



UNifeob
| ESCOLA DE NEGÓCIOS

2023

PROJETO DE CONSULTORIA EMPRESARIAL



UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADO

Spam Mail Prediction using Machine Learning

SÃO JOÃO DA BOA VISTA, SP

OUTUBRO 2023

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADO

Spam Mail Prediction using Machine Learning

MÓDULO - Inteligência Artificial

Inteligência Artificial – Prof. Rodrigo Marudi de Oliveira

Segurança em Sistema Computacionais - Prof. Nivaldo de Andrade

Estudantes:

Breno Dias Madrilles, RA 1012022100029

Renan Flora Dos Santos, RA 1012022100571

SÃO JOÃO DA BOA VISTA, SP
OUTUBRO, 2023

SUMÁRIO

1 INTRODUÇÃO	4
2 DESCRIÇÃO DO PROJETO	5
3 PROJETO DE INTELIGÊNCIA ARTIFICIAL E SEGURANÇA EM SISTEMAS COMPUTACIONAIS	6
3.1 INTELIGÊNCIA ARTIFICIAL	6
3.1.1 Aplicação Prática da Inteligência Artificial	6
3.1.2 Implementação e Técnicas Utilizadas	6
3.2 SEGURANÇA EM SISTEMAS COMPUTACIONAIS	7
3.2.1 Conceitos e Implementação de Segurança	7
3.2.2 Detecção e Prevenção de Ataques	7
4 CONCLUSÃO	9
REFERÊNCIAS	10
ANEXOS	11

1 INTRODUÇÃO

Com o aumento exponencial de e-mails não solicitados, a necessidade de distinguir entre conteúdo legítimo e spam se tornou crucial. Os métodos convencionais muitas vezes falham devido a melhora dos métodos utilizados pelos spammers .

Nesse cenário, o uso de um identificador de spams baseado em machine learning oferece uma solução dinâmica. Esse método analisa automaticamente padrões e características dos e-mails, adaptando-se continuamente para identificar com precisão entre mensagens maliciosas e legítimas.

Ao automatizar essa filtragem, melhora a experiência do usuário ao assegurar que apenas e-mails relevantes e autênticos cheguem à caixa de entrada.

Assim, o uso dessa abordagem não apenas reduz o volume de spam, mas também aprimora a eficiência e a segurança no gerenciamento de e-mails.

2 DESCRIÇÃO DO PROJETO

O desenvolvimento do sistema de machine learning para identificação de spam inicia-se com a coleta de dados, onde se reúne um conjunto de e-mails previamente rotulados como spam e não-spam. Em seguida, realiza-se o pré-processamento dos dados, efetuando a limpeza e normalização para garantir a consistência.

A próxima etapa é a extração de características, onde são identificados atributos relevantes nos e-mails, como palavras-chave, frequência de termos e padrões de formatação.

Após a divisão dos dados em conjuntos de treinamento e teste, inicia-se o treinamento do modelo, permitindo que o modelo aprenda padrões associados a e-mails spam e não-spam. Em seguida, ajustes de parâmetros são realizados, e a validação cruzada é aplicada para garantir a generalização do modelo e evitar sobreajuste.

Na avaliação do desempenho é calculada a acurácia para verificar a eficácia do modelo.

3 PROJETO DE INTELIGÊNCIA ARTIFICIAL E SEGURANÇA EM SISTEMAS COMPUTACIONAIS

3.1 INTELIGÊNCIA ARTIFICIAL

Inteligência artificial é uma área voltada para moldarmos máquinas e fazê-las inteligentes e com isso podemos classificá-la como uma área que permite que sistemas simulem uma inteligência similar à humana, indo além da programação de ordens específicas para tomar decisões de forma autônoma, baseadas em padrões de enormes bancos de dados.

3.1.1 Introdução à Aplicação da IA

Tendo em vista o mundo em que estamos inseridos e como as coisas relacionadas a segurança da informação e dos nossos dados estão submissas a ataques diariamente, pensamos em ligar as duas pontas, tanto a de inteligência artificial quanto a de segurança da informação, para podermos analisar todos os e-mails que recebemos diariamente e poder filtrar os que são seguros e os que não são (spam e afins).

3.1.2 Implementação e Técnicas Utilizadas

Optamos por usar Machine Learning juntamente com Python, para poder realizar a predição e filtragem dos dados recebidos via e-mail, seja eles mensagens ou arquivos. Sendo assim, optamos por conciliar as duas fontes, pois juntas trabalham de uma forma muito eficaz e também devido a facilidade de criação de códigos ou afins para tal feito.

3.2 SEGURANÇA EM SISTEMAS COMPUTACIONAIS

A segurança em sistemas computacionais é uma pauta muito importante a ser discutida, pois gere todas as informações privadas da empresa, seja ela um simples e-mail recebido (que pode ser malicioso) ou até mesmo a senha do banco de dados onde toda informação sobre regra de negócios da empresa está inserida. Pensando nisso, manter um sistema de segurança computacional de boa qualidade e robusto é de suma importância, seja para pequenos ou grandes negócios.

3.2.1 Conceitos e Implementação de Segurança

- **Conceito:** A segurança da informação na empresa sempre foi subdividida para assim não centralizar todos os dados em um único lugar, dificultando a ação maliciosa de terceiros. Mas em exemplos práticos e simples, como uma mensagem nova na caixa de e-mail, notamos que não elas não tinham tanto cuidado assim e decidimos que deveríamos nos preocupar em monitorar mais esse canal, que é utilizado para novas demandas e comunicações externas.
- **Implementação da segurança:** A implementação da segurança foi discutida e muito bem pensada para isso e chegamos na conclusão de que se utilizássemos, Machine Learning resolveria o problema, sendo assim começamos a analisar o contexto para entender como proceder da melhor forma e levar satisfação até o cliente.

3.2.2 Detecção e Prevenção de Ataques

- **Estratégias Adotadas:** Utilizamos como estratégia o treinamento progressivo da IA para que ela pudesse conseguir discernir sobre e-mails que são spam e/ou os que são verdadeiros.
- **Método:** Adotamos um método que faz com que a IA passe por csv contendo uma lista de e-mails dos mais diversos tipos de assuntos possíveis e que ela consiga nos trazer o que é spam ou não.

- **Ferramentas Utilizadas:** Utilizamos Python como ferramenta e algumas de suas bibliotecas como dependências de projeto para realizar tal feito, dentre elas estão: numpy, pandas, train_test_split, Logistic Regression, accuracy_score. Assim também utilizamos o Google Colab para a renderização dos dados.

4 CONCLUSÃO

Neste projeto, exploramos a viabilidade de um protótipo baseado em Machine Learning para identificação eficiente de e-mails spam. A análise de dados rotulados, extração de características relevantes e treinamento do modelo revelaram ótimas capacidades de discernimento entre mensagens indesejadas e legítimas.

A avaliação do desempenho demonstrou que, mesmo em um estágio prototípico, o modelo exibe um bom potencial para melhorar a filtragem de e-mails.

REFERÊNCIAS

GUILLOU, P. Como treinar um modelo de Deep Learning no Colab? Disponível em: <https://medium.com/@pierre_guillou/como-treinar-um-modelo-de-deep-learning-no-colab-e-3cc4da64d9a>. Acesso em: 10 out. 2023.

Machine Learning: Tutorial prático usando apenas o navegador (é sensacional!!!). Disponível em: <<https://www.youtube.com/watch?v=JyGGMMyR3x5I>>. Acesso em: 10 out. 2023.

Como o Gmail Utiliza o Machine Learning para Combater o Spam 2023. Disponível em: <<https://agenciaeon.com.br/inteligencia-artificial/gmail-machine-learning-spam/>>. Acesso em: 11 out. 2023.

ANEXOS

Importando as dependências do projeto

```
[ ] import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Coleção de dados e Pre-processamento

```
[ ] # Carregando os dados do arquivo CSV usando pandas
raw_mail_data = pd.read_csv('/content/mail_data.csv')
```

```
▶ print(raw_mail_data)
```

```
📄 Category Message
0 ham Go until jurong point, crazy.. Available only ...
1 ham Ok lar... Joking wif u oni...
2 spam Free entry in 2 a wkly comp to win FA Cup fina...
3 ham U dun say so early hor... U c already then say...
4 ham Nah I don't think he goes to usf, he lives aro...
... ...
5567 spam This is the 2nd time we have tried 2 contact u...
5568 ham Will ü b going to esplanade fr home?
5569 ham Pity, * was in mood for that. So...any other s...
5570 ham The guy did some bitching but I acted like i'd...
5571 ham Rofl. Its true to its name
```

```
[5572 rows x 2 columns]
```

```
[ ] # Substituindo os dados nulos por uma string nula, assim conseguimos tratar melhor os dados
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)), '')
```

```
[ ] # Consolando as 5 (cinco) primeiras linhas do DataFrame
mail_data.head()
```

```
[ ] # Consolando as 5 (cinco) primeiras linhas do DataFrame
mail_data.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

```
[ ] # Checando os numeros de linhas e colunas na base de dados
mail_data.shape
```

```
(5572, 2)
```

Codificação da camada

```
▶ # Criando regra binaria para camada de spam e não spam, como: spam = 0 e não spam = 1
mail_data.loc[mail_data['Category'] == 'spam', 'Category',] = 0
mail_data.loc[mail_data['Category'] == 'ham', 'Category',] = 1
```

Spam - 0

Não spam - 1

```
[ ] # Separando os dados em textos e camadas
```

```
X = mail_data['Message']
```

```
Y = mail_data['Category']
```

```
[ ] # Consolando a resposta de X
print(X)
```

```
[ ] # Consolando a resposta de X
print(X)

0      Go until jurong point, crazy.. Available only ...
1              Ok lar... Joking wif u oni...
2      Free entry in 2 a wkly comp to win FA Cup fina...
3      U dun say so early hor... U c already then say...
4      Nah I don't think he goes to usf, he lives aro...
      ...
5567   This is the 2nd time we have tried 2 contact u...
5568               Will ü b going to esplanade fr home?
5569   Pity, * was in mood for that. So...any other s...
5570   The guy did some bitching but I acted like i'd...
5571               Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
[ ] # Consolando a resposta de Y
print(Y)

0      1
1      1
2      0
3      1
4      1
      ..
5567   0
5568   1
5569   1
5570   1
5571   1
Name: Category, Length: 5572, dtype: object
```

Dividindo os dados em dados de treinamento e dados de teste

- Item da lista
- Item da lista

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)
```

```
▶ print(X.shape)
print(X_train.shape)
print(X_test.shape)
```

```
(5572,)
(4457,)
(1115,)
```

Clique duas vezes (ou pressione "Enter") para editar

Recurso de extração

```
▶ # Transformando o texto em dados para apresentar vetores que podem ser usados como entrada para a regressão logística

feature_extraction = TfidfVectorizer(min_df = 1, stop_words='english', lowercase=True)

X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

# Convertendo os valores de Y_train e Y_test para inteiros

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

```
[ ] print(X_train)
```

```
3075          Don know. I did't msg him recently.
1787  Do you know why god created gap between your f...
1614          Thnx dude. u guys out 2nite?
4304          Yup i'm free...
3266  44 7732584351, Do you want a New Nokia 3510i c...
...
789    5 Free Top Polyphonic Tones call 087018728737,...
968    What do u want when i come back?.a beautiful n...
1667  Guess who spent all last night phasing in and ...
3321  Eh sorry leh... I din c ur msg. Not sad ahead...
1688  Free Top ringtone -sub to weekly ringtone-get ...
Name: Message, Length: 4457, dtype: object
```

```
[ ] print(X_train_features)

(0, 5413)      0.6198254967574347
(0, 4456)      0.4168658090846482
(0, 2224)      0.413103377943378
(0, 3811)      0.34780165336891333
(0, 2329)      0.38783870336935383
(1, 4080)      0.18880584110891163
(1, 3185)      0.29694482957694585
(1, 3325)      0.31610586766078863
(1, 2957)      0.3398297002864083
(1, 2746)      0.3398297002864083
(1, 918)       0.22871581159877646
(1, 1839)      0.2784903590561455
(1, 2758)      0.3226407885943799
(1, 2956)      0.33036995955537024
(1, 1991)      0.33036995955537024
(1, 3046)      0.2503712792613518
(1, 3811)      0.17419952275504033
(2, 407)       0.509272536051008
(2, 3156)      0.4107239318312698
(2, 2404)      0.45287711070606745
(2, 6601)      0.6056811524587518
(3, 2870)      0.5864269879324768
(3, 7414)      0.8100020912469564
(4, 50)        0.23633754072626942
(4, 5497)      0.15743785051118356
:
(4454, 4602)   0.2669765732445391
(4454, 3142)   0.32014451677763156
(4455, 2247)   0.37052851863170466
(4455, 2469)   0.35441545511837946
(4455, 5646)   0.33545678464631296
(4455, 6810)   0.29731757715898277
(4455, 6091)   0.23103841516927642
(4455, 7113)   0.30536590342067704
(4455, 3872)   0.3108911491788658
(4455, 4715)   0.30714144758811196
(4455, 6916)   0.19636985317119715
(4455, 3922)   0.31287563163368587
(4455, 4456)   0.24920025316220423
(4456, 141)    0.292943737785358
(4456, 647)    0.30133182431707617
(4456, 6311)   0.30133182431707617
(4456, 5569)   0.4619395404299172
(4456, 6028)   0.21034888000987115
(4456, 7154)   0.24083218452280053
(4456, 7150)   0.3677554681447669
```

Modelo de treinamento e Logica de regressão

```
[ ] model = LogisticRegression()
```

```
[ ] # Treinando o modelo de logica de regressão com os dados
model.fit(X_train_features, Y_train)
```

```
▼ LogisticRegression
LogisticRegression()
```

Avaliando o modelo de dados treinados...

```
[ ] # Previsão no treinamento dos dados
```

```
prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)
```

```
[ ] print('Accuracy on training data : ', accuracy_on_training_data)
```

```
Accuracy on training data : 0.9670181736594121
```

```
[ ] # Previsão nos testes de dados
```

```
prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)
```

```
[ ] print('Accuracy on test data : ', accuracy_on_test_data)
```

```
Accuracy on test data : 0.9659192825112107
```

Construindo um sistema preditivo e realizando o teste final

```
[ ] input_mail = ["I've been searching for the right words to thank you for this breather. I promise I won't take your help for granted and will fulfill my promise. You have been wonderful and a blessing at all times"]
# Convertendo os textos em vetores
input_data_features = feature_extraction.transform(input_mail)
# Realizando a previsão dos dados...
prediction = model.predict(input_data_features)
print(prediction)

if (prediction[0]==1):
    print('Ham mail')
else:
    print('Spam mail')
# teste realizado com sucesso!

[1]
Ham mail
```