

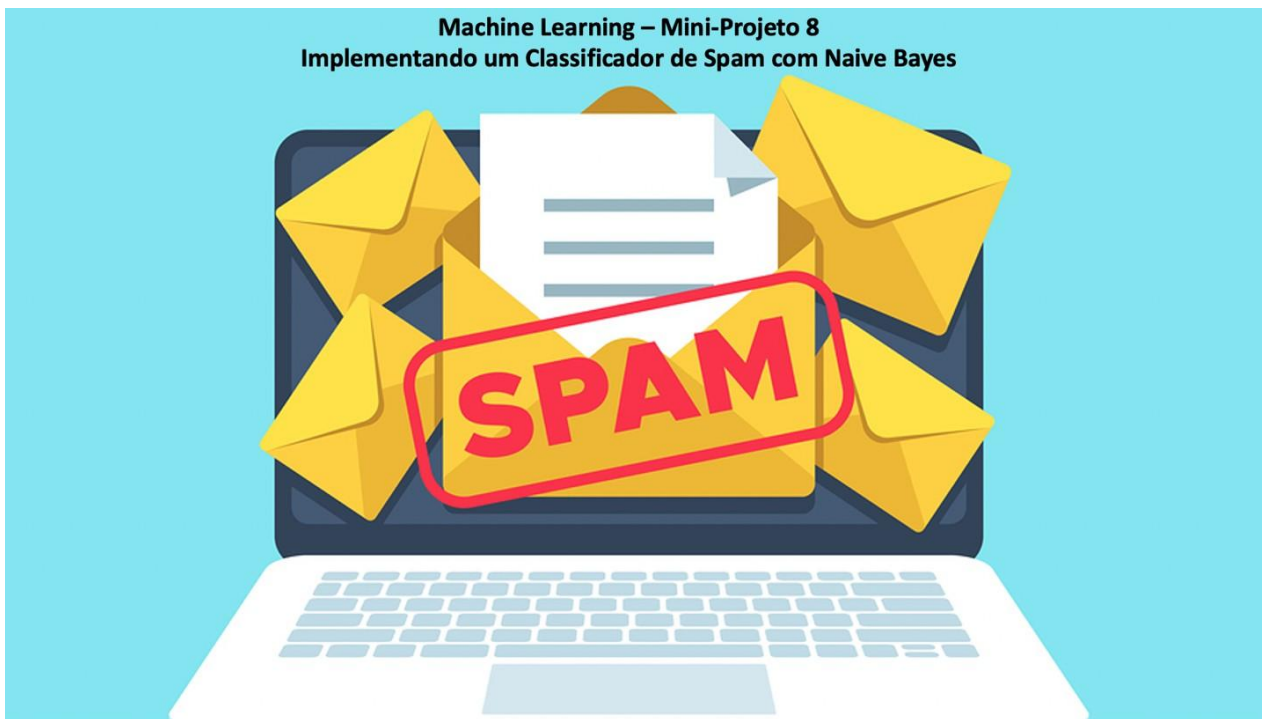


Machine Learning

Machine Learning

Mini-Projeto 8

Implementando um Classificador de Spam com Naive Bayes



O objetivo deste projeto é construir um classificador de Spam usando o algoritmo de classificação Naive Bayes. Construiremos esse algoritmo a partir do zero sem usar bibliotecas, o que será muito útil para construção de aplicações analíticas.

O modelo de documento que usaremos aqui é um modelo de saco de palavras (bag of words). Usaremos dois tipos de modelo bag of words:

A.Com base na presença de palavra (se uma palavra aparece no documento ou não, o que tornará os atributos de entrada binários)

B.Com base na frequência de palavras (frequência de ocorrência de palavra no documento, o que tornará os atributos de entrada contínuos)

Modelo Bag of Words

Um saco de palavras (bag of words) é uma representação de um texto como um agrupamento de palavras, sem qualquer consideração da sua estrutura gramatical ou da ordem das palavras. É simplesmente um histograma sobre as palavras da língua, e cada documento é representado como um vetor sobre estas palavras. As entradas neste vetor simplesmente correspondem à presença ou à ausência da palavra correspondente (quando se utiliza o tipo A acima ou a frequência da ocorrência da palavra quando se usa o caso B acima).

Execução

Nosso classificador é na verdade um aplicativo que será executado via linha de comando. Você precisa ter o interpretador do Python 3 instalado (se você instalou Anaconda, você já possui o interpretador Python 3 instalado). A execução do aplicativo deve ser feito da seguinte forma:

- 1-Abra um terminal ou prompt de comando.
- 2-Navegue até o diretório onde estão os arquivos que você baixou.
- 3-Execute o aplicativo para os dados de treino, a fim de treinar o modelo.
- 4- Execute o aplicativo com os dados de teste, para avaliar o modelo.
- 5- Atingindo o nível de acurácia desejado, seu aplicativo analítico para classificação de Spam está pronto para receber novos conjuntos de dados e realizar a classificação do que é spam e do que não é.

Para executar, digite:



Para treinar o modelo:

```
python spam.py train bayes train/ output/resultado.txt
```

A terminal window titled "1. bash" with a light gray title bar and three colored window control buttons (red, yellow, green) on the left. The terminal background is black, and the text is white. The prompt is "eagle:~/Dropbox/DSA/Projetos/Projeto13 \$". The command entered is "python spam.py train bayes train/ output/resultado.txt". The cursor is at the end of the command line.

python – nome do interpretador

spam.py – nome do seu aplicativo Python (nome do script)

train – tipo de operação do aplicativo, que será executado em modo treinamento

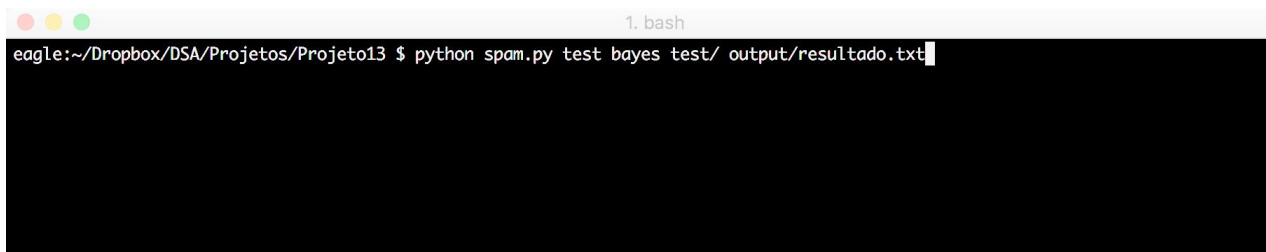
bayes – nome do método de classificação, no caso Naive Bayes, mas você pode mais tarde implementar outros algoritmos

train/ - nome do diretório onde estão os dados de treino

output/resultado.txt – arquivo onde será gravado o resultado do modelo de classificação treinado

Para testar o modelo:

```
python spam.py test bayes test/ output/resultado.txt
```

A terminal window titled "1. bash" with a light gray title bar and three colored window control buttons (red, yellow, green) on the left. The terminal background is black, and the text is white. The prompt is "eagle:~/Dropbox/DSA/Projetos/Projeto13 \$". The command entered is "python spam.py test bayes test/ output/resultado.txt". The cursor is at the end of the command line.

Se você atingiu o nível de acurácia desejado, sua aplicação analítica está pronta e poderá receber novos dados a fim de classificá-los como spam ou não spam.

Seu trabalho agora é fazer a leitura do script atentamente e compreender tudo que está realizado. Você vai reconhecer tudo que aprendeu ao longo do curso, principalmente no capítulo sobre Naive Bayes.

Esse script está completamente automatizado, exatamente como funciona uma aplicação analítica.

Treino:

```

1. bash
eagle:~/Dropbox/DSA/Projetos/Projeto13 $ python spam.py train bayes train/ output/resultado.txt
Treinando o Algoritmo Naive Bayes. Isso pode levar algum tempo. Pegue um café e relaxe...
Primeiro contamos as palavras no dataset de treino. Estou processando. Seja paciente e aguarde...
#### Top 10 palavras mais associadas com spam: ####
#####
#### Top 10 palavras menos associadas com spam: ####
#####
Salvando o modelo no arquivo especificado..
eagle:~/Dropbox/DSA/Projetos/Projeto13 $

```

Teste:

```

1. bash
eagle:~/Dropbox/DSA/Projetos/Projeto13 $ python spam.py test bayes test/ output/resultado.txt
#### Fazendo previsões com a presença de palavras no modelo... ####
Prevedo Acurácia para a classe: notspam
### Total de observações analisadas: 1369
### Observações com Classificação Correta: 1353
### Acurácia: 0.99
Prevedo Acurácia para a classe: spam
### Total de observações analisadas: 1185
### Observações com Classificação Correta: 1139
### Acurácia: 0.96
#####
#### Fazendo previsões com a frequência de palavras no modelo... ####
Prevedo Acurácia para a classe: notspam
### Total de observações analisadas: 1369
### Observações com Classificação Correta: 1348
### Acurácia: 0.98
Prevedo Acurácia para a classe: spam
### Total de observações analisadas: 1185
### Observações com Classificação Correta: 1134
### Acurácia: 0.96
### Obrigado pela sua participação no curso de Machine Learning da Data Science Academy. Esperamos revê-lo em breve ###

```

Este mini-projeto é um estudo dirigido e todos os arquivos do mini-projeto estão em anexo!