

# Universidade Federal de Uberlândia

PGC308A Tópicos Especiais em Sistemas de Computação 2: Internet do Futuro

Rodrigo Moreira – [moreira\\_r@outlook.com](mailto:moreira_r@outlook.com)

2017-2

## 1 Task III: Estimating SLA Conformance and Violation from Device Statistics

1. Model Training - provide the coefficients of your model  $C$ :

Coefficients of the model  $C$ :

$$\Theta_1 = -5.67750597\text{e-}02 \quad \Theta_2 = -4.26722323\text{e-}02$$

$$\Theta_3 = 6.32365126\text{e-}03 \quad \Theta_4 = -4.57938573\text{e-}07$$

$$\Theta_5 = 3.52435923\text{e-}03 \quad \Theta_6 = 3.03572012\text{e-}04$$

$$\Theta_7 = -8.26688033\text{e-}02 \quad \Theta_8 = -7.17041206\text{e-}02$$

$$\Theta_9 = -7.46929691\text{e-}06$$

2. Accuracy of the Classifier  $C$ :

Classification Error (ERR) - Logistic Regression-based  $\approx 0.114$ , i.e.  $\approx 11,4\%$ .

3. Accuracy of the Classifier  $C$  (by considering the Naïve method):

Classification Error (ERR) - Naive-based  $\approx 0.495$ , i.e.  $\approx 49.5\%$ .

4. New classifier extending the linear regression:

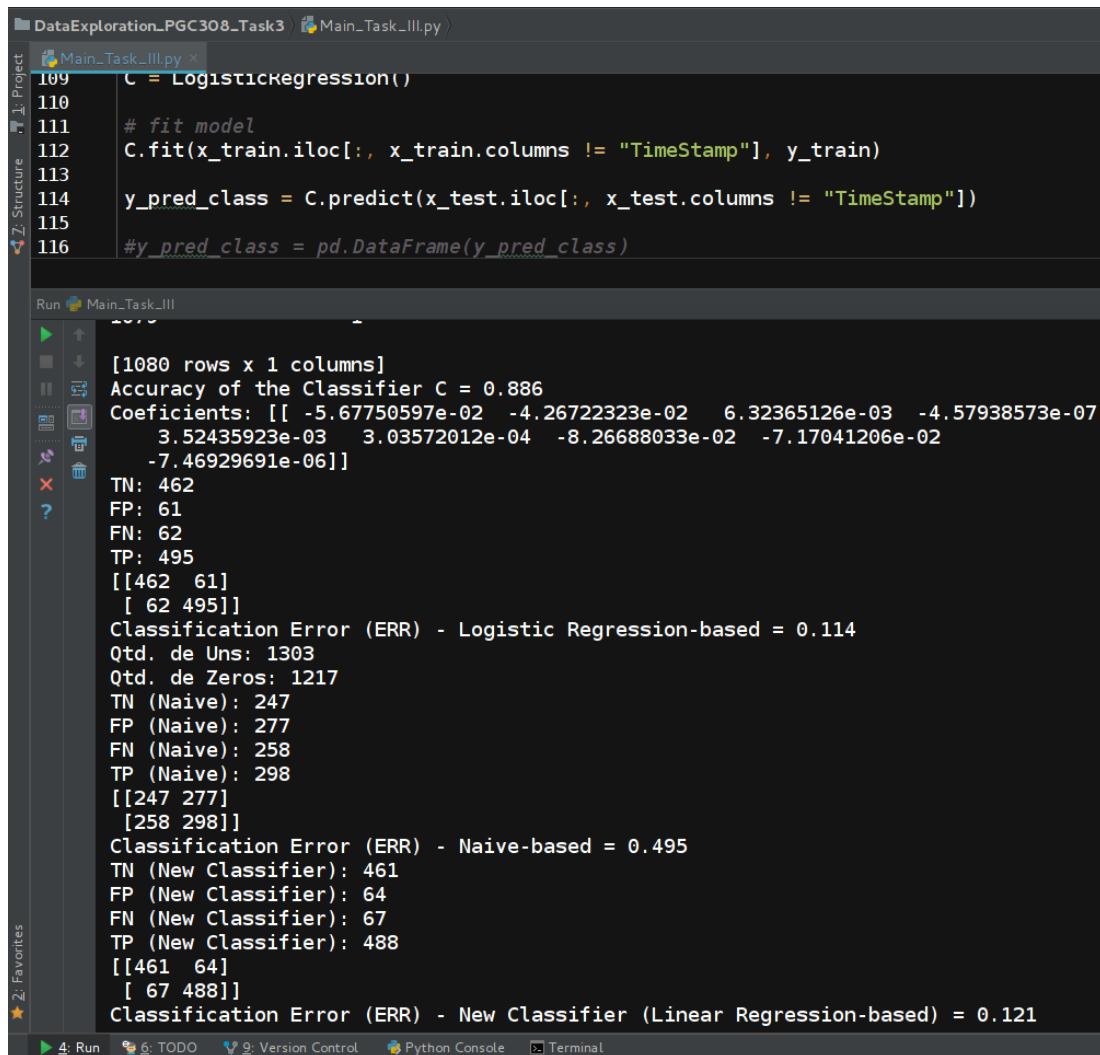
Classification Error (ERR) - New Classifier (Linear Regression-based)  $\approx 0.121$ , i.e.  $\approx 12.1\%$ .

5. Observations and conclusions based on the above work:

The confusion matrix allows visualization of classifier performance. Allowing to check the number of correct classifications as opposed to the classifications predicted for each class. To the resolutions of the questions raised, it is possible to conclude that the accuracy of the classifier using Logistic Regression is sufficiently better than using a Naïve approach. In addition, it is possible to conclude that the accuracy, measured by the metrics (ERR) when using learning-based on Linear Regression as the learning engine for the classifier, we obtain results numerically lower than Logistic Regression. Thus, it is possible to conclude that using the Logistic Regression approach will allow higher classification rates considering the engines proposed for the classifiers.

The codes used to solve these questions are available in the following link: [https://github.com/romoreira/MLN/blob/master/Main\\_Task\\_III.py](https://github.com/romoreira/MLN/blob/master/Main_Task_III.py)

The Figure 1 depicts the output of execution of our Python Script.



The image shows a PyCharm IDE window with a Python script and its execution output. The script defines a Logistic Regression classifier and evaluates its performance against a Naive classifier. The console output displays various metrics including accuracy, coefficients, confusion matrix, and classification error rates.

```
109 C = LogisticRegression()
110
111 # fit model
112 C.fit(x_train.iloc[:, x_train.columns != "TimeStamp"], y_train)
113
114 y_pred_class = C.predict(x_test.iloc[:, x_test.columns != "TimeStamp"])
115
116 #y_pred_class = pd.DataFrame(y_pred_class)
```

```
[1080 rows x 1 columns]
Accuracy of the Classifier C = 0.886
Coefficients: [[ -5.67750597e-02 -4.26722323e-02  6.32365126e-03 -4.57938573e-07
  3.52435923e-03  3.03572012e-04 -8.26688033e-02 -7.17041206e-02
 -7.46929691e-06]]
TN: 462
FP: 61
FN: 62
TP: 495
[[462  61]
 [ 62 495]]
Classification Error (ERR) - Logistic Regression-based = 0.114
Qtd. de Uns: 1303
Qtd. de Zeros: 1217
TN (Naive): 247
FP (Naive): 277
FN (Naive): 258
TP (Naive): 298
[[247 277]
 [258 298]]
Classification Error (ERR) - Naive-based = 0.495
TN (New Classifier): 461
FP (New Classifier): 64
FN (New Classifier): 67
TP (New Classifier): 488
[[461  64]
 [ 67 488]]
Classification Error (ERR) - New Classifier (Linear Regression-based) = 0.121
```

Figure 1: Console Output – Pycharm IntelliJ