Model Development Phase Template

Date	08 August 2025
Skill Wallet ID	**SWUID20250188325**
Project Title	Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
# 1. Logistic Regression
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred_log = log_reg.predict(X_test)
model_results['Logistic Regression'] = {
    'accuracy': accuracy_score(y_test, y_pred_log),
    'report': classification_report(y_test, y_pred_log),
    'confusion': confusion_matrix(y_test, y_pred_log)
}
print(classification_report(y_test, y_pred_log))
print(confusion_matrix(y_test, y_pred_log))
```

```
# 2. Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, max_depth=10, min_samples_split=5, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
model_results['Random Forest'] = {
    'accuracy': accuracy_score(y_test, y_pred_rf),
    'report': classification_report(y_test, y_pred_rf),
    'confusion': confusion_matrix(y_test, y_pred_rf)
}
print(classification_report(y_test, y_pred_rf))
print(confusion_matrix(y_test, y_pred_rf))
```

```python
# 3. Decision Tree Classifier
dt = DecisionTreeClassifier(max_depth=5, min_samples_split=4, random_state=42)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
model_results['Decision Tree'] = {
    'accuracy': accuracy_score(y_test, y_pred_dt),
    'report': classification_report(y_test, y_pred_dt),
    'confusion': confusion_matrix(y_test, y_pred_dt)
}
print(classification_report(y_test, y_pred_dt))
print(confusion_matrix(y_test, y_pred_dt))
```

```python
# 4. Gaussian Naive Bayes   # This is the best in testing with 0.9989
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred_gnb = gnb.predict(X_test)
model_results['Gaussian NB'] = {
    'accuracy': accuracy_score(y_test, y_pred_gnb),
    'report': classification_report(y_test, y_pred_gnb),
    'confusion': confusion_matrix(y_test, y_pred_gnb)
}
print(classification_report(y_test, y_pred_gnb))
print(confusion_matrix(y_test, y_pred_gnb))
```

```python
# 5. Multinomial Naive Bayes (requires non-negative values)
scaler = MinMaxScaler()
X_train_mnb = scaler.fit_transform(X_train)
X_test_mnb = scaler.transform(X_test)

mnb = MultinomialNB()
mnb.fit(X_train_mnb, y_train)
y_pred_mnb = mnb.predict(X_test_mnb)
model_results['Multinomial NB'] = {
    'accuracy': accuracy_score(y_test, y_pred_mnb),
    'report': classification_report(y_test, y_pred_mnb),
    'confusion': confusion_matrix(y_test, y_pred_mnb)
}
print(classification_report(y_test, y_pred_mnb))
print(confusion_matrix(y_test, y_pred_mnb))
```

```python
# Display all accuracies
print("Model Accuracies:")
for name, result in model_results.items():
    print(f"{name}: {result['accuracy']:.4f}")
```

Model Validation and Evaluation Report:

Model Name	Classification Report Screenshot	F1 Score	Confusion Matrix Screenshot
Logistic Regression	precision recall f1-score support 0 0.99 0.96 0.98 139 1 1.00 0.94 0.97 120 4 0.87 1.00 0.93 46 5 0.92 0.98 0.95 60 accuracy 0.96 365 macro avg 0.95 0.97 0.96 365 weighted avg 0.97 0.96 0.96 365	**96%**	[[134 0 0 5] [0 113 7 0] [0 0 46 0] [1 0 0 59]]
Random Forest	precision recall f1-score support 0 1.00 1.00 1.00 139 1 1.00 1.00 1.00 120 4 1.00 1.00 1.00 46 5 1.00 1.00 1.00 60 accuracy 1.00 365 macro avg 1.00 1.00 1.00 365 weighted avg 1.00 1.00 1.00 365	**100%**	[[139 0 0 0] [0 120 0 0] [0 0 46 0] [0 0 0 60]]
Decision Tree	precision recall f1-score support 0 1.00 1.00 1.00 139 1 1.00 1.00 1.00 120 4 1.00 1.00 1.00 46 5 1.00 1.00 1.00 60 accuracy 1.00 365 macro avg 1.00 1.00 1.00 365 weighted avg 1.00 1.00 1.00 365	**100%**	[[139 0 0 0] [0 120 0 0] [0 0 46 0] [0 0 0 60]]
Gaussian Navie Bayes	precision recall f1-score support 0 1.00 1.00 1.00 139 1 1.00 0.67 0.80 120 4 0.53 1.00 0.70 46 5 1.00 1.00 1.00 60 accuracy 0.89 365 macro avg 0.88 0.92 0.87 365 weighted avg 0.94 0.89 0.90 365	**89%**	[[139 0 0 0] [0 80 40 0] [0 0 46 0] [0 0 0 60]]
Multinomial Navie Bayes	precision recall f1-score support 0 0.80 0.84 0.82 139 1 0.85 1.00 0.92 120 4 1.00 0.54 0.70 46 5 0.58 0.52 0.55 60 accuracy 0.80 365 macro avg 0.81 0.73 0.75 365 weighted avg 0.81 0.80 0.79 365	**80%**	[[117 0 0 22] [0 120 0 0] [0 21 25 0] [29 0 0 31]]