Model Development Phase Template

Date	15 March 2024
Team ID	SWTID1720437019
Project Title	Thyroid Classification
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:

```
[39]: from sklearn.ensemble import RandomForestClassifier

      RFclassifier = RandomForestClassifier(max_leaf_nodes=30)
      RFclassifier.fit(x_train, y_train)
```

```
[40]: y_pred = RFclassifier.predict(x_test)

      print(classification_report(y_test, y_pred))
      print(confusion_matrix(y_test, y_pred))
```

SVC Model

```
[45]: from sklearn.svm import SVC
      SVCclassifier = SVC(kernel='linear', max_iter=251)
      SVCclassifier.fit(x_train, y_train)
```

[46]:
```
svc_params = {
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'C': [1, 10, 100],
    'gamma': ['scale', 'auto']
}

grid_svc = GridSearchCV(SVC(), svc_params, cv=5)
grid_svc.fit(x_train, y_train)

print("Best parameters for SVC:", grid_svc.best_params_)
```

XGBClassifier Model

[51]:
```
!pip install xgboost
```

[52]:
```
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y_train_encoded = le.fit_transform(y_train)
xgb = XGBClassifier()
xgb.fit(x_train, y_train_encoded)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

Random Forest	[42]:
	`from sklearn.model_selection import GridSearchCV`
	`rf = {`
	`'n_estimators': [100, 200, 300],`
	`'max_depth': [3, 5, 7, 10, None]`
	`}`
	`RFclassifier = RandomForestClassifier(random_state=42)`
	`grid_rf = GridSearchCV(RFclassifier, rf, cv=5)`
	`grid_rf.fit(x_train, y_train)`
	`print("Best parameters for Random Forest:", grid_rf.best_params_)`
	`y_pred = grid_rf.predict(x_test)`
	`print(classification_report(y_test, y_pred))`
	Best parameters for Random Forest: {'max_depth': None, 'n_estimators': 300}
	precision recall f1-score support
	0 1.00 0.43 0.60 7
	1 0.88 0.95 0.91 74
	2 0.95 0.98 0.97 85
	3 0.89 0.82 0.85 38
	4 0.97 1.00 0.98 122
	5 0.94 0.92 0.93 51
	6 0.99 0.93 0.96 71
	accuracy 0.94 448
	macro avg 0.94 0.86 0.88 448
	weighted avg 0.94 0.94 0.94 448
		94.20%	[43]:
			`print(confusion_matrix(y_test, y_pred))`
			[[3 0 0 0 4 0 0]
			[0 70 1 3 0 0 0]
			[0 2 83 0 0 0 0]
			[0 4 0 31 0 2 1]
			[0 0 0 0 122 0 0]
			[0 2 1 1 0 47 0]
			[0 2 2 0 0 1 66]]

SVC	[47]:
	`y_pred = SVCclassifier.predict(x_test)`
	`print(classification_report(y_test, y_pred))`
	precision recall f1-score support
	0 0.67 0.86 0.75 7
	1 0.79 0.80 0.79 74
	2 0.83 0.74 0.78 85
	3 0.73 0.58 0.65 38
	4 0.89 0.95 0.92 122
	5 0.76 0.75 0.75 51
	6 0.87 0.96 0.91 71
	accuracy 0.83 448
	macro avg 0.79 0.80 0.79 448
	weighted avg 0.83 0.83 0.83 448
		86.61%	[48]:
			`print(confusion_matrix(y_test, y_pred))`
			[[6 0 0 0 1 0 0]
			[1 59 7 3 2 2 0]
			[1 4 63 0 9 3 5]
			[0 6 2 22 0 7 1]
			[1 1 0 0 116 0 4]
			[0 4 3 4 2 38 0]
			[0 1 1 1 0 0 68]]

XGB CLASSIFIER MODEL	[53]:
	`y_test_encoded = le.transform(y_test)`
	`y_pred = xgb.predict(x_test)`
	`print(classification_report(y_test_encoded, y_pred))`
	precision recall f1-score support
	0 1.00 1.00 1.00 7
	1 0.91 0.95 0.93 74
	2 0.95 0.96 0.96 85
	3 0.86 0.84 0.85 38
	4 1.00 1.00 1.00 122
	5 0.92 0.92 0.92 51
	6 0.99 0.94 0.96 71
	accuracy 0.95 448
	macro avg 0.95 0.95 0.95 448
	weighted avg 0.95 0.95 0.95 448
		95.54%	[56]:
			`print(confusion_matrix(y_test_encoded, y_pred))`
			[[5 0 0 0 2 0 0]
			[0 70 1 3 0 0 0]
			[0 0 83 0 0 2 0]
			[0 2 0 35 0 1 0]
			[0 0 0 0 122 0 0]
			[0 2 1 2 0 46 0]
			[0 1 2 0 0 1 67]]