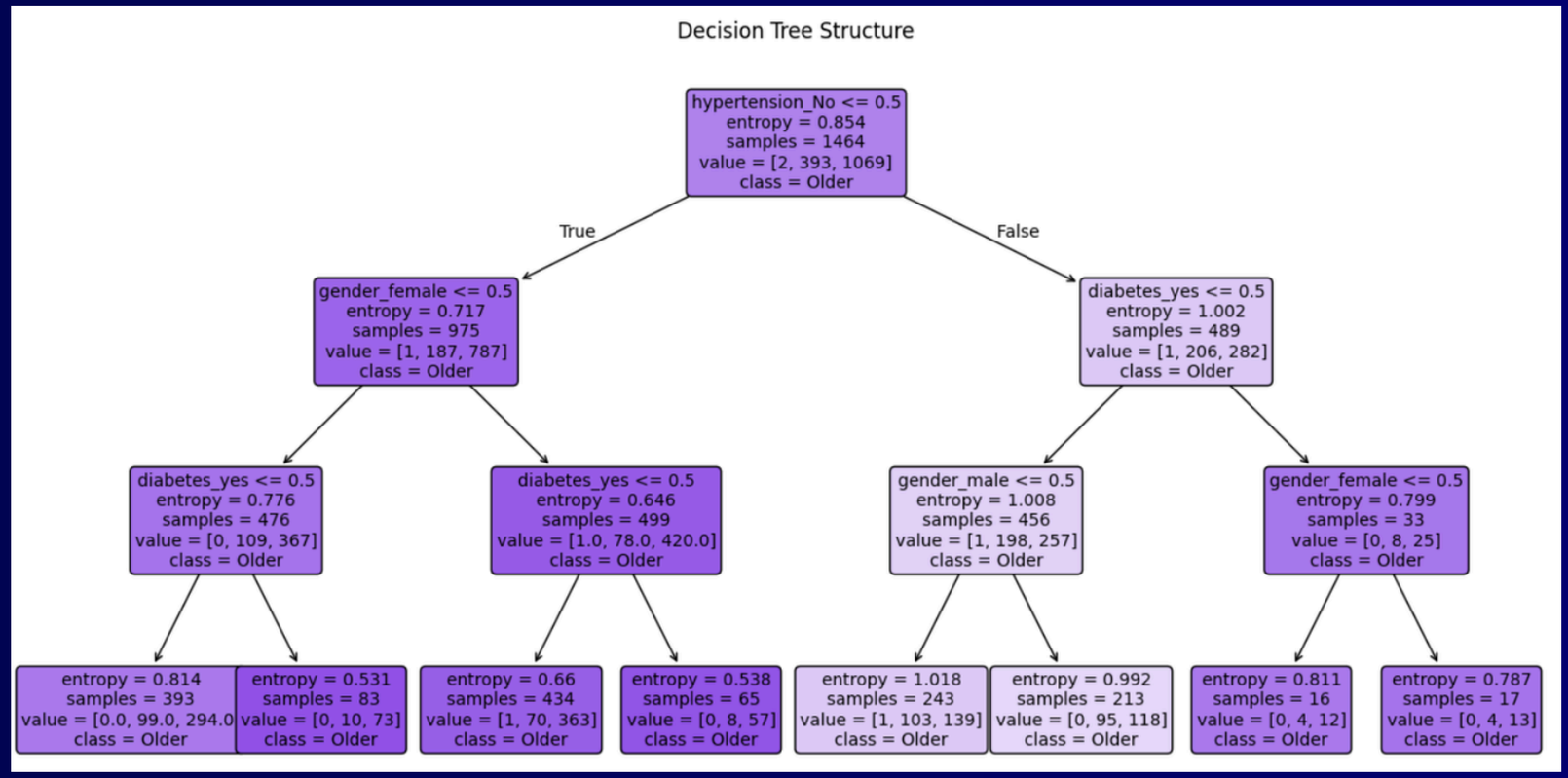
EVALUATION RESULTS

- Model evaluated using:
  - Accuracy score
  - Classification report (precision, recall, F1-score)
  - Confusion matrix (visualized as heatmap)
- Observation from confusion matrix:
  - The model only predicted class 1 (Middle).
  - No predictions for class 0 or 2 → shows bias or limitation in classification.



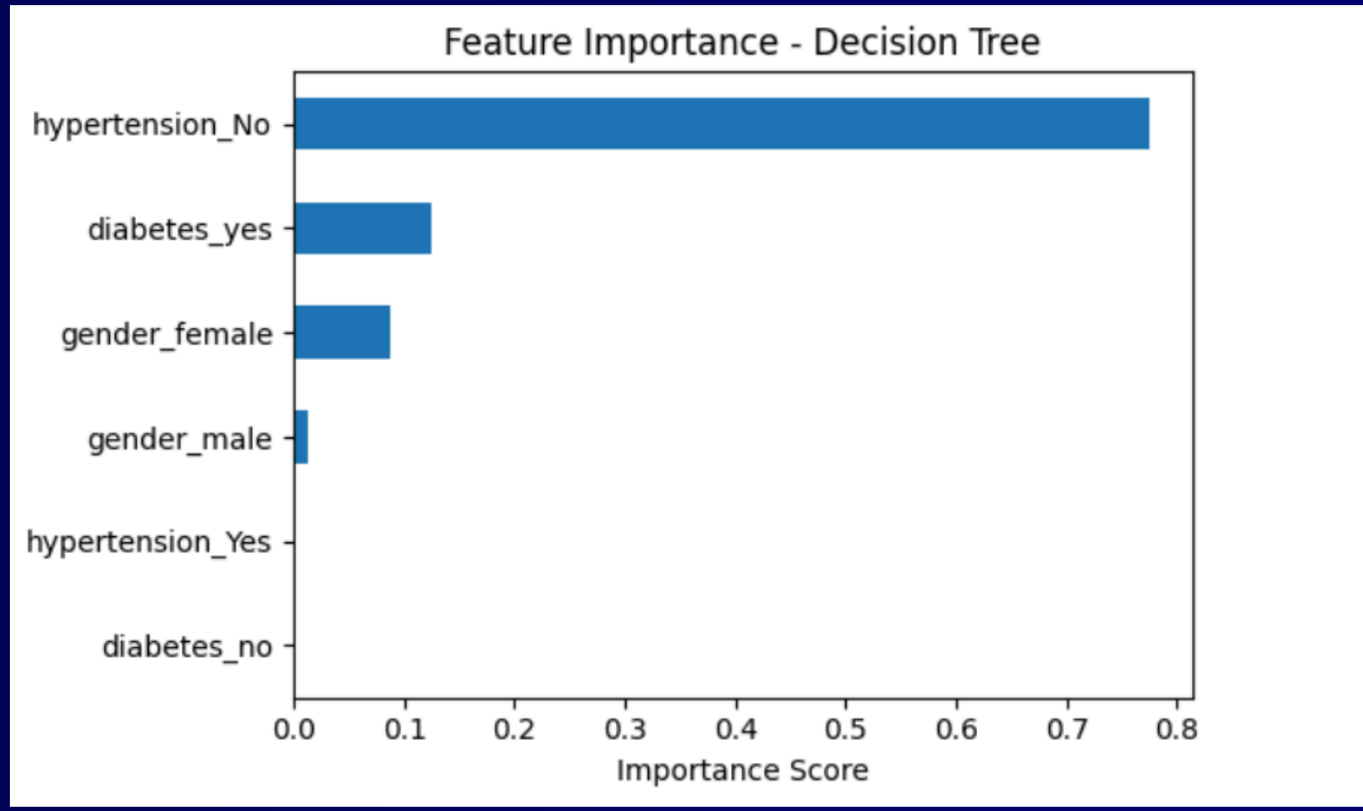
TREE STRUCTURE VISUALIZATION

- Tree visualized using plot_tree().
- Each node shows:
  - Splitting feature and threshold (e.g., hypertension_No ≤ 0.5)
  - Class distribution at that node
  - Entropy (measure of impurity)



FEATURE IMPORTANCE

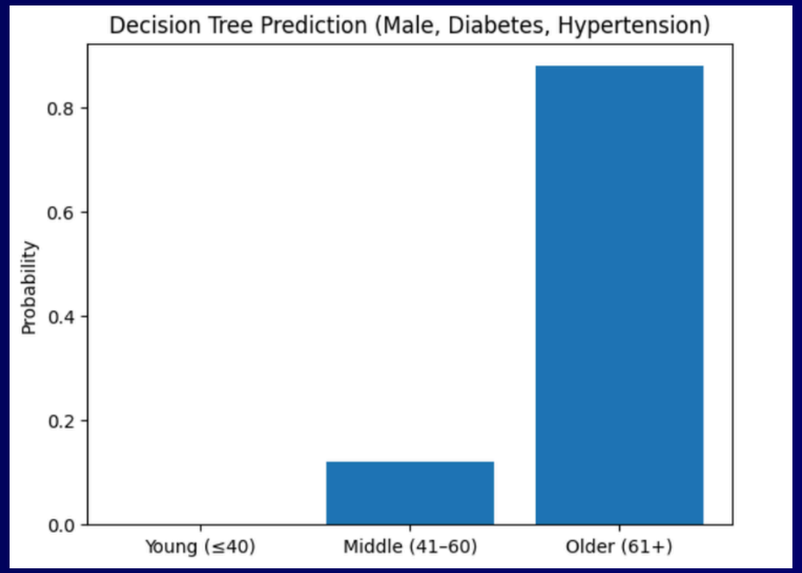
- Calculated importance of each feature in the tree.
- Displayed using a horizontal bar chart.
- Most important features contribute more to splitting decisions.



CASE PREDICTION (MALE, DIABETES, HYPERTENSION)

- A test case was predicted with:
  - gender_male = 1
  - diabetes_yes = 1
  - hypertension_Yes = 1
- Prediction result:
  - 87.95% probability of being Older (age > 60)
  - Very low probability for Young or Middle
  - Visualized using a bar chart of predicted probabilities.

```
Young (≤40): 0.00%
Middle (41-60): 12.05%
Older (61+): 87.95%
```



K-NEAREST NEIGHBORS (KNN)

KNN MODEL OVERVIEW

- Implemented using scikit-learn's KNeighborsClassifier.
- Default number of neighbors: k = 5.
- Trained on scaled features (X_train_scaled).
- Predicts age group classification (0 = Young, 1 = Middle, 2 = Older).

MODEL EVALUATION

- Accuracy and classification report printed.
- Metrics include: precision, recall, F1-score per class.
- Confusion matrix visualized using heatmap.

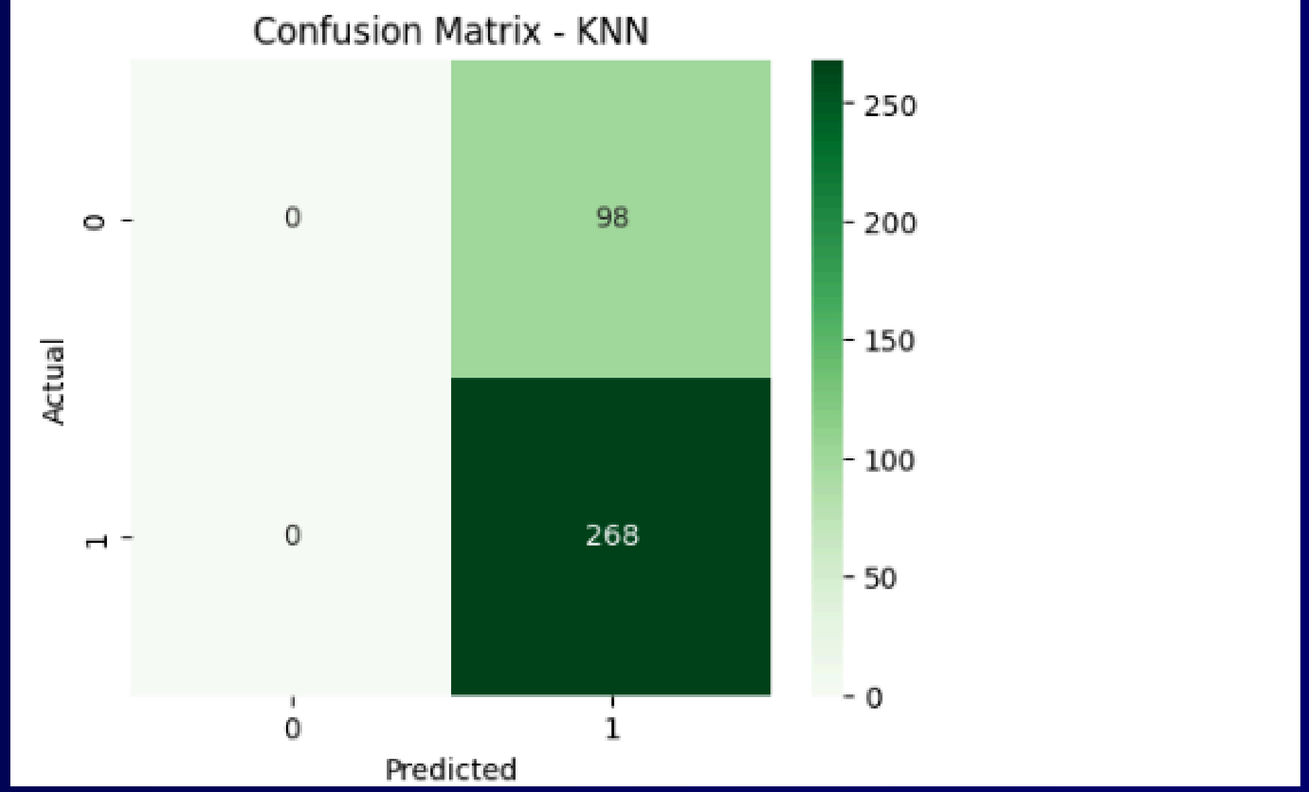
```python
# Step 2: Evaluate the Modelf
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

print("KNN Accuracy:", accuracy_score(y_test, y_pred_knn))
print("KNN Classification Report:")
print(classification_report(y_test, y_pred_knn, zero_division=0))
```

```
KNN Accuracy: 0.73224043715847
KNN Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00        98
           2       0.73      1.00      0.85       268

    accuracy                           0.73       366
   macro avg       0.37      0.50      0.42       366
weighted avg       0.54      0.73      0.62       366
```

MODEL EVALUATION

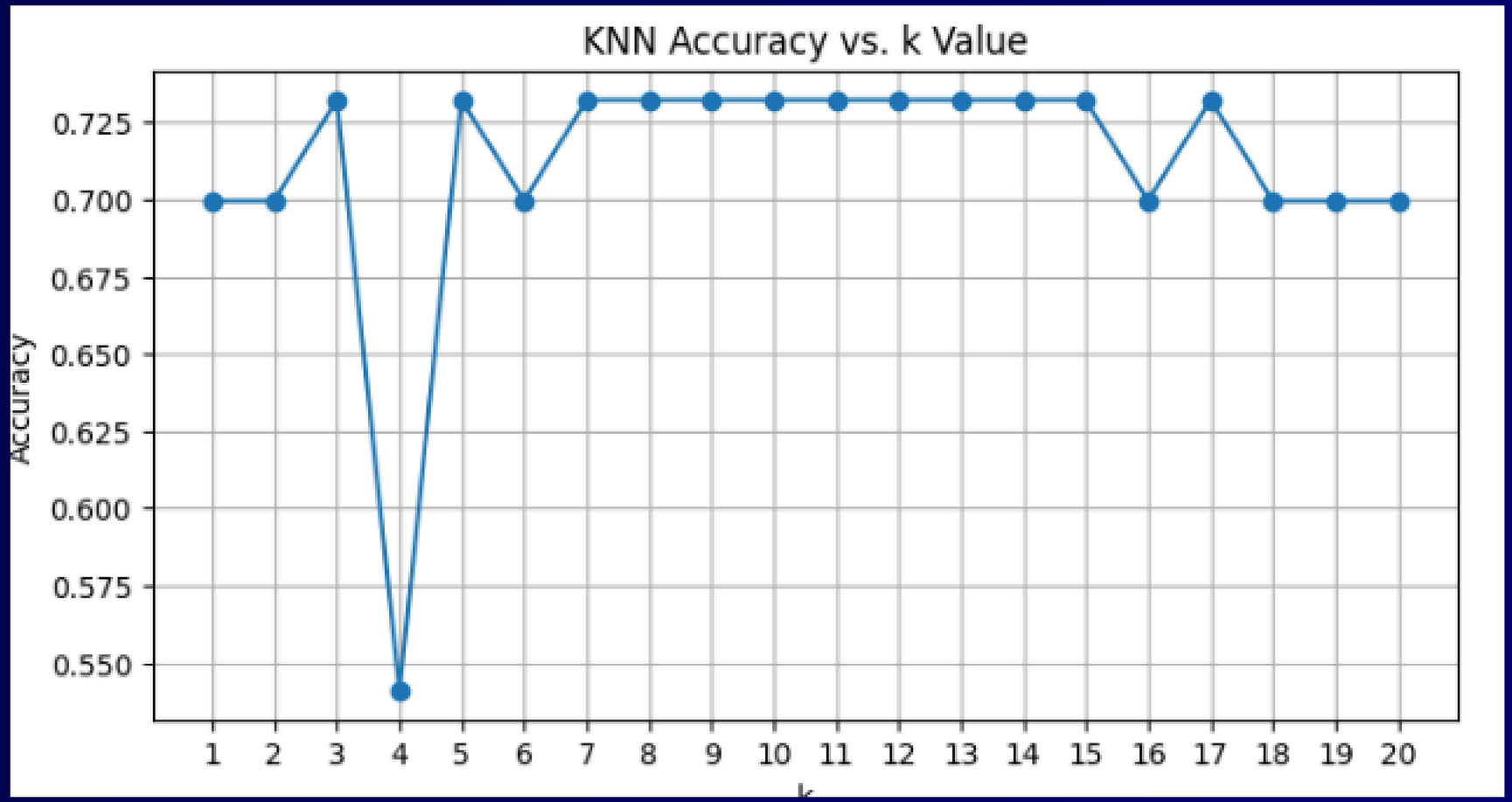
- Accuracy and classification report printed.
- Metrics include: precision, recall, F1-score per class.
- Confusion matrix visualized using heatmap.

```python
# Step 3: Visualize Confusion Matrix

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(5, 4))
sns.heatmap(confusion_matrix(y_test, y_pred_knn), annot=True, fmt='d', cmap='Greens')
plt.title("Confusion Matrix - KNN")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



TUNING K VALUES

- Tested k values from 1 to 20.
- Accuracy scores plotted for each k.
- Helps identify the optimal k for best performance.



REAL-WORLD PREDICTION EXAMPLE

- Sample input: Male, Diabetic, Hypertensive
- Preprocessed and scaled using same pipeline.
- Model predicts class probabilities across all age groups.
- Bar chart displayed for visual interpretation of prediction confidence.

```
# Male, diabetes, hypertension (same input example)
example = pd.DataFrame([[1, 1, 1]], columns=['gender_male', 'diabetes_yes', 'hypertension_Yes'])
for col in X.columns:
    if col not in example.columns:
        example[col] = 0
example = example[X.columns]
example_scaled = scaler.transform(example)

proba_knn = knn_model.predict_proba(example_scaled)[0]
```

SUMMARY OF KNN BEHAVIOR

- KNN is a distance-based classifier, heavily affected by scaling.
- Performance varies based on k-value and class imbalance.
- Can provide intuitive probability outputs for user-friendly interpretation.

Model Comparison: KNN vs Naive Bayes vs Decision Tree

Model	Feature Type	Accuracy (Test Set)
KNN (k=5)	Scaled	0.7322
GaussianNB	Scaled	0.3607
CategoricalNB	Binary (Unscaled)	0.6940
Decision Tree	Binary (Unscaled)	0.7322

- All models aim to classify patients into:
- Age Group 0 (Young ≤ 40)
- Age Group 1 (Middle 41–60)
- Age Group 2 (Older > 60)

CONCLUSION

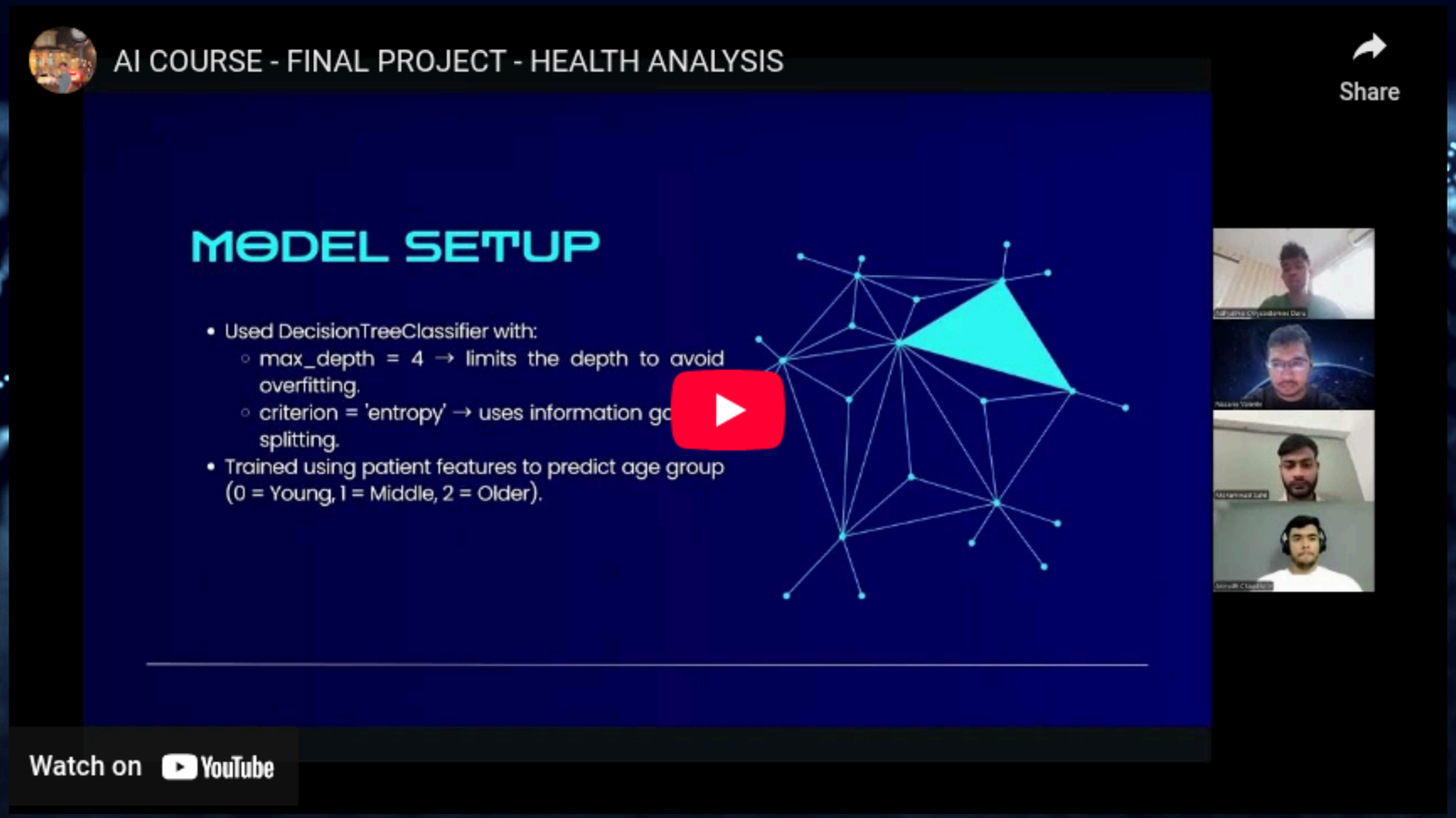
- KNN and Decision Tree achieved the best accuracy (73.22%)
- Categorical Naive Bayes performed fairly well (69.40%)
- Gaussian Naive Bayes had the lowest accuracy (36.07%)
- KNN & DT are recommended due to strong performance on this dataset
- Future work: add more features, handle class imbalance, explore ensemble models

DATASET AND VIDEO PRESENTATION

TotalBrainHealthDataset

Exploring the Interplay Between Lifestyle Factors and Neurological Health .

k kaggle.com




THANK YOU