Model Development Phase Template

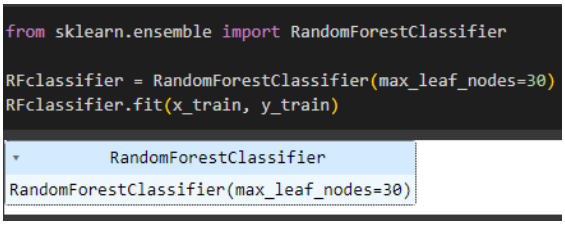
Date	29 June 2024
Team ID	SWTID1720447482
Project Title	THYROID CLASSIFICATION
Maximum Marks	4 Marks

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:

RandomForestClassifier

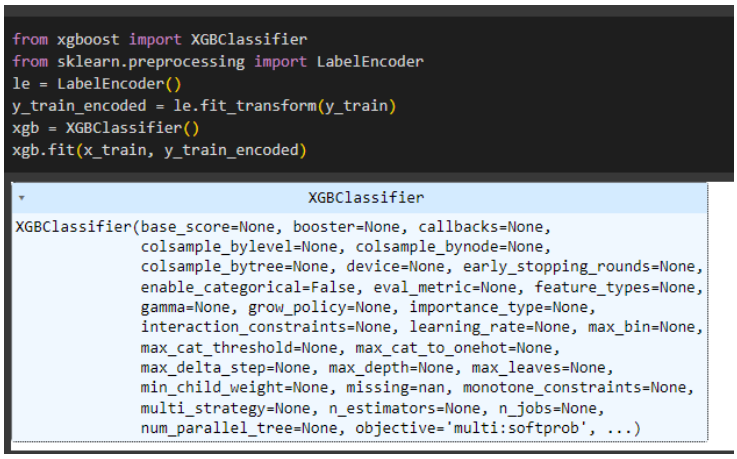


```
from sklearn.ensemble import RandomForestClassifier

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

```
RandomForestClassifier(max_leaf_nodes=30)
```

XGB Classifier



```
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)
```

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, objective='multi:softprob', ...)
```

SVC model

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

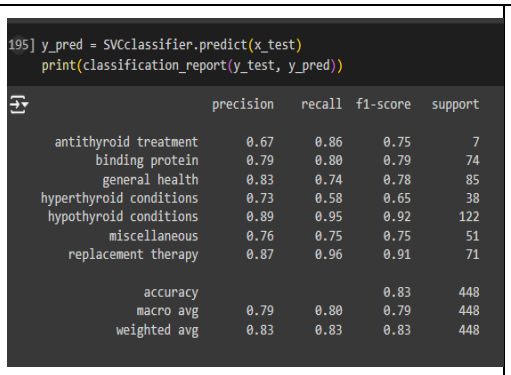
```
SVC
SVC(kernel='linear', max_iter=251)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

Random Forest Classifier	`from sklearn.ensemble import RandomForestClassifier`
	`from sklearn.metrics import classification_report, confusion_matrix # Impo`
	
	`RFclassifier = RandomForestClassifier(max_leaf_nodes=30)`
	`RFclassifier.fit(x_train, y_train)`
	
	`y_pred = RFclassifier.predict(x_test)`
	
	`print(classification_report(y_test, y_pred))`
	`print(confusion_matrix(y_test, y_pred)) # Now you can use confusion_matrix`
	
	`precision recall f1-score support`
	
	`antithyroid treatment 0.00 0.00 0.00 7`
	`binding protein 0.86 0.86 0.86 74`
	`general health 0.88 0.99 0.93 85`
	`hyperthyroid conditions 0.86 0.82 0.84 38`
	`hypothyroid conditions 0.95 1.00 0.97 122`
	`miscellaneous 0.96 0.84 0.90 51`
	`replacement therapy 0.97 0.94 0.96 71`
	
	`accuracy 0.92 448`
	`macro avg 0.78 0.78 0.78 448`
	`weighted avg 0.90 0.92 0.91 448`
	
	`[[ 0 0 1 0 6 0 0]`
	`[ 0 64 6 4 0 0 0]`
	`[ 0 0 84 0 0 0 1]`
	`[ 0 5 0 31 0 1 1]`
	`[ 0 0 0 0 122 0 0]`
	`[ 0 4 2 1 1 43 0]`
	`[ 0 1 2 0 0 1 67]]`
		94.20%	`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]]`

XGB Classifier	`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%	`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]]`

SVC model		86.61%