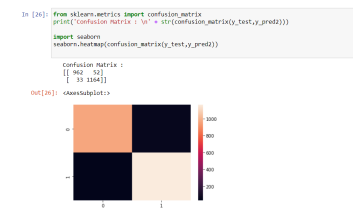
Project Development Phase
Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID23587
Project Name	Project – Web phishing detection
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	**Regression Model:** Decision tree MAE – 0.075 MSE - 0.15 RMSE – 0.387 R2 score – 0.84 **Classification Model:** Confusion Matrix Accuracy Score-0.96	  
2.	Tune the Model	Hyperparameter Tuning - Grid Search Cross Validation	

Metrics :

1.Regression Model:

In [28]:
```
mae = mean_absolute_error(y_test, y_pred2)
mse = mean_squared_error(y_test, y_pred2)
rmse = np.sqrt(mse)
rmsle = np.log(rmse)
n,k = x_train.shape
r2=r2_score(y_test,y_pred2)
adj_r2= 1 - ((1-r2)*(n-1)/(n-k-1))
print(mae,mse,rmse,rmsle,r2,adj_r2)
```

```
0.07507914970601538 0.15015829941203077 0.38750264439359733 -0.9480326059704789 0.8488059398990573 0.8482912659170956
```

2.Classification Model:

In [25]:
```
print('Accuracy Score : ' + str(accuracy_score(y_test,y_pred2)))

from sklearn.metrics import confusion_matrix
print('Confusion Matrix : \n' + str(confusion_matrix(y_test,y_pred2)))
```

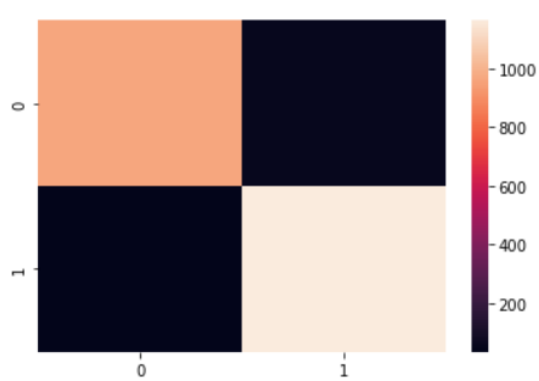
```
Accuracy Score : 0.9624604251469923
Confusion Matrix :
[[ 960   54]
 [  29 1168]]
```

In [26]:
```
from sklearn.metrics import confusion_matrix
print('Confusion Matrix : \n' + str(confusion_matrix(y_test,y_pred2)))

import seaborn
seaborn.heatmap(confusion_matrix(y_test,y_pred2))
```

```
Confusion Matrix :
[[ 962   52]
 [  33 1164]]
```

Out[26]: <AxesSubplot:>



3.Tuning the Model:

In [33]:
```
from sklearn.model_selection import GridSearchCV
grid_values = {'penalty': ['l1', 'l2'],'C':[0.001,.009,0.01,.09,1,5,10,25]}
grid_clf_acc = GridSearchCV(clf, param_grid = grid_values,scoring = 'recall')
grid_clf_acc.fit(x_train, y_train)

y_pred_acc = grid_clf_acc.predict(x_test)

print('Accuracy Score : ' + str(accuracy_score(y_test,y_pred_acc)))
print('Precision Score : ' + str(precision_score(y_test,y_pred_acc)))
print('Recall Score : ' + str(recall_score(y_test,y_pred_acc)))
print('F1 Score : ' + str(f1_score(y_test,y_pred_acc)))
```

```
Accuracy Score : 0.9185888738127544
Precision Score : 0.9130787977254264
Recall Score : 0.9390142021720969
F1 Score : 0.9258649093904447
```