

ACH2016 - INTELIGÊNCIA ARTIFICIAL – 2021
ESPECIFICAÇÃO DO SEGUNDO TRABALHO
Profa. Patrícia Oliveira

Data de apresentação: 13/01/2022

Entrega no e-disciplinas: até o dia 12/01/2022 às 23:59. Devem ser postados: a) o código fonte, o código executável (junto com eventuais instruções de execução do mesmo) e a apresentação em PowerPoint. Basta uma postagem por grupo. Não se esqueçam de colocar a identificação dos integrantes nos arquivos editáveis; b) a avaliação de cada membro da equipe (nota e comentário). Não haverá autoavaliação. Esta postagem é individual.

Observação 1: Este trabalho deve ser realizado por um grupo de no mínimo 3 e no máximo 4 alunos.

Observação 2: Pode ser utilizada qualquer linguagem de programação (Java, Python, etc). Fica vedado o uso de softwares e pacotes utilitários que já contenham algoritmos de aprendizado de máquina implementados (por exemplo, Scikit learn e WEKA).

Observação 3: Plágios, mesmo que parciais, serão punidos com nota zero para os copiadores e copiados envolvidos. O mesmo acontecerá caso seja detectado plágio com arquivos disponíveis na internet.

=====

A. Objetivo geral: Esse trabalho tem como objetivo geral a implementação do algoritmo de aprendizagem Naive Bayes para realizar uma análise de sentimento de *tweets*, de modo a classificá-los como sendo de “sentimento positivo” ou “sentimento negativo”.

B. Conjunto de dados: O conjunto de dados disponível em (<http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip>) contém 1.578.627 *tweets* rotulados, sendo o rótulo 1 para sentimento positivo e o rótulo 0 para sentimento negativo.

Parte 1: Preparação (pré-processamento) dos dados

- a) Façam o download do arquivo contendo o conjunto de dados.
- b) Removam eventuais dados de cabeçalho e assinatura que existirem nos textos.
- c) Construam o vocabulário para os dados de texto.

- d) Transformem cada texto de acordo com a frequência calculada para cada palavra do vocabulário (ver, como exemplo, a atividade sobre Naive Bayes disponível no e-disciplinas).

Parte 2: Implementação do Classificador Naive Bayes

Implementem o classificador Naive Bayes para resolver o problema de classificação dos tweets, de acordo com as seguintes etapas:

Etapla I: Calculem todos os termos de probabilidade necessários, utilizando um conjunto de treinamento, formado a partir de um dos métodos de amostragem (a ser em definidos nas Partes 3 e 4 deste documento).

Etapla II: Computem os valores de classe estimados para cada texto do conjunto de teste e calculem a acurácia do classificador.

A forma com a qual os experimentos foram conduzidos, além dos resultados e impressões obtidos nessa fase, devem ser descritos no relatório final a ser apresentado. Mais detalhes sobre o conteúdo desse relatório serão expostos no final desta especificação.

Parte 3: Treinamento e Avaliação de Desempenho do Classificador Bayesiano utilizando a Abordagem Holdout.

O programa para a Parte 3 deve ser implementado para execução via linha de comando, tendo como argumentos os nomes dos arquivos de entrada (conjunto de dados) e de saída (resultados obtidos), e então:

- a) Carregar o conjunto de dados pré-processados e o vocabulário.
- b) Dividir, aleatoriamente, os dados em conjunto de treinamento e conjunto de teste, em que o conjunto de treinamento deve ter aproximadamente 2/3 do tamanho do conjunto de dados original.
- c) Executar a Etapa I para os dados de treinamento.
- d) Executar a Etapa II para os dados de teste.

O programa pode ser escrito em Java, C/C++, ou qualquer outra linguagem que gere código executável.

Parte 4: Treinamento e Avaliação de Desempenho do Classificador Bayesiano utilizando a Abordagem Crossvalidation.

O programa para a Parte 4 deve ser implementado para execução via linha de comando, tendo como argumentos os nomes dos arquivos de entrada (conjunto de dados) e de saída (resultados obtidos) e, então:

- a) Carregar o conjunto de dados pré-processados e o vocabulário.
- b) Aleatoriamente, dividir esse conjunto em 10 (dez) partições (folds).
- c) Executar o treinamento (e teste) do classificador utilizando o método Crossvalidation.
- d) Registrar, no arquivo de saída, os erros verdadeiros obtidos pelo classificador para cada uma das 10 etapas do Crossvalidation e a estimativa final para o desempenho do modelo (média e erro padrão).

O programa pode ser escrito em Java, C/C++, ou qualquer outra linguagem que gere código executável.

Parte 5: Treinamento e Avaliação de Desempenho do Classificador Bayesiano utilizando a Abordagem Hodout e dados com *stop words* removidas.

Para essa etapa, devem ser removidas do vocabulário as *stop words* contidas no arquivo disponível em: <https://sites.google.com/site/kevinbouge/stopwords-lists>

O programa para a Parte 5 deve ser implementado para execução via linha de comando, tendo como argumentos os nomes dos arquivos de entrada (conjunto de dados com *stop words* removidas) e de saída (resultados obtidos) e, então:

- a) Carregar o conjunto de dados pré-processados (sem *stop words*) e o vocabulário.
- b) Dividir, aleatoriamente, os dados em conjunto de treinamento e conjunto de teste, em que o conjunto de treinamento deve ter aproximadamente 2/3 do tamanho do conjunto de dados original.
- c) Executar a Etapa I para os dados de treinamento.
- d) Executar a Etapa II para os dados de teste.

Devem ser entregues ao final deste trabalho:

- a) Todos os códigos fontes e executáveis referentes às Partes 3 e 4.
- b) Um relatório que inclua:
 - i) **Para a Parte 1:** a descrição de como os dados foram preparados, ou seja, quais foram os passos dados para a construção do vocabulário.
 - ii) **Para a Parte 2:** a descrição e discussão dos experimentos realizados.

- iii) **Para a Parte 3:** uma discussão sobre o desempenho do classificador, levando em conta o erro obtido pelo modelo para o conjunto de teste e a matriz de confusão calculada.
- iv) **Para a Parte 4:** uma análise que discuta os erros obtidos pela rede para cada uma das 10 etapas do Crossvalidation e a estimativa final para o desempenho do modelo.
- v) **Para a Parte 5:** uma discussão que realize a comparação entre o desempenho obtido pelo algoritmo Naive Bayes para os dados originais e o desempenho obtido para os dados com stop words removidas.
- vi) Suas impressões gerais e conclusões.