Sistemas de Informação Multimedia (SIM) – 2019/2020

Trabalho prático

Notas introdutórias:

Este trabalho prático deverá ser realizado por grupos de dois estudantes, ou opcionalmente de forma individual.

O enunciado será publicado em dois momentos, o primeiro no dia 7 de Janeiro de 2020 e o segundo no final do dia 10 de Janeiro depois da aula teórica (no tema de regressão logística).

A maior parte do trabalho será feito nas aulas práticas (15 pontos), com uma pequena componente extra de investigação adicional (5 pontos) a qual será anunciada no segundo enunciado.

Data limite de entrega: 24 de Janeiro de 2020 até as 17h00. Submissão de um único ficheiro Jupyter via Moodle, não será aceite nenhum outro formato.

Este trabalho tem um peso de 40% na avaliação da componente prática da cadeira, sendo que os outros 60% correspondem aos três quizzes práticos.

A implementação deste projeto deve ser diretamente aplicável a outros datasets de qualquer dimensão. Portanto, não implementar código que fique dependente das características deste dataset em particular.

Trabalho a ser realizado (primeiro momento):

1. Descarregar o dataset wine_quality (*wine_q.csv*) disponível no *Moodle*.
2. Usar a Biblioteca *Pandas* para carregar o *dataset* num *Pandas dataframe* em *Jupyter*.
3. Remover possíveis filas duplicadas do *dataframe* (automaticamente, via código).
4. Implementar código para obter os histogramas das diferentes variáveis, e determinar por inspeção visual, quais delas distribuições claramente não são normais (explicar no caderno Jupyter usando células *markdown*).
5. Usar o teste de *Jarque-Bera* em Python para determinar quais distribuições passam o teste de normalidade e quais não. Nota: voltar a descarregar as notas PDF do módulo teórico sobre PCA, já que existe uma pequena correção na secção que introduz o uso de Jarque-Bera em Python. Devem implementar código iterativo sobre as diferentes variáveis (não fazer variável a variável manualmente).
6. Calcular e visualizar a matriz de correlação do *dataset*.
7. Analisar os resultados dos pontos (5) e (6) e explicar porque aplicaria, ou não, PCA para analisar estes dados.
8. Separar (via código aplicado ao *dataframe*) o dataset em variáveis independentes (as onze primeiras) e variável dependente (quality). O resultado deve ser dois objetos *Numpy.*
9. Aplicar PCA nos dados das variáveis independentes, tomando em conta o procedimento inteiro, que inclui o pre-processamento dos dados. É recomendado usar o *StandardScaler* de *sklearn.* Explicar em markdown se o resultado obtido é consistente, ou não, com a recomendação feita em (7).
10. Implementar código que use funções disponíveis em *sklearn* para separar o dataset em *training* e *testing sub-datasets* que possam ser usados por algoritmos de *Machine Learning* para encontrar um classificador binário (o que será o foco da segunda parte do projeto).