Universidade Federal do Rio Grande do Sul
Instituto de Informática
Departamento de Informática Aplicada

INF01017 - Aprendizado de Máquina
Prof. Anderson Rocha Tavares

Trabalho 1
Aprendizado Supervisionado

1. Objetivo

O trabalho o objetivo é permitir que os alunos explorem diferentes algoritmos de aprendizado supervisionado para uma tarefa de classificação de interesse próprio, validando os modelos treinados com uma estratégia de **k-fold cross validation** e métricas apropriadas de avaliação de modelos.

O trabalho será desenvolvido em grupos de 3 a 5 alunos. Cada grupo deve:

- Escolher um conjunto de dados (dataset) e uma pergunta de pesquisa associada. Como sugestão, alguns repositórios que podem ser buscados: PMLB[1], Kaggle[2] e UCI ML Repository[3]. Observe o número de instâncias por classe, especialmente a classe minoritária, e evite conjuntos com menos de 10 instâncias em alguma das classes;
- Avaliar a necessidade de pré-processar os dados, como, por exemplo, normalizar atributos, tratar valores faltantes, etc. A necessidade pode ser decorrente das características originais dos dados ou por especificidades dos algoritmos escolhidos;
- Selecionar **ao menos três** algoritmos de aprendizado supervisionado cobertos pelo conteúdo programático da disciplina para treinamento de modelos preditivos focados na pergunta de interesse. Os grupos podem optar por outros algoritmos que não sejam vistos na disciplina em adição a estes. Os algoritmos de aprendizado não precisam ser implementados pelos grupos: podem ser utilizadas bibliotecas de aprendizado de máquina que os implementem;
- Comparar o desempenho dos algoritmos escolhidos. Para isso, o grupo deve usar a estratégia de k-fold cross validation, com diferentes valores de k, gerar matriz(es) de confusão e calcular métricas de desempenho. Os grupos deverão reportar, no mínimo, desempenho médio e desvio padrão obtido através do processo de validação cruzada, discutindo os resultados de forma comparativa entre os algoritmos de aprendizado usados. O uso de gráficos como boxplot para avaliar distribuição de desempenho (e outros selecionados a critério do grupo) é desejável e será valorizado na avaliação do trabalho;
- Desenvolver um relatório sobre o trabalho desenvolvido, no qual deverão constar, no mínimo, as seguintes informações:
    - Uma definição clara do objetivo do trabalho desenvolvido, ou seja, dos dados utilizados para treinamento dos modelos e da pergunta de pesquisa que o grupo visou

[1] https://epistasislab.github.io/pmlb/
[2] https://www.kaggle.com/
[3] https://archive.ics.uci.edu/datasets

investigar com algoritmos de AM. Por exemplo: "*Nosso trabalho busca desenvolver um modelo de aprendizado de máquina para prever o tipo de vinho (tinto ou branco) a partir de um conjunto de características físico-quimicas. Para tanto, utilizamos o conjunto de dados <breve descrição> disponível em <link para download>.*";

- uma explicação sobre os dados utilizados, incluindo características como número de instâncias, número de atributos, número de classes, número de instâncias por classe, tipo dos atributos usados (numéricos ou categóricos), se existem valores faltantes, distribuição dos valores dos atributos numéricos, etc;
- uma menção à linguagem de programação e outros recursos computacionais utilizados no desenvolvimento do trabalho, incluindo pacotes e bibliotecas. O grupo pode desenvolver o código em ambiente Colab, desde que fique fácil de executar e observar os resultados; ou pode desenvolver o código para ser executado localmente. Em qualquer caso, o grupo deve fornecer instruções claras para garantir que as dependências necessárias possam ser instaladas facilmente e o código desenvolvido possa ser executado de maneira simples (e.g. que a execução de um algoritmo seja feita com 1 clique ou um comando em terminal);
- uma descrição da metodologia usada para desenvolvimento dos modelos preditivos, incluindo a etapa de pré-processamento dos dados, os algoritmos escolhidos, os valores de hiperparâmetros utilizados no treinamento destes algoritmos (e eventuais estratégias adotadas para otimizar estes hiperparâmetros), além de outras técnicas que podem ser aplicadas de forma opcional a fim de aprimorar o desenvolvimento dos modelos, como redução de dimensionalidade e balanceamento de dados;
- uma descrição e análise dos resultados obtidos. Por análise, entende-se que além de utilizar recursos visuais como gráficos e tabelas para relatar os resultados alcançados com os algoritmos utilizados, os grupos também irão discutir estes achados, discorrendo sobre aspectos como: todos os algoritmos conseguiram resolver a tarefa de classificação abordada? com base nas métricas de desempenho aplicadas, o desempenho dos algoritmos parece satisfatório? como o desempenho variou para estes algoritmos no processo de validação cruzada? algum algoritmo se saiu particularmente melhor, seja por alcançar desempenhos mais altos ou mais consistentes? os algoritmos conseguiram aprender bem para todas as classes do problema, ou demonstraram maior dificuldade com alguma classe ou algum tipo de erro (como falsos positivos ou falsos negativos)? A fim de determinar o algoritmo com desempenho mais satisfatório, recomenda-se que os alunos escolham uma métrica alvo a ser otimizada, com base na pergunta de pesquisa abordada. Esta escolha deve ser apresentada no relatório e consistente com o dataset em questão (e.g. se há dados desbalanceados);
- uma conclusão sobre o trabalho desenvolvido e os resultados alcançados, bem como uma breve discussão de eventuais limitações do trabalho e dos modelos treinados.

Os grupos podem se basear em material visto na disciplina ou consultado em outras fontes para elaborar a metodologia de treinamento dos modelos. Entretanto, a implementação do pipeline de treinamento dos modelos deve ser desenvolvida pelos grupos. Ou seja, não podem ser usados códigos prontos disponibilizados em repositórios como Kaggle ou outros. Se houver quaisquer dúvidas sobre a possibilidade de uso de algum recurso computacional ou software, por favor consultem antes de fazê-lo.

2. Entrega

Os grupos deverão enviar um arquivo .zip pelo moodle, contendo:

- O relatório em PDF;
- O código fonte em um subdiretório 'src' dentro do arquivo .zip. O código fonte pode ser implementado em qualquer linguagem de programação de preferência dos alunos, mas deverá ser enviado com instruções simples e precisas de como compilar (se necessário), e rodar o código (um arquivo Readme.md, por exemplo). Caso o código tenha sido feito no Colab, baixe o .ipynb e envie no .zip. Não é válido o envio da URL do Colab.

Importante: caso a implementação seja no Colab, a simples exportação/impressão de um notebook como PDF não será aceito como relatório. O relatório deve ser um documento devidamente estruturado contendo as informações solicitadas acima.

A estrutura do arquivo `.zip` a ser entregue deve ser a seguinte:

```
relatorio.pdf
src           <-- subdiretorio contendo seu codigo
|-- Readme.md <-- com as instrucoes
\-- [arquivos e subdiretorios de sua implementacao]
```

3. Critérios de correção

- Completude do trabalho e atendimento aos requisitos definidos neste enunciado (ver Seção 1);
- Metodologia: avaliação da seleção de hiperparâmetros, o k-fold cross-validation, se a matriz de confusão foi construída corretamente, se as métricas escolhidas estão apropriadas e se as conclusões relatadas são consistentes;
- Corretude da implementação: se o código executado atinge os resultados reportados e a facilidade de especificar a execução de um algoritmo com um k customizado na validação cruzada;
- Tempo de resposta: não iremos avaliar o tempo de uma eventual otimização de hiperparâmetros, pois ela envolve muitas execuções. Mas haverá penalização caso a execução de 1 fold de uma parametrização específica demore mais que 5 minutos;
- Qualidade do relatório final, em termos da estrutura, da escrita e da discussão dos itens pedidos no enunciado.

Observações gerais

- O trabalho deve ser feito em grupos.
- O tempo de execução é estipulado tendo como referência uma máquina GNU/Linux com a configuração: processador Core i5 2.93 Ghz e 4Gb de memória RAM.
- Fiquem atentos à política de plágio!

Política de Plágio

O uso de assistentes de IA para programar é permitido, mas os grupos não podem copiar uns aos

outros. Os grupos poderão apenas discutir questões de alto nível relativas a resolução do problema em questão. Poderão discutir, por exemplo, questões sobre os algoritmos utilizados, o ajuste dehiperparâmetros, etc.

Usamos rotineiramente um sistema anti-plágio que compara o código-fonte desenvolvido pelos grupos uns com os outros e com soluções enviadas em edições passadas da disciplina.

Qualquer nível de plágio entre grupos poderá resultar em nota zero no trabalho. Todos os envolvidos (não apenas os que copiaram) serão penalizados. Esta política de avaliação não é aberta a debate. Se você tiver quaisquer dúvidas se uma determinada prática pode ou não, ser considerada plágio, não assuma nada: pergunte ao professor(a) e monitor(es).

Note que, considerando-se os pesos das avaliações desta disciplina (especificados e descritos no Plano de Ensino), nota zero em qualquer um dos trabalhos tem grande impacto na nota, e pode impedir o acesso à prova de recuperação. Ou seja: caso seja detectado plágio, há o risco direto de reprovação.