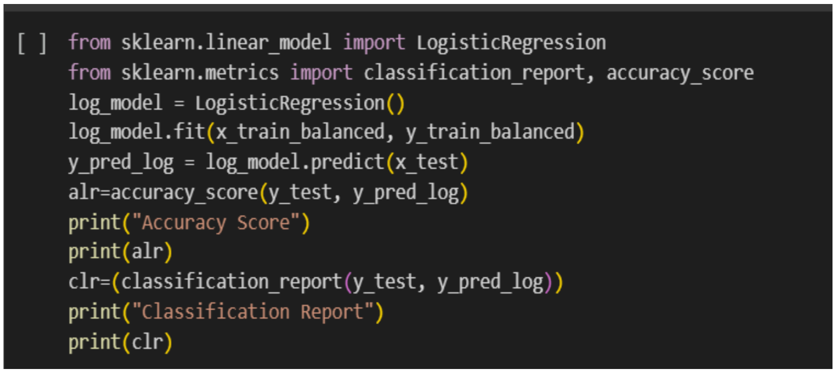
Model Development Phase Template

Date	9 August 2025
Skillwallet ID	SWUID20250188620
Project Title	Anemia Sense: Leveraging Machine Learning for Precise Anemia Recognition
Maximum Marks	4 Marks

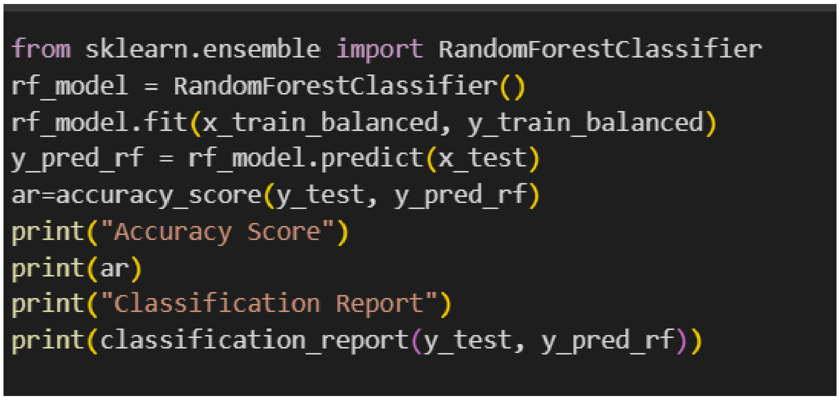
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be 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.

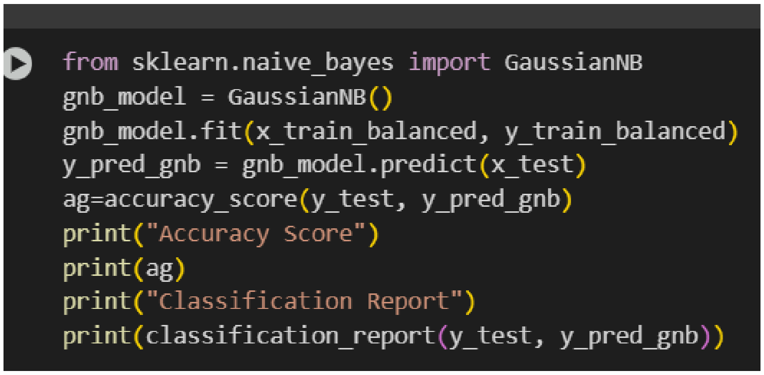
Initial Model Training Code:



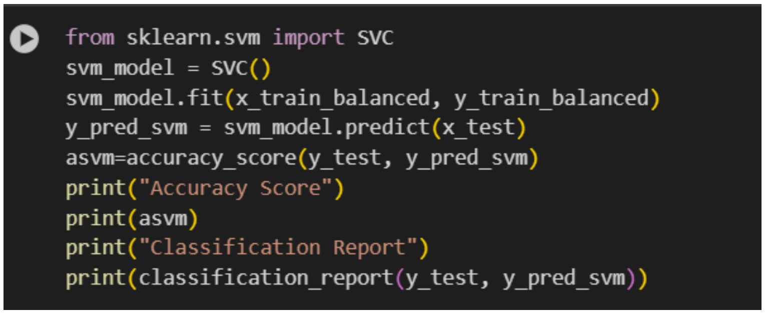
```
[ ]  from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import classification_report, accuracy_score
     log_model = LogisticRegression()
     log_model.fit(x_train_balanced, y_train_balanced)
     y_pred_log = log_model.predict(x_test)
     alr=accuracy_score(y_test, y_pred_log)
     print("Accuracy Score")
     print(alr)
     clr=(classification_report(y_test, y_pred_log))
     print("Classification Report")
     print(clr)
```



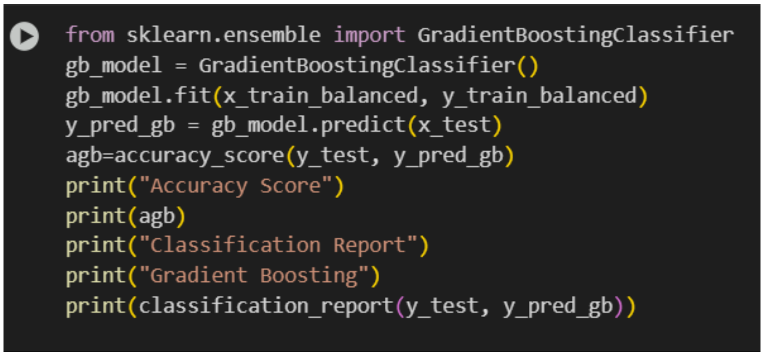
```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
rf_model.fit(x_train_balanced, y_train_balanced)
y_pred_rf = rf_model.predict(x_test)
ar=accuracy_score(y_test, y_pred_rf)
print("Accuracy Score")
print(ar)
print("Classification Report")
print(classification_report(y_test, y_pred_rf))
```



```
from sklearn.naive_bayes import GaussianNB
gnb_model = GaussianNB()
gnb_model.fit(x_train_balanced, y_train_balanced)
y_pred_gnb = gnb_model.predict(x_test)
ag=accuracy_score(y_test, y_pred_gnb)
print("Accuracy Score")
print(ag)
print("Classification Report")
print(classification_report(y_test, y_pred_gnb))
```

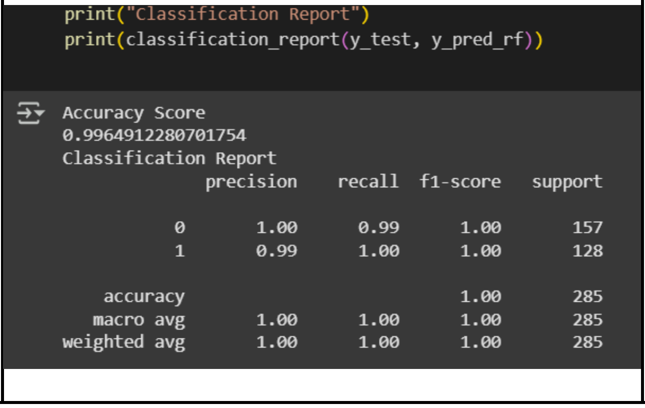
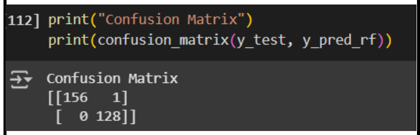


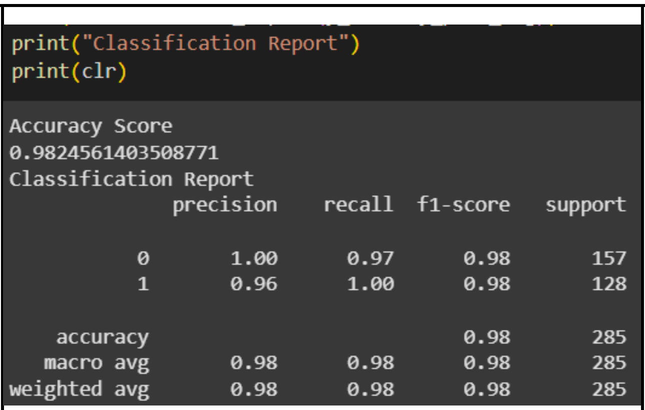
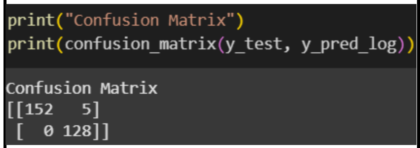
```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(x_train_balanced, y_train_balanced)
y_pred_svm = svm_model.predict(x_test)
asvm=accuracy_score(y_test, y_pred_svm)
print("Accuracy Score")
print(asvm)
print("Classification Report")
print(classification_report(y_test, y_pred_svm))
```

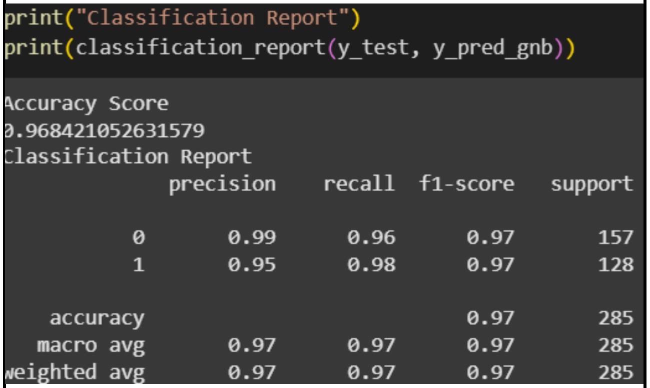
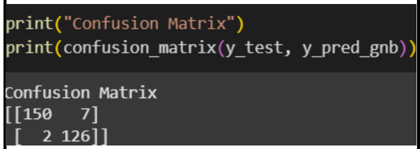


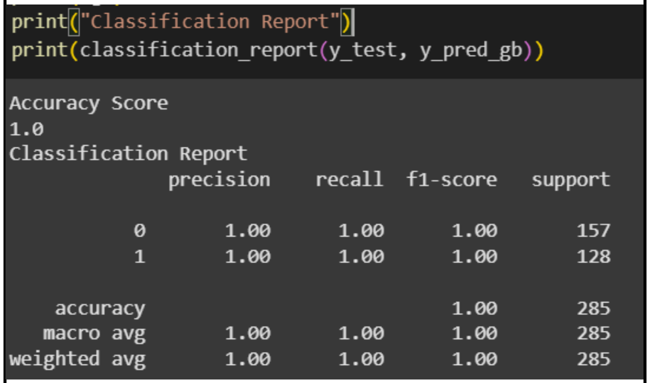
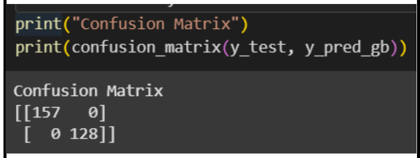
```
from sklearn.ensemble import GradientBoostingClassifier
gb_model = GradientBoostingClassifier()
gb_model.fit(x_train_balanced, y_train_balanced)
y_pred_gb = gb_model.predict(x_test)
agb=accuracy_score(y_test, y_pred_gb)
print("Accuracy Score")
print(agb)
print("Classification Report")
print("Gradient Boosting")
print(classification_report(y_test, y_pred_gb))
```

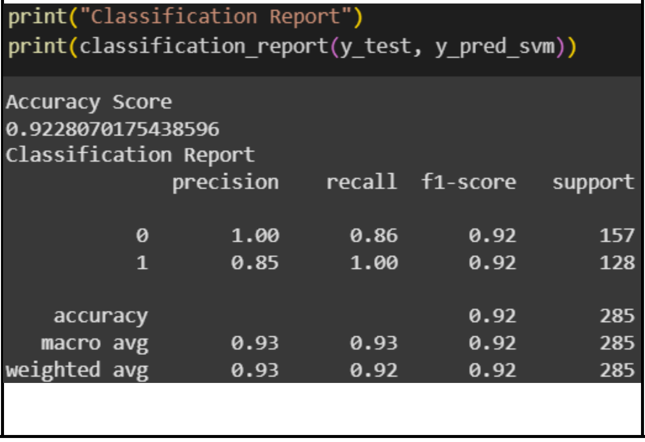
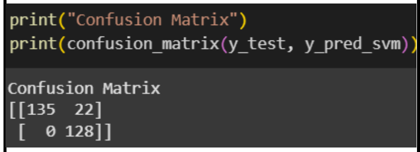
Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	 print("Classification Report") print(classification_report(y_test, y_pred_rf)) Accuracy Score 0.9964912280701754 Classification Report precision recall f1-score support 0 1.00 0.99 1.00 157 1 0.99 1.00 1.00 128 accuracy 1.00 285 macro avg 1.00 1.00 1.00 285 weighted avg 1.00 1.00 1.00 285	99%	 112] print("Confusion Matrix") print(confusion_matrix(y_test, y_pred_rf)) Confusion Matrix [[156 1] [0 128]]

Logistic Regression	
	print("Classification Report")
	print(clr)
	Accuracy Score
	0.9824561403508771
	Classification Report
	              precision recall f1-score support
	           0 1.00 0.97 0.98 157
	           1 0.96 1.00 0.98 128
	    accuracy 0.98 285
	   macro avg 0.98 0.98 0.98 285
	weighted avg 0.98 0.98 0.98 285
		98%	
			print("Confusion Matrix")
			print(confusion_matrix(y_test, y_pred_log))
			Confusion Matrix
			[[152 5]
			 [0 128]]

GaussianNB	
	print("Classification Report")
	print(classification_report(y_test, y_pred_gnb))
	Accuracy Score
	0.9684210526315789
	Classification Report
	              precision recall f1-score support
	           0 0.99 0.96 0.97 157
	           1 0.95 0.98 0.97 128
	    accuracy 0.97 285
	   macro avg 0.97 0.97 0.97 285
	weighted avg 0.97 0.97 0.97 285
		96%	
			print("Confusion Matrix")
			print(confusion_matrix(y_test, y_pred_gnb))
			Confusion Matrix
			[[150 7]
			 [2 126]]

Gradient Boosting	
	print("Classification Report")
	print(classification_report(y_test, y_pred_gb))
	Accuracy Score
	1.0
	Classification Report
	              precision recall f1-score support
	           0 1.00 1.00 1.00 157
	           1 1.00 1.00 1.00 128
	    accuracy 1.00 285
	   macro avg 1.00 1.00 1.00 285
	weighted avg 1.00 1.00 1.00 285
		100%	
			print("Confusion Matrix")
			print(confusion_matrix(y_test, y_pred_gb))
			Confusion Matrix
			[[157 0]
			 [0 128]]

SVM	
	print("Classification Report")
	print(classification_report(y_test, y_pred_svm))
	Accuracy Score
	0.9228070175438596
	Classification Report
	              precision recall f1-score support
	           0 1.00 0.86 0.92 157
	           1 0.85 1.00 0.92 128
	    accuracy 0.92 285
	   macro avg 0.93 0.93 0.92 285
	weighted avg 0.93 0.92 0.92 285
		92%	
			print("Confusion Matrix")
			print(confusion_matrix(y_test, y_pred_svm))
			Confusion Matrix
			[[135 22]
			 [0 128]]