INF1771 - INTELIGÊNCIA ARTIFICIAL
TRABALHO 3 – APRENDIZADO DE MÁQUINA

O objetivo do Trabalho 3 é desenvolver um sistema de **analise de sentimento** para o reconhecimento de opiniões positivas e negativas em avaliações de filmes.

Exemplo de avaliação positiva:

"I gave this film my rare 10 stars. When I first began watching it and realized it would not be a film with a strong plot line I almost turned it off. I am very glad I didn't. This is a character driven film, a true story, which revolves mainly around the life of Rachel "Nanny" Crosby, a strong, beautiful (inside and out)Black woman and how she touched the lives of so many in the community of Lackawanna. Highly interesting not only its strong characterizations of Nanny and the people who lived at her boardinghouse, but also it gives us a look at what life and community were like for African Americans in the 1950's, prior to integration, and the good and bad sides of segregation and how it ultimately affected and changed the Black community. In addition to excellent performances by all members of the cast, there is some fine singing and dancing from that era."

Exemplo de avaliação negativa:

"I have not seen many low budget films I must admit, but this is the worst movie ever probably, the main character the old man talked like, he had a lobotomy and lost the power to speak more than one word every 5 seconds, a 5 year old could act better. The story had the most awful plot, and well the army guy had put what he thought was army like and then just went over the top, I only watched it to laugh at how bad it was, and hoped it was leading onto the real movie. I can't believe it was under the 2 night rental thing at blockbusters, instead of a please take this for free and get it out of our sight. I think there was one semi decent actor other than the woman, I think the only thing OK with the budget was the make-up, but they show every important scene of the film in the beginning music bit. Awful simply awful."

O seu sistema deve ser capaz de reconhecer automaticamente avaliações positivas e negativas de acordo com o texto fornecido pelo avaliador.

Para essa tarefa, você tem a disposição um conjunto de 50 mil exemplos de avaliações de filmes extraídos do IMDB (25 mil avaliações positivas e 25 mil avaliações negativas). Para o desenvolvimento do sistema, você deve utilizar um método de **aprendizado de máquina supervisionado**, visto que temos um conjunto rotulado de exemplos para treinamento (avaliações de filmes). Para isso, você deve seguir os seguintes passos:

1) Definir quais serão os atributos que serão usados para descrever os exemplos de treinamento.

2) Criar um programa para extrair os atributos das avaliações de filmes para gerar um conjunto de treinamento e um conjunto de validação. Normalmente gera-se um único conjunto de dados e depois se divide ele em dois conjuntos (treinamento e validação). Cada avaliação de filme será um exemplo de treinamento.

3) Utilizar 4 algoritmos de aprendizado supervisionado, treinando-os com o conjunto de treinamento e depois realizando classificação do conjunto de testes para verificar qual algoritmo apresenta a melhor taxa de reconhecimento. Devem ser utilizados os seguintes algoritmos:

    - Árvores de Decisão
    - K-Nearest Neighbor (KNN)
    - Support Vector Machine (SVM)
    - Rede Neural (usando backpropagation)

    Se a taxa de reconhecimento estiver muito baixa para todos os algoritmos, deve-se retornar para a etapa 1 e selecionar melhor os atributos para descrever os exemplos de treinamento.

Informações Adicionais:

- Durante a etapa de teste dos 4 algoritmos não é necessário implementar todos os classificadores para realizar os experimentos. É permitida a utilização de ferramentas de aprendizado de máquina, como por exemplo, o Weka (http://www.cs.waikato.ac.nz/ml/weka/).

- download do conjunto de dados de avaliações de filmes pode ser feito acessando o seguinte link:

  http://www.inf.puc-rio.br/~abaffa/inf1771/movie_review_dataset.zip

- Ao realizar experimentos no Weka, sempre faça cross-validation com pelo menos 10 folds.
- A implementação dos algoritmos de préprocessamento pode ser feita em qualquer linguagem (C/C++, C#, Java...).

- Você deve decidir quais atributos serão utilizados pelos classificadores. A classificação pode piorar ou melhorar dependendo do conjunto de atributos utilizado.

Dicas:

- modelo *bag-of-words* pode ser uma boa opção para extrair características para descrever os exemplos de treinamento:

http://en.wikipedia.org/wiki/Bag-of-words_model

- Para se obter melhores resultados, é possível combinar o modelo *bag-of-words* com o TF-IDF:

http://en.wikipedia.org/wiki/Tf%E2%80%93idf

- Se as abordagens estatísticas não estiverem dando bons resultados, é possível adotar um modelo semântico. Um possível caminho é a utilização do Stanford Parser:

http://nlp.stanford.edu/software/lex-parser.shtml
http://nlp.stanford.edu:8080/parser/index.jsp (versão online para testes)

- Outra abordagem que auxilia nos resultados é indexar as palavras semelhantes para evitar ruídos na análise. Palavras no singular ou no plural, assim como palavras em gêneros diferentes, simbolizam a mesma semântica. A distância de Mahalanobis, e os algoritmos "Dynamic Time Warp", "Soundex" ou "Distância de Edição" analizam matemáticamente duas series de dados para verificar semelhanças. Um possível algoritmo a ser usado é o "StrikeAMatch" de Catalysoft:

http://www.catalysoft.com/articles/StrikeAMatch.html

Forma de Avaliação:

Será avaliado se todas as etapas do processo foram cumpridas corretamente. A avaliação também será baseada na **apresentação dos resultados** durante a aula.
Essa apresentação deverá conter:

- Descrição da modelagem dos exemplos de treinamento:
- Atributos selecionados para descrever os exemplos;
    - Justificativa para a escolha dos atributos;

    - Estrutura dos exemplos;

- Descrição dos experimentos realizados:
    - Variações na modelagem dos exemplos;
    - Variações no conjunto treinamento e testes;
    - Variação nos parâmetros dos algoritmos;

- Comparação dos algoritmos analisados:
    - Taxa de reconhecimento;
    - Tempo gasto no processo de treinamento;
    - Tempo gasto no processo de classificação de um exemplo desconhecido;

Bônus:

- O trabalho que conseguir a maior taxa de reconhecimento receberá 2.0 pontos extras na nota.

Data de Entrega:
06/12

Forma de Entrega:

Os trabalhos devem ser **apresentados** na aula do dia 06/dez (dezembro) e o código deverá ser disponibilizado em repositório git.

Não serão aceitos trabalhos enviados depois desta data.