レポート提出票

科目名: 情報工学実験３

実験課題名: 課題２　パターン認識

実施日: 2024年 5月 16日

学籍番号: 4622045

氏名: 小澤　翼

共同実験者:

1　背景

今回扱うパターン認識は、特徴を抽出し数式にする段階と特徴に基づいてデータをあらかじめ定められたクラスに分類する段階があり、特徴をクラスに分類するために識別器を作成する必要がある。

2　目的

Python で表データ分析を行い、できるようになる。

3　課題

3.1　課題 1

大エリア (big_area_name) が A もしくは B に属するビジネスホテルを、基本料金 (base_price) が高い順に並べ、TOP10 をホテル ID、順位、基本料金がわかるように表示せよ。

以下はそのソースコードである。

ソースコード 1: 課題 1

```
#hotel_tb から大エリア名が A,B のホテルを抽出し、hotel_AB_tb という変数に代入する
hotel_AB_tb=hotel_tb.query('big_area_name == "A" or big_area_name == "B"')

#hotel_AB_tb からビジネスホテル(is_business が True のホテル)を抽出
hotel_AB_bus_tb=hotel_AB_tb.query('is_business == True')

#地価が高い順にランキングをつけて、hotel_AB_bus_tb に rank という列で追加
hotel_AB_bus_tb['rank']=hotel_AB_bus_tb['base_price'].rank(ascending=False,
    method='min')

#rank 順にデータを昇順ソート. ソートした結果をresult という変数に代入
result=hotel_AB_bus_tb.sort_values('rank',ascending=True).reset_index()

#TOP10 の表示
result.head(10)
```

このソースコードの結果は以下の通りである。

表 1: 課題 1 の結果

	index	hotel_id	base_price	big_area_name	small_area_name	hotel_latitude	hotel_longitude	is_business	rank
0	268	h_269	67600	A	A-3	35.815809	139.937240	True	1.0
1	192	h_193	58500	B	B-3	35.641363	139.594342	True	2.0
2	194	h_195	50200	A	A-3	35.909876	139.934702	True	3.0
3	5	h_6	49500	A	A-3	35.912764	139.731281	True	4.0
4	272	h_273	44900	A	A-3	35.809547	139.940654	True	5.0
5	160	h_161	44600	A	A-1	35.716076	139.839687	True	6.0
6	275	h_276	39500	B	B-1	35.437891	139.595387	True	7.0
7	79	h_80	36700	B	B-1	35.439007	139.695035	True	8.0
8	242	h_243	27700	A	A-1	35.712357	139.939032	True	9.0
9	1	h_2	26400	A	A-1	35.715320	139.939446	True	10.0

3.2 課題 2

customer_tb から顧客情報をランダムに 1 行抽出する。抽出した顧客の家と顧客が訪れたことのあるホテルの「距離」を計算し、その平均を表示せよ。(平均と customer_id が分かるように表示)

ここで「距離」とは、横軸を経度、縦軸を緯度とした二次元平面上の 2 点間のユークリッド距離とする。つまり、顧客 A の家が (経度, 緯度)=(120,30)、ホテル B が (経度, 緯度)=(123,34) であるとき、その「距離」は

$$\sqrt{(120-123)^2+(30-34)^2}=5$$

である。

以下はそのソースコードである。

ソースコード 2: 課題 2

```
  #customer_tb からランダムに 1 行抽出し、target という変数に代入する
target=customer_tb.sample(n=1)

#
    reserve_tb と target を customer_id に着目して結合し、merge_rc_tb という変数に代入する

merge_rc_tb=pd.merge(reserve_tb,target,on='customer_id',how='inner')

#target が訪れたことのあるホテルを抽出する
#
    merge_rc_tb と hotel_tb を hotel_id に着目して結合し、merge_cus_tb という変数に代入する

merge_cus_tb=pd.merge(merge_rc_tb,hotel_tb,on='hotel_id',how='inner')

#家からホテルまでの距離を計算し、merge_cus_tb に distance という列で追加
merge_cus_tb['distance']=((merge_cus_tb['home_latitude']-merge_cus_tb['
    hotel_latitude'])**2 + (merge_cus_tb['home_longitude']-merge_cus_tb['
    hotel_longitude'])**2)**0.5

#mean 関数でホテルの距離の平均を求め、distance_avg として追加
result=target
result['distance_avg']=merge_cus_tb['distance'].mean()

#結果の表示
result.fillna(0, inplace=True)
result
```

このソースコードの結果は以下の通りである。

表 2: 課題 2 の結果

	customer_id	age	sex	home_latitude	home_longitude	distance_avg
938	c_939	60	woman	32.280107	130.295499	10.503535

3.3 課題 3.A

大エリア E に属するホテルについて、緯度 (hotel_latitude), 経度 (hotel_longitude) を利用して小エリア (small_area_name) を分類する規則を自身で作成する。作成した規則に基づいてホテルを分類し、その結果と実際の小エリアを比較してその精度を求めよ。
レポートには、作成した規則とその精度がわかるように記載せよ。
規則の例）緯度が 30 度以上なら E-1、経度が 130 度未満であれば E-3 など
以下はそのソースコードである。

ソースコード 3: 課題 3.A

```
#hotel_tb から大エリアが E のホテルを抽出し、hotel_E_tb という変数に代入する
hotel_E_tb=hotel_tb.query('big_area_name=="E"')
#hotel_E_tb に含まれるホテルの位置をプロットして確認
hotel_E_tb.plot(x='hotel_longitude', y='hotel_latitude', kind='scatter')
#小エリアごとに緯度･経度の最大値･最小値を求める
hotel_E_tb.groupby('small_area_name') .agg({'hotel_latitude': ['max', 'min
    '],'hotel_longitude': ['max', 'min']}).reset_index()
#ルールの作成
def rule(x, y):
    if x < 136.94:
        if y < 35.24:
            return "E-1"
        else:
            return "E-3"
    else:
        if y < 35.24:
            return "E-2"
        else:
            return "E-4"
#作成したルールにしたがってホテルを分類し､分類結果をhotel_E_tb に付与
small_area_class = list(map(rule, hotel_E_tb['hotel_longitude'],hotel_E_tb['
    hotel_latitude']))
hotel_E_tb['small_area_class'] = small_area_class
hotel_E_tb
#分類の精度を調べる
#accuracy_score を使用するために ndarray 型の変数にする
big_area_true = hotel_E_tb['small_area_name'].to_numpy()
big_area_pred = hotel_E_tb['small_area_class'].to_numpy()

# scikit-learn で計算する
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(big_area_true, big_area_pred)
#精度の表示
print(accuracy)
```

以下のグラフと数値はそれぞれホテルの位置を縦軸を経度、横軸を緯度としてまとめたものとルールに基づく精度である。

図 1: ホテルの位置

ソースコード 4: 課題 3.A の精度

```
1.0
```

したがって、今回扱ったルールは正しかったといえる。

3.4 課題 3.B

どの小エリアに属しているか分かっていないホテルの小エリアを特定したいと思う。そのためには、今分かっているホテルの位置を図のように書き出し、そこで目視で明らかにまとまっていると思われる範囲をグルーピングし、そのグループ 1 つ 1 つを小エリアとし、分かっていないホテルをその小エリアに属させればよいと考える。

4 まとめ

今回の実験で Python を扱う能力を高めることが出来た。