Atividade 2 - Atributos Categóricos e Valores Faltantes

Prof. Dr. Juliano Henrique Foleis

Descrição da Atividade

Nesta atividade você vai implementar dois sistemas de classificação usando uma base de dados que contém atributos categóricos e valores faltantes. Sua implementação deve ser feita em Python em um caderno no Jupyter.

Nesta atividade vamos trabalhar com um subconjunto da base de dados "Mushrooms". Esta base de dados é famosa por ter apenas atributos categóricos. Cada instância descreve uma amostra de cogumelo. Cada amostra possui 22 atributos. Seu objetivo é construir um classificador que seja capaz de reconhecer dado cogumelo como comestível (*edible*) ou venenoso (*poisonous*).

Documente cada um dos passos indicados a seguir no Jupyter:

1. Codifique o atributo de saída (*class*) da seguinte forma: $e \rightarrow 0$ e $p \rightarrow 1$. Isto será útil para o cálculo da métrica *f1_score*, mais adiante.
2. Realize a imputação para os valores faltantes. Os valores faltantes estão em apenas na coluna "stalk-root". A estratégia de imputação fica a seu critério. Também há a possibilidade de excluir esta coluna.
3. Faça a codificação dos atributos categóricos. O arquivo `agaricus-lepiota.names` explica a significado e os valores relativos a cada atributo da base de dados. De acordo com o significado e os valores de cada atributo decida qual é o codificador mais adequado.
4. Avalie o desempenho do classificador KNN usando validação cruzada em dois níveis, conforme discutimos na Semana 4. A validação cruzada no primeiro deve ser em 10 vias, enquanto no segundo nível deve ser em 5 vias. **Dica:** no primeiro nível você deve usar StratifiedKFold para gerar os particionamentos, e no segundo nível você deve usar `GridSearchCV`. A validação cruzada no segundo nível deve selecionar o melhor k. Utilize a métrica *f1-score* da classe positiva (*poisonous*) para avaliar o desempenho do classificador em ambos níveis. **Dica 1:** use o parâmetro `scoring` no construtor do GridSearchCV para escolher a métrica de desempenho. **Dica 2:** a função `f1_score` do módulo `sklearn.metrics` calcula o *f1_score* e os parâmetros são os mesmos que usamos com `accuracy_score`.
5. Avalie o desempenho do classificador SVM usando validação cruzada em dois níveis, da mesma forma que no item *3*. A validação cruzada no segundo nível deve selecionar a melhor combinação de C e *gamma* (γ) de acordo com o que vimos na aula síncrona. Use o kernel `rbf`.
6. Faça o teste da hipótese nula (pelo Teste-T) para verificar se os resultados obtidos com o KNN e com a SVM são estatisticamente diferentes com 95% de confiança. Interprete o resultado do teste.
7. Você usaria algum classificador que criou para decidir se comeria ou não um cogumelo classificado por ele? Justifique usando o desempenho obtido e o resultado do teste de hipótese.

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 e 5 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*.
- A entrega deverá ser realizada via Moodle, na *Atividade da Semana 5*.
- **Prazo para entrega:** 23/08/2021 às 23:55.
- O trabalho é individual.
- Não é permitido alterar o arquivo que contém a base de dados (**agaricus_lepiota_small_c.csv**)!

BONS ESTUDOS!