PROJETO 2

CLASSIFICADOR NAIVE BAYES

CLASSIFICADOR NAIVE BAYES PARA CLASSIFICAR MENSAGEM SPAM OU HAM

1. PROBLEMA

O Classificador Naive-Bayes, o qual se baseia no uso do teorema de Bayes, é largamente utilizado em filtros anti-spam de e-mails. O classificador permite calcular qual a probabilidade de uma mensagem ser SPAM considerando as palavras em seu conteúdo e, de forma complementar, permite calcular a probabilidade de uma mensagem ser HAM dada as palavras descritas na mensagem.

Para realizar o MVP (minimum viable product) do projeto, você precisa programar uma versão do classificador que "aprende" o que é uma mensagem SPAM considerando uma base de treinamento e comparar o desempenho dos resultados com uma base de testes.

2. Separação da base de dados em Treinamento e Teste

A base de dados deve ser separada em duas partes, aleatoriamente, considerando:

- 75% dos dados para a parte Treinamento. Essa parte será considerada para criar um classificador que aprenda quais são as palavras pertencentes às mensagens SPAM e quais são as pertencentes às mensagens HAM.
- 25% dos dados para a parte Teste. Essa parte será considerada para classificar as mensagens como SPAM ou HAM utilizando o classificador criado no item anterior.

Obs.: Apenas aqui sua dupla pode usar alguma biblioteca que possua um comando já pronto que realiza essa separação na base de dados.

3. Classificador Naive-Bayes

Para desenvolver o Classificador, faça:

1. Com a base de treinamento montada:
    a. Limpe as mensagens removendo caracteres que não afetam a classificação: :; "; '; (,), etc.
    b. Corrija a separação de espaços entre palavras e/ou emojis.
    c. Proponha outras limpezas/transformações que não afetem a qualidade da informação.
2. Calcule as probabilidades marginais e condicionais necessárias para classificar uma nova mensagem. Ou seja, calcule:

    $P(SPAM)$: probabilidade de uma mensagem ser SPAM;
    $P(HAM) = P(SPAM^C)$: probabilidade de uma mensagem ser HAM;
    $P(word \mid SPAM)$: probabilidade de uma palavra ($word$) acontecer se a mensagem na base treinamento é considerada SPAM;
    $P(word \mid HAM)$: probabilidade de uma palavra acontecer se a mensagem na base treinamento é considerada HAM.

3. Agora, para classificar uma mensagem da base de teste, terá que saber calcular as probabilidades:

    $P(SPAM \mid mensagem) = ?$ (1)
    $P(HAM \mid mensagem) = ?$ (2)

    Nesse caso, estude atentamente como fazer isso utilizando o link:
    https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/

4. Para testar o seu Classificador, considere a base de teste. Para tanto, sua dupla deve extrair as seguintes medidas:

    % de falsos positivos (mensagens marcadas como SPAM mas não são SPAM);
    % de positivos verdadeiros (mensagens marcadas como SPAM e são SPAM);
    % de falsos negativos (mensagens marcadas como não SPAM mas são SPAM);
    % de negativos verdadeiros (mensagens marcadas como não SPAM e não são SPAM).

5. Discuta a qualidade do seu Classificador!

4. Curva ROC

Para definir se uma mensagem será classificada como SPAM ou HAM, é necessário escolher uma regra de predição utilizando as probabilidades descritas nas equações **(1)** e **(2)**. Certamente sua dupla escolheu alguma regra no Classificador criado anteriormente.

De qualquer forma, é intuitivo pensar que:

- ✓ se o valor de $P(SPAM \mid mensagem)$ for grande (perto de 1), a mensagem deva ser classificada como SPAM; e
- ✓ se $P(SPAM \mid mensagem)$ for pequeno (perto de 0), a mensagem deva ser classificada como não SPAM (ou HAM).

Mas como determinar o ponto que para os valores dessa probabilidade próximos de 0,5, por exemplo. Esse ponto é conhecido como ponto de corte. A curva ROC pode ser uma alternativa para determinar o melhor ponto de corte.

Considerando a base de treinamento e a base de teste definidas no tópico acima **Classificador Naive-Bayes**, construa a curva ROC alternando, no seu Classificador, o ponto de corte entre valores de 0 e 1. Essa alteração deverá refletir mudanças nos itens 3 e 4 apenas.

Antes, porém, faça a leitura atenta dos links abaixo:

http://www.portalaction.com.br/analise-de-regressao/45-predicao

http://crsouza.com/2009/07/13/analise-de-poder-discriminativo-atraves-de-curvas-roc/

5. Qualidade do Classificador alterando a base de treinamento

Um importante passo no aprendizado de máquina é trabalhar com uma boa base de dados para o treinamento e teste do seu classificador. Entretanto, é razoável pensar que a divisão de dados utilizada no seu Classificador representa uma entre muitas possíveis combinações em dividir a base de dados total em ¾ para treinamento e ¼ para teste.

Assim sendo, aqui o objetivo é avaliar como a base de dados treinamento pode interferir numa melhor ou não tão boa classificação das mensagens contidas na base de teste.

Nesse caso, faça:

1. Repita 10.000 vezes o tópico **Separação da base de dados em Treinamento e Teste**;
2. Para cada base separada, faça os itens de 1 a 4 descritos no tópico **Classificador Naive-Bayes** e guarde os percentuais de acertos (= % de positivos verdadeiros + % de negativos verdadeiros);
3. Construa um histograma com esses percentuais de acertos;
4. Discuta o resultado do histograma refletindo sobre possíveis vantagens ou desvantagens sobre construir um Classificador considerando uma única vez a divisão da base de dados em treinamento e em teste.

REGRAS:

1. O Projeto 2 é estritamente INDIVIDUAL ou em DUPLA;
2. Use o **arquivo layout** disponibilizado na pasta Projeto2 do GitHub ou Blackboard;
3. O projeto deve ser claro e organizado respeitando o que foi pedido com interpretações e/ou conclusões quando solicitadas.

   A estrutura do documento deve ser clara e de fácil compreensão da linha de raciocínio. Nesse caso, o notebook não deve haver excesso de impressões não discutidas ou de códigos não utilizados.

   Após finalização do projeto, aconselhamos que sua dupla faça uma análise geral e salve com outro nome, limpe seu IPython Notebook apenas com os resultados relevantes e melhore seu texto.
4. Seu projeto deve ser adicionado no seu GitHub dentro de uma pasta chamada Projeto2.
5. O arquivo DEVE ser com extensão .ipynb

CRONOGRAMA:

DATA	**Finalização:**
21/03	APS3: Check descrito na Aula 08 Agende seu horário no link https://doodle.com/poll/cpf4hct47af25nwt
22/03	Aula Estúdio É importante que o projeto já esteja quase concluído
23/03	PROJETO 2 FINALIZADO Fazer git push em seu Github até 23:59 do Projeto 2 finalizado.

RUBRICS DE AVALIAÇÃO DO OBJETIVO DE APRENDIZADO

Objetivo de aprendizado	Insatisfatório (I)	Em desenvolvimento (D)	Essencial (C)	Proficiente (B)	Avançado (A)
Aplicar teoria de probabilidade adequadamente. **(Especificar as distribuições de probabilidades adequadas para as variáveis quantitativas discretas e/ou contínuas)**	Não consegue trabalhar com bases de dados de forma proficiente. Apresenta problemas com arquivos, formatos de arquivos ou não tem habilidades básicas de separação dos dados.	Consegue abrir e separar a base de dados em treinamento e em teste. O notebook é praticamente uma coleção de comandos com pouca análise e sem chegar a construir o classificador adequadamente.	Descreve com riqueza de detalhes o tópico **Classificador Naive-Bayes** descrito no Projeto 2. É capaz de separar a base de dados e programar um algoritmo de classificação que considera o teorema de Bayes.	Executa os comportamentos da rubrica C de maneira excepcional. Descreve com riqueza de detalhes o tópico **Curva ROC** descrito no Projeto 2 **Ou** Descreve com riqueza de detalhes o tópico **Qualidade do Classificador alterando a base de treinamento** descrito no Projeto 2. Comunica os resultados com bastante clareza.	Executa os comportamentos da rubrica B de maneira excepcional. Descreve com riqueza de detalhes **tanto** o tópico **Curva ROC** **como** o tópico **Qualidade do Classificador alterando a base de treinamento** descrito no Projeto 2. Expõe os resultados obtidos de maneira excepcional.