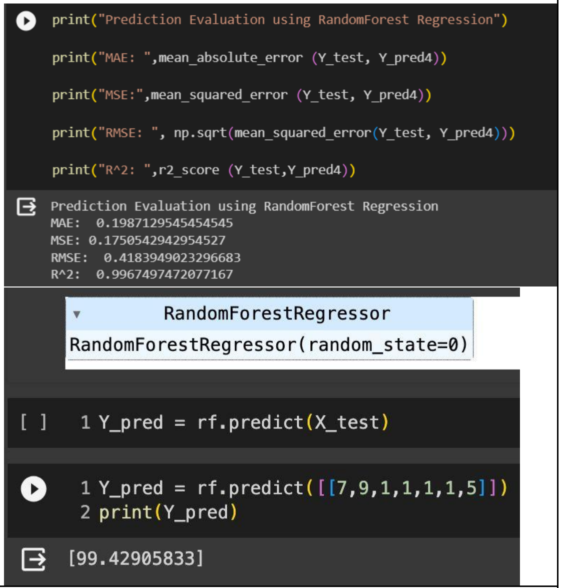
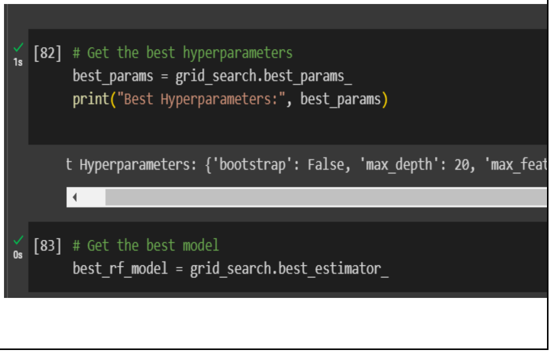
Project Development Phase
Model Performance Test

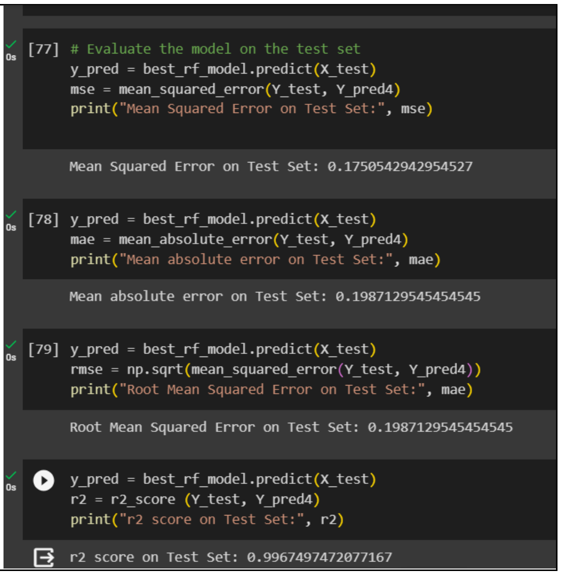
Date	14 November 2023
Team ID	591606
Project Name	ENVISIONING SUCCESS: Predicting University Scores using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Regression Model:**
		MAE - , MSE - , RMSE - , R2 score -
			After performing all the Regression Models we found the best model as Random Forest Regression with best MAE, MSE, RMSE, R2 Scores
			
			`print("Prediction Evaluation using RandomForest Regression")`
			`print("MAE: ",mean_absolute_error (Y_test, Y_pred4))`
			`print("MSE:",mean_squared_error (Y_test, Y_pred4))`
			`print("RMSE: ", np.sqrt(mean_squared_error(Y_test, Y_pred4)))`
			`print("R^2: ",r2_score (Y_test,Y_pred4))`
			`Prediction Evaluation using RandomForest Regression`
			`MAE:  0.1987129545454545`
			`MSE: 0.1750542942954527`
			`RMSE:  0.4183949023296683`
			`R^2:  0.9967497472077167`
			`RandomForestRegressor`
			`RandomForestRegressor(random_state=0)`
			`[ ] Y_pred = rf.predict(X_test)`
			`Y_pred = rf.predict([[7,9,1,1,1,1,5]])`
			`print(Y_pred)`
			`[99.42905833]`

2.	Tune the Model	Hyperparameter Tuning -
		Validation Method -
			Using GridSearch for random forest model
			
			`[82] # Get the best hyperparameters`
			`best_params = grid_search.best_params_`
			`print("Best Hyperparameters:", best_params)`
			`t Hyperparameters: {'bootstrap': False, 'max_depth': 20, 'max_feat`
			`[83] # Get the best model`
			`best_rf_model = grid_search.best_estimator_`

```
[77] # Evaluate the model on the test set
     y_pred = best_rf_model.predict(X_test)
     mse = mean_squared_error(Y_test, Y_pred4)
     print("Mean Squared Error on Test Set:", mse)

     Mean Squared Error on Test Set: 0.1750542942954527

[78] y_pred = best_rf_model.predict(X_test)
     mae = mean_absolute_error(Y_test, Y_pred4)
     print("Mean absolute error on Test Set:", mae)

     Mean absolute error on Test Set: 0.1987129545454545

[79] y_pred = best_rf_model.predict(X_test)
     rmse = np.sqrt(mean_squared_error(Y_test, Y_pred4))
     print("Root Mean Squared Error on Test Set:", mae)

     Root Mean Squared Error on Test Set: 0.1987129545454545

     y_pred = best_rf_model.predict(X_test)
     r2 = r2_score (Y_test, Y_pred4)
     print("r2 score on Test Set:", r2)

     r2 score on Test Set: 0.9967497472077167
```