Centro Universitário de Brasília – UniCEUB
Faculdade de Tecnologia e Ciências Sociais Aplicadas (FATECS)

AUGUSTO OLIVEIRA SILVA

MANUAL DO USUÁRIO

ENGENHARIA REVERSA DE MALWARE USANDO MACHINE LEARNING: IMPACTOS E APLICABILIDADE NA ÁREA DA CIBERSEGURANÇA

Brasília
2025

1. INTRODUÇÃO

Este manual descreve como utilizar os recursos do projeto de pesquisa “Engenharia Reversa de Malware Usando Machine Learning”. O projeto visa identificar tráfego malicioso em redes por meio de modelos de aprendizado de máquina que foram desenvolvidos e treinados com base nos conjuntos de dados CIC-IDS2017 e CSE-CIC-IDS2018.

A aplicação desenvolvida em Streamlit funciona como uma interface interativa que permite ao usuário aplicar, de forma prática, os modelos treinados durante a pesquisa. Através do envio de um arquivo .csv contendo dados de tráfego de rede, o sistema identifica automaticamente se o problema é de classificação binária (benigno vs. malicioso) ou multiclasse (tipos específicos de ataques) e disponibiliza os modelos correspondentes para análise. Além de realizar a previsão, a aplicação apresenta visualizações gráficas — como gráficos de barras, pizza e mapas de calor — que facilitam a interpretação dos resultados. Esta interface permite simular, de maneira bem simples e acessível, como os modelos poderiam ser integrados em um sistema de monitoramento de segurança real.

2. FUNCIONALIDADES

Algumas funcionalidades da aplicação desenvolvida, são:

- Upload de arquivos no formato .CSV que contém dados de tráfego de rede;
- Detecção automática do tipo de classificação possível;
- Escolha de modelos de Machine Learning previamente treinados;
- Visualização dos resultados da previsão através de gráficos;
- Interface interativa.

3. REQUISITOS:

Para conseguir rodar e aproveitar ao máximo a aplicação desenvolvida, é necessário ter cumprir alguns requisitos técnicos, uma vez que a aplicação foi desenvolvida com o intuito de testar os modelos e realizar uma simples interação com o usuário. São requisitos necessários:

- Ter instalado na máquina o Python 3.8+;
- Ter instalado na máquina o Pip (instalação normalmente ocorre junto com o python);
- Ter o Git/Github instalado na máquina;
- Ter feito o download do arquivo **modelos.pkl** e colocado o mesmo na pasta **/notebooks**;
- Os dados de teste devem estar no formato **.csv**.

4. ESTRUTURA DO PROJETO

Para seguir boas práticas de desenvolvimento o projeto foi montado em uma estrutura relativamente simples e dividido em pastas de dados de teste (amostra), notebooks e scripts respectivamente. Além disso, contém também um arquivo README.md e outro requirements.txt. Segue abaixo estrutura do projeto:

```
Data-Science-Capstone/
├── amostras/
│   └── amostra_1.csv                          # Amostra de Teste 1
│   └── amostra_2.csv                          # Amostra de Teste 2
│   └── amostra_3.csv                          # Amostra de Teste 3
│   └── amostra_4.csv                          # Amostra de Teste 4
│   └── amostra_5.csv                          # Amostra de Teste 5
│   └── amostra_6.csv                          # Amostra de Teste 6
│
├── notebooks/
│   └── 1-data-preprocessing.ipynb             # Coleta e Preparação dos Dados
│   └── 2-exploratory-data-analysis.ipynb      # Análise Exploratória dos Dados
│   └── 3-feature-engineering                  # Engenharia de Features
│   └── 4-ml-models                            # Modelagem ML
│   └── modelos.pkl                            # Modelos treinados
│
├── scripts/
│   ├── app.py                                 # Aplicação Streamlit
│   └── visuals.py                             # Funções visuais do app
│
├── README.md                                  # Arquivo de apresentação
├── requirements.txt                           # Pacotes necessários
```

5. INSTALAÇÃO E EXECUÇÃO DO APLICATIVO

Todo o tutorial a seguir está descrito de forma mais clara e de melhor visualização no repositório do GitHub https://github.com/gut0oliveira/Data-Science-Capstone

5.1. Clone o repositório

Abra um novo terminal e coloque esse código:

```
git clone https://github.com/gut0oliveira/data-science-capstone.git
```

Em seguida, este:

```
cd data-science-capstone
```

5.2. Instale as dependências

Depois de executar os códigos acima, instale as dependências necessárias:

```
pip install -r requirements.txt
```

5.3. Baixe o arquivo de modelos (.pkl)

Devido ao tamanho do arquivo **`modelos.pkl`**, o mesmo não pode ser enviado ao GitHub por ser maior que 25MB.

ATENÇÃO!!
`O arquivo tem 111MB, então o Google pode mostrar um alerta.`
`Pode prosseguir com segurança clicando em 'Fazer o download mesmo assim'`

Clique aqui para baixar o arquivo: modelos.pkl
Depois de baixar, coloque o arquivo dentro da pasta **`/notebooks`** do projeto.

5.4. Navegue até a pasta /scripts do projeto

Clique com o botão direito na pasta **`/scripts`** e selecione **`Copy Path`**
Abra o terminal e cole o código, como no exemplo abaixo:

```
cd caminho/copiado/para/a/pasta/scripts
```

5.5. Rode o Streamlit

Após acessar a pasta **`/scripts`**, coloque esse código abaixo no terminal:

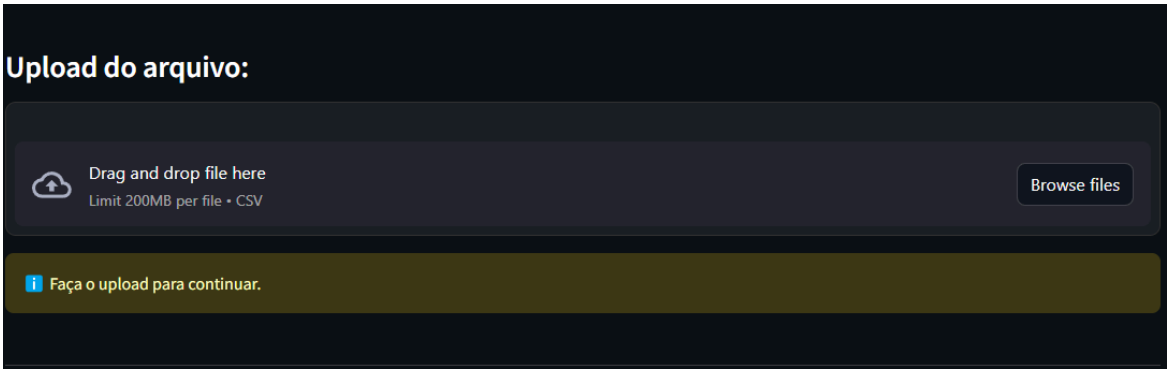
```
streamlit run app.py
```

Abra o link gerado no navegador (http://localhost:...) para interagir com a aplicação.

6. COMO UTILIZAR A INTERFACE

6.1. Primeira Etapa

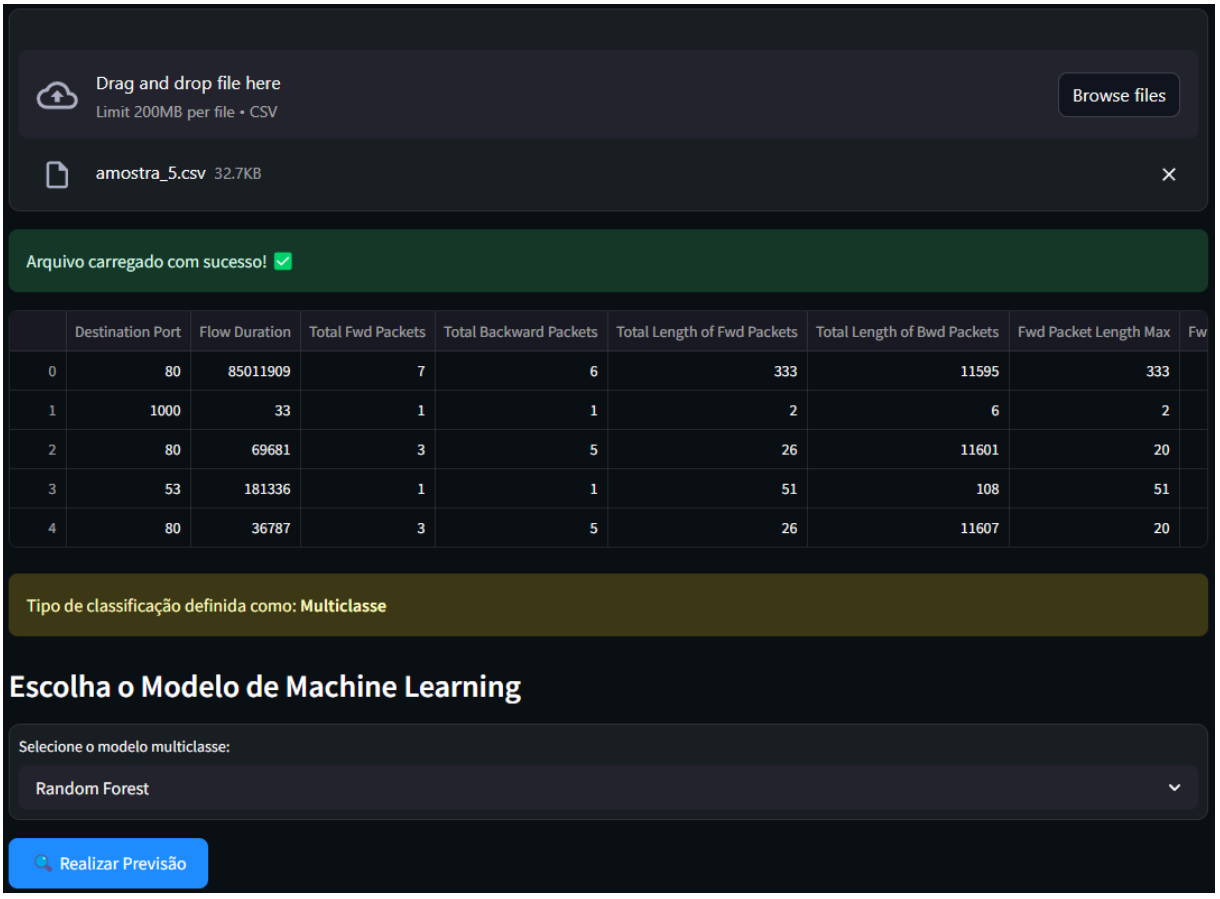
Upload do arquivo csv, conforme imagem abaixo:



Nessa etapa, para que a aplicação funcione corretamente, utilize um dos 6 arquivos disponíveis na pasta **`/amostras`**. Os arquivos dessa pasta são amostras aleatórias de dados.

6.2. Segunda Etapa

Após selecionar um arquivo, essa será a tela:



	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fw
0	80	85011909	7	6	333	11595	333	
1	1000	33	1	1	2	6	2	
2	80	69681	3	5	26	11601	20	
3	53	181336	1	1	51	108	51	
4	80	36787	3	5	26	11607	20	

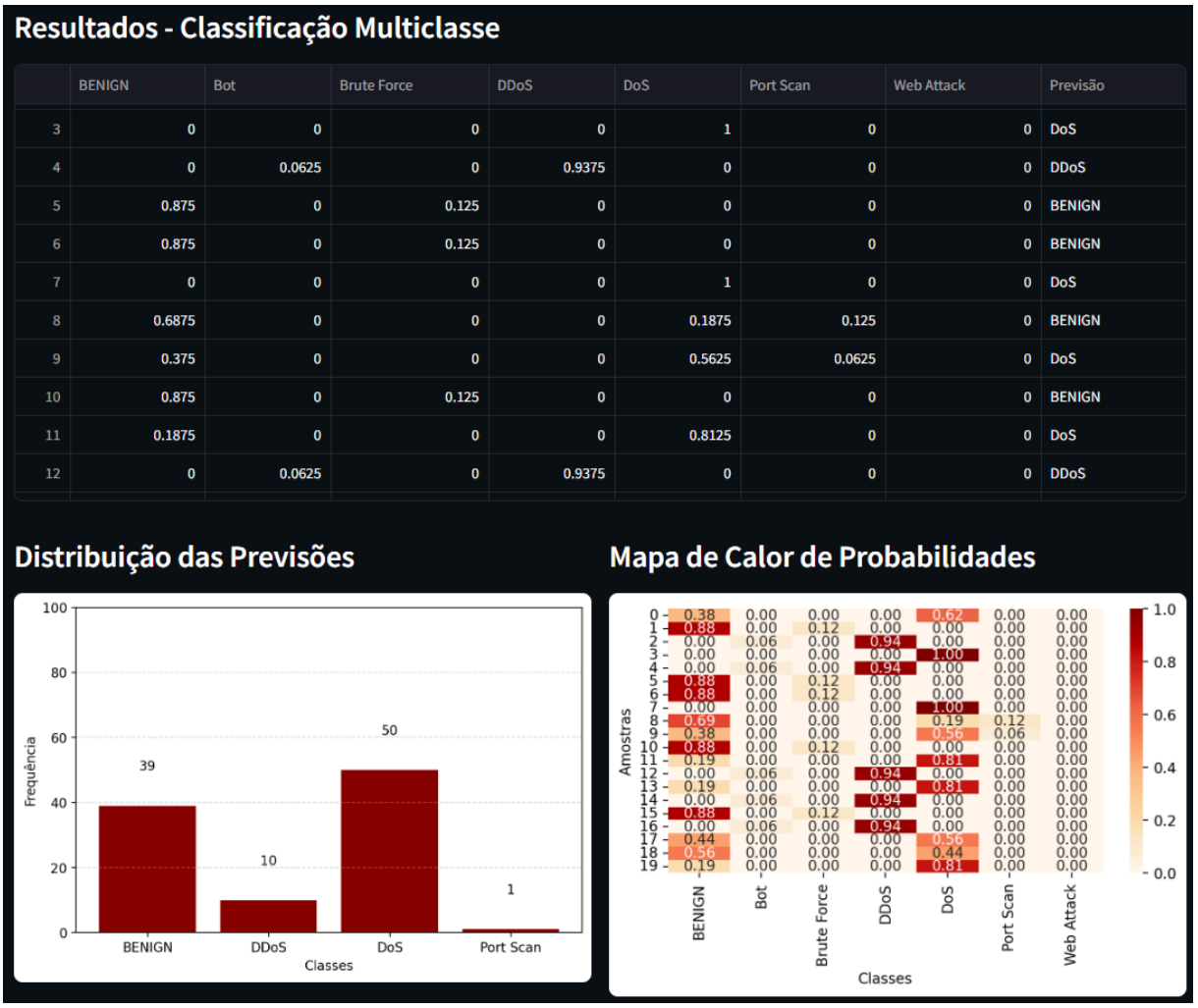
Conforme visto na imagem, a aplicação retorna um alerta de sucesso, quando o arquivo está válido, caso contrário, mostrará um alerta de erro. Além disso, é possível ver uma tabela com o arquivo .csv e o tipo de classificação que o modelo será capaz de fazer.

6.3. Terceira Etapa

Essa etapa consiste na escolha do modelo, para classificação binária, está disponível os modelos de **Regressão Logística** e **Support Vector Machine**, já para a classificação multi-classe, os modelos **Random Forest**, **K-Nearest Neighbours** e **XGBoost estão disponíveis**.

6.4. Quarta Etapa

Após escolher o modelo e selecionar o botão de "Realizar Previsão", a aplicação realizará a previsão e retornará para o usuário, uma tabela e gráficos mostrando os resultados obtidos, conforme visto na imagem.



	BENIGN	Bot	Brute Force	DDoS	DoS	Port Scan	Web Attack	Previsão
3	0	0	0	0	1	0	0	DoS
4	0	0.0625	0	0.9375	0	0	0	DDoS
5	0.875	0	0.125	0	0	0	0	BENIGN
6	0.875	0	0.125	0	0	0	0	BENIGN
7	0	0	0	0	1	0	0	DoS
8	0.6875	0	0	0	0.1875	0.125	0	BENIGN
9	0.375	0	0	0	0.5625	0.0625	0	DoS
10	0.875	0	0.125	0	0	0	0	BENIGN
11	0.1875	0	0	0	0.8125	0	0	DoS
12	0	0.0625	0	0.9375	0	0	0	DDoS

7. MENU LATERAL

A aplicação conta também com um simples menu lateral que contém uma breve descrição da aplicação e, também, as etapas de desenvolvimento, sendo elas:

1. Pré-processamento de Dados;
2. Análise Exploratória de Dados;
3. Engenharia de Atributos;
4. Treinamento e Avaliação.