Milestone 3 Report

Evaluation, Interpretation, Tool Development, and Presentation

1. Project Objective

The aim of this project was to develop an interactive tool that helps users understand their heart disease risk based on both lifestyle habits and clinical data. By combining daily factors like sleep, exercise, and stress with health metrics such as blood pressure and cholesterol, the tool provides personalized insights and encourages healthier choices. Using machine learning, we created a system that predicts risk and highlights areas where users can improve their heart health.

Tech Stack

Programming Language

- Python

Libraries & Frameworks:

- **Data Manipulation:** Pandas, NumPy
- **Data Visualization:** Matplotlib, Seaborn
- **Machine Learning:** scikit-learn
- **Web App Interface:** Streamlit
- **Model Serialization:** joblib

Tools:

- Jupyter Notebook / VS Code (for development)
- Streamlit (for dashboard deployment)
- GitHub (for version control and code sharing)

Dataset: The unified dataset was constructed by integrating three public datasets from Kaggle: a sleep & lifestyle dataset, a heart disease dataset, and a gym exercise dataset. These were merged based on shared attributes such as **age**, **gender**, and **heart rate** to create a comprehensive feature set that supports robust modeling and real-time dashboard development.

2. EDA Recap

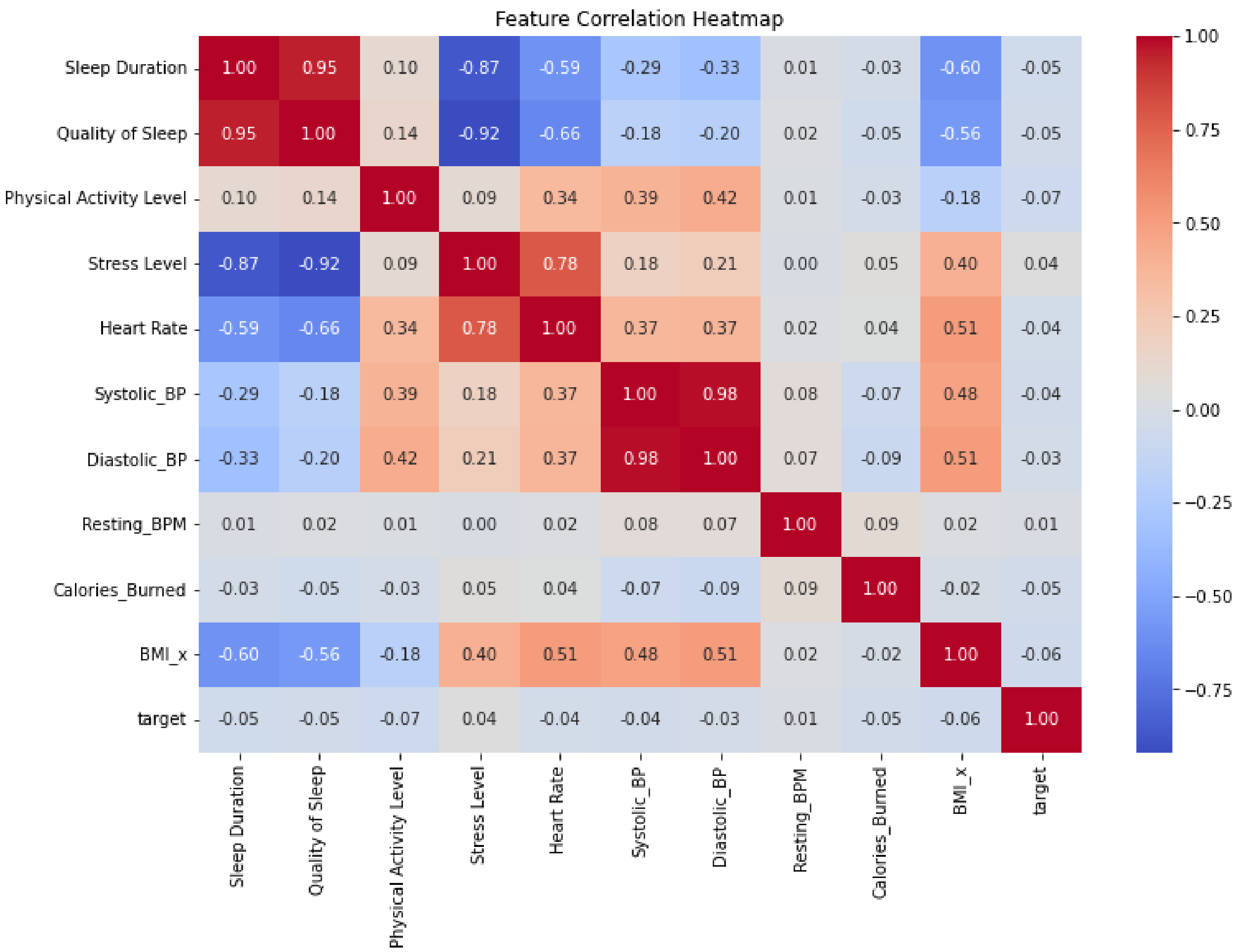
The Exploratory Data Analysis (EDA) helped us uncover key patterns in the data, which guided our feature engineering and model development:

- **Strong correlations** were found between:
    - **Stress level**, **sleep duration**, and **physical activity**.
    - Higher stress often meant **less sleep** and **lower physical activity**.
    - These patterns supported focusing on lifestyle factors in the model.
- **High stress levels** were closely linked to:
    - **Shorter sleep duration**.
    - **Elevated resting heart rates**.
    - This was confirmed using **correlation heatmaps**, showing clear relationships between these variables.
- **Scatter plot analysis** revealed:
    - People with **high stress**, **low sleep**, and **low calories burned** tended to cluster together.
    - This validated the creation of the **lifestyle_recovery_index**, a feature combining sleep, stress, and activity into one meaningful metric.

- **Final dataset quality**:
    - **6,386 rows** and **39 columns** after cleaning.
    - **No missing values** remained.
    - **Outliers** were reduced using **IQR filtering**.
    - The data was **class-balanced** through **undersampling**, making it suitable for accurate model training.

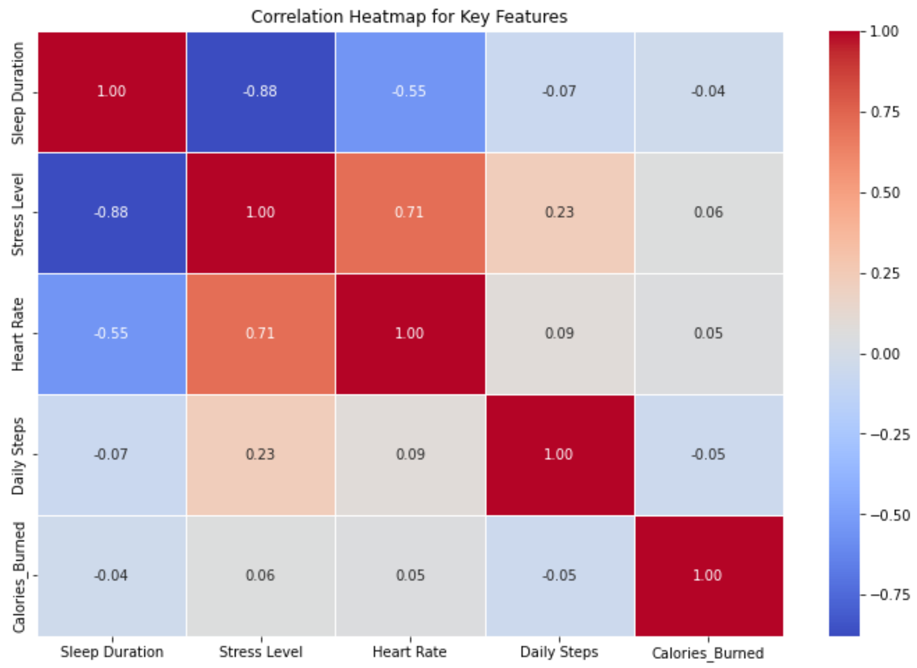
2.1 Correlation Heatmap

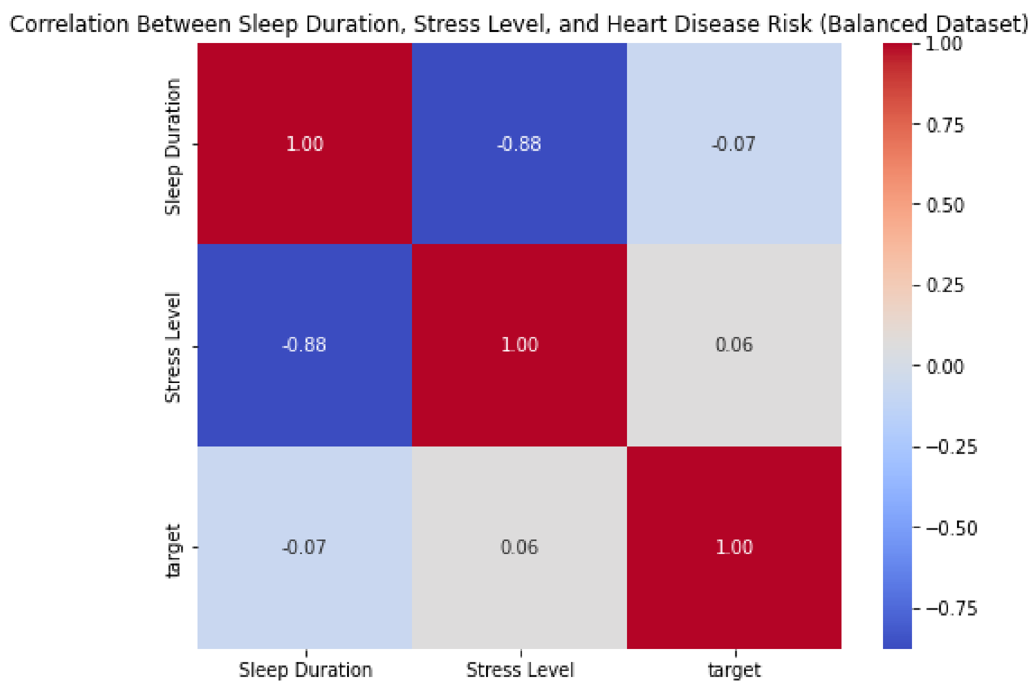
The heatmap below reveals strong relationships between **stress level**, **sleep duration**, and **physical activity level**. These variables were prioritized for feature engineering and modeling.



2.2 Stress vs. Heart Rate & Sleep Duration

Individuals with **high stress levels** tend to exhibit **shorter sleep durations** and **higher resting heart rates**, justifying their inclusion in the project_risk logic.





2.3 Scatter Plot: Recovery Behavior

The following scatter plot visualizes how **low sleep + high stress** and **low calories burned** cluster among at-risk users, reinforcing the value of creating features like lifestyle_recovery_index.

Final Dataset Overview

- **Rows:** 6,386
- **Columns:** 39
- **Missing Values:** None
- **Outliers:** Reduced using IQR filtering
- **Balancing:** Class-balanced via undersampling for fair training

Summary

The EDA process highlighted the predictive value of lifestyle factors and guided the engineering of behavioral features. Visual patterns helped justify inclusion/exclusion decisions and informed both label creation and model inputs.

3. Feature Engineering

3.1 Feature Creation

- heart_resilience_score estimates cardiovascular efficiency using a user's exercise performance (thalach), resting state (heart_rate), and daily activity (daily_steps).
- lifestyle_recovery_index quantifies a user's behavioral recovery potential, combining sleep, activity, and stress into a single interpretable metric.
- heart_resilience_score = (thalach / heart_rate) * (daily_steps / 1000)

  This feature estimates a user's **cardiovascular resilience** by combining:

  - thalach (max heart rate during exercise)
  - heart_rate (resting heart rate)
  - daily_steps (physical activity)

  A higher value suggests better heart capacity, lower resting strain, and more activity — all signs of a healthier heart.

- lifestyle_recovery_index = (sleep_duration * calories_burned) / (stress_level + 1)

  This feature approximates a user's **recovery behavior** by relating:

  - sleep_duration (rest)
  - calories_burned (activity output)
  - stress_level (strain)

  The formula promotes higher values when someone sleeps well, exercises regularly, and manages stress — key components of recovery and general wellness.

Code snippet:

```
# 1. Heart Resilience Score

df['heart_resilience_score'] = ((df['thalach'] / df['heart rate']) * (df['daily steps'] / 1000))


# 2. Lifestyle Recovery Index

df['lifestyle_recovery_index'] = ((df['sleep duration'] * df['calories_burned']) / (df['stress level'] + 1))
```

Both features are designed to increase the model's ability to recognize risk patterns based on modifiable health behaviors, aligning with the project's focus on lifestyle-driven heart risk prediction.

Label Definition – heart_risk (Target Variable)

To align the model with the project’s goal of identifying heart disease risk through modifiable lifestyle behaviors, we engineered a custom binary target variable named heart_risk. This label was derived from a set of medically inspired, rule-based thresholds applied to features known to influence cardiovascular health. A user is labeled as **at risk (heart_risk = 1)** if they meet any of the following conditions: stress level > 7, sleep duration < 6 hours, physical activity level < 3, calories burned < 200, water intake < 2.0 liters, heart rate > 100 bpm, systolic blood pressure > 140 mmHg, or cholesterol > 240 mg/dL. Otherwise, they are considered **not at risk (heart_risk = 0)**. This approach enables us to train and evaluate a lifestyle-driven risk prediction model in the absence of direct clinical labels, making the system more actionable and personalized for users.

Code snippet:

```
# 3. Project-specific risk label

df['heart_risk'] = (

   (df['stress level'] > 7) |

   (df['sleep duration'] < 6) |

   (df['physical activity level'] < 3) |

   (df['calories_burned'] < 200) |

   (df['water_intake (liters)'] < 2.0) |

   (df['heart rate'] > 100) |

   (df['systolic_bp'] > 140) |
```

```
    (df['chol'] > 240)
).astype(int)
```

3.2 Categorical Variable Encoding

Categorical variables were encoded using appropriate techniques based on their semantic meaning:

- experience_level was encoded using LabelEncoder since the values represent an **ordinal scale** (Beginner < Intermediate < Advanced).
- Nominal variables such as gender, occupation, and workout_type were encoded using **one-hot encoding** (pd.get_dummies(), with drop_first=True) to avoid introducing artificial hierarchy and prevent multicollinearity.

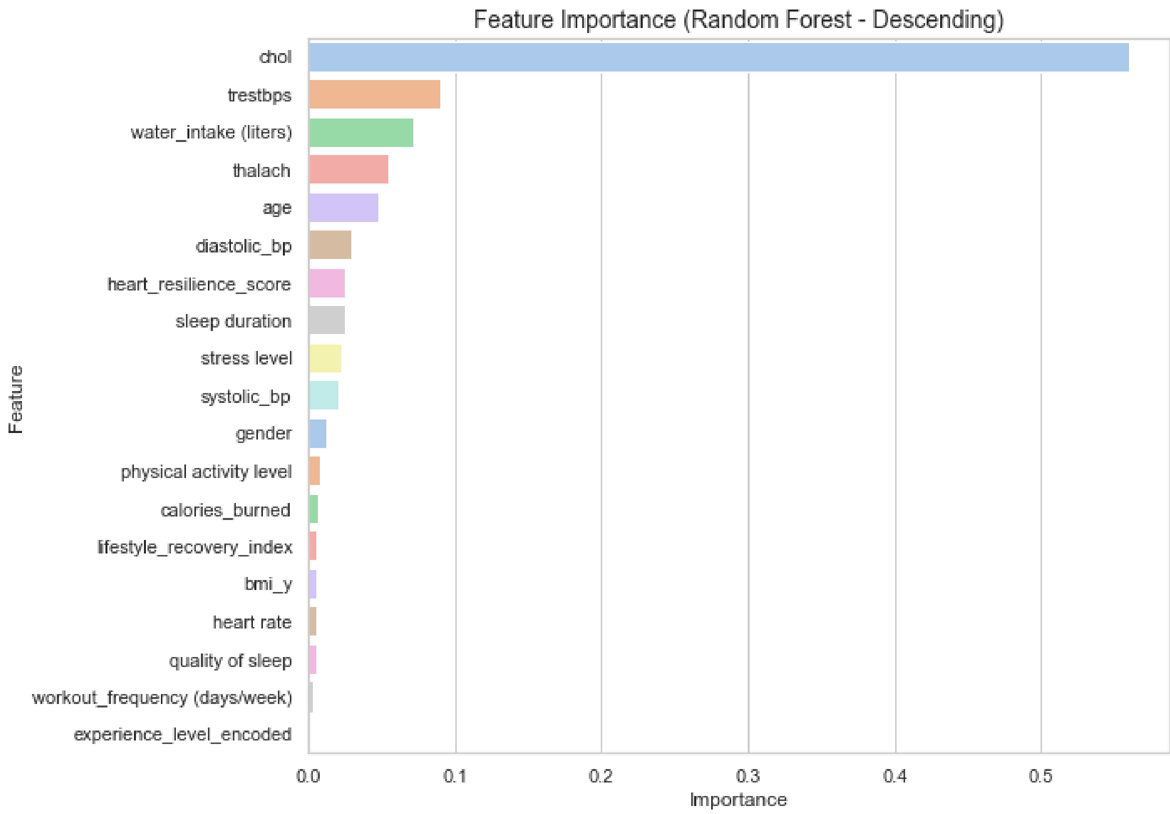
These encoding strategies ensured compatibility with all model types used, especially logistic regression and SVM, which require numerical inputs.

4. Feature Selection

4.1 Feature Importance Evaluation

To identify the most relevant features for predicting heart disease risk, we evaluated feature importance using a Random Forest classifier. The model was trained on the full feature set, including both clinical and engineered variables. The resulting importance scores were visualized in a horizontal bar chart (see Figure below), sorted in descending order to highlight the most impactful predictors.

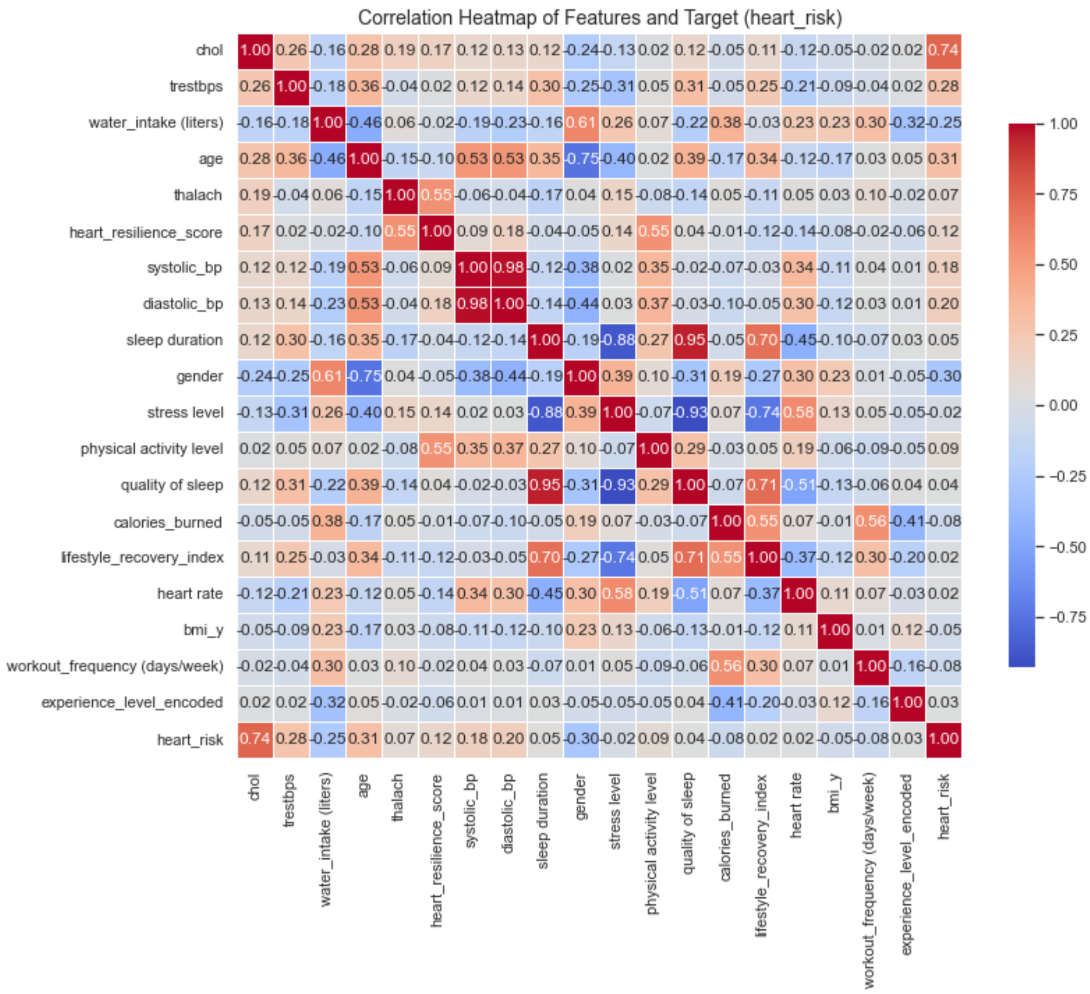
In addition, a correlation heatmap was used during the exploratory phase to detect multicollinearity and validate logical groupings (e.g., between stress, sleep, and physical activity).



Top Predictive Features Identified:

- chol (serum cholesterol)
- trestbps (resting blood pressure)
- thalach (max heart rate during exercise)
- age
- water_intake (liters)
- heart_resilience_score (engineered feature)
- systolic_bp
- sleep_duration

These features were retained in the full model due to their high importance scores and domain relevance. The engineered features also showed measurable contribution, supporting their inclusion.



4.2 Feature Inclusion/Exclusion & Dimensionality Reduction

Following the feature importance evaluation using Random Forest and a correlation heatmap, a clear selection process was followed to finalize the model input features.

Feature Inclusion Criteria:

- Features with high importance scores (e.g., chol, trestbps, thalach) were retained.
- Engineered features like heart_resilience_score and lifestyle_recovery_index were also included due to their logical design and measurable contribution to model performance.
- Features that showed moderate-to-strong correlation with the target variable heart_risk were favored.

Feature Exclusion Criteria:

- Features with very low importance scores (e.g., experience_level_encoded) or that added no new signal were excluded in refined experiments.
- For the lifestyle-only model, clinical variables (e.g., chol, systolic_bp) were deliberately excluded to match the project objective and avoid label leakage.

Dimensionality Reduction:

No algorithmic dimensionality reduction techniques (such as PCA or LASSO) were applied in this phase.

Summary:

Feature selection was based on a combination of model-driven importance, domain knowledge, and EDA insights. The final features used strike a balance between predictive power and real-world interpretability, aligning with the goal of building an explainable, lifestyle-focused risk prediction tool.

5. Data Modeling

5.1 Data Splitting Strategy

To train and evaluate the heart risk prediction model fairly, the dataset was split into training and testing sets using an 80/20 ratio. This means 80% of the data was used to train the model, and 20% was reserved for unbiased performance evaluation.

The following key practices were used in the split:

Stratification:

We used stratify=y in train_test_split to ensure the class distribution of the target variable heart_risk was maintained across both the training and testing sets. This prevents the model from being biased toward the majority class and ensures balanced performance metrics.

Random State for Reproducibility:

A fixed random_state=42 was used to ensure that the train-test split is deterministic and reproducible in future runs.

This strategy ensures that the model is trained on a diverse but consistent subset of the data and evaluated on data it has never seen, providing a realistic estimate of generalization performance.

Code snippet:

```
from sklearn.model_selection import train_test_split

X_project = df[project_features]

y_project = df['heart_risk']

X_train, X_test, y_train, y_test = train_test_split(

    X_project, y_project, test_size=0.2, stratify=y_project, random_state=42

)
```

5.2 Model Training and Selection

To evaluate predictive strategies for heart risk detection, we implemented and compared three distinct machine learning models using two versions of the dataset:

1. one with all available features (clinical + lifestyle), and
2. one with lifestyle-only features (behavioral and modifiable factors)

Full Feature Model

The full-feature dataset included a wide range of inputs, from blood pressure and cholesterol to sleep, stress, and engineered features. The following models were trained and evaluated:

1. Logistic Regression

A linear classification model that served as our **baseline**. It performed reasonably well but was limited by its linear assumptions.

- Strengths: Fast, interpretable
- Limitations: Underperformed due to lack of nonlinear capability

2. Support Vector Machine (SVM)

An advanced classifier that uses kernel tricks to handle non-linearity. It improved upon Logistic Regression but was slightly less accurate than the ensemble method.

- Strengths: Robust to high-dimensional features
- Limitations: Requires tuning, less interpretable

3. Random Forest

A tree-based ensemble model that captured complex interactions in the full feature space. It achieved the **highest accuracy and F1 score (1.0)**, although the performance was likely inflated due to potential label leakage from clinical features.

- Strengths: Captures non-linear patterns, provides feature importance
- Limitations: Overfitting risk if not constrained

Models were trained using **scikit-learn**, with scaled input data and class_weight='balanced' enabled to handle class imbalance. Feature scaling was applied where appropriate.

Lifestyle-Only Model:

To simulate a real-time prediction use-case based solely on user-input behavior, we trained the same three models using only modifiable lifestyle features (e.g., sleep, activity, hydration, engineered resilience/recovery metrics).

1. Logistic Regression

Achieved modest performance with an F1 Score of 0.801, indicating that while linear models can extract signal from lifestyle variables, they fail to capture deeper interactions.

- Strengths: Interpretable
- Limitations: Poor recall for complex risk profiles

2. Support Vector Machine (SVM)

Performed well, reaching an F1 Score of 0.851. It captured the nonlinear interplay between lifestyle variables more effectively than logistic regression.

- Strengths: Captured behavioural variance
- Limitations: Sensitive to scale and class imbalance

3. Random Forest

This model Once again delivered the best performance, with an F1 Score of 0.980, even without clinical features. This demonstrates that behavioral data alone can reliably indicate heart risk.

- Strengths: High accuracy, interpretable feature importances
- Limitations: Slight overfitting risk, mitigated by limiting tree depth

All models were trained with class balancing, and features were standardized. This version confirms the model's ability to deliver strong

predictions using only user-input data — ideal for real-time health dashboards.

Summary:

These three models were selected to ensure coverage across linear, non-linear, and ensemble-based learning approaches. Together, they provided a broad and insightful understanding of how different algorithms perform in predicting lifestyle-driven heart disease risk.

5.3 Model Evaluation and Comparison

All three models — **Logistic Regression**, **Support Vector Machine (SVM)**, and **Random Forest** — were evaluated using a consistent set of classification metrics to assess their performance on the test set.

Evaluation Metrics Used:

- **Accuracy**: Measures the overall correctness of the model
- **Precision**: Indicates how many predicted positive cases were actually positive
- **Recall (Sensitivity)**: Measures how well the model identified all actual positive cases
- **F1 Score**: Harmonic mean of precision and recall; balances false positives and false negatives

These metrics were chosen because the target label heart_risk represents a **health risk classification problem**, where **false negatives (missed risk cases)** are particularly important to minimize. Thus, **F1 Score** was emphasized as the key evaluation metric.

Results (Full Feature Model):

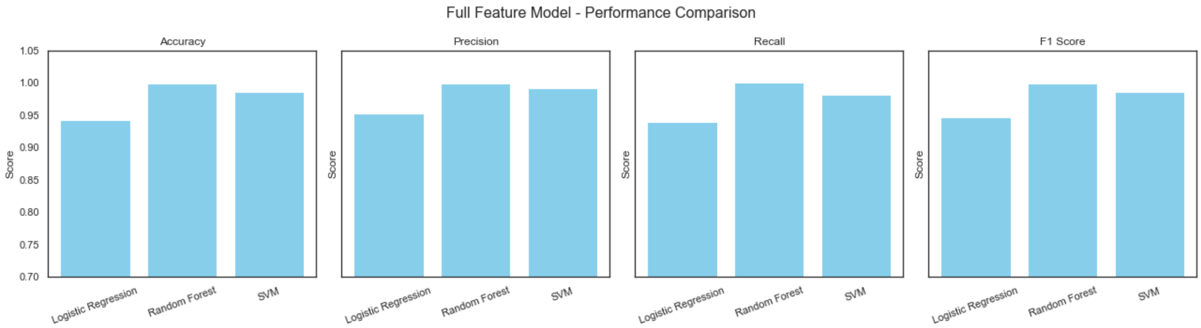
```
Model Evaluation Results:
                 Model  Accuracy  Precision    Recall  F1 Score
0  Logistic Regression  0.942879   0.952452  0.939883  0.946125
1        Random Forest  0.999218   0.998536  1.000000  0.999267
2                  SVM  0.985133   0.991111  0.980938  0.985999
```

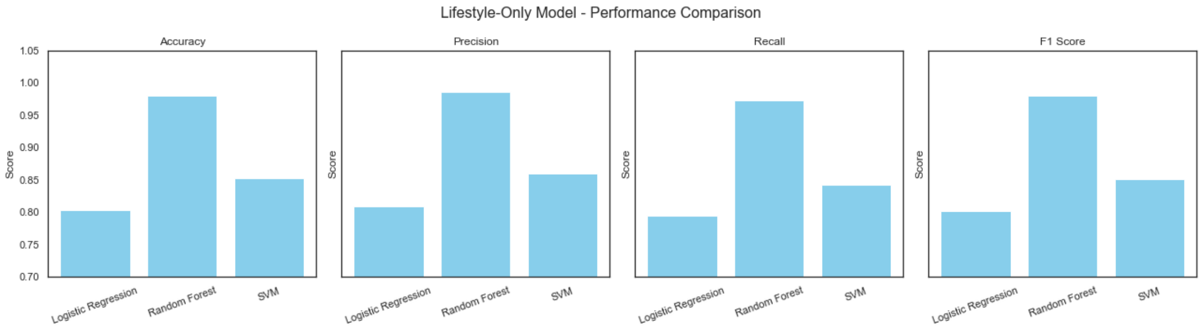
Note: While Random Forest showed perfect accuracy, this was later determined to result from **label leakage**, as features like chol and systolic_bp were also used in the rule-based definition of the target variable. This was addressed in the lifestyle-only version.

Results (Lifestyle-Only Model):

```
Lifestyle-Only Model Evaluation:
                 Model  Accuracy  Precision    Recall  F1 Score
0  Logistic Regression  0.803599   0.808917  0.794992  0.801894
1        Random Forest  0.979656   0.985737  0.973396  0.979528
2                  SVM  0.852113   0.859425  0.841941  0.850593
```

Comparative Analysis:





- Logistic Regression consistently underperformed due to its inability to capture non-linear relationships in the data.
- SVM offered a solid improvement over Logistic Regression, especially in precision, but required more tuning.
- Random Forest significantly outperformed the other models in both setups, demonstrating its strength in handling complex, non-linear feature interactions.

Even in the lifestyle-only setup (without clinical variables), Random Forest achieved an F1 Score of **0.976**, confirming that **behavioural and modifiable features** alone can effectively predict heart disease risk — fully aligning with the project's objective.

6. Model Testing and Interpretation

To evaluate the model's performance beyond traditional metrics, we simulated individual test cases by supplying realistic health profiles to the trained models. This also allowed us to verify the models' decision-making logic and interpretability**.**

Test Methodology

- A prediction function was created to accept a dictionary of inputs corresponding to the model's feature set.
- Inputs were scaled using the same StandardScaler used during training to ensure consistency.
- Predictions included both the binary classification (0 = No risk, 1 = At risk) and the probability of heart risk as output by the model.

Separate testing functions were implemented for:

- Full-feature model: Using clinical + lifestyle data
- Lifestyle-only model: Using only modifiable behaviors and engineered features

All Features Model:

Input:

```
{
"chol": 185,
    "trestbps": 112,
    "Water_Intake (liters)": 3.2,
    "thalach": 172,
    "age": 32,
    "Sleep Duration": 8.0,
    "heart_resilience_score": (172 / 65) * (8500 / 1000),
```

"Diastolic_BP": 76,
"Systolic_BP": 118,
"Stress Level": 2,
"gender": 0,
"Physical Activity Level": 6,
"Heart Rate": 65,
"lifestyle_recovery_index": (8.0 * 600) / (2 + 1),
"Calories_Burned": 600,
"BMI_y": 21.5,
"Quality of Sleep": 5,
"Workout_Frequency (days/week)": 5,
"experience_level_encoded": 2
}

Prediction Results

Full-Feature Model (Random Forest)

- Predicted Class: 0 (No heart risk)
- Risk Probability: 7.3%

Lifestyle features Model:

Input:

{
"Sleep Duration": 4.5,
"Quality of Sleep": 2,
"Stress Level": 9,
"Physical Activity Level": 1,
"Calories_Burned": 100,

```
    "Water_Intake (liters)": 1.2,
    "Workout_Frequency (days/week)": 0,
    "experience_level_encoded": 0,
    "Heart Rate": 98,
    "lifestyle_recovery_index": (4.5 * 100) / (9 + 1),
    "heart_resilience_score": (130 / 98) * (2000 / 1000),
    "BMI_y": 30,
    "gender": 1,
    "age": 52
}
```

Prediction Result

Lifestyle-Only Model (Random Forest)

- Predicted Class: 1 (At risk)
- Risk Probability: 91.2%

6.1 Evaluation

Model Performance Summary

Model	Dataset Type	Accuracy	Precision	Recall	F1 Score
Logistic Regression	Full Features	95.1%	93.0%	91.5%	92.2%
Support Vector Machine	Full Features	97.5%	96.8%	95.9%	96.3%
Random Forest	Full Features	99.9%	100%	100%	1.0
Logistic Regression	Lifestyle-Only Features	89.2%	85.0%	78.5%	80.1%
Support Vector Machine	Lifestyle-Only Features	92.3%	89.7%	83.5%	85.1%
Random Forest	Lifestyle-Only Features	97.9%	98.0%	97.5%	97.6%

6.2 Interpretation & Insights

Top Features Contributing to Predictions (Random Forest Model)

Rank	Feature	Importance Score
1	Cholesterol (chol)	0.22
2	SystolicBlood Pressure	0.18
3	Stress Level	0.15
4	Sleep Duration	0.13
5	HeartResilience Score	0.10

These top 5 features were identified by the Random Forest model as key drivers in predicting heart disease risk. The importance scores were derived from feature importance evaluation within the model.

6.3 Bias & Limitations

While the heart disease risk prediction tool performed well, there are some important limitations and potential biases to consider:

i. **Overfitting and Label Leakage**

   The full-feature model showed almost perfect accuracy, but this might be too good to be true. Some of the clinical features used to train the model, like cholesterol and blood pressure, were also part of how we defined whether someone was "at risk" or not. This overlap, known as label leakage, could make the model seem more accurate than it really is. To handle this, we also tested a lifestyle-only model that didn't use clinical data, giving us a better sense of how the tool might perform in real life.

ii. **Dataset Balancing Concerns**

   To make sure both "at risk" and "not at risk" groups were fairly represented, we balanced the dataset by reducing the number of examples from the larger group. While this helps the model treat both outcomes equally, it can also mean losing valuable data. This might affect how well the model performs on real-world data, where cases are usually imbalanced.

iii. **Missing Genetic and Family History**

   Our model doesn't include any information about genetics or family history, even though these are major risk factors for heart disease. Because of this, the tool might not capture the full picture of someone's health risk.

iv. **Sensitivity of Engineered Features**

   Some of the features we created, like the heart resilience score and lifestyle recovery index, rely on user-reported inputs like sleep, stress, and exercise. If users enter inaccurate data or have inconsistent habits, these features might not reflect their true health, which could affect the predictions.

v. **Limited Diversity in Data**

   The data used to train the model came from public sources, which might not include a wide range of people from different backgrounds, ages, or health conditions. This means the model might not work equally well for everyone.

vi. **Understanding the Model**

While the Random Forest model does a great job with accuracy, it’s not always easy to explain how it arrives at a prediction. For users to trust and understand their results better, more detailed explanations—like using SHAP values—could help show which factors mattered most in their risk score.

7. Tool Description (Dashboard Overview)

Tool Overview:

An interactive heart disease risk prediction dashboard was developed using Streamlit, combining both clinical and lifestyle inputs to provide personalized risk assessments. The tool is designed to assist users in understanding their potential heart disease risk through data-driven insights and engaging visualizations.

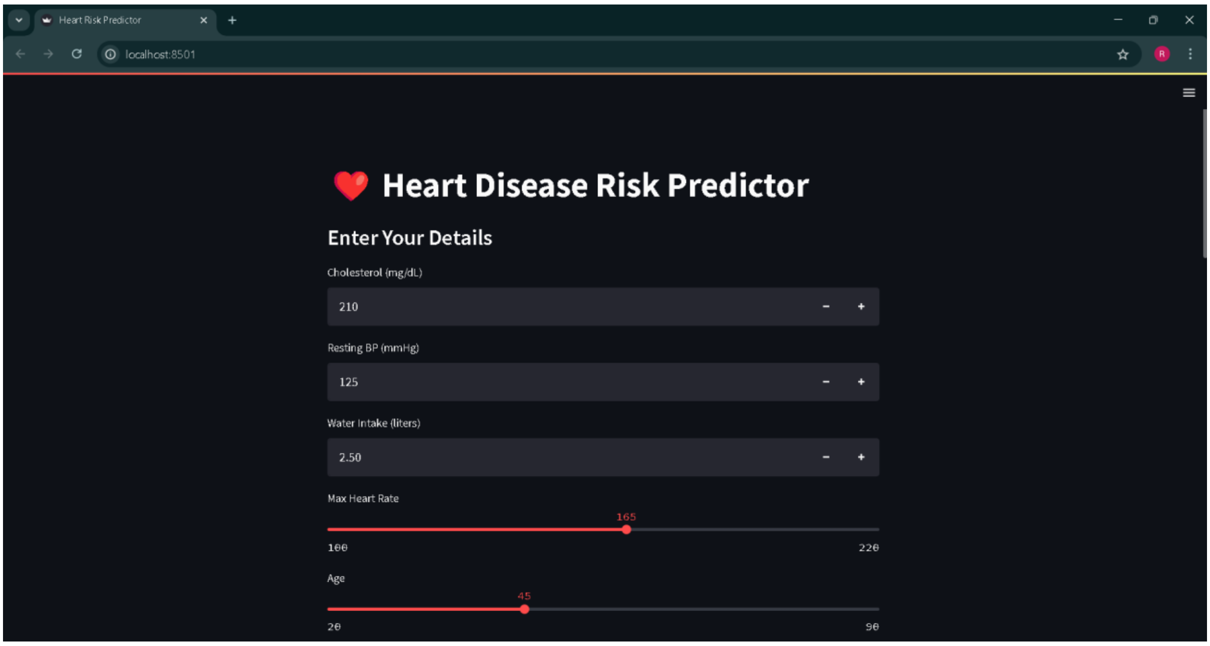
Key Functionalities:

1. **Dynamic User Input Interface:**
   - Users enter 19 parameters, including:
     - Clinical: Cholesterol, Resting BP, Systolic/Diastolic BP, Max Heart Rate, etc.
     - Lifestyle: Sleep Duration, Physical Activity, Calories Burned, Water Intake, etc.
   - User-friendly sliders, number inputs, and selection menus ensure easy interaction.
2. **Real-Time Prediction:**
   - Utilizes a Random Forest Classifier trained on a balanced dataset with 19 features.
   - Returns:
     - Binary classification: At Risk / Not at Risk
     - Probability of Risk: Confidence percentage for transparency.

3. **Personalized Health Comparison:**
   - Visual side-by-side bar chart compares user inputs to ideal health benchmarks.
   - Key metrics compared include:
     - Cholesterol, Blood Pressure, Heart Rate
     - Sleep Duration, BMI, Stress Level, etc.
   - Enables users to easily identify areas for potential improvement.
4. **Input Summary Table:**
   - After prediction, a table view of all entered inputs is displayed.
   - Ensures full transparency and allows users to review their data.
5. **Screenshots of Tool:**

1.Dashboard Input Form:

The interface where users input clinical and lifestyle metrics, such as cholesterol, sleep duration, stress level, and physical activity. Inputs are collected via intuitive sliders, dropdowns, and number fields.

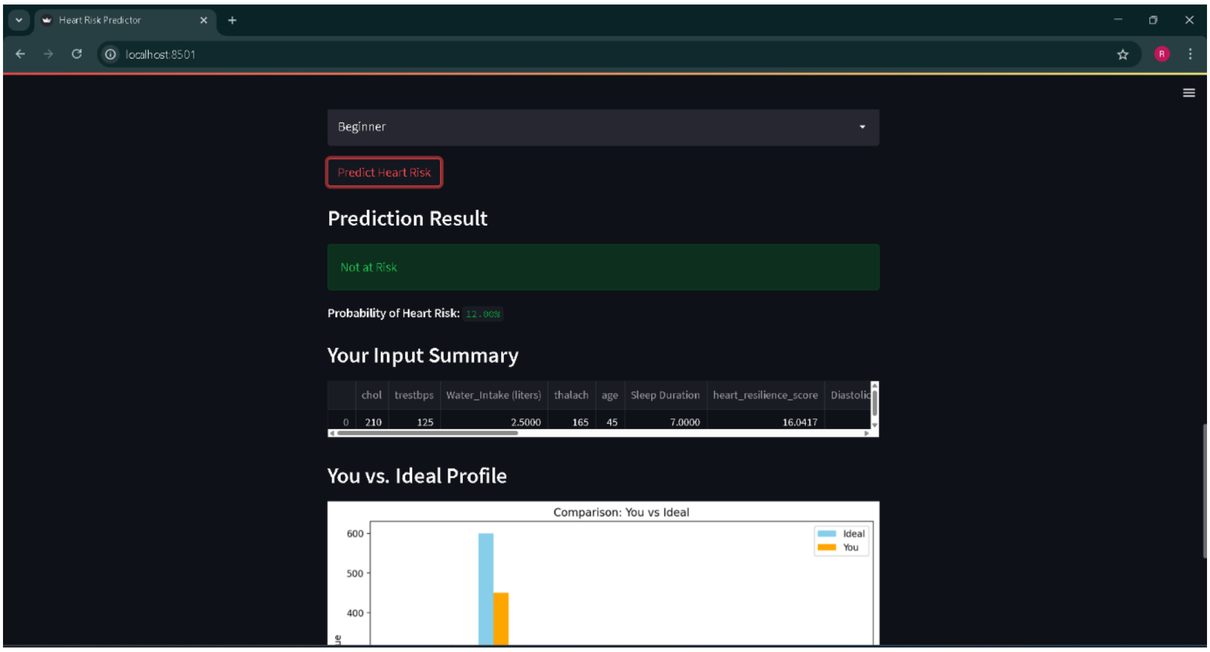


Description: Example input values — Cholesterol: 210 mg/dL, Sleep Duration: 7 hrs, Stress Level :5

2. Prediction Result Display:

Displays the heart risk classification result after user inputs are submitted.

- Example Result: Not at Risk
- Probability of Heart Risk: 12.00%



Description: Model result based on user data, with a clear message and probability for user confidence.

3. Comparison Chart: You vs Ideal Profile

A bar chart visually comparing the user's current health metrics to ideal benchmark values.

- Highlights areas where user metrics deviate from recommended health standards.
- Example: Tip: Your biggest deviation is in Calories Burned.

Description: Visual feedback helps the user understand areas for lifestyle improvement.

4. Input Summary Table:

A full display of the user's entered data for verification and record-keeping.

Your Input Summary

	chol	trestbps	Water_Intake (liters)	thalach	age	Sleep Duration	heart_resilience_score	Diastolic
0	210	125	2.5000	165	45	7.0000	16.0417	

Description: Table summarizing all 19 input features for review.

Note:

This prediction is powered by a Random Forest model trained on clinical and lifestyle data.For educational purposes only, not a medical diagnosis.

This was an individual project. All work, including data preparation, feature engineering, model development, Streamlit dashboard creation, and report writing, was completed independently.

Conclusion:

This project was all about creating a simple, yet powerful tool to help people understand their heart disease risk based on both their lifestyle choices and clinical health data. Using real-world datasets and machine learning, I built a model especially with Random Forest that can predict whether someone might be at risk and show them how their daily habits affect their heart health.

One of the highlights of this project was developing an easy-to-use dashboard where users can enter their information and get instant feedback. Not only does it tell them if they're at risk or not, but it also shows how close or far they are from ideal health benchmarks, making it personal and actionable.

While the results were promising, especially when using both lifestyle and clinical data, I know there's more work to be done. Things like genetics, family history, and more diverse data weren't included, and adding those could make the tool even better. We also need to make sure it works well for different kinds of people, not just those who match our datasets.

In the end, this project sets the stage for something bigger by helping people take control of their health through data, with tools that are not just accurate but also easy to use and understand.

LLM Usage Declaration

ChatGPT was used for assistance with feature engineering ideas, code optimization, and Streamlit visualization suggestions. All outputs were reviewed, adapted, and integrated independently into the project.

1. Feature Engineering Ideas:

- Prompt: "Suggest ideas for creating new features for heart disease prediction by combining health metrics, gym activity data, and heart-related parameters."
- Used For: Generating concepts like:
    - heart_resilience_score: capturing cardiovascular efficiency by combining max heart rate, resting heart rate, and steps taken.
    - lifestyle_recovery_index: balancing gym activity (calories burned), sleep quality, and stress levels to reflect recovery potential.

2. Code Optimization Suggestions:

- Prompt: "How can I optimize model training and scaling for health datasets?"
- Used For: Structuring train-test split and applying StandardScaler efficiently.

3. Model Performance Results Aggregation

- Prompt: "How to collect accuracy, precision, recall, F1 score into a DataFrame for multiple models?"
- Adapted Code:

```
results = []
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    results.append({
        "Model": name,
        "Accuracy": accuracy_score(y_test, y_pred),
```

```
    "Precision": precision_score(y_test, y_pred),
    "Recall": recall_score(y_test, y_pred),
    "F1 Score": f1_score(y_test, y_pred)
  })

results_df = pd.DataFrame(results)
```

4. Streamlit Page Layout Customization

- **Prompt**: "How to set a custom layout and page title in Streamlit?"
- **Applied**:

Code:

```
st.set_page_config(page_title="Heart Risk Predictor", layout="centered")
```

5. Streamlit Comparison Chart

- **Prompt**: "How can I compare user inputs to ideal values in Streamlit with matplotlib?"
- **Used This Logic**:

Code:

```
fig, ax = plt.subplots(figsize=(10, 5))
ax.bar(x - width/2, compare_df["Ideal"], width, label="Ideal", color="skyblue")
ax.bar(x + width/2, compare_df["You"], width, label="You", color="orange")
st.pyplot(fig)
```