



Ahmedabad
University

CSE 523 Machine Learning
Weekly Report Progress
(08-04-23)

Topic - Password strength checker

Group name - Predictors_4.0

NAME	ENROLLMENT NUMBER
Zalak Shah	AU2040217
Freya Shah	AU2040228
Freya Modi	AU2040222
Avinash Raval	AU2040031

Up til last week we completed training of the logistic regression model. So, this week we tested the model against the test dataset and evaluated its performance.

We did the following things:

- Tested the model against the X_{test} data.

Testing

```
[ ] def test(X, y, theta):  
    m = len(y)  
    h = sigmoid(np.dot(X, theta))  
    predictions = (h >= 0.8).astype(int)  
    accuracy = np.mean(predictions == y)  
  
    return accuracy  
print(weights)  
acc = test(X_test, y_test, weights)  
print(acc)  
  
[ 0.          0.62438398 11.37436986  2.26979361  2.16028848  1.57822711  
 0.90403387  0.24002652  0.62282317  0.13341262]  
0.7431231706588615
```

- Then we found the performance of the model using $f1_score$, precision, recall and accuracy.

```
# test the model on the data  
from sklearn.metrics import accuracy_score, recall_score, f1_score, precision_score  
  
y_pred = sigmoid(X_test @ weights)  
y_pred = np.round(y_pred)  
accuracy = accuracy_score(y_test, y_pred)  
precision = precision_score(y_test, y_pred, average='weighted')  
recall = recall_score(y_test, y_pred, average='weighted')  
f1 = f1_score(y_test, y_pred, average='weighted')  
print("Accuracy:", accuracy)  
print("Precision:", precision)  
print("Recall:", recall)  
print("F1 Score:", f1)  
  
Accuracy: 0.7431231706588615  
Precision: 0.5522320467700794  
Recall: 0.7431231706588615  
F1 Score: 0.6336121922598826
```

- After this we checked the model for overfitting by finding its performance on the test and train dataset.

```
▶ y_pred_test = sigmoid(X_test @ weights)
  y_pred_train = sigmoid(X_train @ weights)
  y_pred_test = np.round(y_pred_test)
  y_pred_train = np.round(y_pred_train)
  accuracy_test = accuracy_score(y_test, y_pred_test)
  accuracy_train = accuracy_score(y_train, y_pred_train)
  print("accuracy on the training set:", accuracy_train)
  print("accuracy on the testing set:", accuracy_test)
```

```
☞ accuracy on the training set: 0.7415850307627979
  accuracy on the testing set: 0.7431231706588615
```