Manual de Utilizador

Projeto de Reconhecimento de Padrões

Rain Prediction@Australia

Este manual foi escrito para que permita ao leitor uma replicação da experiência (completamente descrita no relatório) e também a reprodução e verificação dos resultados lá apresentados.

A experiência foi realizada com o propósito de previsão de chuva para o dia seguinte em 44 locais Australianos.

Com este manual, queremos transmitir essencialmente como executar o programa desenvolvido.

Dependências

O código - desenvolvido em Python 3 - deve estar inserido num ambiente que contenha as seguintes livrarias:

- tksklearn
- sklearn
- pandas
- numpy
- matplotlib

main_interface.py

Este é o ficheiro principal do projeto. Contém todo o código necessário para a inicialização da interface gráfica do projeto de modo a ser possível escolher várias configurações, treinar os classificadores, e por fim, visualizar os resultados.

É ainda possível, clicando no botão “Give me the stats”, ter acesso a uma visualização (histogramas e boxplots) e avaliação da normalidade das features.

Ao ser executado com o comando python main_interface.py, irá aparecer uma janela com a GUI. Neste ponto, é possível que o utilizador escolha várias opções, tais como o tipo de *scaler,* o tipo de seleção de *features,* o tipo de redução de dimensionalidade, e por fim o classificador a ser utilizado. É também possível uma escolha da localização onde o utilizador quer que a previsão de chuva seja feita.

Depois de o utilizador ter escolhido as configurações desejadas, deverá clicar em *Run!* e esperar que os resultados sejam exibidos na tela.

data_load.py

Este ficheiro tem como função a leitura e tratamento dos dados. Nele os dados do ficheiro weatherAUS.csv são carregados para um *data frame* utilizando a biblioteca *pandas*.

Na parte de tratamento de dados, são eliminadas colunas não necessárias ao problema e são também convertidas Strings “yes” e “no” para inteiros 0 e 1.

No final, é retornado um data frame com todas as *features* necessárias e com o *target.*

scaler.py

Este ficheiro é composto por duas funções relacionadas ao *scaler*, o min_max_scaler, e o standard_scaler. Ambas as funções tem como argumento um array com as *features* e como objectivo fazer o scaling dos dados.

No final, é retornado um array com as *features* devidamente modificadas.

feature_selection.py

Este ficheiro é dedicado às funções de seleção de *features.* Nele encontram-se quatro funções responsáveis pela seleção de *features* utilizando métodos/testes estatísticos como, f_classification, chi_squared, kruskal e pearson_correlation (devidamente explicados no Relatório).

Todas as funções têm como argumentos as *features* e o *target* do problema e retornam um *array* com as *features* selecionadas.

dimensionality_reduction.py

Este ficheiro contém duas funções responsáveis pela redução de Dimensionalidade baseadas em PCA e LDA. Ambas as funções têm com argumento um array das features, porém a função de LDA necessita também de um array com o target.

Por fim, é retornado um array com as features devidamente reduzidas.

classification.py

Este ficheiro contém sete classificadores. Entre eles encontram-se classificadores como Linear Discriminant Analysis, Nearest Centroid (euclidean MDC), Random Forest, Decision Tree, Gaussian Naive Bayes, K-Nearest Neighbors e por último, Support Vector Machine Classifier.

Todos eles necessitam de um *array* das *features* e um *array* com o *target* e retornam arrays com métricas. Destes arrays, decidimos utilizar os arrays de accuracy e roc auc curve. Para isso, extraímos a média de accuracy, a best accuracy overall, desvio padrão da accuracy, média de roc auc, o best roc auc overall e o desvio padrão do roc auc.