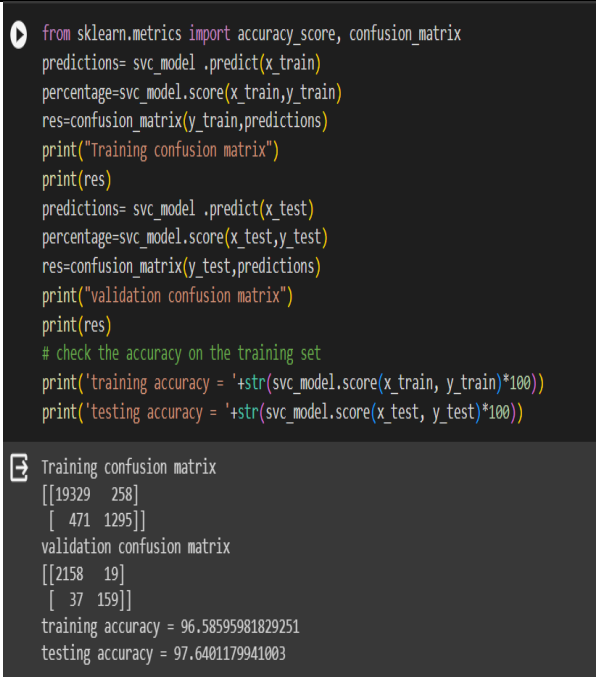


## Project Development Phase Model Performance Test

Date	10 November 2023
Team ID	592049
Project Name	PREDICTING LUMPY SKIN DISEASE
Maximum Marks	10 Marks

### Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	<b>Regression Model:</b> MAE - , MSE - , RMSE - , R2 score -  <b>Classification Model:</b> Confusion Matrix - , Accuray Score- & Classification Report -	 <pre> from sklearn.metrics import accuracy_score, confusion_matrix predictions= svc_model .predict(x_train) percentage=svc_model.score(x_train,y_train) res=confusion_matrix(y_train,predictions) print("Training confusion matrix") print(res) predictions= svc_model .predict(x_test) percentage=svc_model.score(x_test,y_test) res=confusion_matrix(y_test,predictions) print("validation confusion matrix") print(res) # check the accuracy on the training set print('training accuracy = '+str(svc_model.score(x_train, y_train)*100)) print('testing accuracy = '+str(svc_model.score(x_test, y_test)*100)) </pre> <pre> Training confusion matrix [[19329  258]  [ 471 1295]] validation confusion matrix [[2158  19]  [ 37 159]] training accuracy = 96.58595981829251 testing accuracy = 97.6401179941003 </pre>

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<pre> import pandas as pd from sklearn.model_selection import train_test_split from sklearn.svm import SVC from sklearn.preprocessing import StandardScaler from sklearn.pipeline import make_pipeline import pickle from sklearn.model_selection import train_test_split  X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42) # Load your dataset df = pd.read_csv('/content/archive (10).zip') # Replace 'your_dataset.csv' with the actual filename  # One-hot encode categorical columns df = pd.get_dummies(df, columns=['region', 'country'])  # Assuming 'target_column' is the column you want to predict X = df.drop('wet', axis=1) y = df['wet']  # Split the data into training and testing sets X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  # Define the SVC model with a pipeline including standard scaling model = make_pipeline(StandardScaler(), SVC())  # Save the model with open('tuned_svc_model.pkl', 'wb') as file:     pickle.dump(model, file) </pre>
----	----------------	---	---