

## 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;

$P(word | SPAM)$ : probabilidade de uma palavra (*word*) acontecer se a mensagem na base treinamento é considerada SPAM;

$P(word | HAM)$ : 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:

$$P(SPAM | mensagem) = ? \quad (1)$$

$$P(HAM | mensagem) = ? \quad (2)$$

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. 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  $\frac{3}{4}$  para treinamento e  $\frac{1}{4}$  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 em até TRIO;
2. Use o **arquivo layout** disponibilizado na pasta Projeto2 do GitHub ou Blackboard;
3. Vocês poderão **consultar livros, internet e anotações DESTE SEMESTRE pertencentes apenas aos integrantes do grupo**. Todavia, nenhuma forma de plágio será tolerada e pode acarretar a reprovação de todos os integrantes do grupo. Ainda, é extremamente importante ter ciência e real compreensão sobre a definição de plágio tanto quando se envolve texto com código. Leia o [Manual do Aluno disponível no Portal do Aluno do Insper](#) e as regras disponíveis [aqui](#).
4. 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 seu trio faça uma análise geral e salve com outro nome, limpe seu IPython Notebook apenas com os resultados relevantes e melhore seu texto.**
5. Seu projeto deve ser adicionado no seu GitHub dentro de uma pasta chamada Projeto2.
6. O arquivo DEVE ser com extensão .ipynb

## **CRONOGRAMA:**

DATA	Finalização:
14/03	Aula Estúdio
19/03 e 21/3	APS4: Check a ser disponibilizado É importante que o projeto já esteja quase concluído
<b>20/03</b>	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)
<p><b>Aplicar teoria de probabilidade adequadamente.</b></p> <p><b>(Especificar as distribuições de probabilidades adequadas para as variáveis quantitativas discretas e/ou contínuas)</b></p>	<p>Não consegue trabalhar com bases de dados de forma proficiente.</p> <p>Apresenta problemas com arquivos, formatos de arquivos ou não tem habilidades básicas de separação dos dados.</p>	<p>Consegue abrir e separar a base de dados em treinamento e em teste.</p> <p>O notebook é praticamente uma coleção de comandos com pouca análise e sem chegar a construir o classificador adequadamente.</p>	<p>Descreve com riqueza de detalhes o tópico <b><u>Classificador Naive Bayes</u></b> descrito no Projeto 2.</p> <p>É capaz de separar a base de dados e programar um algoritmo de classificação que considera o teorema de Bayes.</p>	<p>Executa os comportamentos da rubrica C de maneira excepcional.</p> <p>Comunica os resultados com bastante clareza.</p>	<p>Executa os comportamentos da rubrica B de maneira excepcional.</p> <p>Descreve com riqueza de detalhes o tópico <b><u>Qualidade do Classificador alterando a base de treinamento</u></b> descrito no Projeto 2.</p> <p>Expõe os resultados obtidos de maneira excepcional.</p>