Atividade de lab - Regressão Logsítica

Daniel Santana e Isaque Fernando

Atividade 1: Analisar as Questions 1, 2, 3, 4

Question 1

A Question 1 do arquivo `.ipynb` visa fazer uma espécie de pré-processamento dos dados, através de uma análise inicial dos mesmos.

Inicialmente, esta questão nos incentiva a analisar os dados existentes no arquivo de dados, uma vez que este possui um número muito grande de colunas. Através da função `value_counts`, pudemos observar que, apesar de ser um dataset bem extenso, possui apenas floats como features, e uma coluna para a label.

O segundo passo desta questão é levar o aluno a perceber a necessidade (ou não) de normalizar os valores das features. Como todas já tinham valores entre -1 e 1, não foi necessária a normalização.

O terceiro passo visa remover quaisquer *bias*, verificando se a quantidade de atividades está balanceada, ou seja, se temos a mesma quantidade de dados de entrada representando cada atividade. Com esta análise, é notável que os dados estão de certa forma balanceados, já que a menor quantidade de dados que temos é de 1406 para a atividade `WALKING_DOWNSTAIRS`, e a maior quantidade é 1944, para a atividade `LAYING`.

Como último passo deste pré-processamento, temos de substituir o `object` representado pelas labels por números inteiros. Para isto, usamos o módulo `LabelEncoder` do `Sklearn`, que verifica a quantidade de labels diferentes que o dataset possui, e atribui um número inteiro para cada uma. Neste passo, alteramos a coluna `Activit` do dataset, de forma a substituir a *string* da label pelo número inteiro retornado pelo método `fit_transform` do `LabelEncoder`.

Question 2

A Question 2 tem por objetivo correlacionar as muitas features do dataset, criando um histograma de correlação para identificar as variáveis que tem relação positiva ou negativa umas com as outras, dada a sua label.

A matriz de correlação gerada pela função `corr()` é simplificada, obtendo-se os índices da diagonal principal e tornando todos os valores abaixo dela `None`. Isto pode ser feito, pois, abaixo da diagonal principal, basicamente tem-se apenas valores repetidos de correlação, que já foram listados acima.

Após a simplificação descrita acima, a matriz é convertida em um `DataFrame`, e as suas colunas são renomeadas para terem nomes mais intuitivos. Então, o histograma de correlações é plotado, e a matriz é ordenada para exibir as variáveis mais relacionadas (cuja correlação absoluta é maior que 0.8).

Question 3

Esta Question tem por objetivo separar o dataset em dados de treino e de teste. Para isto, usamos o módulo `StratifiedShuffleSplit` do `ScikitLearn`, de forma a manter a frequência de classes preditas. Após a separação, conferimos as frequências antes de seguir para a regressão. A taxa usada foi de

70% treino e 30% teste para o dataset. Neste passo, definimos as variáveis `X_train` e `Y_train` que indicam as features e labels de treino, respetivamente, e `X_test` e `Y_test`, que representam as features e labels de teste respetivamente.

Question 4

Finalmente, usamos a regressão logística para prever valores de Activity, dados as features que são *inputs* do celular dos usuários. Por padrão, usamos os parâmetros `penalty='l2', C=10.0`, como mostrado nos slides da disciplina. Estes parâmetros serão explicados em aulas posteriores. Houve também a necessidade de se aumentar a quantidade de iterações, uma vez que o algoritmo demorou um tempo considerável para convergir. Este tempo provavelmente se deve ao tamanho do dataset, uma vez que os dados estavam normalizados, balanceados e pré-processados corretamente, conforme nos garantimos nas Questions anteriores. Desta forma, aumentando o parâmetro `max_iter` ao inicializar o modelo, obtivemos sucesso na convergência do algoritmo.

Question 5

Após todos estes passos, basta agora avaliar o modelo treinado, calculando a sua acurácia. Visto que a acurácia é a quantidade de classificações corretas que um modelo classificou, iremos comparar a variável `y_predict` com o `y_test` obtendo assim a acurácia do modelo. Segundo o cálculo da acurácia obtivemos um total de 98% de acurácia com o modelo. Esse valor mostra uma boa acertividade do modelo utilizado. Porém como temos até o momento apenas essa métrica, não sabemos ainda como este mesmo modelo se comportará com a generalização se mostrando assim um modelo robusto.