

Data Mining mini project

40847011S 高子翔

(1) 資料集特徵資料說明

1. no_times_pregnant
懷孕次數 · ratio
2. glucose_concentration
口服定量葡萄糖 2 小時的血漿葡萄糖濃度 · ratio
3. blood_pressure
舒張壓 (mm Hg) · ratio
4. skin_fold_thickness
三頭肌皮脂厚度 (mm) · 用於測量是否為肥胖 · ratio
5. serum_insulin
餐後血清胰島素 (mu U/ml) · ratio
6. BMI
身體質量指數 (公斤體重/(公尺身高)²) · ratio
7. diabetes_pedigree
糖尿病族譜系數 · ratio
8. age
年齡 · ratio
9. diabetes
是否患有糖尿病 · nominal

(2)

其中 glucose_concentration、blood_pressure、skin_fold_thickness、serum_insulin、BMI、diabetes_pedigree 不應為 0，所以需要前處理將為 0 的視為 missing data 改掉，而我選擇使用平均值取代。

```
# preprocessing
NaN_col_names = ["glucose_concentration", "blood_pressure", "skin_fold_thickness", "serum_insulin", "bmi", "diabetes_pedigree"]
train[NaN_col_names] = train[NaN_col_names].replace(0, np.NaN)
test[NaN_col_names] = test[NaN_col_names].replace(0, np.NaN)

# 得到各項中間值後填入 missing data
train_medians = train.median()
train = train.fillna(train_medians)

test_medians = train.median()
test = test.fillna(test_medians)

y_train = train["diabetes"]
x_train = train.drop(["diabetes", "p_id"], axis = 1)
only_id = pd.DataFrame(test["p_id"])
test = test.drop(["p_id"], axis = 1)
```

為了避免極值造成訓練模型過於偏差，將所有的資料標準化。

```
# 數據標準化
Scaler = StandardScaler()
x_train = Scaler.fit_transform(x_train)
test = Scaler.fit_transform(test)
```

由於每項可能都跟糖尿病有關，使用 id 以外的所有特徵

(3)


使用 KNN，在面對這種數量級的資料時，不會需要太多的執行時間，也能有不錯的預測結果

(4)

總共調整了三次 KNN 的 n-neighbors，分別為 1, 2, 3 其中以 3 的表現最好。

```
# 訓練模型
neigh = KNeighborsClassifier(n_neighbors = 3)
neigh.fit(x_train, y_train)
```

(5)

 result.csv Complete (after deadline) · 40m ago	0.67532	0.67532	<input type="checkbox"/>
 result.csv Complete (after deadline) · 41m ago	0.70779	0.70779	<input type="checkbox"/>
 result.csv Complete (after deadline) · 41m ago	0.68181	0.68181	<input type="checkbox"/>

由上而下分別為 n-neighbors 為 1, 3, 2 時的結果表現