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		Logout
1.	IVICTIOS	Model:	File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ip	kernel) O
			□ □	
		MAE - 0.0479567,	<pre>In [41]: print(classification_report(y_test_target,y_predict_svm))</pre>	
		MSE - 0.0046170,	precision recall fi-score support	
		RMSE - 0.067948,	0 0.93 0.98 0.95 51 1 0.96 0.86 0.91 29	
	R2 score – 0.8212	accuracy 0.94 88 macro avg 0.94 0.92 0.93 88 weighted avg 0.94 0.94 0.94 88		
		In [42]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix print("wheccuracy_score: %" %(accuracy_score (y_test_target.y_predict_swn) * 1809)) print("Recall_score: %F" %(recall_score (y_test_target.y_predict_swn) * 1809)) print("Got_score: %F" %(recall_score (y_test_target.y_predict_swn) * 1809)) print(confusion_matrix(y_test_target.y_predict_swn))		
		Accuracy score: 93.7508000 Recall score: 86.206907 RO: score: 92.123056		
			[[50 1] [4 15]]	
			<pre>In [49]: print('Linear Regression :') print('RZ: 'rZ ycorne(y_test, ln_pred)) print('RZ: 'rZ ycorne(y_test, ln_pred)) print('RSE:, setricio.mema_absolute_error(y_test, ln_pred)) print('RSE:, setricio.mema_squared_error(y_test, ln_pred)) print('RSE:, np.sqr(testic.mema_squared_error(y_test, ln_pred)))</pre>	
			Linear Regression : R2: 0.8212082591480991 MAE: 0.8212082591480991 MSE: 0.004017003377356913 MSEE: 0.004017003377356913 MSEE: 0.0704585374050232	
			Without Heisenin Admit Elschlift. Desdister	
2.	Building	Linear	Jupyter University Admit Eligibility Predictor Last Checkpoint: Last Wednesday at 6:30 PM (unsaved changes) File File More Insert Cell Karnel Widness Nath Total Denne 3 (in) Purpos 3 (Logout
2.	Building the model	Linear Regression,	Jupyter University Admit Eligibility Predictor Last Checkpoint Last Wednesday at 6:30 PM (unsaved changes) File	
2.	•		File Edit Vew Insert Cell Karnel Widgets Help Trusted Python 3 (in 1	
2.	•	Regression,	File Edit View Insert Call Kamel Widgets Help Trouted Python 3 (i) S + 3 € 2	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgets Help Trouted Python 3 (i)	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgets Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearn.calibration import CalibratedClassifierCV from sklearn.svm import LinearSVC model.pun - LinearSVC) model.pun - LinearSVC()	
2.	•	Regression,	File Edit View Insert Call Kamel Widgets Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38] From sklearm.calibration import CalibratedClassifierCV from sklearm.svm import LinearSVC nodel_svm = LinearSVC() nodel_svm =	
2.	•	Regression,	File Edit View Insert Call Kamel Widgets Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearm.callbration import CallbratedClassifierCV from sklearm.svm import LinearSVC nodel_svm = LinearSVC() nodel_svm = LinearSVC() nodel_svm = LinearSVC() to nodel_svm = CallbratedClassifierCV(nodel_svm) out[38]: CallbratedClassifierCV(base_estimator-LinearSVC())	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgets Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearn.calibration import CalibratedClassifierCV from sklearn.svm import LinearSVC model_svm - CalibratedClassifierCV(model_svm) model_svm - CalibratedClassifierCV(model_svm) model_svm - CalibratedClassifierCV(model_svm) model_svm.fit(x_train,y_train_target) Out [38]: CalibratedClassifierCV(pase_estinator=LinearSVC()) In [38]: y_predict_svm - model_svm.predict(x_test)	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgels Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearm.calibration import CalibratedClassifierCV from sklearm.calibration import CalibratedClassifierCV from sklearm.svm import LinearSVC model_svm - LinearSVC() model_svm - CalibratedClassifierCV(model_svm) model_svm.fit(x_train_x_train_target) Out[38]: CalibratedClassifierCV(base_estimator=LinearSVC()) In [38]: y_predict_svm = model_svm.predict(x_test) Linear Regression	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgets Help Trusted Python 3 (t) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearn.calibration import Calibrate/ClassifierCV from sklearn.swn import LinearSVC model_svm - calibrate/ClassifierCV(model_svm) model_svm - calibrate/ClassifierCV(model_svm) model_svm.fit(c_trustn,y_trustn_targex) Out[38]: Calibrate/ClassifierCV(base_estimator-tinearSVC()) In [39]: y_predict_svm = model_svm.predict(x_test) Linear Regression In [43]: from sklearn.linear_model_import Linear*Regression In [43]: from sklearn.linear_model_import Linear*Regression In [43]: from sklearn.linear_model_import Linear*Regression	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgels Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearn.calibration import CalibratedClassifierCV from sklearn.svm import LinearSVC model_svm - LinearSVC) model_svm - LinearSVC() model_svm - CalibratedClassifierCV(model_svm) model_svm.fit(x_tvain,y_tvain_target) Out[38]: calibratedClassifierCV(base_estimator=LinearSVC()) In [38]: [x_predict_svm - model_svm, predict(x_test) Linear Regression In [43]: from sklearn.linear_model import LinearRegression	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgels Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearm.calibration import CalibratedClassifierCV from sklearm.svm import LinearSVC model_svm - LinearSVC) model_svm - LinearSVC() model_svm - LinearSVC() model_svm - LinearSVC() model_svm - LinearSVC() In [38]: (alibratedClassifierCV(model_svm) model_svm.fit(x_train,y_train_target) Union [38]: (x_predict_svm - model_svm.predict(x_test) Linear Regression In [43]: from sklearm.linear_model import LinearRegression In [43]: from sklearm.linear_model import LinearRegression In [43]: LinearRegression()	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgels Help Trusted Python 3 (i) Training And Testing The Model Linear Support Vector Classification In [38]: from sklearn.calibration import CalibratedClassifierCV from sklearn.svm import LinearSVC model_svm - LinearSVC) model_svm - LinearSVC() model_svm - CalibratedClassifierCV(model_svm) model_svm.fit(x_tvain,y_tvain_target) Out[38]: calibratedClassifierCV(base_estimator=LinearSVC()) In [38]: [x_predict_svm - model_svm, predict(x_test) Linear Regression In [43]: from sklearn.linear_model import LinearRegression	
2.	•	Regression,	File Edit View Insert Cell Kamel Widgets Help Trusted Python 3 (t) Training And Testing The Model Linear Support Vector Classification In [38]: from sklaarm.calibration Import CalibratedClassifierCV from sklaarm.calibration Import LinearSVC model_sva = LinearSVC) model_sva = LinearSVC) model_sva = LinearSVC) model_sva = LinearSVC) model_sva = LinearSVC (and state the state that the state t	