ScikitLearn 操作記錄單(ScikitLearn pre-Processing)

組別: Team16　　學號: 41071102H　　姓名: 徐敏皓

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/

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()	此函式會將缺失值以"mean", "median"等方式將其填補。 SimpleImputer (missing_values , strategy , fill_value , add_indicator) missing value: 需要被填補的缺失值，預設為 np.nan，即處理 NaN 值。 strategy: 填補策略，包含'mean', 'median', 'most_frequent', or 'constant'。 fill_value: 當 stategy = 'constant'時需要指定的常數。預設為 None。

			add_indicator: 是否添加一個二元指標列，預設為 False。
Sampling	sklearn.model_selection	train_test_split()	將資料集作分割，分割成訓練集和測試集，可設定兩者比例。 train_test_split(x, y, test_size, train_size, random_state, shuffle, stratify) x: 特徵 dataset。 y: 標籤 dataset。 test_size: 用來指定 testing data 的大小，可以用 floating point 表示(0.0~1.0 之間)。 train_size: 用來指定 trainting data 的大小，與 test_size 互補，通常不需同時設置。 random_state: 控制隨機數生成的種子。 shuffle: 是否再分割 data 前打亂 data，預設為 True。 stratify: 用來根據標籤進行分層抽樣，保持訓練集和測
	sklearn.utils.random	sample_without_replacement()	

			試集中類別比例一致，常與分類問題一起使用。 將資料集做取樣，取出的資料中不會有重複選取的問題。 sample_without_replacement(n_population, n_sample, method, random_state) n_population: 母體大小(有多少樣本可選)。 n_sample: 要抽取的樣本數量。 method: 控制採樣方法的選擇。 random_state: 控制隨機性的種子。
Binarize	sklearn.preprocessing	Binarizer()	將數值資料做二元化，設定一個門檻(threshold)，大於這個門檻的值會變成 1，小於等於這個值會變成 0。 Binarizer(threshold, copy) threshold: 預設為 0，大於這個值的 data 會被轉換為 1，否則轉換為 0。
	sklearn.preprocessing	OneHotEncoder()	

			copy: 預設為 True，如果設成 False，在原本 data 上進行操作，不會產生新的 data 副本。 將 attributes 做 one-hot 編碼。 OneHotEncoder(categories, drop, sparse_output, dtype, handle_unknown) categories: 當設成 auto，OneHotEncoder 會自動檢測每個特徵中的類別。 drop: first 或 if_binary，可選擇丟棄某個類別來避免共線性。 sparse_output: 預設為 True，傳回稀疏矩陣。 dtype: 指定返回矩陣的資料型別，預設為 np.float64。 handle_unknown: 預設為 error，當資料中有遇到未見過的類別時，是否丟出錯誤或忽略。

Discretization	sklearn.preprocessing	K-bins discretization()	將 attributes 做區間化，可設定要有多少個區間，每個區間要等寬或等量。 KBinsDiscretizer(n_bins, encode, strategy) n_bins: 指定每個特徵的區間數量。預設為 5。 encode: 決定將結果轉換為何種格式。 strategy: 指定如何劃分 data。
Standardize	sklearn.preprocessing	StandardScaler() Scale()	將 attributes 做標準化。其結果平均值為 0，標準差為 1。 StandardScalar(copy, with_mean, with_std) copy: 預設為 True。決定是否在標準化時進行數據的拷貝。 with_mean: 預設為 True。是否將 data 的平均值調整為 0。 with_std: 預設為 True。是否將 data 的標準差調整為 1。

			將 attributes 做標準化。其結果平均值為 0，標準差為 1。 Scale(x, axis, with_mean, with_std, copy) x: 需要進行標準化的 data axis: 指定沿著哪一個軸標準化，預設為 0。 with_mean: 預設為 True。是否將 data 的平均值調整為 0。 with_std: 預設為 True。是否將 data 的標準差調整為 1。 copy: 預設為 True。是否在輸出中返回 data 的副本。
Normalize	sklearn.preprocessing	MinMaxScaler()	將 attributes 做正規化。其預設結果所有數值範圍會落在 [0, 1]，可自行調整。， MinMaxScaler(feature_range, copy, clip) feature_range: 指定縮放後的數值範圍。預設為[0, 1]， 可自行調整為[min, max]。 copy: 預設為 True，表示輸入的 data 將會被複製。若設 為 False，縮放會直接在原 data 上進行改變。

			clip: 預設為 False，若設為 True，會將 data 限制在指定範圍內，避免縮放後的數值超出範圍。
Dimension reduction	sklearn.decomposition	PCA()	將 attributes 做降維，可設定降維後要保留多少數量的 attributes。 PCA(n_components, copy, whiten, svd_solver, tol, iterated_power, random_state) n_components: 指定希望保留的主成分數量。 copy: 預設為 True，表示在處理 data 時不修改原始 data。 whiten: 預設為 False，若設為 True，則會對每個主成分進行白化。白化可去除特徵間的相關性，但可能失去一些訊息。 svd_solver: 指定奇異值分解(SVD)的方法。 tol: 針對 arpack 解法時設定的容忍度。 iterated_power: 隨機化 SVD 中迭代的次數。

			random_state: 當使用 randomized SVD 解法時，用來設置隨機數生成器的種子。
Feature selection	sklearn.feature_selection	VarianceThreshold()	刪除方差未達指定閥值特徵。預設會刪掉所有方差為 0 的特徵。 VarianceThreshold(threshold, copy) threshold: 類型 float，預設值 0。這個參數設置了方差的閥值，方差低於這個閥值的特徵會被刪除。 copy: 類型 bool，預設值 True。如果設為 True，將會返回輸入 data 的副本，不會更改原始 data。
	sklearn.feature_selection	SelectKBest() chi2	根據評分，選擇評分較高的 k 個特徵。 SelectKBest(score_func, k) score_func: 用於計算特徵得分的函數。預設為 f_classif。 k: 整數，指定要選擇的最佳特徵數量。

			評估特徵與目標間的關聯性。透過卡方統計量和 p 值判斷特徵的重要性。 chi2(x, y, score_func, n_feature_to_select, threshold) x: 特徵矩陣。 y: 目標標籤。 score_func: 用於計算特徵得分的函數，預設為 chi2。 n_feature_to_select: 需要選擇的特徵數量。 threshold: 進行特徵選擇的閥值，特徵的得分大於等於此閥值的特徵會被選種。
	sklearn.feature_selection	SelectFromModel()	基於模型的重要性或係數來自動選擇特徵的方法。 SelectFromModel(estimator, threshold, prefit, norm_order, max_features) estimator: 用於特徵選擇的估算器，需要具備 feature_importances_或 coef_屬性。 threshold: 用於選擇特徵的重要性閥值，預設為 mean。

			prefit: 布林值，指示估算器使否已經擬和，預設為 False。如果為 True，則在擬和前不會再次擬和估算器 norm_order: 標準化的方式，指定用於計算特徵重要性 的 Lp 範數。 max_features: 整數或浮點數，限制選擇的特徵數量。當 為浮點數時，將其視為比例。
	sklearn.feature_selection	RFE()	遞迴的方式篩選出最重要的特徵。 RFE(estimator, n_features_to_select, step, verbose, importance_getter, cv) estimator: 用於計算特徵重要性的模型。 n_features_to_select: 要選擇的特徵數量。如果未指定， 則選擇所有特徵數量的 1/2。 step: 在每次遞迴中要刪除的特徵數量或比例。 verbose: 整數，預設為 0，表示是否顯示進程訊息，1

			表示顯示，0 表示不顯示。 importance_getter: 用於獲取特徵重要性的方式。 cv: 用於交叉驗證的參數。

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