Detecção de Domínios Maliciosos

Universidade Federal do Paraná - UFPR
Abner Fontebom Bissolli Costa
Michele Venturin

Dataset Escolhido e Distribuição dos Dados

Detaset e Motivação

- **Dataset:**

  Conjunto 1M de domínios, previamente classificados como benignos, *malware*, *phishing* e *spam*. (CIC-Bell-DNS2021)

- **Motivação:**

"A cada ano, muitas grandes corporações são afetadas por essas ameaças, resultando em enormes perdas financeiras em um único ataque. Portanto, detectar e classificar um domínio malicioso em tempo hábil é essencial."

Exemplos

Domínio	Class. Multiclasse	Class. Binária
stockholm.se	Benigno	Benigno
ahackaday.io	Benigno	Benigno
arrow.com	Benigno	Benigno
paypal.de.daten.sicherheit-benutzer.top	Malware	Malicioso
businessbattle.tk	Phishing	Malicioso
k-slee.com	Spam	Malicioso

Problema

- Problema:

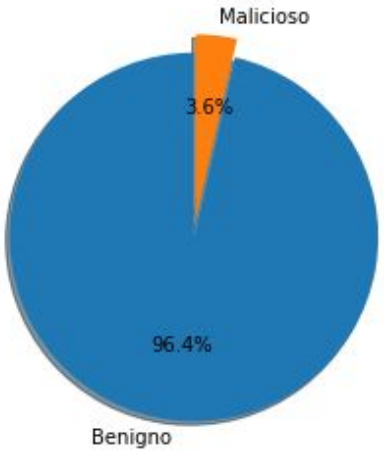
  Classificar corretamente domínios entre benignos e maliciosos, sem gerar muitos falsos alertas e ao mesmo tempo bloquear uma quantidade significante de domínios maliciosos

- Proposta:

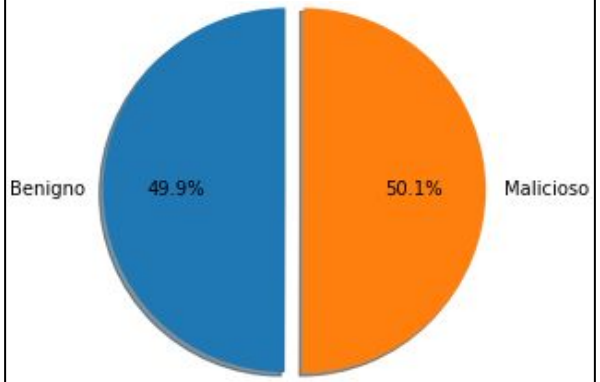
  Gerar um algoritmo classificar que atuará em juntamente com uma lista branca.

Distribuição dos Dados

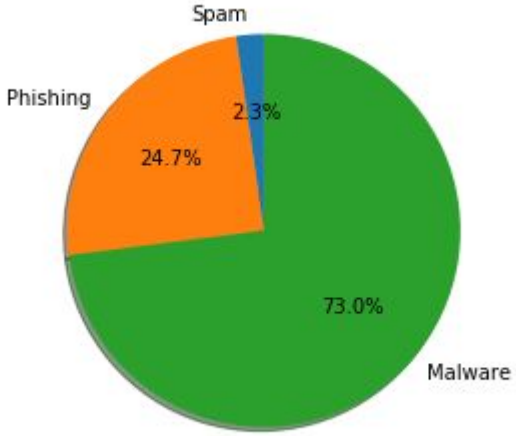
Distribuição Original (Be/Mw)



Distribuição Balanceada (Be/Mw) (Treino/Teste)



Distribuição Classes Maliciosos (Mw/Ph/Sp)



Atributos e Características

Atributos

Os atributos foram gerados utilizando a biblioteca *tld* para Python3, a qual realiza o parsing do domínio.

Características

Características Léxicas (45)

Tamanho Subdomínio+SLD.
Entropia da string.
Número de caracteres não alfanuméricos (‘.’ e ‘-’)
Valor TF-IDF do TLD.
Tamanho da maior sequência de mesmos caracteres.

Características de Terceiros (13)

Nº de nomes registrados
Informações Privadas
Valor TF-IDF do TLD.
Nº de servidores registrado.
Menor distância do SLD para SLD famosos (Alexa Rank).

Codificação e Normalização

- Para a codificação de características textuais, foram utilizados os algoritmos *TF-IDF* e *LabelEncoder* (*sklearn*), com pequenas modificações.

- Após a codificação das variáveis textuais, foi realizada uma normalização utilizando utilizando a função *Normalizer* (*sklearn*).

Algoritmos Testados

Algoritmos Testados

Florestas Randômicas

Parâmetro	Definição
n_estimators	1000
max_depth	30
class_weight	“balanced”
random_state	42

Perceptron Multicamadas (MLP)

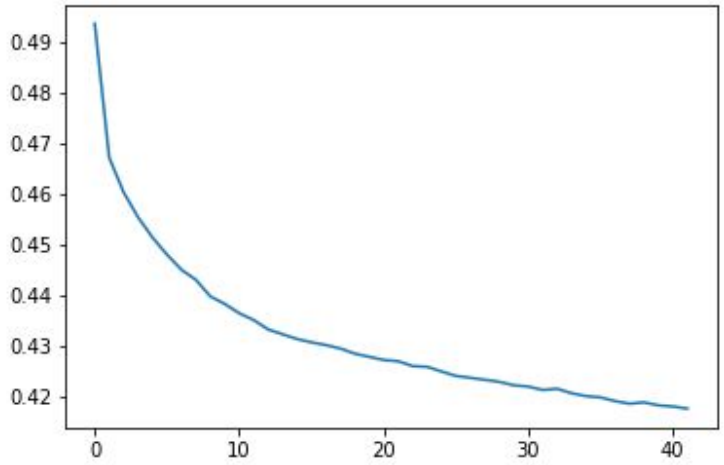
Parâmetro	Definição
n_neighbors	8
metric	“minkowski”
weights	“uniform”

***K-Nearest Neighbors* (KNN)**

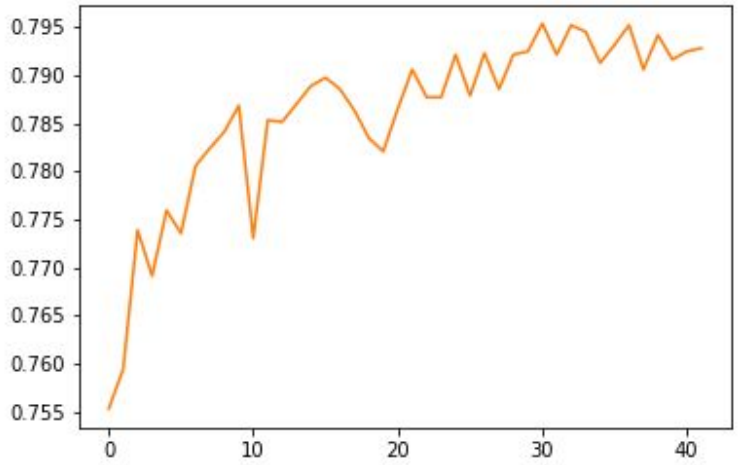
Parâmetro	Definição
hidden_layer_sizes	$(50, 100, 50)$
random_state	42
max_iter	1000
learning_rate_init	0.001
learning_rate	“adaptive”
validation_fraction	0.1
batch_size	32
early_stopping	True
verbose	True

Algoritmos Testados - *Tuning*

MLP – *Loss Value*

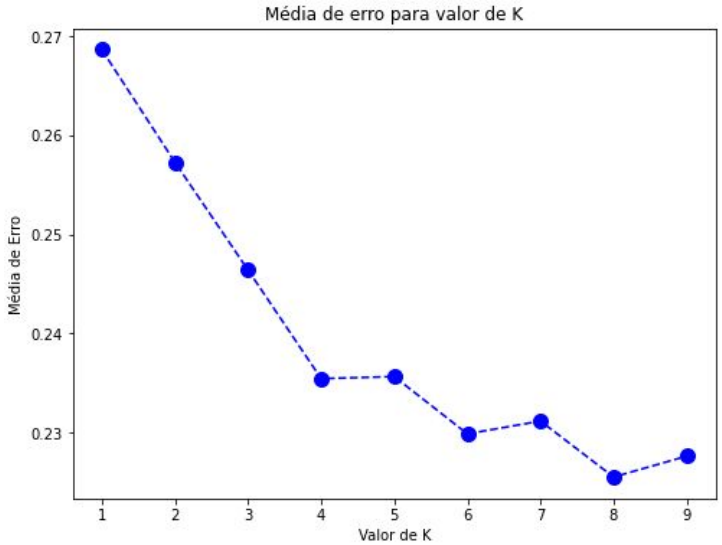


MLP – *Validation Score*



Algoritmos Testados - *Tuning*

KNN - Número de Vizinhos



Limiares definidos

Algoritmo	Limiar
Florestas Randômicas	0.435
MLP	0.466
KNN	0.4

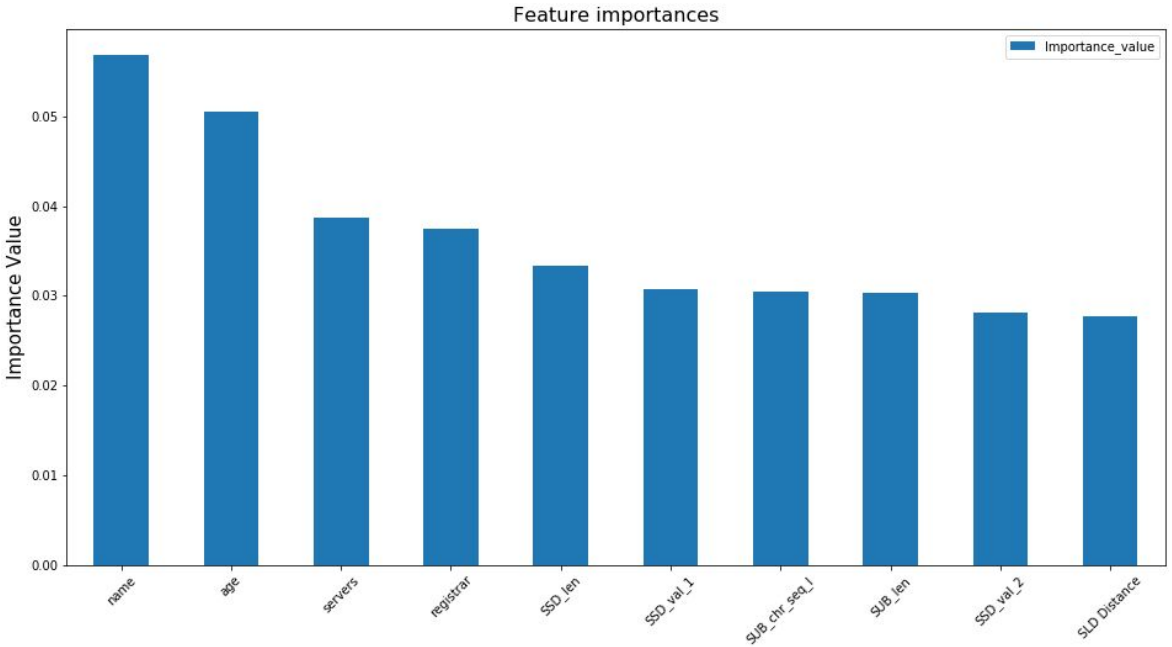
Algoritmos Testados - Importância das Características

- **Características Léxicas:**

  70%

- **Características de Terceiros:**

  30%



Algoritmos Testados - Importância das Características

Número de Nomes Registrados.
Idade do Domínio.
Número de Servidores Registrados.
Número de Registradores Registrados.
Tamanho do SSD.
Menor número de caracteres distintos que compõe a maioria do SSD.
Tamanho da maior sequência de mesmos caracteres no SUB.
Tamanho do SUB.
Porcentagem do SSD que os 5 caracteres mais utilizados compreendem.
Menor Distância do SLD para SLD Famosos.

Resultados e Análise

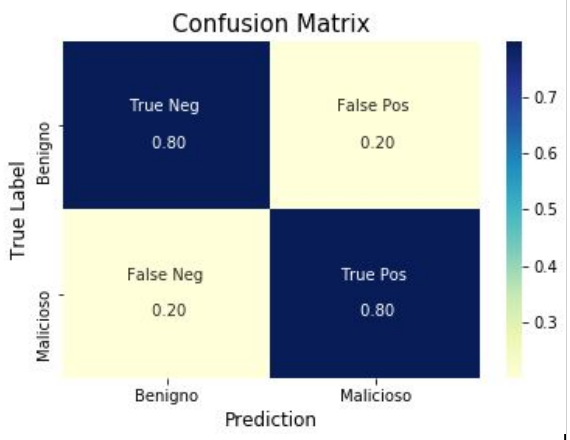
Resultados - Métricas

Algoritmo	Acurácia	*Precision*	*Recall*	*F1-Score*
Florestas Randômicas	0.799	0.799	0.798	0.799
Neurônio Multicamadas	0.797	0.798	0.795	0.796
K-Nearest Neighbors	0.763	0.748	0.792	0.769

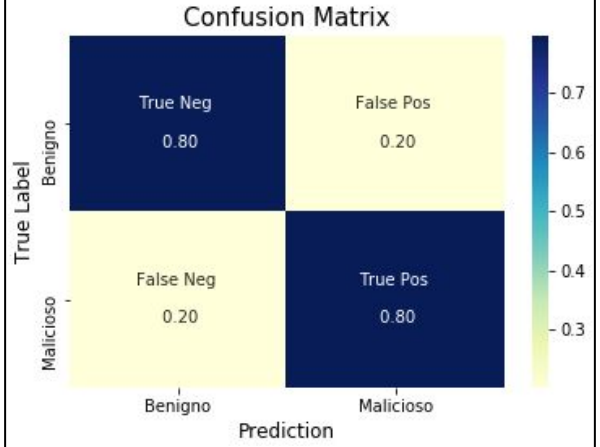
Algoritmo	Pasta 1	Pasta 2	Pasta 3	Pasta 4	Pasta 5	Média
Florestas Randômicas	0.799	0.796	0.797	0.793	0.797	0.796
Neurônio Multicamadas	0.792	0.788	0.793	0.786	0.794	0.791
K-Nearest Neighbors	0.772	0.769	0.771	0.764	0.765	0.768

Resultados - Matrizes de Confusão

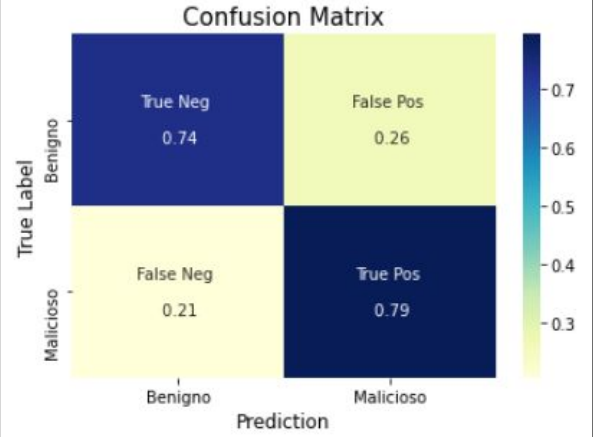
Florestas Randômicas



Perceptron Multicamadas (MLP)

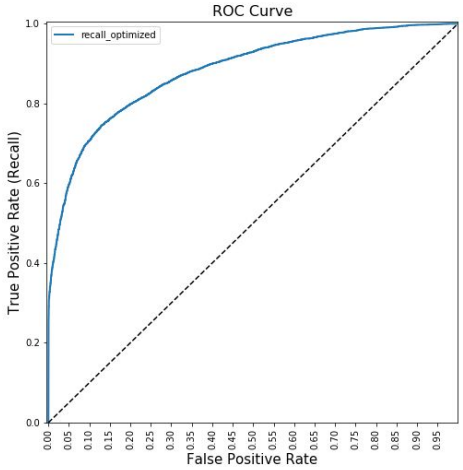


***K-Nearest Neighbors* (KNN)**

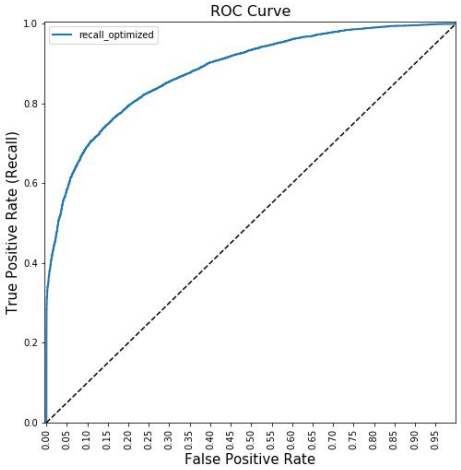


Resultados - Curvas ROC

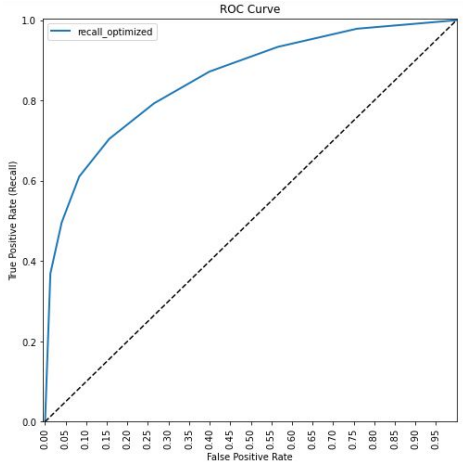
Florestas Randômicas



Perceptron Multicamadas (MLP)



***K-Nearest Neighbors* (KNN)**



Análise

- Através dos algoritmos testados, foi possível observar que todos os resultados ficam próximos a 80%.

- Este fato nos levou a acreditar que as características geradas não são capazes de solucionar o problema.

Análise

- Apenas dois tipos de características foram utilizadas **léxicas** e de **terceiro**. No entanto outras características podem ser exploradas, como dados da seção de resposta da resposta do DNS.

- Também é possível utilizar redes neurais recorrentes com palavras ou subdivisões do domínio, ou então utilizando técnicas como *n-grams*.

Demonstração

Referências

- Amazon (2021). Top 1M domains. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. Acessado em 07/12/2021.

- Bezerra, M. A. et al. (2015). Uma investigação do uso de características na detecção de urls.

Referências

- Davison, J., Moffitt, T., and and, H. L. (2019). Webroot cybersecurity. threat report - mid-year update.

- Mahdavifar, S., Maleki, N., Lashkari, A. H., Broda, M., and Razavi, A. H. (2021). Classifying malicious domains using dns traffic analysis.

Referências

- Olivo, C. K., Santin, A., and Oliveira, L. (2010). Avaliação de características para detecção de phishing de e-mail. Pontifícia Universidade Católica do Paraná, Curitiba-PR, Brasil.

Referências

- Ceschin, F., Oliveira, L. S., and Grégio, A. (2019). Aprendizado de máquina para segurança: Algoritmos e aplicações. Em Minicursos do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Capítulo 2, páginas 41-90. SBSeg.