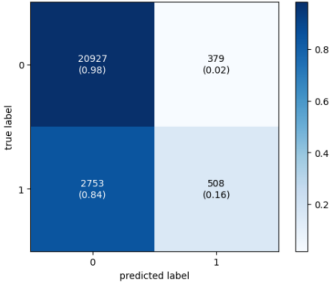


Model Performance Test

Date	4 November 2023
Team ID	592321
Project Name	Diabetes Prediction using Machine Learning

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	<p>Regression Model: MAE - , MSE - , RMSE - , R2 score -</p> <p>Classification Model: Confusion Matrix - , Accuracy Score- & Classification Report -</p>	<p>Random Forest Classifier</p> <pre>[] rf=RandomForestClassifier(max_depth=12,n_estimators=10,random_state=30) rf.fit(x_train,y_train) [] rf=RandomForestClassifier(max_depth=12,n_estimators=10,random_state=30) [] y_pred=rf.predict(x_test) print('Training set score: {:.4f}'.format(rf.score(x_train, y_train))) print('Test set score: {:.4f}'.format(rf.score(x_test, y_test))) Training set score: 0.9227 Test set score: 0.8859 [] mse=mean_squared_error(y_test, y_pred) print('Mean Squared Error: ' + str(mse)) rmse = (mean_squared_error(y_test, y_pred))**(0.5) print('Root Mean Squared Error: ' + str(rmse)) Mean Squared Error: 0.1341230185425975 Root Mean Squared Error: 0.3661280594893495</pre> <p>Decision Tree Classifier</p> <pre>[] dt=DecisionTreeClassifier(max_depth=12) dt.fit(x_train,y_train) [] dt=DecisionTreeClassifier(max_depth=12) [] y_pred=dt.predict(x_test) print('Training set score: {:.4f}'.format(dt.score(x_train, y_train))) print('Test set score: {:.4f}'.format(dt.score(x_test, y_test))) Training set score: 0.9355 Test set score: 0.8293 [] mse=mean_squared_error(y_test, y_pred) print('Mean Squared Error: ' + str(mse)) rmse = (mean_squared_error(y_test, y_pred))**(0.5) print('Root Mean Squared Error: ' + str(rmse)) Mean Squared Error: 0.1786761102391693 Root Mean Squared Error: 0.4131265304994693 cr1 = confusion_matrix(y_test,y_pred) plot_confusion_matrix(conf_mat=cr1, show_absolute=True, show_normalized=True, colorbar=True) plt.show()</pre>

			<div>Logistic Regression</div> <pre>[] lg=LogisticRegression(max_iter=1500) lg.fit(x_train,y_train) + LogisticRegression LogisticRegression(max_iter=1500) [] y_pred=lg.predict(x_test) print('Training set score: {:.4f}'.format(lg.score(x_train, y_train))) print('Test set score: {:.4f}'.format(lg.score(x_test, y_test))) [] Training set score: 0.8704 Test set score: 0.8725 [] mse =mean_squared_error(y_test, y_pred) print('Mean Squared Error: ' + str(mse)) rmse =(mean_squared_error(y_test, y_pred))**(0.5) print('Root Mean Squared Error: ' + str(rmse)) [] Mean Squared Error: 0.12748889578434486 Root Mean Squared Error: 0.357047490812489</pre> 
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<pre>lg=LogisticRegression(max_iter=1500) rf=RandomForestClassifier(max_depth=12,n_estimators=10,random_state=20) dt=DecisionTreeClassifier(max_depth=12)</pre>