PROJETO 1

CLASSIFICADOR NAIVE-BAYES COM TEXTO NATURAL

CLASSIFICADOR AUTOMÁTICO DE TEXTOS

Com os mercados digitais e redes sociais, muitas empresas recebem críticas na forma de textos corridos. A vantagem desse tipo de feedback é que os clientes podem escrever o que quiserem. A desvantagem é que é difícil organizar tanta informação, de forma que aspectos importantes podem ser perdidos em meio às mensagens.

No caso de críticas a livros, é claro que há críticas que são acionáveis, como “O conteúdo só é bom para iniciantes no assunto” (significando que este cliente em questão gostaria de um livro com maior profundidade), e outras críticas que não são acionáveis, como “O autor fala, fala e não diz nada” (o que pode querer dizer muitas coisas), e é difícil tomar atitudes quanto a isso. O problema é que, dentre algumas críticas acionáveis, há uma multidão de críticas não-acionáveis.

Pensando numa solução para este problema, a equipe de projeto deve criar um sistema automático que encontre os reviews acionáveis dentre aqueles que foram falados. Algo ainda melhor seria encontrar quem é o responsável (o autor ou a editora) por lidar com a crítica. Talvez o grupo de trabalho possa propor alguma análise ainda mais acurada quanto ao que deve ser feito com cada uma das críticas – deseja encontrar aquelas que são mal-educadas? Encontrar as que são realmente relevantes para a compra? Isso ficará a critério do grupo.

Com base em seus conhecimentos de Ciência dos Dados, você lembrou do Teorema de Bayes, mais especificamente do Classificador Naive-Bayes, que é largamente utilizado em filtros anti-spam de e-mails, por exemplo. Esse classificador permite calcular qual a probabilidade de uma mensagem ser acionável dada as palavras em seu conteúdo.

Para realizar a POC (*proof-of-concept*) do projeto, você precisa implementar uma versão do classificador que "aprende" o que é uma crítica acionável com uma base de treinamento e compara a performance dos resultados com uma base de testes.

Após validado, o seu protótipo poderia, porque não, também capturar e classificar automaticamente as mensagens da plataforma.

OBTENÇÃO DOS MENSAGENS

As mensagens serão obtidas de forma **automática** com o uso de um programa em Python já fornecido pelos professores!

Todos os detalhes de captura dos dados estão descritos no Jupyter Notebook na pasta **notebooks/scraping**/.

Ao executar o notebook e escolher o assunto, serão criados os arquivos **dados_treino.xlsx e dados_teste.xlsx** na pasta **do seu notebook** contendo as mensagens que serão utilizadas no projeto.

ETAPAS DO PROJETO

Para entregar um projeto de sucesso, você deve seguir os seguintes passos:

1. Coleta dos exemplos

Usando o notebook fornecido como exemplo, baixe a base de dados de reviews de livros da Amazon

2. Definindo a variável Target

Com base nas mensagens, será necessário criar uma variável **Target**. A variável **Target** representa o que se deseja aprender a partir dos dados. Em um cenário de negócios, a variável **Target** viria da necessidade apontada por alguma área cliente da empresa, entretanto, neste projeto ela deverá ser definida pelo grupo. No caso deste trabalho, a variável Target poderia ser a classificação dos reviews como ACIONÁVEL ou NÃO-ACIONÁVEL, por exemplo.

Alguns exemplos de assuntos e variável *target*:

- Em mensagens sobre o assunto **Petrobras**, a variável **Target** poderia ser a classificação dos tweets como **NEUTROS**, **POSITIVOS** ou **NEGATIVOS**. Neste caso, os tweets Neutros poderiam ser aqueles onde a Petrobras não é assunto principal e é citada superficialmente. Já os tweets NEGATIVOS poderiam ser aquelas que comentam de assuntos danosos para a marca Petrobras (danos ao meio ambiente, acidentes de trabalho, questões jurídicas, etc.). Por fim, os POSITIVOS seriam aqueles que comentam sobre ações benéficas para a marca Petrobras (ações sociais, boas práticas ESG, lucros, etc.)

- Um fundo de investimentos quer monitorar o cenário **Crypto**. Assim, a variável **Target** envolveria duas categorias (**RELEVANTE**, **IRRELEVANTE**). Os tweets RELEVANTES seriam aqueles que comentam sobre queda ou crescimento no preço de ativos. Os IRRELEVANTES seriam os tweets que comentam sobre qualquer outro assunto, relacionado ou não ao universo Crypto.
- Uma varejista quer identificar se as avaliações feitas nos **produtos** realmente são uma avaliação do produto (**RELEVANTE**) ou se são uma avaliação do serviço prestado (**IRRELEVANTE**), como problemas no envio ou produtos danificados.

O grupo ficará livre para definir: o **assunto** que deseja classificar; e a quantidade de categorias (ver rubrica) para construção da variável **Target**.

3. Classificando as mensagens na coragem

Agora você deve abrir o arquivo Excel com as mensagens capturadas e classificar cada uma conforme a definição utilizada pelo grupo para a variável **Target**. Guarde essa classificação na coluna **Target** (não renomeie esta coluna e a crie caso necessário), utilizando números inteiros a partir de zero para identificar as classes (Ex: relevante utiliza valor 1 e irrelevante utiliza valor 0).
Faça o mesmo na planilha de **Teste**.

Um ponto de atenção nesta etapa é evitar que cada membro do grupo classifique as mensagens com critérios diferentes. Conversem e garantam que o critério que define a variável **Target** esteja bem definido entre os membros do grupo. Também é recomendado que todos os membros do grupo atuem tanto na classificação do treino quanto do teste e revisem a classificação dos colegas para diminuir a incidência de viés.

4. Montando o classificador Naive-Bayes

Considerando apenas as mensagens da planilha **Treinamento**, o objetivo aqui é ensinar o seu classificador quais são as palavras mais comuns (frequentes) nas mensagens de cada categoria.
Nesse caso, seu código deve conter preferencialmente:

- ✔ Limpeza de mensagens removendo os caracteres: enter, :, ", ', (,), etc.
- ✔ Proposta de outras limpezas/transformações que não afetem a qualidade da informação.
- ✔ Suavização de Laplace: link1 (com leitura até **antes** da seção "Creating a naive bayes classifier with Monkeylearn") e link2.

5. Verificando a *performance*

Considerando agora apenas as mensagens da planilha **Teste**, seu objetivo aqui é testar a qualidade do seu classificador.

Para tanto, você deve extrair as seguintes contagens:

- ✔ Porcentagem de verdadeiros positivos (Ex: mensagens relevantes e que são classificadas como relevantes)
- ✔ Porcentagem de falsos positivos (Ex: mensagens irrelevantes e que são classificadas como relevantes)
- ✔ Porcentagem de verdadeiros negativos (Ex: mensagens irrelevantes e que são classificadas como irrelevantes)
- ✔ Porcentagem de falsos negativos (Ex: mensagens relevantes e que são classificadas como irrelevantes)
- ✔ Acurácia (mensagens corretamente classificadas, independente da categoria)

Opcionalmente:

- ✔ Criar categorias intermediárias de relevância baseado na diferença de probabilidades. Exemplo: muito relevante, relevante, neutro, irrelevante e muito irrelevante.

6. Concluindo

Faça um comparativo qualitativo sobre os percentuais obtidos para que possa discutir a *performance* do seu classificador.

Explique como são tratadas as mensagens com dupla negação e sarcasmo.

Proponha um plano de expansão. Por que eles devem continuar financiando o seu projeto?

Opcionalmente:

- ✔ Propor diferentes cenários de uso para o classificador Naive-Bayes. Pense em outros cenários sem intersecção com este projeto.
- ✔ Sugerir e explicar melhorias reais no classificador com indicações concretas de como implementar (não é preciso codificar, mas indicar como fazer. Indique material de pesquisa sobre o assunto).

7. Qualidade do Classificador a partir de novas separações das notícias entre Treinamento e Teste

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 o total de notícias em 300 para treinamento e 200 para teste.

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

Nesse caso, faça:

- ✔ Junte todas as mensagens do **Treinamento** e do **Teste** em único *dataframe* (vamos supor que sejam 500) e separe, de forma aleatória, 300 mensagens para ficar na base de dados treinamento e 200 na base de dados teste. **Obs.: Apenas aqui sua dupla poderá usar alguma biblioteca que possua um comando já pronto que realiza essa separação na base de dados (procure no google "split em train e test")**;
- ✔ Para cada base separada, faça os itens de 3 a 4 descritos no tópico **Etapas do projeto** e guarde os percentuais de acertos (= % de positivos verdadeiros + % de negativos verdadeiros);
- ✔ Repita os dois passos acima 100 vezes.

Construa um histograma com esses percentuais de acertos e 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 1 é em DUPLA. No caso de TRIO, terá rubrica diferente para seguir.
2. O projeto será corrigido conforme os critérios da rubrica.
3. Use os **notebooks** disponibilizados no github classroom.
4. Os entregáveis deverão ser colocados no Blackboard:

   ✔ Arquivos notebooks com o código para obter as notícias e com código do classificador, seguindo layout dos notebooks disponibilizados na pasta Projeto 1.

   ✔ Arquivo Excel com as notícias de treinamento e teste totalmente classificadas.

   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 de variáveis e de *dataframe*.

   Aconselhamos fazer uma análise geral e, após finalizada, salve com outro nome, limpe seu IPython Notebook apenas com os resultados relevantes e melhore seu texto.

ENTREGAS

As entregas deverão ser feitas via Blackboard, nos locais relacionados à atividade. Caso etapas sejam atrasadas, haverá desconto conforme disponível no cronograma.

CRONOGRAMA

DATA	Finalização:
01/09 (sexta)	Cadastro do grupo no Blackboard: ✔ Dupla ou trio formado.
05/09 (terça)	Deve estar no Blackboard até 23h59: ✔ Arquivos Excel **dados_treino.xlsx** e **dados_teste.xlsx** contendo a base de treinamento e teste **sem classificação manual (Target).**
12/09 (terça)	Deve estar no Blackboard até às 23h59 com as seguintes evidências: ✔ Arquivos Excel **dados_treino.xlsx** e **dados_teste.xlsx** contendo a contendo a base de treinamento e teste **já classificados manualmente**.
14/09 (quinta)	Deve estar no Blackboard até às 23h59 com as seguintes evidências: ✔ Arquivos Excel **dados_treino.xlsx** e **dados_teste.xlsx** contendo a as mensagens de treinamento e teste totalmente classificadas manualmente. ✔ **Arquivo relatorio.ipynb** com o código do classificador e análise dos resultados, seguindo *layout*.

RUBRICA

NÍVEL	DESCRIÇÃO
I	Não entregou Entregou, mas não tem sequer a base de dados para treinamento e teste A base não tem rótulos feitos manualmente (Target)
D	Tem a base de dados para treinamento e testes, mas o classificador não funciona Classificou manualmente (Target) menos que 500 notícias. Existem rotinas para cálculos de probabilidades, mas as fórmulas ou cálculos estão errados, ou não funciona
C	Entregou; Tem a base de dados para treinamento e testes; Limpou: \n, :, '', ', (,), etc Rotinas funcionam, mas a análise não ficou completa; ou não ficou boa O notebook tem excesso de blocos de código ou impressões não discutidas Possui pequenos erros TANTO na suavização de Laplace (Smoothing) QUANTO no Naïve Bayes (Ex: não usar a frequência correta das palavras, esquecer da priori)
B	Entregou; Tem a base de dados para treinamento e testes; Limpou: \n, :, '', ', (,), etc. Produziu um texto de qualidade na análise crítica da performance do classificador (item Concluindo do enunciado) Utilizou métricas adequadas para a análise da qualidade do classificador (item Verificando a performance do enunciado) Mas existe um pequeno erro na suavização de Laplace (Smoothing) OU no Naïve Bayes (não em ambos)
CASO SEU PROJETO SE ENQUADRE EM ALGUM DOS NÍVEIS ACIMA, ENTÃO OS ITENS AVANÇADOS SERÃO IGNORADOS; **SENÃO, SEU NÍVEL SERÁ PELA CONTAGEM DE ITENS AVANÇADOS:** **B+ : 3 itens** **A : 4 ou 5 itens** **A+ : 6 ou 7 itens**	IMPLEMENTOU outras limpezas e transformações que não afetem a qualidade da informação contida nas notícias. Ex: stemming, lemmatization, stopwords
	CONSIDEROU mais de duas categorias na variável Target e INCREMENTOU a quantidade de notícias, mantendo pelo menos 250 notícias por categoria **(OBRIGATÓRIO PARA TRIOS, sem contar como item avançado)**
	CONSTRUIU o cálculo das probabilidades corretamente utilizando bigramas E apresentou referência sobre o método utilizado.
	EXPLICOU porquê não pode usar o próprio classificador para gerar mais amostras de treinamento
	PROPÔS diferentes cenários para Naïve Bayes fora do contexto do projeto (pelo menos dois cenários, exceto aqueles já apresentados em sala pelos professores: por exemplo, filtro de spam)
	SUGERIU e EXPLICOU melhorias reais com indicações concretas de como implementar (indicar como fazer e indicar material de pesquisa)
	FEZ o item Qualidade do Classificador a partir de novas separações das Notícias entre Treinamento e Teste descrito no enunciado do projeto **(OBRIGATÓRIO para conceitos A ou A+)**