Disciplina: Aprendizagem de Máquina
Período: 2023.1
Professor: César Lincoln Cavalcante Mattos

Lista 5 - SVM e comitês de modelos

Instruções

- Com exceção dos casos explicitamente indicados, os algoritmos e modelos devem ser implementados do início em qualquer linguagem de programação (Python, R, Octave...).
- Pacotes auxiliares (sklearn, matplotlib, etc) podem ser usados somente para facilitar a manipulação dos dados e criar gráficos.
- A entrega da solução pode ser feita via pdf ou Jupyter notebook pelo SIGAA.

Questão 1

Considere o conjunto de dados disponível em **californiabin.csv**, organizado em 9 colunas, sendo as 8 primeiras colunas os atributos e a última coluna a saída. Os 8 atributos são usados na predição de preços de casas em distritos da Califórnia na década de 1990. A saída é binária: 0, para abaixo da mediana dos preços; 1, para acima da mediana dos preços. Maiores detalhes sobre os dados podem ser conferidos em `https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset`.

a) Considerando uma divisão de 80% dos padrões para treinamento e 20% para teste, avalie modelos de classificação binária nos dados em questão. Para tanto, use as abordagens abaixo:

   - **SVM**: Escolha um *kernel* RBF e use *grid-search* para ajustar os hiperparâmetros C (valores $2^{-5}, 2^{-3}, 2^{-1}, \ldots, 2^{11}, 2^{13}, 2^{15}$) e γ (valores $2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^{1}, 2^{2}, 2^{3}$).
   - **Random Forest**: Use *grid-search* para ajustar o número de classificadores base (valores 10, 20, ..., 180, 190, 200) e a máxima profundidade (*max depth*, valores 4, 6, 8, 10 ou máxima (None no sklearn)).

b) Para cada modelo campeão, reporte os hiperparâmetros selecionados e as métricas de **acurácia**, **revocação**, **precisão** e **F1-score** nos dados de teste. Plote também a **curva ROC** e a **curva Precision-Recall**.

Observações:

→ Use validações cruzadas em 10 *folds* no interior do *grid-search*.

→ Não esqueça de retreinar o modelo final com os hiperparâmetros otimizados usando tanto os dados de treino quanto de validação.

→ Você pode usar implementações já existentes dos modelos acima, como do sklearn.