

ScikitLearn 操作記錄單(ScikitLearn pre-Processing)

組別: _team15_____ 學號: _41147046S_____ 姓名: _楊子萱_____

1. 請根據以下教學資源操作: <http://www.cse.msu.edu/~ptan/dmbook/tutorials/tutorial4/tutorial4.html>
若對 python 及 pandas 不熟悉, 請從 <http://www.cse.msu.edu/~ptan/dmbook/software/> 中學習更前面的教學內容。
2. 請自行查詢了解下列 scikit-learn 模組的功能作用 <https://scikit-learn.org/stable/>

實作功能: 除最後 4 個外皆有實作, 若遇到選用的資料集不太適合實作此功能時則會自定義一個小的資料

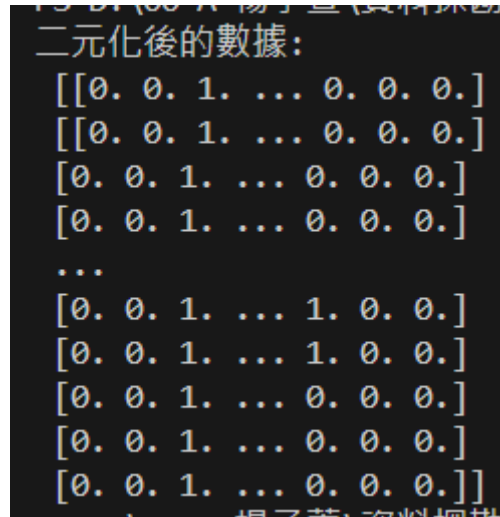
採用資料集: <https://www.kaggle.com/datasets/harish24/music-genre-classification/data?select=dataset.csv>

程式碼連結: https://docs.google.com/document/d/10K6qggJ36yInPYby4mzAwieGRN_bMPWUopK2C664Rh4/edit?usp=sharing

連結點入為 google document 且功能皆已被註解, 若要執行程式, 需複製貼上到.py, 並且把被註解掉的功能解開執行

Pre-processing	Module	Function	試寫程式, 實驗該函式所提供功能及主要參數設定效果並簡要敘述
Missing data	sklearn.preprocessing	Imputer()	<pre>imputer = SimpleImputer(strategy='mean') df[columns_to_impute] = imputer.fit_transform(df[columns_to_impute])</pre> <p>功能: 把資料集中缺少的值依"給定條件"補齊</p> <p>參數設定效果:</p> <p>可指定要補齊的列(輸入列的名稱)</p> <p>給定條件有整列的平均值、中位數、重數等, 也可直接填 0</p> <p>簡要敘述: 返回填補缺失值後的數據</p>
Sampling	sklearn.model_selection	train_test_split()	<p>train_test_split():</p> <pre>train_test_split(*arrays, test_size=None, train_size=None,</pre>

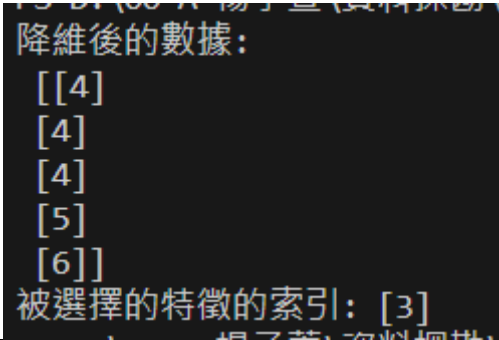
	sklearn.utils.random	sample_without_replacement()	<p>random_state=None, shuffle=True, stratify=None)</p> <p>功能: 將數據集分割為訓練集和測試集。這樣的分割使得我們能夠在訓練模型時使用訓練集進行訓練，然後用測試集來評估模型的性能</p> <p>參數設定效果:</p> <p>*arrays: 需要拆分的數據，通常包括特徵變量和標籤變量。</p> <p>test_size: 測試集的比例或數量。如果是小數，表示分配比例，默認值為 0.25。</p> <p>train_size: 訓練集的比例或數量，通常與 test_size 二選一設置。</p> <p>random_state: 隨機種子，保證每次分割出的訓練集和測試集都是一樣的，方便比較實驗結果</p> <p>shuffle: 是否在分割前打亂數據，默認為 True。</p> <p>stratify: 幫助確保分割後的訓練集和測試集在目標變量（標籤）的類別分佈上與原始數據集保持一致，避免不平衡數據集影響預測</p> <p>簡要敘述: 返回四個數組：訓練特徵集、測試特徵集、訓練標籤集、測試標籤集</p> <p>sample_without_replacement():</p> <p>sklearn.utils.random.sample_without_replacement(n_population, n_samples, random_state=None)</p> <p>功能: 用於從一個範圍或數據集中不放回地隨機抽取樣本</p> <p>參數設定效果:</p> <p>n_population: 總樣本空間的大小，也就是總共可以選擇的元素數量。</p>
--	----------------------	------------------------------	--

			<p>n_samples: 希望隨機抽取的樣本數量。</p> <p>random_state: 隨機種子，用於確保每次運行結果一致（類似於 train_test_split 中的 random_state，可選）</p> <p>簡要敘述: 返回一個隨機選取的樣本索引的數組</p>
Binarize	sklearn.preprocessing	Binarizer()	<p>Binarizer():</p> <p>Binarizer(threshold=0.0)</p> <p>功能: 將數值特徵轉換為二進制格式（0 和 1）</p> <p>參數設定效果:</p> <p>threshold: 一個浮點數，表示閾值。小於或等於該閾值的值將被轉換為 0，大於該閾值的值將被轉換為 1。預設值為 0.0</p> <p>簡要敘述: 返回一個二元化後的數組，原始數據中小於或等於閾值的元素將變為 0，大於閾值的元素將變為 1</p> <p>結果顯示:</p>  <p>OneHotEncoder():</p>
	sklearn.preprocessing	OneHotEncoder()	

			<p>OneHotEncoder(sparse=True, dtype=np.float64, handle_unknown='error')</p> <p>功能: 將類別特徵轉換為一組二元 (0 和 1) 特徵的工具，稱為獨熱編碼 (One-Hot Encoding)</p> <p>參數設定效果:</p> <p>sparse: 如果設置為 True，則返回稀疏矩陣；如果設置為 False，則返回普通的數組 (默認值為 True)。</p> <p>dtype: 設置返回數據的類型 (默認為 np.float64)。</p> <p>handle_unknown: 當遇到未知類別時的處理方式，可以設置為 'error' (默認), 'ignore' 等。</p> <p>簡要敘述: 返回編碼後的數組，其中每一行對應於原始數據中的一個樣本，每一列對應於一個類別特徵</p>
Discretization	sklearn.preprocessing	K-bins discretization()	<p>KBinsDiscretizer(n_bins=5, encode='onehot', strategy='uniform')</p> <p>功能: 可將連續的數據轉換為離散數據</p> <p>參數設定效果:</p> <p>n_bins: 整數或數組，指定要將特徵分為多少個 bin。默認值是 5。</p> <p>encode: 字符串，指定如何編碼 bin。選項有：</p> <ul style="list-style-type: none"> 'onehot': 返回一個獨熱編碼的稀疏矩陣 (默認) 'onehot-dense': 返回一個獨熱編碼的密集矩陣。 'ordinal': 返回每個樣本所屬的 bin 索引。 <p>strategy: 字符串，指定 bin 劃分策略。選項有：</p> <ul style="list-style-type: none"> 'uniform': 每個 bin 的寬度相等。 'quantile': 每個 bin 包含相同數量的樣本。 'kmeans': 基於 K-means 聚類進行分箱

			簡要敘述: 返回離散化後的數據
Standardize	sklearn.preprocessing	StandardScaler() Scale()	StandardScaler(): 功能: 將數據的每個特徵縮放到均值為 0，標準差為 1 的範圍 參數設定效果: 沒有參數需要設置 簡要敘述: 可印出經過處理後的 data 的平均值、標準差等 結果: <pre> 標準化後的數據: file_name chroma_stft rms spectral_centroid spectral_bandwidth rolloff zero_crossing_rate ... mfcc15 mfcc16 mfcc17 mfcc18 mfcc19 mfcc20 label 0 blues_00000.wav -0.351481 0.010723 -0.583303 -0.466892 0.288166 -0.207057 ... 0.265298 -0.082798 0.592837 -0.232318 0.807811 0.481498 blues 1 blues_00001.wav -0.461466 -0.533266 -0.939066 -0.307668 0.688869 -1.138238 ... -0.439656 -0.066294 0.714717 -0.455190 0.543824 0.436835 blues 2 blues_00002.wav -0.184484 0.080832 -0.907419 -0.541088 0.972811 -0.453772 ... 1.007722 0.285748 -0.820553 -0.680787 -0.294285 -0.295113 blues [5 rows x 28 columns] 均值: [3.78668707e-01 1.80928628e-01 2.20184236e+01 2.04225963e+03 8.37728136e+01 1.83017298e-01 1.44479128e+02 9.3923380e+01 -8.92188888e+00 3.62918606e+01 -1.14662706e+00 1.46340819e+01 -5.12962378e+00 1.48118678e+01 -6.99575146e+00 7.78658078e+00 6.40111807e+00 4.47160398e+00 -4.79121353e+00 1.78154580e+00 -3.47027553e+00 1.145788812e+00 -1.96743054e+00 5.473629880e-01 2.32877912e+00 1.40881656e+00] 標準差: [0.16648515e-02 6.50523893e-02 7.15603277e+02 5.268074428e+02 1.57398245e+01 4.18135468e-02 1.08185529e+02 1.13162366e+01 2.16881648e+01 1.66386501e+01 1.22164418e+01 1.18188793e+01 9.93666666e+00 1.05588766e+01 8.28807796e+00 7.93488680e+00 6.81588884e+00 6.71395268e+00 6.16783236e+00 5.40028875e+00 4.87389576e+00 4.57081956e+00 4.54817641e+00 3.86733311e+00] </pre>
Normalize	sklearn.preprocessing	MinMaxScaler()	功能: 通過線性變換將數據的最小值和最大值分別映射到目標範圍的兩端 參數設定效果: feature_range: 指定縮放範圍，默認為 (0, 1) 簡要敘述:

			$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \times (\max - \min) + \min$ <p>其中，X 是輸入數據，X_min 和 X_max 分別是該特徵的最小值和最大值。</p>
Dimension reduction	sklearn.decomposition	PCA()	<p>功能: 用於降維，可將高維數據映射到較低維度空間來簡化數據集</p> <p>參數設定效果:</p> <p>n_components: 指定要保留的主成分數量，可以是整數（要保留的主成分的數量）或介於 0 和 1 之間的浮點數（保留的變異百分比）</p> <p>簡要敘述: 數據的維度將從原來的多維縮減到你指定的維度</p> <p>結果:</p> <p>降維後的數據:</p> <pre>[[1.19522463 -1.17845138] [0.10621898 -1.40719492] [1.64178635 -2.41998842] ... [4.27810006 -2.15438184] [0.57177576 -2.55924837] [1.76445572 -2.29477638]]</pre>
Feature selection	sklearn.feature_selection	VarianceThreshold()	<p>VarianceThreshold(threshold=0.0)</p> <p>功能: 通過去除低變異的特徵來進行特徵選擇</p> <p>參數設定效果:</p> <p>threshold: 設置特徵的變異閾值。默認值為 0.0，表示將刪除變異為 0 的特徵，可以設置為其他值來刪除變</p>

			<p>異低於該值的特徵</p> <p>簡要敘述: 計算每個特徵的變異性後，去除變異低於指定閾值的特徵</p> <p>結果:</p> 
	sklearn.feature_selection	SelectKBest() chi2	
	sklearn.feature_selection	SelectFromModel()	
	sklearn.feature_selection	RFE()	

其他參考資源:

- machine learning 參考書: "[Introduction to Machine Learning with Python](#)"

之 github code https://github.com/amueller/introduction_to_ml_with_python

- Scikit Learn 官方 documentation: <https://scikit-learn.org/stable/>
- 自尋搜尋其他可信網路資源