상권 데이터 업종 분포 분석

Col: 시도명, 행정동명, 상권업종중분류명, 상권업종소분류명...

분석 방식: (Defult group: '시도명', '행정동명')
1) 동일 중분류 中 동일 소분류 높은 비율 내림차순
2) 동일 중분류와 소분류 中 높은 중분류 개수 내림차순
3) 가중치(weighted_middle_category) 계산 후 높은 비율 내림차순

함수:
1. recommend_regions_sorted: 동일 중분류 中 동일 소분류 비율 출력

	시도명	행정동명	상권업종중분류명	상권업종소분류명	count	middle_category_count	subcategory_ratio
2106	서울특별시	개포1동	한식	백반/한정식	4	4	1.000000
45696	서울특별시	이촌2동	한식	백반/한정식	6	6	1.000000
18302	서울특별시	반포2동	한식	백반/한정식	12	14	0.857143
42075	서울특별시	오륜동	한식	백반/한정식	11	13	0.846154
16893	서울특별시	무악동	한식	백반/한정식	9	11	0.818182
2325	서울특별시	개포3동	한식	백반/한정식	20	26	0.769231
21693	서울특별시	부암동	한식	백반/한정식	20	26	0.769231
43026	서울특별시	용산2가동	한식	백반/한정식	26	34	0.764706
27349	서울특별시	서빙고동	한식	백반/한정식	13	17	0.764706
24529	서울특별시	삼청동	한식	백반/한정식	48	63	0.761905

1) 그룹화 및 개수 계산: 데이터프레임을 특정 컬럼(시도명, 행정동명 등)으로 그룹화하여 소분류별 개수를 계산.

2) 중분류 총 개수 계산:지역별로 중분류의 총 개수를 계산하여, 중분류의 전체 크기를 파악

3) 병합(Merge): 계산된 중분류 총 개수를 원본 데이터와 병합하여 각 소분류 데이터에 중분류 총 개수를 추가

4) 소분류 필터링: 사용자가 입력한 특정 소분류 값(input_example)에 해당하는 데이터만 필터링하여 분석.

5) 비율 계산: 각 지역에서 소분류가 중분류에서 차지하는 비율을 계산합니다. 이 비율은 소분류의 중요도 표시.

2. recommend_regions_with_ratio_and_middle_count:

middle_category_count를 기준으로 내림차순 정렬

	시도명	행정동명	상권업종중분류명	상권업종소분류명	count	middle_category_count	subcategory_ratio
40203	서울특별시	역삼1동	한식	백반/한정식	494	895	0.551955
50353	서울특별시	종로1.2.3.4가동	한식	백반/한정식	481	863	0.557358
39970	서울특별시	여의동	한식	백반/한정식	428	711	0.601969
27136	서울특별시	서교동	한식	백반/한정식	348	622	0.559486
777	서울특별시	가산동	한식	백반/한정식	227	485	0.468041
990	서울특별시	가양1동	한식	백반/한정식	221	462	0.478355
41391	서울특별시	영등포동	한식	백반/한정식	219	400	0.547500
8673	서울특별시	논현2동	한식	백반/한정식	183	384	0.476562
15763	서울특별시	명동	한식	백반/한정식	198	382	0.518325
28113	서울특별시	서초3동	한식	백반/한정식	202	368	0.548913

3. recommend_regions_with_weighted_ratio: 가중치 계산, combined_score 내림차순

	시도명	행정동명	상권업종중분류명	상권업종소분류명	count	middle_category_count	subcategory_ratio	weighted_middle_category	combined_score
40203	서울특별시	역삼1동	한식	백반/한정식	494	895	0.551955	8.347007	8.898963
50353	서울특별시	종로1.2.3.4가동	한식	백반/한정식	481	863	0.557358	8.048567	8.605925
39970	서울특별시	여의동	한식	백반/한정식	428	711	0.601969	6.630975	7.232944
27136	서울특별시	서교동	한식	백반/한정식	348	622	0.559486	5.800937	6.360423
777	서울특별시	가산동	한식	백반/한정식	227	485	0.468041	4.523239	4.991280
990	서울특별시	가양1동	한식	백반/한정식	221	462	0.478355	4.308735	4.787090
41391	서울특별시	영등포동	한식	백반/한정식	219	400	0.547500	3.730506	4.278006
15763	서울특별시	명동	한식	백반/한정식	198	382	0.518325	3.562633	4.080958
8673	서울특별시	논현2동	한식	백반/한정식	183	384	0.476562	3.581286	4.057848
28113	서울특별시	서초3동	한식	백반/한정식	202	368	0.548913	3.432066	3.980979

주요 변경 사항

1) 가중치 계산:

middle_category_count의 값에 가중치를 부여하기 위해 각 지역의 중분류 총 개수를 전체 평균(average_middle_count)으로 나눔.

* 결과적으로 weighted_middle_category가 계산.

2) 새로운 지표(combined_score) 생성:

subcategory_ratio와 weighted_middle_category를 더하여 새로운 지표(combined_score)를 생성.

*** 의도:**

1) 상대적인 중요성 반영:

- middle_category_count가 클수록 해당 지역이 더 중요한 상권일 가능성이 높음.
- 지역별 middle_category_count의 절대적인 크기는 지역마다 다르므로, 전체 평균을 기준으로 나누어 상대적인 중요성을 반영.

2) 정규화:

- 규모(middle_category_count)가 다르기에 값을 평균으로 나누면 모든 값이 "평균 대비 몇 배인지"를 나타내는 상대적 지표로 바뀌어, 과도한 값 차이를 줄임을 기대.

$$\text{weighted\_middle\_category} = \frac{\text{middle\_category\_count}}{\text{average\_middle\_category\_count}}$$

4) 한계 및 수정사항

- 비율로만 따지기에 분석이 유의미한가를 고민
- 예시 입력값(input_exmple) 사용 중
- 파이썬(jupyter note) 사용으로 spark 변환 필요
- 지역확대 필요(현재 서울지역 데이터만 사용 중)