Centro Universitário Facens

Engenharia da Computação
Inteligência Artificial

Prof. Dr. Renato Moraes Silva

Avaliação Final (AF)

Instruções gerais

1. Leia atentamente a descrição do trabalho para garantir que você está executando o que foi pedido.
    - O não atendimento de qualquer item descrito neste documento, implicará perda de nota.
2. O trabalho pode ser feito em grupo de no **mínimo dois integrantes** e no **máximo até cinco integrantes**.
    - Somente um integrante do grupo deverá enviar a atividade no Canvas.
    - O grupo poderá ser composto por pessoas de qualquer uma das duas turmas da disciplina de Inteligência Artificial da Engenharia de Computação
3. **Participação na competição**: este trabalho contempla a participação em uma competição no Kaggle envolvendo todos os grupos da disciplina. A participação é obrigatória. Todos os componentes do grupo devem se inscrever na competição e formar uma equipe nela.
4. Coloque células explicativas nos notebooks e comentários no código Python para ajudar seu professor a entender o que foi feito;
5. **Erros de compilação/execução**: irão gerar nota zero no item que o problema ocorrer. Antes de submeter o trabalho, certifique-se que não há erros de código em nenhum notebook ou *script* do Python. Nos notebooks, uma forma de se certificar disso é usar a opção ”Reiniciar Kernel e executar todas as células” do Jupyter ou a opção ”Reiniciar e executar tudo” do Google Colab.
6. **Tentativa de fraude**: cópia da Internet ou entre grupos implicará em nota zero para todos os alunos de todos os grupos envolvidos.
    - Não é permitido usar qualquer informação da base de teste da competição no treinamento dos métodos. Isso será considerado tentativa de fraude e resultará em nota 0.
    - É permitido copiar qualquer trecho dos *notebooks* desenvolvidos ao longo da disciplina e disponibilizados no Canvas. Isso não será considerado fraude.

Descrição do projeto

Neste projeto, será abordado o problema de classificação de notícias. Para isso, você deverá usar a base de dados disponível no link mostrado a seguir: *link da base de dados*.

O problema e a bases de dados estão melhor detalhados no Kaggle, no link a seguir: link da competição.

Protocolo experimental, análise dos métodos e resultados

Seu objetivo será analisar qual combinação de método de aprendizado de máquina e técnica de representação vetorial é a mais adequada para esse problema.

Você deverá testar, no mínimo, os seguintes métodos:

- Naive Bayes multinomial
- Regressão Logística
- Floresta aleatória
- Perceptron multicamadas (MLP, do inglês, *multilayer perceptron*)
- Uma rede neural recorrente unidirecional (*long short-term memory* ou *gated recurrent unit*)
- Uma rede neural recorrente bidirecional (*long short-term memory* ou *gated recurrent unit*)

Nos experimentos com os métodos Naive Bayes multinomial, Regressão Logística e Floresta aleatória, você deverá testar, no mínimo, as seguintes técnicas de representação vetorial:

- Baseadas em *bag-of-words*: *term-frequency* (TF), (term frequency-inverse document frequency) (TF-IDF) e binário;
- *Word embeddings*
  - Treinando com a própria base de dados
  - Usando algum modelo de *embeddings* pré-treinado disponibilizado por algum grupo de pesquisa
  - Você pode usar word2vec ou qualquer outro modelo de embeddings que desejar

Nos experimentos com a rede neural recorrente e o MLP, é obrigatório testar, no mínimo, *word embeddings* treinadas na própria base e pré-treinadas, conforme descrito anteriormente.

Alguns modelos acima não podem ser aplicados diretamente no modelo de representação solicitado. Nesse caso, você deverá fazer alguma transformação nos valores para que a aplicação do método seja possível.

Para todos os modelos, você deverá usar alguma técnica de escolha de hiperparâmetro, como a busca em grade, algoritmos genéticos ou busca aleatória (esse último é o mais rápido). Alguns exemplos de hiperparâmetros que podem ser avaliados por esse método incluem o número de árvores na floresta aleatória, o custo na regressão logística e parâmetros impactantes nas redes neurais como o número de camadas, a taxa de dropout, a função de ativação, normalização, entre outros. O critério de escolha da melhor configuração é a medida **AUC** (*area under the curve*).

Para facilitar sua organização e execução dos experimentos, sugere-se que você crie funções ou classes para cada etapa, facilitando que essas funções sejam chamadas no *pipeline* de experimentos.

Formato de Entrega

Você deverá submeter no Canvas um arquivo .zip contendo:

- Um relatório em PDF de no máximo 10 páginas contendo os itens descritos a seguir.
  - Nome, RA e turma de todos os componentes do grupo.
  - Descrição da metodologia que você aplicou: métodos, técnicas de representação de texto, técnicas de pré-processamento, etc.
  - Uma seção de resultados, apresentando uma tabela obrigatória contendo a medida **macro F1** e a **AUC** (*area under the ROC curve*) de cada um dos métodos solicitados, conforme o exemplo mostrado na Tabela 1.

Tabela 1: Valores das medidas macro F1 e AUC obtidas para cada combinação de método e técnica de representação.

	Representação 1		...		Representação n	
	F1	AUC	...	...	F1	AUC
Método 1	0.00	0.00	...	...	0.00	0.00
Método 2	0.00	0.00	...	...	0.00	0.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Método n	0.00	0.00	...	...	0.00	0.00

  - A tabela deverá ser obtida a partir dos dados de treinamento, já que os dados de teste da competição não fornecem a classe correta.
  - Na tabela acima, se você não executou o método algum método de representação, basta colocar o símbolo "$--$" no lugar do valor.
  - Caso você ache necessário, também é permitido adicionar tabelas extras que ajudem a entender o que foi feito ou os resultados obtidos.
  - Será **cobrada** explicação dos resultados. Tente explicar motivos e razões que ajudem a explicar as diferenças entre os métodos.
  - Não será cobrado nenhum tipo de formatação ou o uso de referências.
- Um notebook para cada um dos métodos que você implementar, com nomes que ajudem a identificá-los, mas começando pela palavra main. Por exemplo, o notebook em que estão os testes com o método *naive* Bayes pode ter o nome main_naiveBayes.ipynb.
  - Também é permitido criar notebooks separados para algumas etapas independentes, tais como a análise de dados.
- Parte da implementação poderá ser feita em scripts Python (.py), que poderão ser importados pelos códigos dos notebooks principais.
- Você não deve enviar a base dentro do arquivo .zip que irá submeter. Espera-se que seu código possua uma variável inicial que contenha o *path* da pasta com os arquivos da competição fornecidos anteriormente e que todo o resto seja executado pelo algoritmo a partir disso: descompactação, importação, tratamento, etc.

O que será avaliado?

A nota do projeto será atribuída à análise dos itens apresentados a seguir.

- Relatório com a metodologia e análise dos resultados.
  - Não precisa escrever introdução e conclusão. Foque apenas na metodologia e resultados.
  - Será verificado se todos os testes estão descritos na metologia e se foram realmente contemplados nos algoritmos.
  - Análise dos resultados: você deve apresentar uma análise dos resultados baseada na tabela das medidas macro F1 e AUC. Tente apresentar explicações para valores interessantes ou inesperados e porque um determinado método obteve resultados melhores e piores do que outro quando testado com uma ou mais técnicas de representação. Uma dica que pode ajudar a encontrar explicações é análisar os documentos classificados errados e relacionar às medidas obtidas com base nesses erros.
  - Na análise dos resultadosé é esperado também que você explique quais motivos o levaram a escolher os dois resultados finais da competição.
  - Será analisada a escrita: erros ortográficos, coesão e clareza do texto.
- Análise exploratória para melhor entendimento dos dados;
  - Estatísticas da base de dados, nuvem de palavras, apresentação das palavras mais relevantes obtidas com a técnica *Information Gain*, visualização das embeddings usando t-SNE, etc.
- Aplicação correta dos conceitos de aprendizado de máquina;
- Avaliação de diferentes técnicas de pré-processamento relevantes para o problema;
  - Técnicas de processamento de linguagem natural e outros relevantes para problemas de aprendizado de máquina como balanceamento das classes, transformação de atributos, etc.
- Ajuste de parâmetros e correção de problemas de *overfitting* ou *underfitting*;
- Análise de desempenho individual por meio das principais métricas, como F1 e AUC.

Composição da nota

A nota final do projeto será calculada por:

$$AF = (I + C)$$

sendo que:

- I corresponde a nota na implementação (0.0 a 10.0);

- C corresponde a nota adicional dependendo da posição (Pos) final na competição (no *Private LeaderBoard*) entre as n equipes. Ela será calculada por:

$$C = \begin{cases} 10.0, & \text{se Pos = 1} \\ 10.0, & \text{se Pos = 2} \\ 2.0, & \text{se Pos = 3} \\ -1.0, & \text{se Pos = n} \\ 0, & \text{se 3<Pos<n} \end{cases}$$

- Caso a sua nota fica acima de 10, ela será limitada em 10. Caso sua nota fique abaixo de 0, ela será limitada em 0.