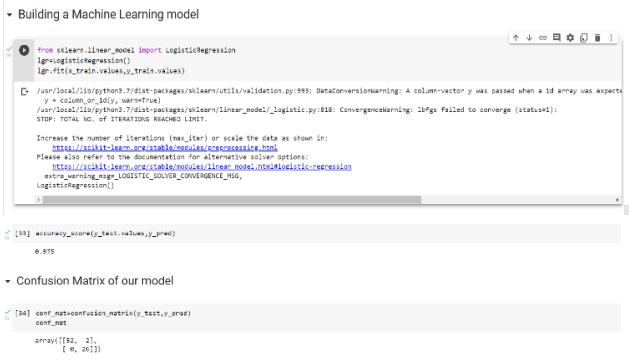


Project Development Phase Model Performance Test

Date	23 November 2022
Team ID	PNT2022TMID39626
Project Name	Project – Early detection of chronic kidney disease using machine learning.
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	<p>Regression Model: Logistic Regression Model.</p> <p>Classification Model: Confusion Matrix – array([[52, 2],[0, 26]])</p> <p>Accuracy Score – 0.975</p>	 <pre> Building a Machine Learning model from sklearn.linear_model import LogisticRegression logisticRegression() log.fit(train_values,y_train.values) /usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:93: DataConversionWarning: A column-vector y was passed when a 1d array was expected: y = column_or_1d(y, warn=True) /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1): STOP: TOTAL NO. of ITERATIONS REACHED LIMIT. Increase the number of iterations (max_iter) or scale the data as shown in: https://scikit-learn.org/stable/modules/preprocessing.html Please also refer to the documentation for alternative solver options: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression extra_warning_msg="LOGISTIC_SOLVER_CONVERGENCE_MSG", logisticRegression() [33]: accuracy_score(y_test_values,y_pred) 0.975 Confusion Matrix of our model [34]: conf_matrix(confusion_matrix(y_test,y_pred) conf_mat array([[52, 2], [0, 26]]) </pre>

2.

Tune the Model

Splitting the data into train and test.

Label Encoding.

Independent and

Dependent variables.

Splitting the data into train and test

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(320, 8)
(320, 1)
(80, 8)
(80, 1)
```

Label Encoder

```
from sklearn.preprocessing import LabelEncoder
```

```
for i in catcols:
    print('LABEL ENCODING OF:', i)
    le = LabelEncoder()
    print(data[i])
    data[i] = le.fit_transform(data[i])
    print(data[i])
    print('--')
```

LABEL ENCODING OF: appetite

Counter({'good': 118, 'poor': 82})

Counter((0, 118), (1, 82))

LABEL ENCODING OF: doctorvis

Counter({'no present': 176, 'present': 82})

Counter((0, 176), (1, 82))

LABEL ENCODING OF: red_blood_cells

Counter({'normal': 176, 'abnormal': 144})

Counter((1, 176), (0, 144))

LABEL ENCODING OF: pus_cells

Counter({'normal': 208, 'abnormal': 112})

Counter((1, 208), (0, 112))

Creating Independent and Dependent

```
features = ['red_blood_cells', 'pus_cells', 'total glucose random', 'blood_sugar', 'total_cholesterol', 'triglycerides', 'coronary_artery_disease']
x = pd.DataFrame(data, columns=features)
y = pd.DataFrame(data, columns=['class'])
print(x.shape)
print(y.shape)
```

```
(400, 8)
(400, 1)
```