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.

ESCOLHA DA BASE DE DADOS

ATÉ TRIO (de 1 a 3 alunos no grupo)

A base de dados foi extraído da plataforma Kaggle e tem o seguinte contexto:

Essa base de dados foi compilada para abranger artigos textuais que incluem terminologia comum, conceitos e definições nas áreas de ciência da computação, inteligência artificial e segurança cibernética. O conjunto de dados contém textos gerados por humanos, provenientes de diferentes dicionários e enciclopédias de ciência da computação, bem como textos gerados pelo ChatGPT da OpenAI em resposta a perguntas postadas manualmente.

O conjunto de dados (base de dados original) sentence_level_data.csv possui um total de 7344 entradas, compostas por 4008 sentenças geradas por IA e 3336 geradas por humanos. Os textos gerados por inteligência artificial (IA) e por humanos foram combinados em uma única coluna, recebendo rótulos apropriados, em outra coluna, indicados por: **0 = Gerado por Humano** e **1 = Gerado por IA**).

Aqui, use **obrigatoriamente** o arquivo Cria base de dados Treino e Teste – TRIO.ipynb para obter mensagens aleatórias da base de dados original. Ao rodar esse notebook, serão criados dois arquivos no seu computador no mesmo diretório que salvou este notebook. Esses novos arquivos, com extensão csv, terão o username que colocar ao rodar o notebook. Por exemplo,

dados_treino_TRIO_mariakv.csv

dados_teste_TRIO_mariakv.csv

ATÉ QUARTETO (de 1 a 4 alunos no grupo)

A base de dados foi extraído da plataforma Kaggle e tem o seguinte contexto:

O conjunto de dados consiste em avaliações (*reviews*) de passageiros, as quais fornecem informações sobre a satisfação do cliente quanto a qualidade do serviço.

O conjunto de dados (base de dados original) Airline Passenger Reviews.csv possui um total de 64017 avaliações, segmentadas em três categorias considerando a metodologia *Net Prometer Score (NPS)*: **Detractor** (Detrator); **Promoter** (Promotor) e **Other** (Outro). Essas categorias ajudam a avaliar o nível geral de satisfação do cliente.

Vamos explicar o significado de cada termo:

Detractor (Detrator):

Significado: Refere-se aos clientes que expressaram insatisfação ou descontentamento significativo com o serviço ou produto.

Classificação: Geralmente, os clientes que atribuem uma pontuação de 0 a 6 (em uma escala de 0 a 10) são considerados detratores.

Promoter (Promotor):
Significado: Representa os clientes extremamente satisfeitos e leais ao serviço ou produto.
Classificação: Os clientes que atribuem uma pontuação de 9 ou 10 são considerados promotores. Esses clientes são vistos como defensores entusiasmados da marca.

Other (Outro):
Significado: Engloba as respostas que não se enquadram nas categorias específicas de detratores ou promotores.
Classificação: Geralmente inclui as pontuações intermediárias, como 7 e 8 na escala de 0 a 10.

Aqui, use **obrigatoriamente** o arquivo Cria base de dados Treino e Teste – QUARTETO.ipynb para obter mensagens aleatórias da base de dados original. Ao rodar esse notebook, serão criados dois arquivos no seu computador no mesmo diretório que salvou este notebook. Esses novos arquivos, com extensão csv, terão o *username* que colocar ao rodar o notebook. Por exemplo,

dados_treino_QUARTETO_mariakv.csv

dados_teste_QUARTETO_mariakv.csv

ETAPAS DO PROJETO

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

1. Criação das base de dados de treinamento e de teste

Usando o notebook de acordo com a quantidade de alunos no grupo, crie os conjuntos de dados necessários para contruir seu classificador a partir do Teorema de Bayes.

2. Montando o classificador Naive-Bayes

Use o arquivo **Projeto1 Template.ipynb** disponibilizado no Blackboard como template para construir o Projeto 1 conforme demanda abaixo.

Considerando apenas as mensagens classificadas armazenadas no arquivo **dados_treino_....csv**, 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.

3. Verificando a *performance*

Considerando agora apenas as mensagens classificadas armazenadas no arquivo **dados_teste_....csv**, 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)

4. 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).

5. Qualidade do Classificador a partir de novas separações das mensagens 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 mensagens em treinamento e em teste.

Assim sendo, aqui o objetivo é avaliar como as mensagens 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* (TRIO: 2100; QUARTETO: 6000) e separe, de forma aleatória, em 70% de mensagens para ficar na base de dados treinamento e 30% para ficar na base de dados teste. **Obs.: Apenas aqui seu grupo 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 2 a 3 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.

Ciência dos dados - ENGENHARIA 2024 / 1º. Semestre

Profa. Maria Kelly Venezuela

REGRAS

1. O Projeto 1 é em até TRIO. No caso de QUARTETO, terá base de dados e rubrica diferentes para seguir.
2. O projeto será corrigido conforme os critérios da rubrica.
3. Use os **notebooks** disponibilizados no Blackboard.
4. Os entregáveis deverão ser colocados no Blackboard:

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

   ✔ Arquivos csv treinamento e teste.

   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.

Ciência dos dados - ENGENHARIA 2024 / 1º. Semestre

Profa. Maria Kelly Venezuela

CRONOGRAMA

DATA	Finalização:
27/02 (terça)	Cadastro do grupo no Blackboard: ✔ Trio ou quarteto formado.
28/02 (quarta)	Deve estar no Blackboard até 23h59: ✔ Leiam a seção CRIAÇÃO DAS BASE DE DADOS DE TREINAMENTO E DE TESTE para fazer este item. ✔ Anexar no Blackboard os arquivos: **dados_treino_....csv** e **dados_teste_....csv** contendo as bases de treinamento e teste**.**
29/02 (quinta)	✔ Aula estúdio (sem entregável)
14/03 (terça) **FINAL**	Deve estar no Blackboard até às 23h59 com as seguintes evidências: ✔ Arquivos Excel **dados_treino_....csv** e **dados_teste_....csv** contendo a as mensagens de treinamento e teste. ✔ Arquivo **Projeto1 Template.ipynb** com o código do classificador e análise dos resultados, seguindo *layout* descrito nesse notebook.

Ciência dos dados - ENGENHARIA 2024 / 1º. Semestre

Profa. Maria Kelly Venezuela

RUBRICA

NÍVEL	DESCRIÇÃO
I	Não entregou Entregou, mas não tem sequer a base de dados para treinamento e teste
D	Entregou; Tem a base de dados para treinamento e testes, mas o classificador não funciona Existem rotinas para cálculos de probabilidades, mas as fórmulas ou cálculos estão errados, ou não funciona. **Qualquer outra rotina (comandos) feitos a mais serão desconsiderados na correção, caso os descritos cima estejam válidos.**
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) **Qualquer outra rotina (comandos) feitos a mais serão desconsiderados na correção, caso os descritos cima estejam válidos.**
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) **Qualquer outra rotina (comandos) feitos a mais serão desconsiderados na correção, caso os descritos cima estejam válidos.**
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 arquivo com três categorias na classificação das variáveis (**OBRIGATÓRIO PARA QUARTETOS, 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 diferentes, 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 mensagens entre Treinamento e Teste descrito no enunciado do projeto **(OBRIGATÓRIO para conceitos A ou A+)**