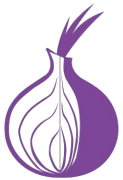
Classificação de tráfego TOR-nonTOR

Djenifer Renata Pereira

Ciência de Dados para Segurança
2021

Objetivo

A partir de amostras de fluxos de 10 segundos de uma rede encriptada, conseguir classificar esse fluxo como Tor ou Não-Tor.



Introdução

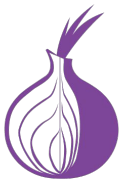
- Tor-nonTor dataset (ISCXTor2016) [1]
- Arquivo usado: SelectedFeatures-10s-TOR-NonTOR.csv
- 29 colunas

- Source IP
- Source Port
- Destination IP
- Destination Port
- Protocol

- Flow Duration
- Flow Bytes/s
- Flow Packets/s
- label

- Flow IAT (4)
- Fwd IAT (4)
- Bwd IAT (4)
- Active (4)
- IDLE (4)

mean, std, max, min



Exploração de dados

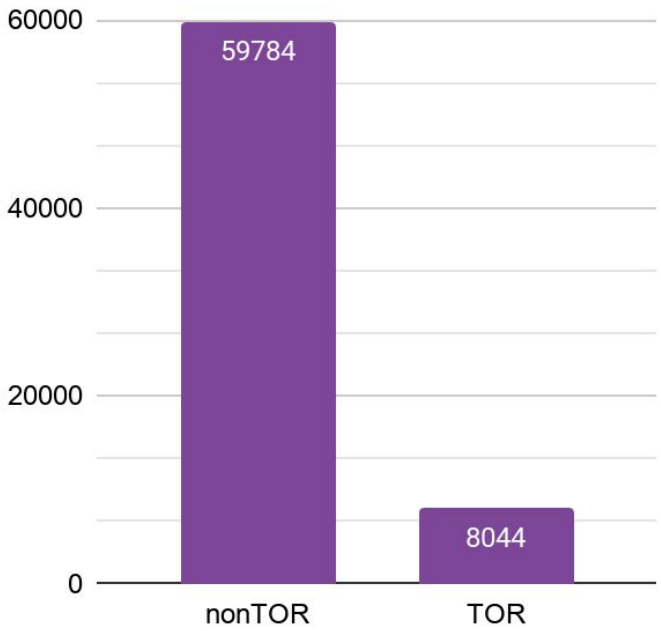
- Dados majoritariamente numéricos
- Má distribuição das classes
- Colunas pouco relevantes (10 + 4)

tráfego

- Flow Bytes/s
- Flow Packets/s
- Active (4)
- IDLE(4)

origem/destino

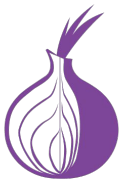
- Source IP
- Source Port
- Destination IP
- Destination Port



Extração de características

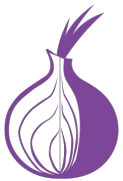
- Matriz de correlação entre 14 colunas
- Atributos `flow_iat_<x>` altamente relacionados a `fwd_iat_<x>`
- Colunas `<y>_iat_mean` altamente relacionadas a `<y>_iat_min`
- Colunas `<z>_iat_std` altamente relacionadas a `<z>_iat_max`
- Remoção de 8 colunas (15 - 8 = 7)

- Protocol
- Flow Duration
- Fwd IAT (max, min)
- Bwd IAT (max, min)
- label



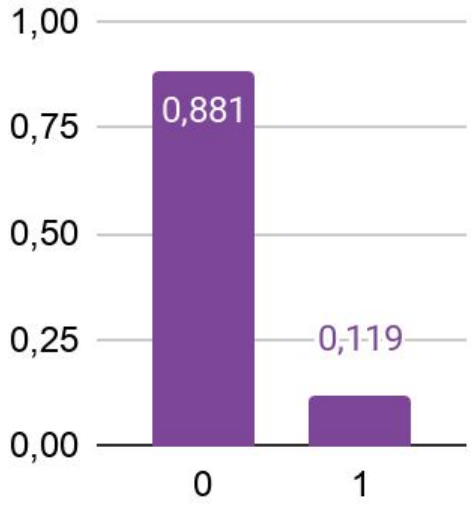
Machine Learning

Passos a serem seguidos

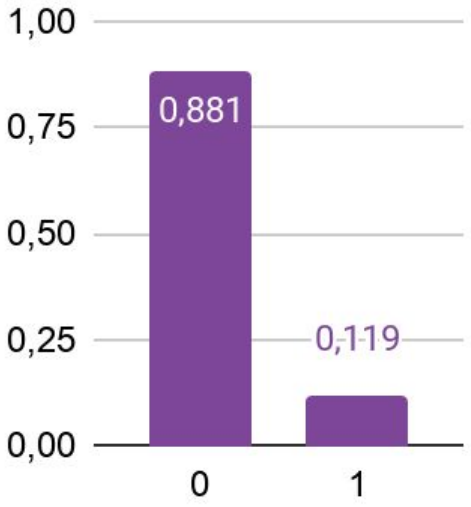


1. Separar o dataset em 80% e 20%
2. Treinar com a parte de 80%
    a. Validar modelos com split 80/20
    b. Validar modelos com KFold de 5
3. Testar com os 20%

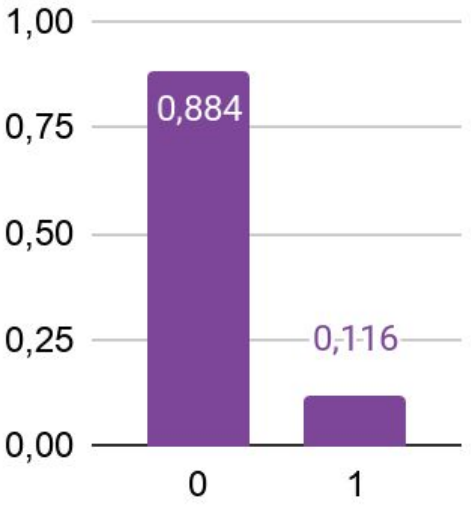
1. Separação do dataset



Dataset inteiro



Dataset 80%



Dataset 20%

2. Treinamento

- Normalização usando MinMaxScaler
- Foram treinados 4 modelos

KNN

n_neighbors	4
weights	‘distance’

Random Forest

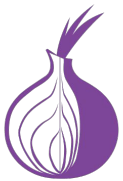
n_estimators	75
random_state	21

SVM

class_weight	‘balanced’
random_state	21
cache_size	500

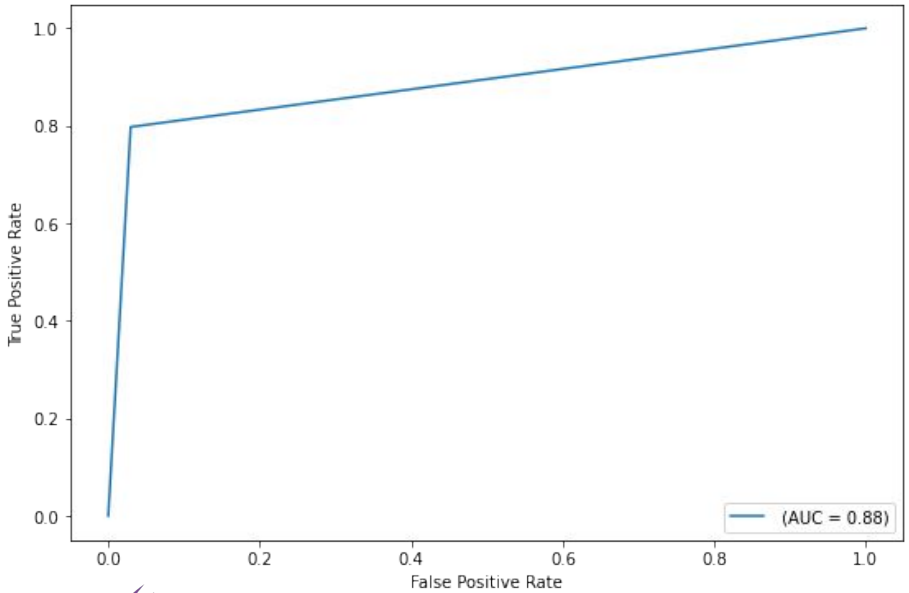
Perceptron

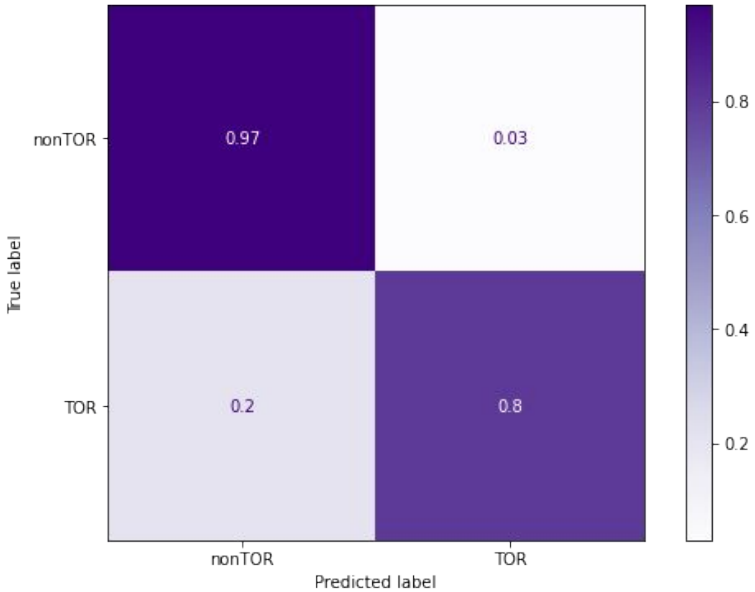
penalty	‘elasticnet’
alpha	0.0005
random_state	21



➔ KNN

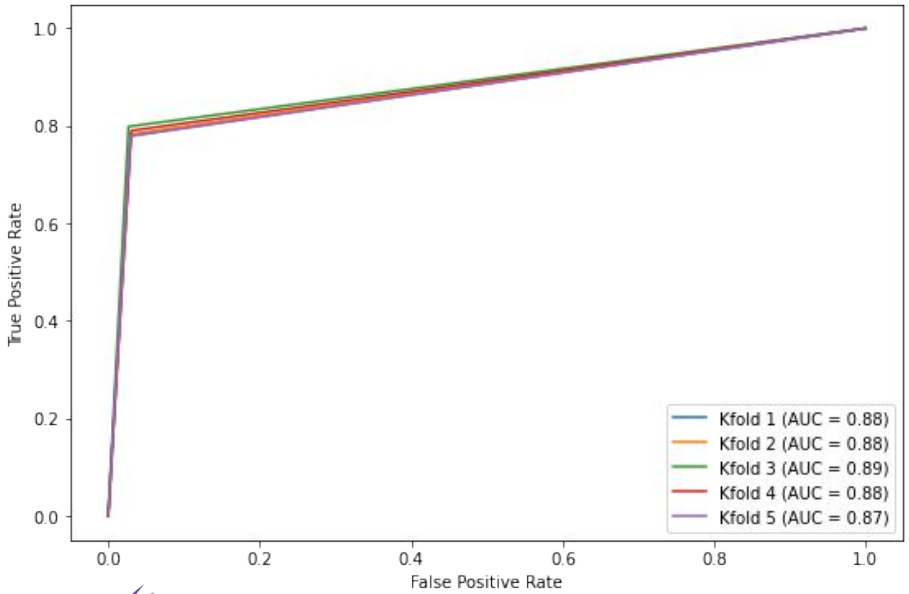
80/20

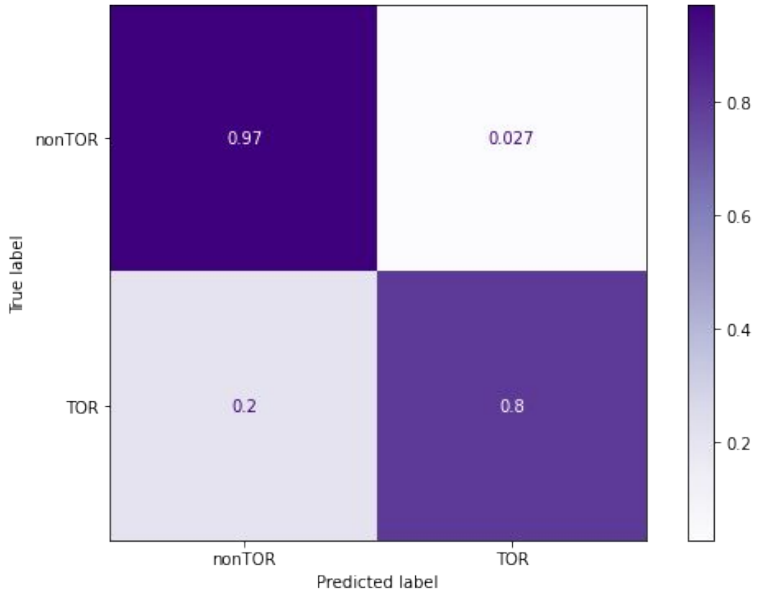

True Positive Rate
False Positive Rate
(AUC = 0.88)
0.0
0.2
0.4
0.6
0.8
1.0


nonTOR
TOR
True label
Predicted label
0.97
0.03
0.2
0.8

➔ KNN

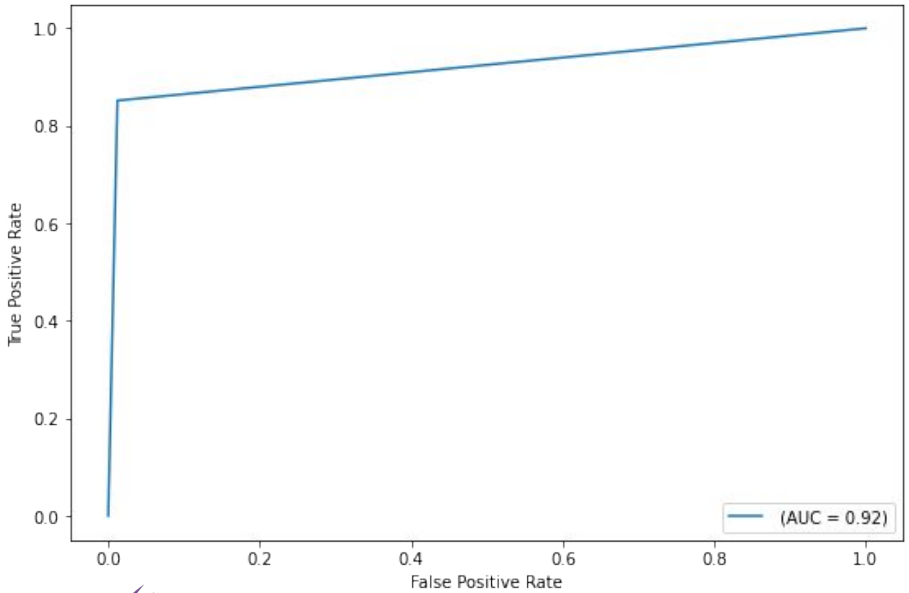
KFold 5

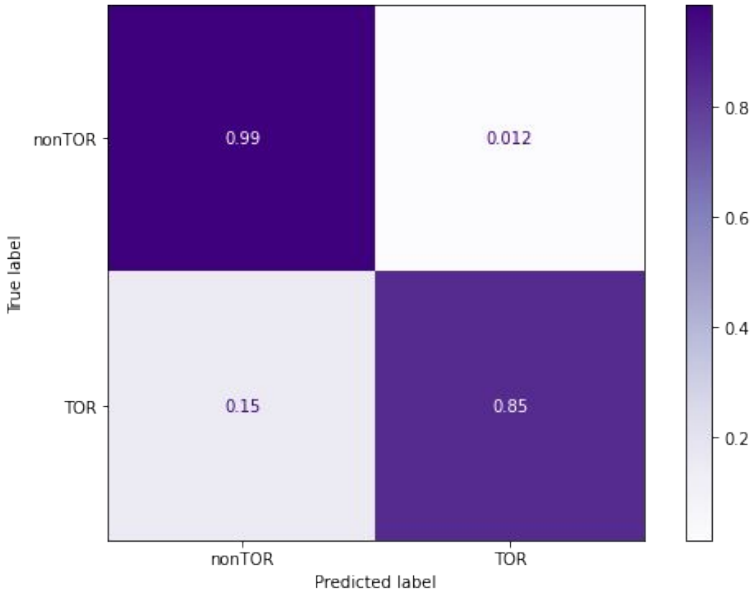

True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Kfold 1 (AUC = 0.88)
Kfold 2 (AUC = 0.88)
Kfold 3 (AUC = 0.89)
Kfold 4 (AUC = 0.88)
Kfold 5 (AUC = 0.87)


nonTOR
TOR
True label
Predicted label
0.97
0.027
0.2
0.8
0.8
0.6
0.4
0.2

➔ Random Forest

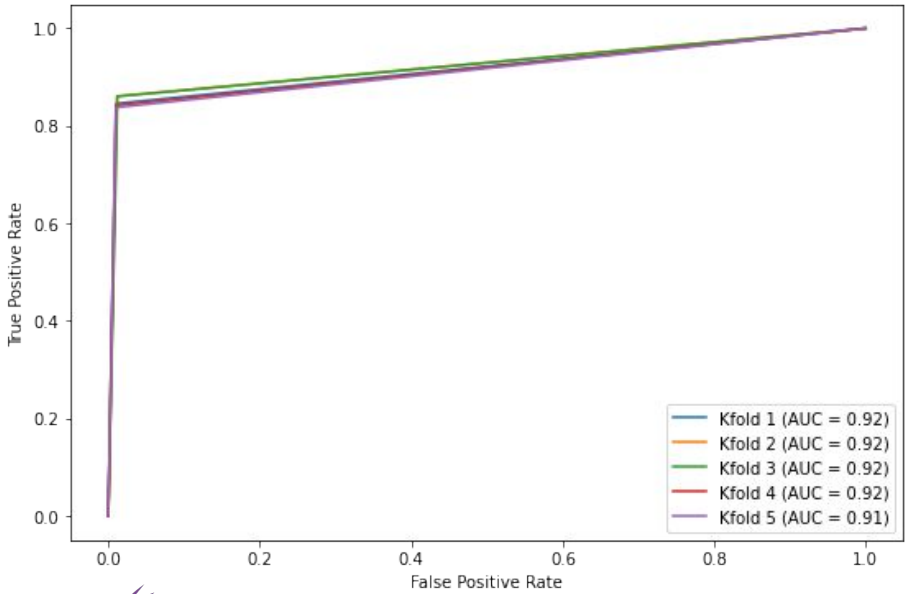
80/20

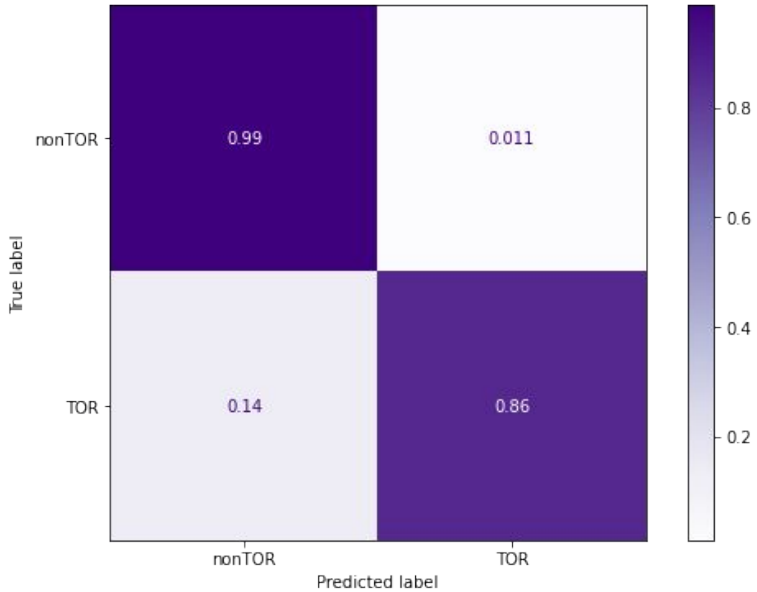




➔ Random Forest

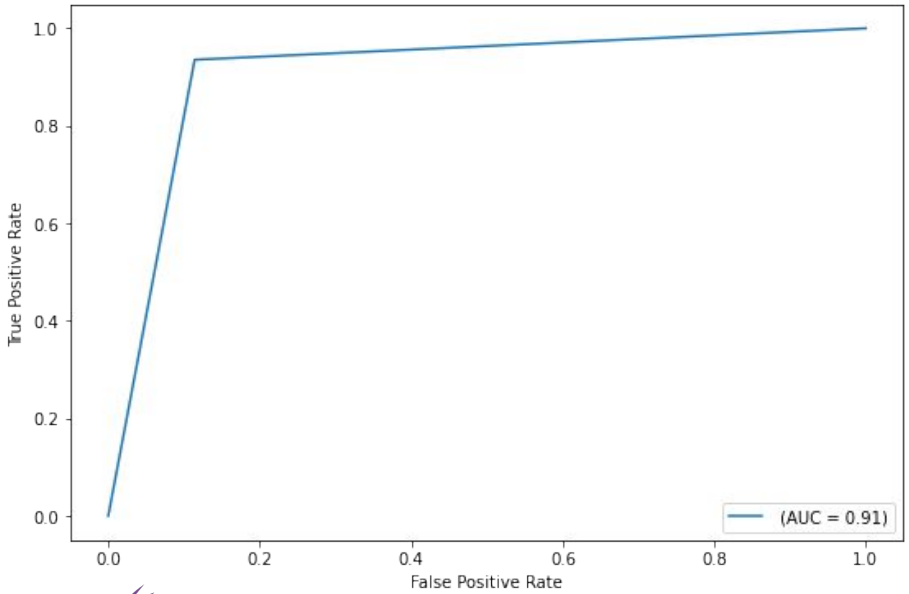
KFold 5

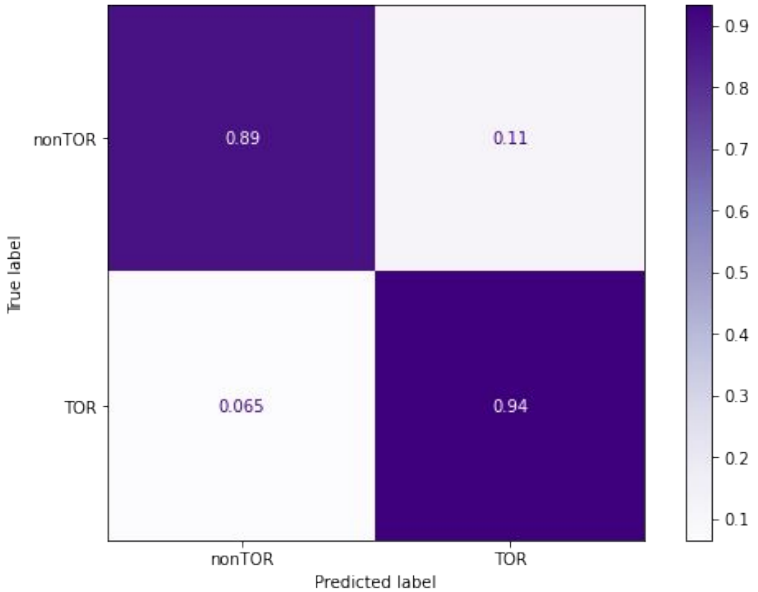




➔ SVM

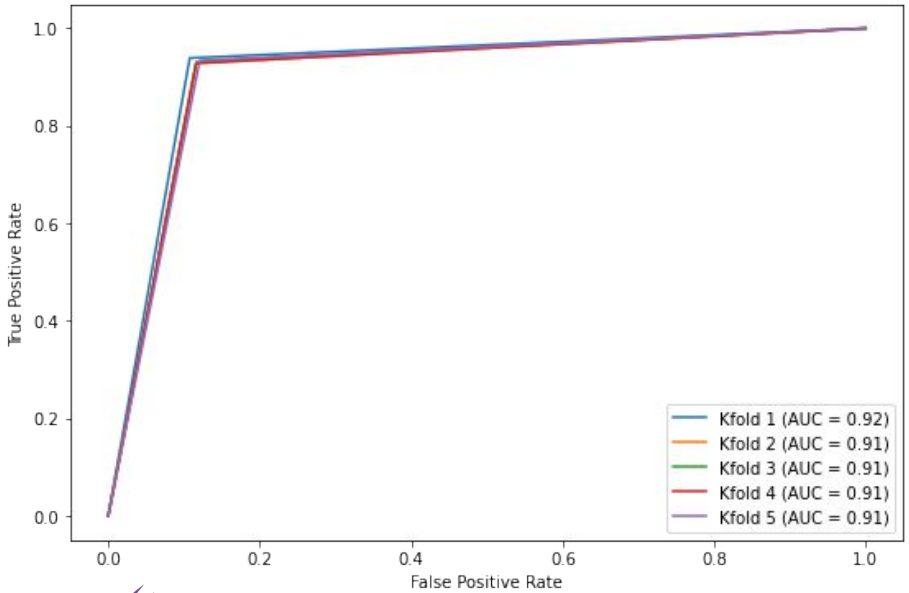
80/20

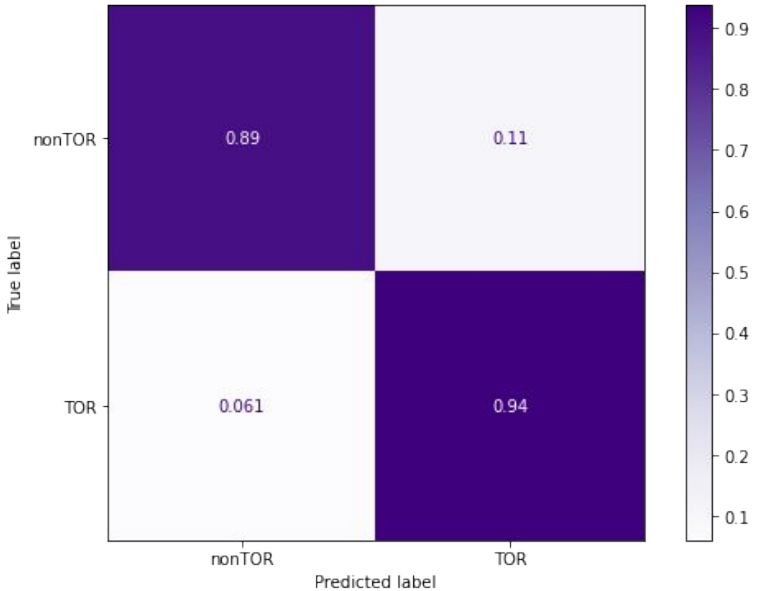

True Positive Rate
False Positive Rate
(AUC = 0.91)
0.0
0.2
0.4
0.6
0.8
1.0


True label
Predicted label
nonTOR
TOR
0.89
0.11
0.065
0.94
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

➔ SVM

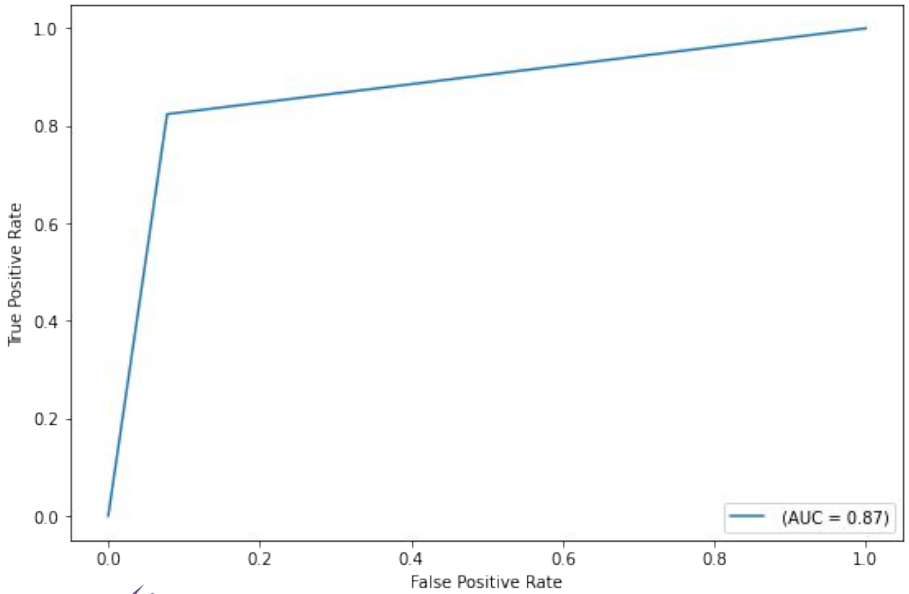
KFold 5

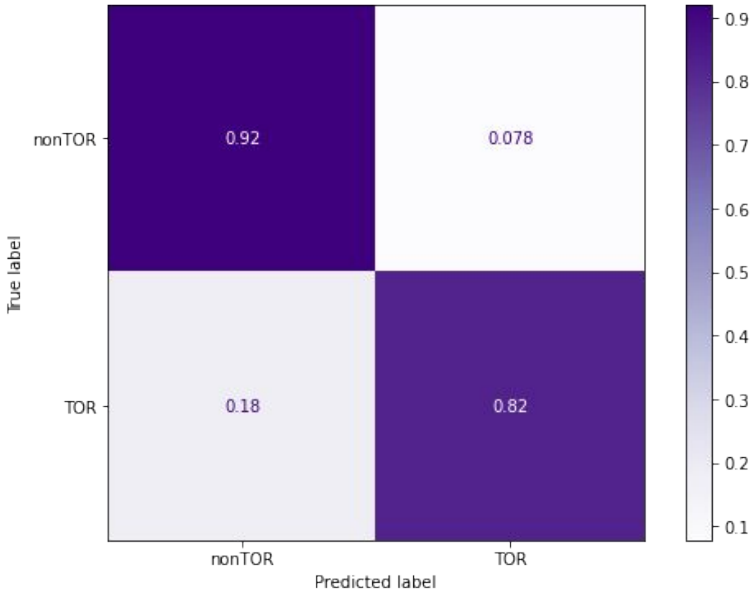

True Positive Rate
False Positive Rate
Kfold 1 (AUC = 0.92)
Kfold 2 (AUC = 0.91)
Kfold 3 (AUC = 0.91)
Kfold 4 (AUC = 0.91)
Kfold 5 (AUC = 0.91)


nonTOR
TOR
0.89
0.11
0.061
0.94
True label
Predicted label

➔ Perceptron

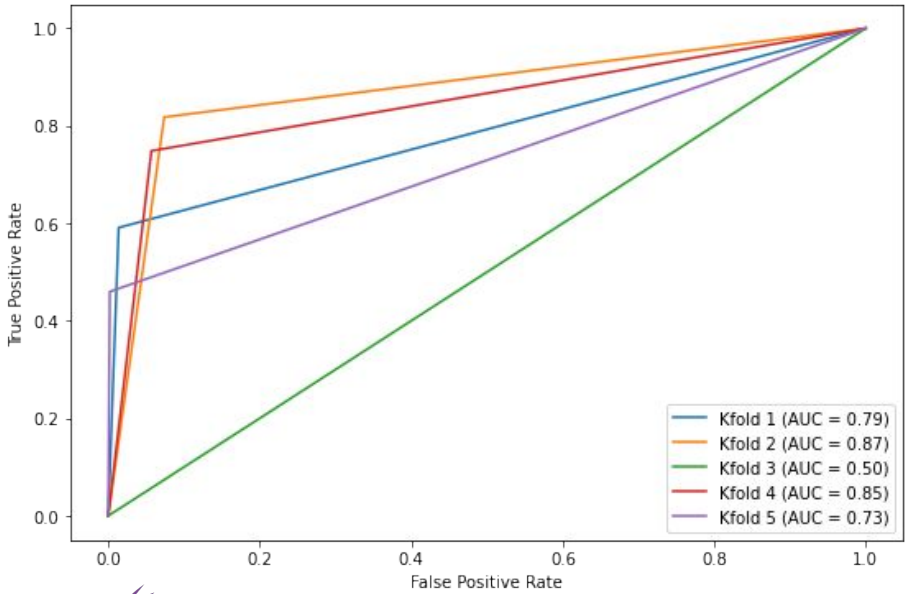
80/20

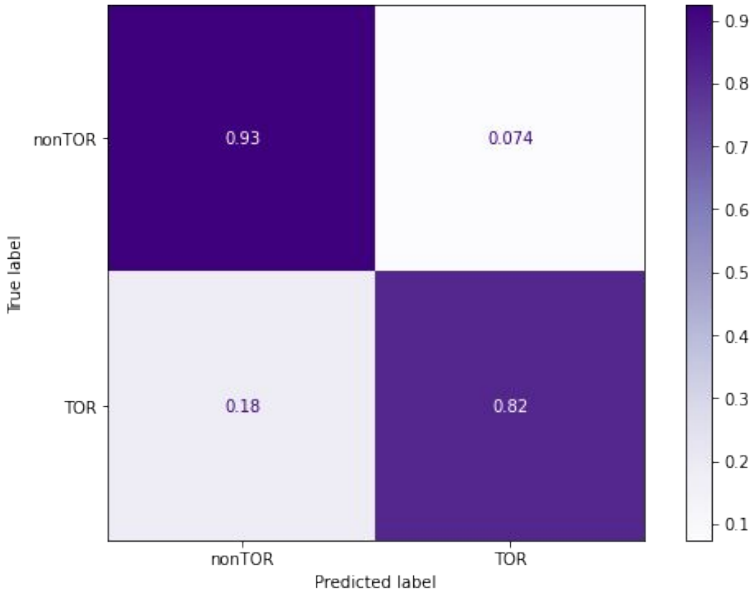




➔ Perceptron

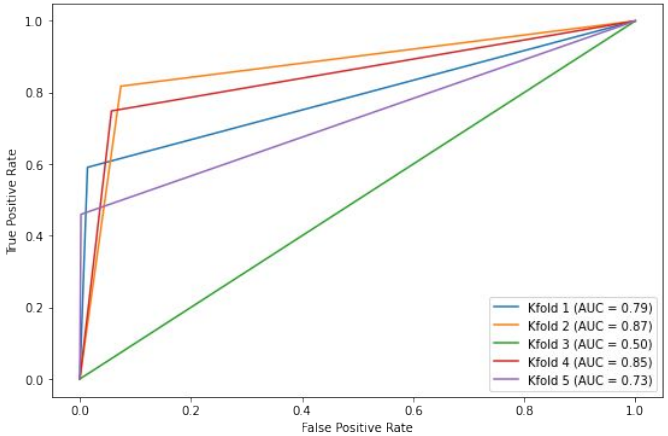
KFold 5

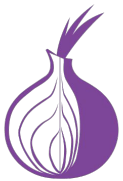




Observações do Treinamento

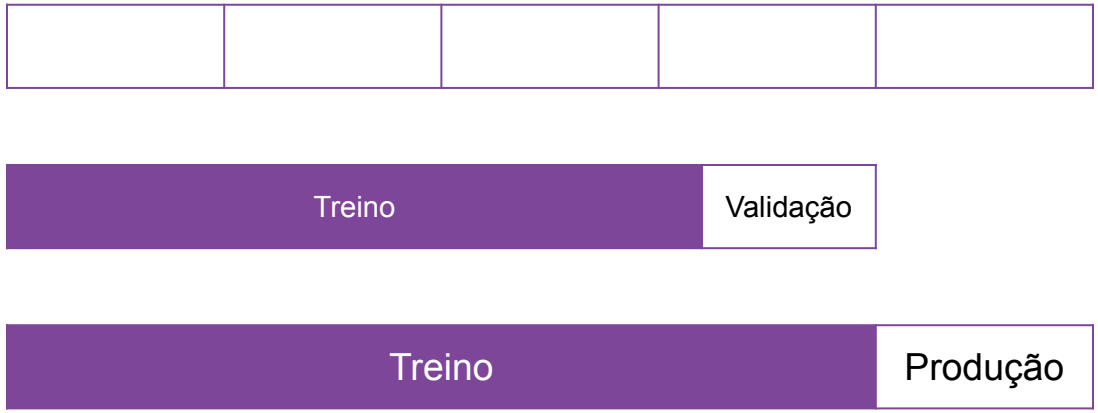
- AUC dos modelos ficaram acima de 0.8
- TP e TN também ficaram acima de 0.8
- Divergência nos treinos do perceptron usando KFold
- O tempo de treino foi relativamente rápido
  - exceto no SVM (30s)



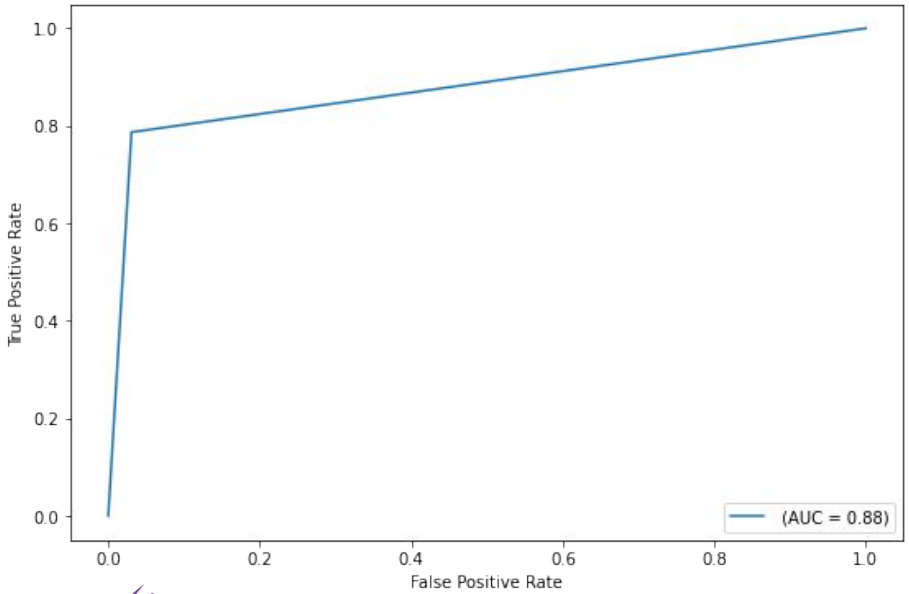


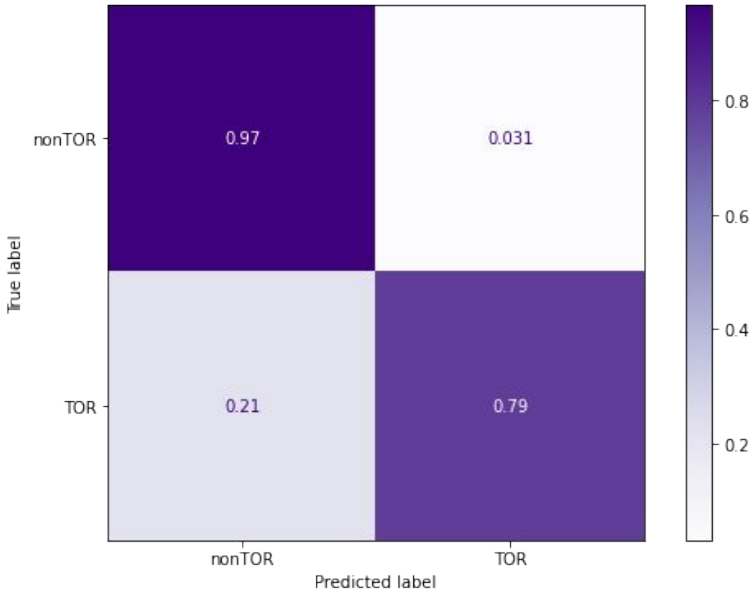
3. Teste

- Usando a parte de 20% como dados de produção
- Aplicado aos modelos treinados com a porção inteira de 80%

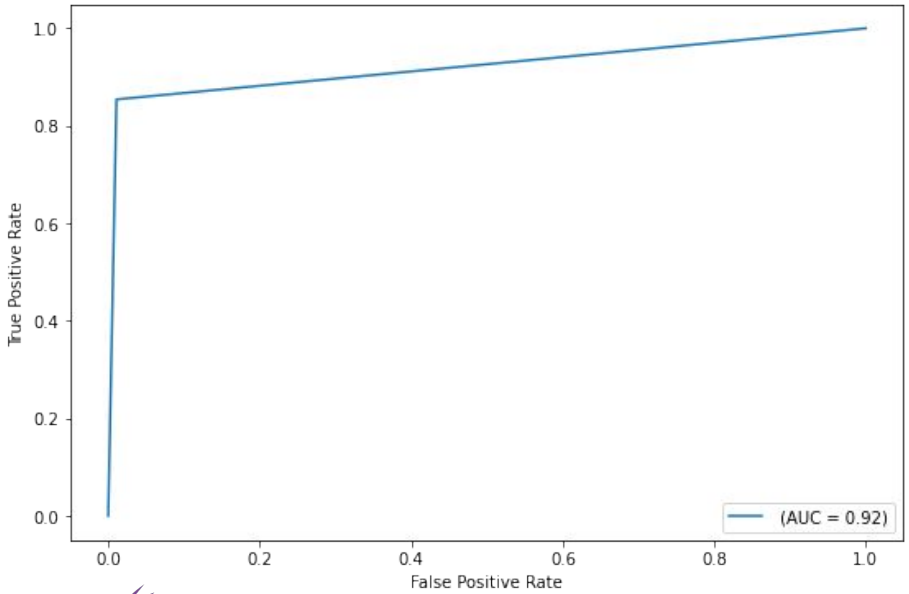


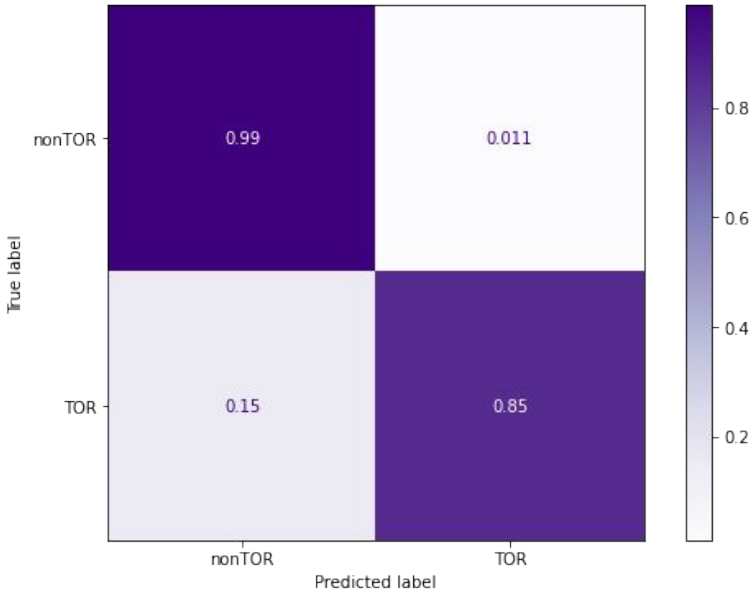
➔ KNN


True Positive Rate
False Positive Rate
(AUC = 0.88)
0.0
0.2
0.4
0.6
0.8
1.0

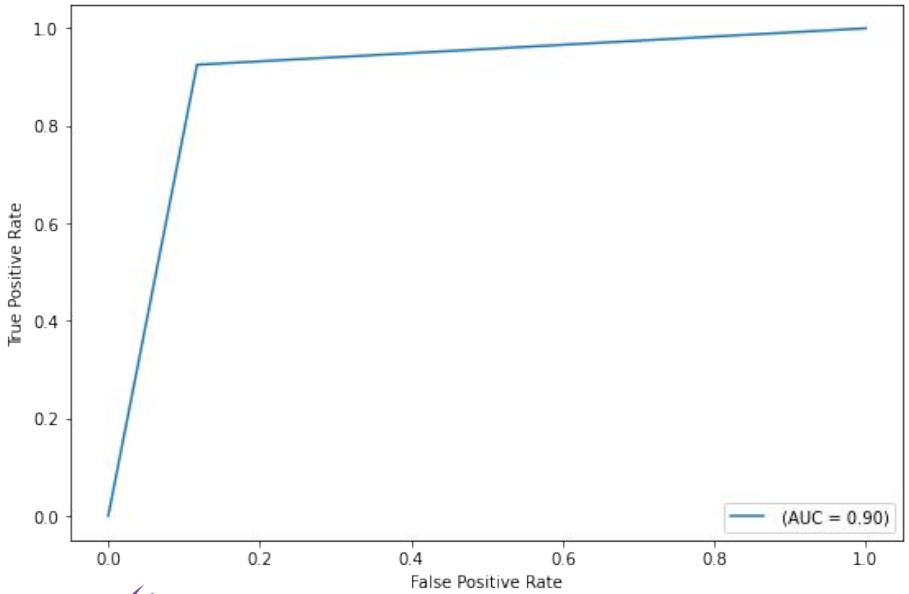

True label
Predicted label
nonTOR
TOR
0.97
0.031
0.21
0.79
0.8
0.6
0.4
0.2

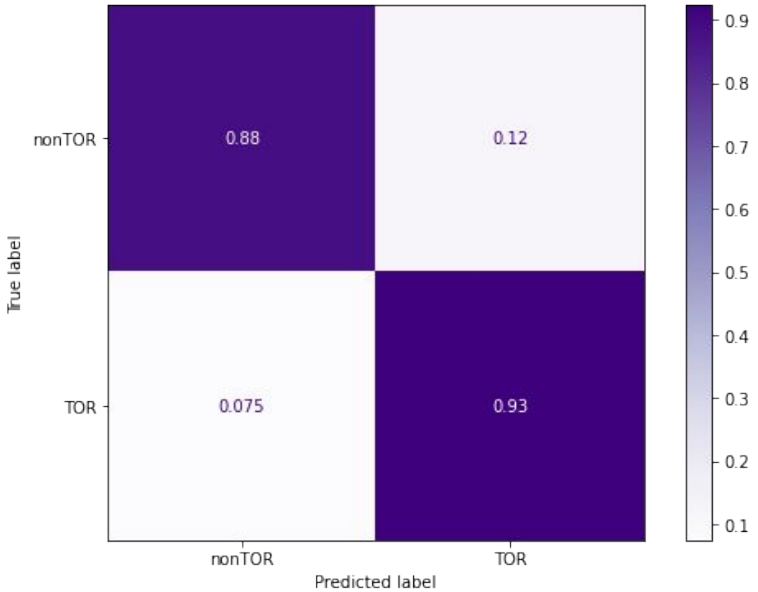
➔ Random Forest



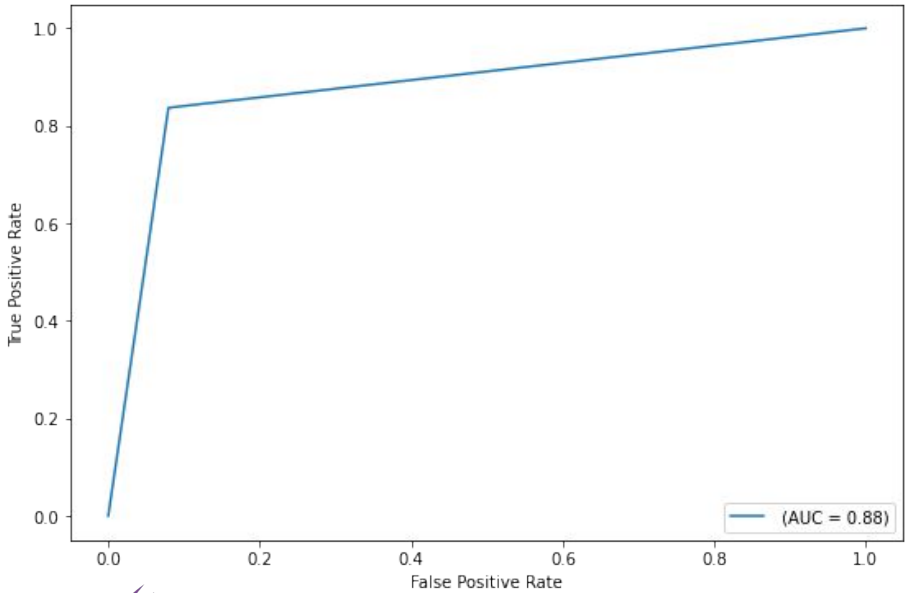


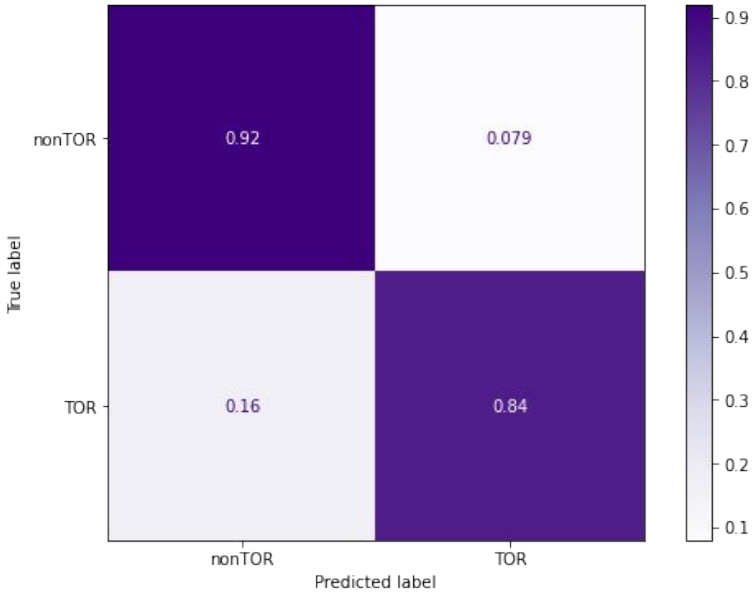
➔ SVM


True Positive Rate
False Positive Rate
(AUC = 0.90)
0.0
0.2
0.4
0.6
0.8
1.0

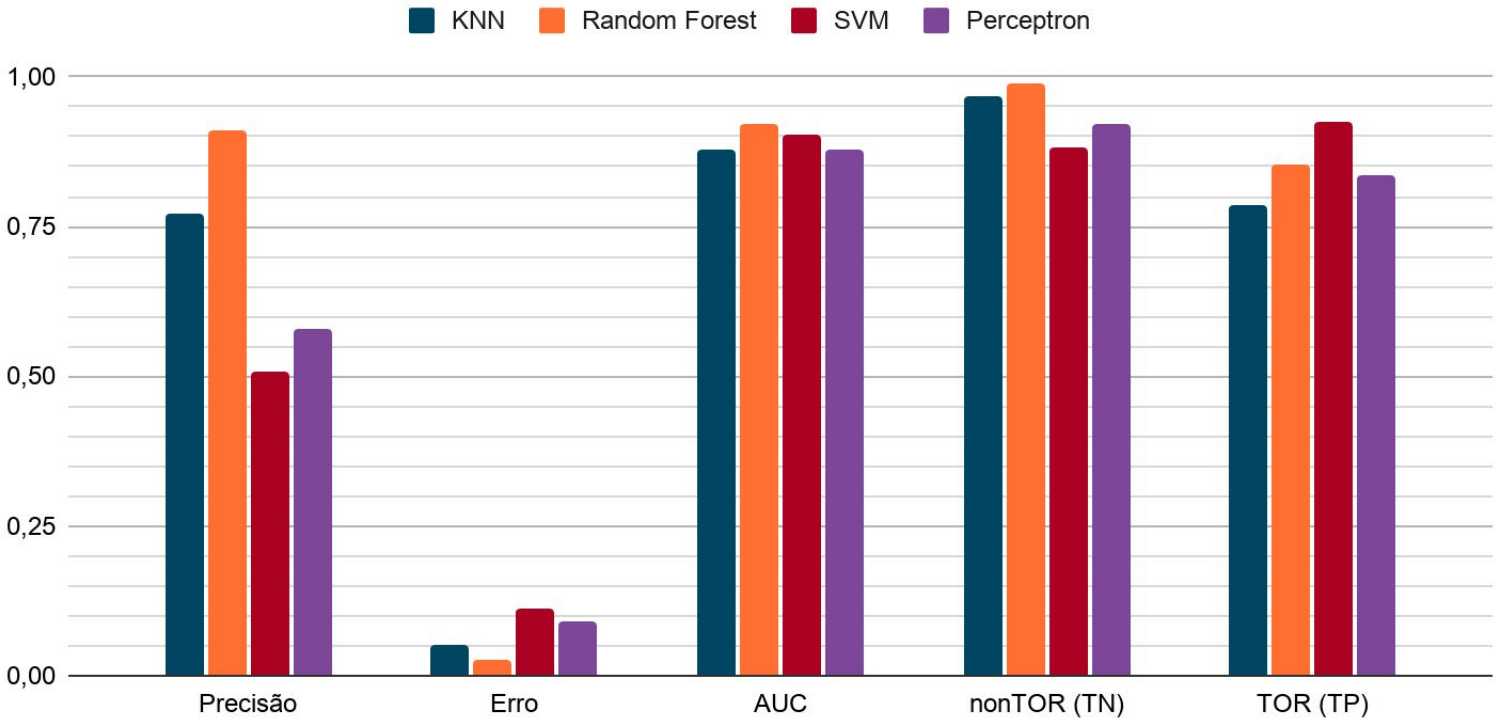

True label
Predicted label
nonTOR
TOR
0.88
0.12
0.075
0.93
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

➔ Perceptron


True Positive Rate
False Positive Rate
(AUC = 0.88)

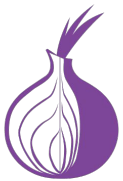

nonTOR
TOR
True label
0.92
0.079
0.16
0.84
nonTOR
TOR
Predicted label

Resultado dos testes



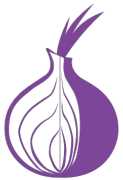
Conclusão

- Random Forest parece a melhor solução para nosso problema, no geral
  - Tem a melhor precisão
  - Menor erro
  - Maior área sob a curva ROC.
- Observação
  - O dataset é desbalanceado
  - Random Forest tem maior taxa de acerto em nonTOR
  - SVM tem maior taxa de acerto em TOR



Em resumo

O que usar?



- Random Forest
  - Acertar mais tráfego nonTOR
  - Sem tempo para treino e/ou retreino
  - Detecção na casa de milisegundos
- SVM
  - Acertar mais tráfego TOR

Links e Referências

- Github: https://github.com/djeni98/ciencia-dados-tor
- [1]
  - https://www.unb.ca/cic/datasets/tor.html
  - Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun and Ali A. Ghorbani, "Characterization of Tor Traffic Using Time Based Features", In the proceeding of the 3rd International Conference on Information System Security and Privacy, SCITEPRESS, Porto, Portugal, 2017