Model Development Phase

Date	7 August2025
Skill wallet ID	SWUID20250185217
Project Title	Anemia Sense: Leveraging Machine Learning for Precise Anemia Recognition
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be presented in future updates through clear and well-documented screenshots, providing transparency in the implementation process. The Model Validation and Evaluation Report will comprehensively showcase performance metrics for multiple machine learning models, including classification reports, accuracy scores, and confusion matrices. These results will be accompanied by their respective visual representations and screenshots, ensuring clarity, comparability, and ease of interpretation for all stakeholders.

Initial Model Training Code:

```
# Logistic Regression
logistic_regression = LogisticRegression()
logistic_regression.fit(x_train, y_train)
y_pred = logistic_regression.predict(x_test)
acc_lr = accuracy_score(y_test, y_pred)
c_lr = classification_report(y_test, y_pred)

print("Logistic Regression Accuracy Score:", acc_lr)
print(c_lr)
```

```
# Random Forest
random_forest = RandomForestClassifier()
random_forest.fit(x_train, y_train)
y_pred = random_forest.predict(x_test)
acc_rf = accuracy_score(y_test, y_pred)
c_rf = classification_report(y_test, y_pred)

print("Random Forest Accuracy Score:", acc_rf)
print(c_rf)
```

```
# Naive Bayes
NB = GaussianNB()
NB.fit(x_train, y_train)
y_pred = NB.predict(x_test)
acc_nb = accuracy_score(y_test, y_pred)
c_nb = classification_report(y_test, y_pred)

print("Naive Bayes Accuracy Score:", acc_nb)
print(c_nb)
```

```
from sklearn.svm import SVC

support_vector=SVC()
support_vector.fit(x_train,y_train)
y_pred= support_vector.predict(x_test)

acc_svc = accuracy_score(y_test,y_pred)
c_svc = classification_report(y_test,y_pred)

print('Accuracy Score: ',acc_svc)
print(c_svc)
```

```
from sklearn.ensemble import GradientBoostingClassifier

gbc = GradientBoostingClassifier()
gbc.fit(x_train,y_train)
y_pred= gbc.predict(x_test)

acc_gbc = accuracy_score(y_test,y_pred)
c_gbc = classification_report(y_test,y_pred)

print('Accuracy Score: ',acc_gbc)
print(c_gbc)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	Random Forest Accuracy Score: 1.0 precision recall f1-score support 0 1.00 1.00 1.00 113 1 1.00 1.00 1.00 135 accuracy 1.00 248 macro avg 1.00 1.00 1.00 248 weighted avg 1.00 1.00 1.00 248	100%	Confusion Matrix [[113 0] [0 135]]

Logistic Regression	Logistic Regression Accuracy Score: 0.9919354838709677 precision recall f1-score support 0 1.00 0.98 0.99 113 1 0.99 1.00 0.99 135 accuracy 0.99 248 macro avg 0.99 0.99 0.99 248 weighted avg 0.99 0.99 0.99 248	99%	Confusion Matrix [[111 2] [0 135]]
Gaussian NB	Naive Bayes Accuracy Score: 0.9798387096774194 precision recall f1-score support 0 0.99 0.96 0.98 113 1 0.97 0.99 0.98 135 accuracy 0.98 248 macro avg 0.98 0.98 0.98 248 weighted avg 0.98 0.98 0.98 248	97%	Confusion Matrix [[109 4] [1 134]]
Gradient Boosting	Accuracy Score: 1.0 precision recall f1-score support 0 1.00 1.00 1.00 113 1 1.00 1.00 1.00 135 accuracy 1.00 248 macro avg 1.00 1.00 1.00 248 weighted avg 1.00 1.00 1.00 248	100%	Confusion Matrix [[113 0] [0 135]]
SVM	Accuracy Score: 0.9395161290322581 precision recall f1-score support 0 0.99 0.88 0.93 113 1 0.91 0.99 0.95 135 accuracy 0.94 248 macro avg 0.95 0.93 0.94 248 weighted avg 0.94 0.94 0.94 248	93%	Confusion Matrix [[99 14] [1 134]]