

## Model Development Phase Template

Date	09 JULY 2024
Team ID	739734
Project Title	Evolving efficient classification patterns in Lymphography
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:

```
y_pred = svr.predict(x_test)
print("Prediction Evaluation using SVR Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))
```

```
y_pred = lassoReg.predict(x_test)
print("Prediction Evaluation using lasso Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))
```

```
# Assuming 'x_test' is available in the environment and is a pandas DataFrame or a NumPy array.
y_pred = linReg.predict(x_test) # Predict on the entire x_test dataset
```

```
print("Prediction Evaluation using Linear Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))
```

```
y_pred = dt.predict(x_test)
print("Prediction Evaluation using Random Regression")
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R-squared:', r2_score(y_test, y_pred))
```

75]

## Model Validation and Evaluation Report:

Model	Classification Report	Accuracy
Random Forest	<pre>accuracy_score(y_test,prediction)  0.8666666666666667</pre>	0.86
Linear regression	<pre>accuracy = linReg.score(x_test,y_test) print(accuracy)  0.30857468451341064</pre>	0.30
Lasso regression	<pre>y_pred = lassoReg.predict(x_test)  (variable) accuracy: Float accuracy = lassoReg.score(x_test,y_test) print(accuracy)  -1.3421120758942114</pre>	-1.42

Decision tree

```
> accuracy = dt.score(x_test,y_test)
> print(accuracy)
-8.92565055762082
```

-8.92