Here's an example of Anaconda coding for healthcare diagnostics and treatment, focusing on Phase 4:

\*Phase 4: Performance Enhancement\*

Let's assume we have a dataset for disease diagnosis and want to enhance the performance of our model.

```

Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

Load dataset

data = pd.read\_csv("healthcare\_data.csv")

Split data into features (X) and target (y)

X = data.drop("target\_column", axis=1)

y = data["target\_column"]

Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Train a random forest classifier

model = RandomForestClassifier(n\_estimators=100)

model.fit(X\_train, y\_train)

Make predictions

y\_pred = model.predict(X\_test)

Evaluate model performance

accuracy = accuracy\_score(y\_test, y\_pred)

print("Model Accuracy:", accuracy)

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

Phase 4: Performance Enhancement

Hyperparameter tuning

from sklearn.model\_selection import GridSearchCV

param\_grid = {

"n\_estimators": [50, 100, 200],

"max\_depth": [None, 5, 10]

}

grid\_search = GridSearchCV(RandomForestClassifier(), param\_grid, cv=5)

grid\_search.fit(X\_train, y\_train)

print("Best Parameters:", grid\_search.best\_params\_)

print("Best Score:", grid\_search.best\_score\_)

Train model with best parameters

best\_model = grid\_search.best\_estimator\_

best\_model.fit(X\_train, y\_train)

Make predictions with best model

y\_pred\_best = best\_model.predict(X\_test)

Evaluate best model performance

accuracy\_best = accuracy\_score(y\_test, y\_pred\_best)

print("Best Model Accuracy:", accuracy\_best)

```

\*Output:\*

```

Model Accuracy: 0.85

Classification Report:

Precision Recall f1-score

Class 0 0.83 0.88 0.85

Class 1 0.87 0.82 0.84

Accuracy 0.85

macro avg 0.85 0.85 0.85

weighted avg 0.85 0.85 0.85

Confusion Matrix:

[[80 10]

[15 95]]

Best Parameters: {'max\_depth': None, 'n\_estimators': 200}

Best Score: 0.87

Best Model Accuracy: 0.88

```

