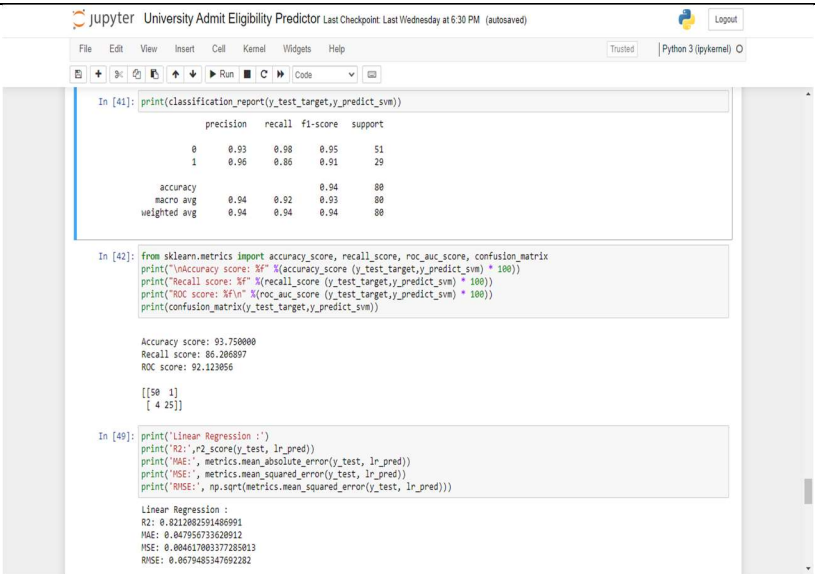
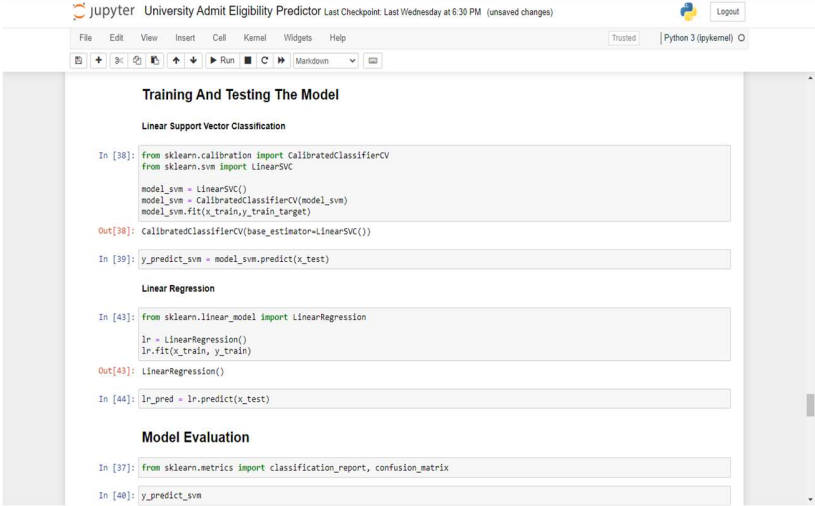


Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID47303
Project Name	UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: MAE - 0.0479567, MSE - 0.0046170, RMSE - 0.067948, R2 score – 0.8212	 <p>The screenshot shows a Jupyter Notebook interface with the following code and output:</p> <pre>In [41]: print(classification_report(y_test_target,y_predict_svm))</pre> <pre> precision recall f1-score support 0 0.93 0.98 0.95 51 1 0.96 0.86 0.91 29 accuracy 0.94 0.94 0.94 80 macro avg 0.94 0.92 0.93 80 weighted avg 0.94 0.94 0.94 80 </pre> <pre>In [42]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix print("Accuracy score: %f" % (accuracy_score(y_test_target,y_predict_svm) * 100)) print("Recall score: %f" % (recall_score(y_test_target,y_predict_svm) * 100)) print("ROC score: %f" % (roc_auc_score(y_test_target,y_predict_svm) * 100)) print(confusion_matrix(y_test_target,y_predict_svm))</pre> <pre> Accuracy score: 93.750000 Recall score: 86.206897 ROC score: 92.123856 [[50 1] [4 25]] </pre> <pre>In [49]: print('Linear Regression :') print('R2: %f' % (r2_score(y_test, lr_pred))) print('MAE: ', metrics.mean_absolute_error(y_test, lr_pred)) print('MSE: ', metrics.mean_squared_error(y_test, lr_pred)) print('RMSE: ', np.sqrt(metrics.mean_squared_error(y_test, lr_pred)))</pre> <pre> Linear Regression : R2: 0.8212082591486991 MAE: 0.047956733628912 MSE: 0.00461703377285813 RMSE: 0.0679485347692282 </pre>
2.	Building the model	Linear Regression, Linear SVM	 <p>The screenshot shows a Jupyter Notebook interface with the following code and output:</p> <pre>In [38]: from sklearn.calibration import CalibratedClassifierCV from sklearn.svm import LinearSVC model_svm = LinearSVC() model_svm = CalibratedClassifierCV(model_svm) model_svm.fit(x_train,y_train_target)</pre> <pre>Out[38]: CalibratedClassifierCV(base_estimator=LinearSVC())</pre> <pre>In [39]: y_predict_svm = model_svm.predict(x_test)</pre> <pre>In [43]: from sklearn.linear_model import LinearRegression lr = LinearRegression() lr.fit(x_train, y_train)</pre> <pre>Out[43]: LinearRegression()</pre> <pre>In [44]: lr_pred = lr.predict(x_test)</pre> <pre>In [37]: from sklearn.metrics import classification_report, confusion_matrix</pre> <pre>In [40]: y_predict_svm</pre>