Atividade 5 - Métodos de Classificação Baseados em Árvore

Prof. Dr. Juliano Henrique Foleis

Descrição da Atividade

Nesta atividade você vai implementar dois sistemas de classificação usando métodos de classificação baseados em árvore. Sua implementação deve ser feita em Python em um caderno no Jupyter.

Nesta atividade vamos trabalhar com a base de dados "Video Games Rating by ESRB". Seu objetivo é fazer um classificador decide a classificação etária de jogos de video game baseando-se em tags como "Alcohol Reference", "Blood", "Suggestive Themes" e "Violence", por exemplo. As classificações etárias são: E (*Everyone*), ET (*Everyone 10+*), T (*Teen*) e M (*Mature*). Visite o link do *kaggle* acima para conhecer o significado de cada *tag*. Note que esta é uma base de dados que usa apenas tags já codificadas com 0 ou 1, onde 0 indica que a *tag* não está associada ao jogo e 1 indica que a *tag* está associada ao jogo. A coluna `title` tem o nome do jogo, que pode ser aproveitado de alguma forma ou simplesmente descartada.

Documente cada um dos passos indicados a seguir no Jupyter:

1. Visualize o espaço formado de características usando todos os atributos ligados às tags. Use PCA para reduzir a dimensionalidade.
2. Avalie o desempenho do classificador `Decision Tree` usando validação cruzada em dois níveis, conforme discutimos nas aulas de otimização de hiperparâmetros. A validação cruzada no primeiro deve ser em 10 vias, enquanto no segundo nível deve ser em 5 vias. A validação cruzada no segundo nível deve selecionar a melhor combinação de hiperparâmetros. Os parâmetros a serem otimizados são os mesmos que estudamos em sala de aula (Tópico 7). Utilize a métrica *weighted f1-score* para avaliar o desempenho do classificador. Imprima os resultados usando a função `classification_report`. Imprima a soma das matrizes de confusão dos folds com a melhor combinação de parâmetros.

Dica: no primeiro nível você deve usar StratifiedKFold para gerar os particionamentos, e no segundo nível você deve usar `GridSearchCV`. **Dica:** use o parâmetro `scoring` no construtor do GridSearchCV para escolher a métrica de desempenho.

3. Usando a melhor combinação de parâmetros obtida acima, treine a árvore usando todos os dados do dataset. Em seguida, mostre a árvore (usando a função `plot_tree`), e depois escreva a estrutura de regras de decisão induzidas pelo algoritmo. Note que as regras de decisão podem ser estruturadas na forma de *if*'s aninhados.
4. Avalie o desempenho do classificador `Random Forest` usando validação cruzada em dois níveis, da mesma forma que no item *2*. Os parâmetros a serem otimizados são os mesmos que estudamos em sala de aula (Tópico 8).
5. Avalie o desempenho do classificador `KNN` usando validação cruzada em dois níveis, da mesma forma que no item *2*. A validação cruzada no segundo nível deve selecionar o melhor `k`. Use a métrica de distância `euclidean`.
6. Avalie o desempenho do classificador `SVM` usando validação cruzada em dois níveis, da mesma forma que no item *2*. A validação cruzada no segundo nível deve selecionar a melhor combinação de C e *gamma* (γ) de acordo com o que vimos na aula sobre SVM. Use o kernel `rbf`.
7. Faça o teste da hipótese nula (pelo Teste-T) para verificar se a diferença entre o melhor e o pior resultado obtido (entre os melhores resultados com Decision Tree, Random Forest, KNN e SVM) é

estatisticamente significativa com 95% de confiança. Interprete o resultado do teste.

Em vários dos passos acima existem muitas decisões que podem ser tomadas que afetam o desempenho dos classificadores. Justifique suas escolhas. Experimente variações e tente desenvolver um sistema que acerte o máximo possível!

Instruções e Entrega

- A maioria dos passos acima estão prontos nos cadernos das Semanas 4–7 disponibilizados no GitHub.
- Capriche no seu *notebook*: coloque textos explicativos, faça gráficos que julgar necessário, etc. Aproveite para aprender como usar as ferramentas!
- A atividade deve ser feita em um Jupyter Notebook. Você pode usar o *Google Colab* se quiser, mas é necessário entregar o arquivo *.ipynb*. Caso coloque código em arquivos *.py* por favor entregar junto com o *.ipynb* em um arquivo *.zip*.
- A entrega deverá ser realizada via Moodle, na *Atividade 5*.
- **Prazo para entrega:** 21/6/2022 às 23:55.
- O trabalho é individual.
- Não é permitido alterar o arquivo que contém a base de dados (**Video_games_esrb_rating.csv**)!

BONS ESTUDOS!