

## Performance testing

Date	18-NOV-2022
Project Name	Web phishing detection
Team ID	PNT2022TMID43023
Marks	10marks

### 1.Metrics:

#### Regression Model:

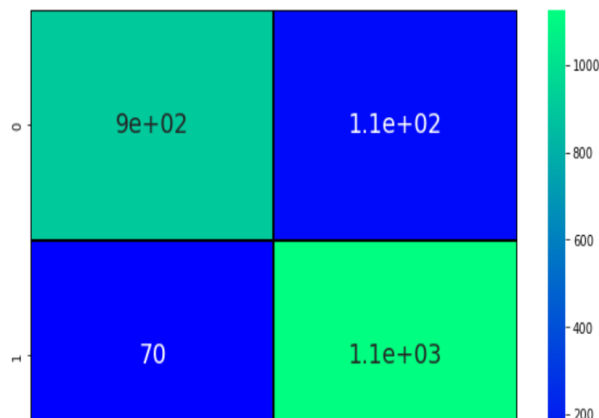
MAE-, MSE-, RMSE-, R2 Score

#### Classification Model:

Confusion Matrix-, Accuracy score- & Classification report-

```
In [14]: cm=confusion_matrix(y_test,y_pred1)
plt.figure(figsize=(10,6))
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
#print(round(accuracy_score(prediction3,y_test)*100,2))
#print('Testing Accuracy for knn',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for logistic regression',(TP/(TP+FN)))
print('Testing Specificity for logistic regression',(TN/(TN+FP)))
print('Testing Precision for logistic regression',(TP/(TP+FP)))
print('Testing accuracy for logistic regression',accuracy_score(y_test,y_pred1))
```

Testing Sensitivity for logistic regression 0.9279835390946503  
Testing Specificity for logistic regression 0.9096045197740112  
Testing Precision for logistic regression 0.8895463510848126  
Testing accuracy for logistic regression 0.9176843057440073



## 2.TUNE THE MODEL:

Hyperparameters tuning-

Validation Method

```
In [58]: #HYPERPARAMETER TUNING  
grid.fit(X_train, y_train)
```

```
Out[58]: 

GridSearchCV  
GridSearchCV(cv=5,  
             estimator=GradientBoostingClassifier(learning_rate=0.7,  
                                                  max_depth=4),  
             param_grid={'max_features': array([1, 2, 3, 4, 5]),  
                        'n_estimators': array([ 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130,  
140, 150, 160, 170, 180, 190, 200])})  
             estimator: GradientBoostingClassifier  
             GradientBoostingClassifier(learning_rate=0.7, max_depth=4)  
             GradientBoostingClassifier  
             GradientBoostingClassifier(learning_rate=0.7, max_depth=4)


```

```
In [59]: print("The best parameters are %s with a score of %0.2f"  
            % (grid.best_params_, grid.best_score_))
```

```
The best parameters are {'max_features': 5, 'n_estimators': 200} with a score of 0.97
```