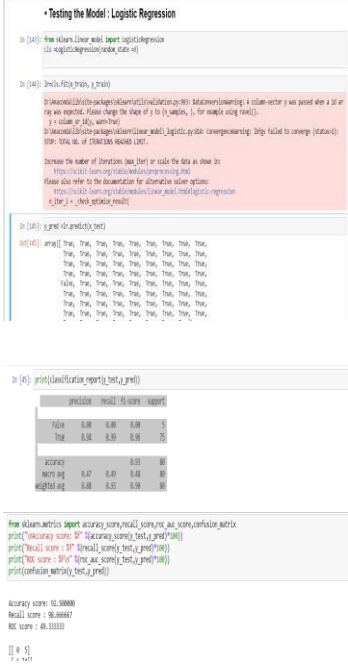



Project Development Phase Model Performance Test

Date	16 November 2022
Team ID	PNT2022TMID32283
Project Name	Project – University Admit Eligibility Predictor
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	<p>Regression Model: Logistic Regression</p> <p>Classification Model:</p> <ul style="list-style-type: none"> Confusion Matrix - $\begin{bmatrix} 0 & 5 \\ 1 & 74 \end{bmatrix}$ Accuracy Score - 92.50000 Recall Score - 98.666667 ROC Score - 49.333333 Classification Report : <pre>precision - 0.88 support - 80 f1-score - 0.90 recall - 0.93</pre> 	 <pre> • Testing the Model : Logistic Regression In [34]: from sklearn.linear_model import LogisticRegression logit = LogisticRegression(max_iter=1000) In [35]: logit.fit(X_train, y_train) In [36]: from sklearn.metrics import classification_report, confusion_matrix y_pred = logit.predict(X_test) print(classification_report(y_test, y_pred)) precision recall f1-score support 0.00 0.00 0.00 1 1.00 0.98 0.99 75 accuracy: 0.92 weighted avg: 0.92 0.98 0.99 80 Confusion matrix: [[0 5] [1 74]] </pre>
2.	Tune The Model	<p>Hyper parameter Tuning:(GridSearchCV)</p> <p>clf.best_score_ - 0.921875</p> <p>Validation Method – GridSearchCV (estimator=SVC ()</p>	 <pre> from sklearn.model_selection import GridSearchCV # Define the parameter grid parameters = { 'C': [0.1, 1, 10, 100, 1000], 'gamma': [0.001, 0.01, 0.1, 1, 10] } # Create the GridSearchCV object grid_search = GridSearchCV(SVC(), parameters, cv=5) # Fit the model grid_search.fit(X_train, y_train) # Print the best parameters and the best score print("Best parameters found: ", grid_search.best_params_) print("Best score: ", grid_search.best_score_) </pre>