

Model Development Phase

Date	06 July 2024
Team ID	739824
Project Title	SmartLender – Envisioning Success: Predicting University Scores With Machine Learning
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 a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code :

```
linReg = LinearRegression()  
linReg.fit(x_train,y_train)
```



▼ LinearRegression
LinearRegression()

```
[38] y_pred = linReg.predict(x_test)
```

✓
1s [40] `lassoReg = linear_model.Lasso(alpha = 0.1)`
`lassoReg.fit(x,y)`



▼ Lasso
Lasso(alpha=0.1)

✓
0s [41] `y_pred = lassoReg.predict(x_test)`

✓
0s [43] `svr = SVR().fit(x,y)`

✓
0s [44] `y_pred = svr.predict(x_test)`

✓
0s [46] `dt = DecisionTreeRegressor(random_state = 0)`
`dt.fit(x,y)`



▼ DecisionTreeRegressor
DecisionTreeRegressor(random_state=0)

✓
0s [47] `y_pred = dt.predict(x_test)`

✓
0s [49] rf = RandomForestRegressor(n_estimators = 100 , random_state = 0)
rf.fit(x,y)



RandomForestRegressor
RandomForestRegressor(random_state=0)

✓
0s [50] y_pred = rf.predict(x_test)

Model Validation and Evaluation Report :

Model	Summary	Training and Validation Performance Metrics
Linear Regression	<pre># Assuming 'y_test' is available in the environment and is a pandas DataFrame or a NumPy array y_pred = linreg.predict(x_test) # Result 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))</pre>	<pre>Prediction Evaluation using Linear Regression Mean Absolute Error: 0.9264657671450711 Mean Squared Error: 1.7890643253705259 Root Mean Squared Error: 1.33750092294066 R-squared: 0.7439493774502185</pre>
Lasso Regression	<pre>[50] y_pred = lassoeg.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))</pre>	<pre>Prediction Evaluation using Lasso Regression Mean Absolute Error: 0.9352851280018133 Mean Squared Error: 1.7857764888364731 Root Mean Squared Error: 1.3363294886433303 R-squared: 0.7444399332854582</pre>
Support Vector Machine	<pre>[50] y_pred = svm.predict(x_test) print("Prediction Evaluation using Support Vector 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))</pre>	<pre>Prediction Evaluation using Support Vector Regression Mean Absolute Error: 0.5454348693726399 Mean Squared Error: 1.3812917299771091 Root Mean Squared Error: 1.179878460150395 R-squared: 0.8223400225968596</pre>
Decision Tree	<pre>[50] y_pred = dt.predict(x_test) print("Prediction Evaluation using Decision 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))</pre>	<pre>Prediction Evaluation using Decision Regression Mean Absolute Error: 5.264475724048743e-15 Mean Squared Error: 2.7561365735867209e-28 Root Mean Squared Error: 1.6603616106833456e-14 R-squared: 1.0</pre>

Random Forest

```
[60] y_pred = rf.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))
```

```
➡ Prediction Evaluation using Random Regression
Mean Absolute Error: 0.010686590909090909
Mean Squared Error: 0.0009053502244319952
Root Mean Squared Error: 0.0300891870127082
R-squared: 0.9998704251212489
```