

AI Term-Project (Prof.최강타)

안선영

20201696

Topic. 프랜차이즈 매장 위치 분석 및 선형회귀분석을 통한 최적의 입지 찾기

Content.

1. 문제 정의
2. 데이터 수집
3. 데이터 전처리
4. 데이터 분석
 - 4-1. 분석에 사용된 방법에 대한 소개
 - 4-2. 분석 방법에 대한 선택 이유
 - 4-3. 분석 결과
5. 데이터 시각화
6. 결론 AND/OR 제언
7. 출처



1. 문제 정의

2021년 기준 한국 자영업자의 비중은 23.9%로 OECD 30개국의 평균인 17.0%보다 높은 수준이라고 한다. 대한민국 사람이라면 퇴직 후 치킨집을 차릴 것이라는 농담은 한 번쯤 들어봤거나 생각해봤을 법한 이야기이다. 이러한 현실을 생각하며 인공지능 기말 프로젝트에서 프랜차이즈 매장 위치 분석 및 선형회귀분석을 통한 최적의 입지를 찾아보고자 하였다. 매장은 폐업율이 낮은 편에 속하는 배스킨라빈스와 던킨도너츠를 선정하였다. 공공데이터 포털을 활용하여 서울시의 특정 상권에서 "배스킨라빈스"와 "던킨도너츠"의 입점 현황 및 위치를 이해한 뒤, 그 후 선형회귀분석을 사용하여 경도와 위도 간의 관계를 모델링하여 새로운 위치의 위도를 예측하고자 하였다.

2. 데이터 수집

공공데이터 포털에서 소상공인시장진흥공단_상가(상권)정보를 다운로드받은 뒤 seoul.csv 파일을 생성하여 로드한다. seoul.csv는 상호명, 업종코드, 업종명, 지번주소, 도로명주소, 경도, 위도 등의 정보를 포함하고 있다.

3. 데이터 전처리

먼저, 결측치 확인 후 시각화하였다. 필요한 컬럼(col)만을 선택하고, 서울시에 해당하는 데이터만 필터링하였다. 신고된 상호명의 경우 브랜드명을 결정하고 결측치를 처리한다. 상호명을 통일시킨 뒤 배스킨라빈스와 던킨도너츠를 포함하는 데이터 추출한다. 그 후 상권업종대분류명이 음식인 데이터만 남기고 나머지는 제외하며 데이터 전처리작업을 수행했다.

4. 데이터 분석

4-1. 분석에 사용된 방법에 대한 소개

- 1) 데이터 전처리를 위해 pandas, numpy를 사용하였다.
- 2) countplot를 사용하여 브랜드별 분포된 매장의 개수를 확인하였다.
- 3) pivot_table, countplot를 사용하여 시군구별 브랜드별 상호 수 카운트 및 시각화 하였다.
- 4) scatterplot 및 jointplot을 사용하여 브랜드별 위치를 시각화 하였다.
- 5) Marker, MarkerCluster를 사용하여 지도상 브랜드별 위치를 시각화 하였다.
- 6) 마지막으로 Scikit-Learn의 LinearRegression모델을 사용하여 선형회귀분석을 통한 경도와 위도의 관계 모델링 하였다.

4-2. 분석 방법에 대한 선택 이유

데이터 전처리, 시각화, 경도와 위도를 활용한 선형회귀분석 단계로 프로세스를 나누었다. 각 브랜드의

상호수, 시군구별 브랜드 상호수, 브랜드의 위치분포 시각화하는 작업으로 각각의 분석방법을 사용하였고 선형회귀모델은 경도와 위도 사이의 관계를 모델링하기에 적합하며, 간단한 모델로서 설명이 용이하여 분석 방법으로 사용하게 되었다.

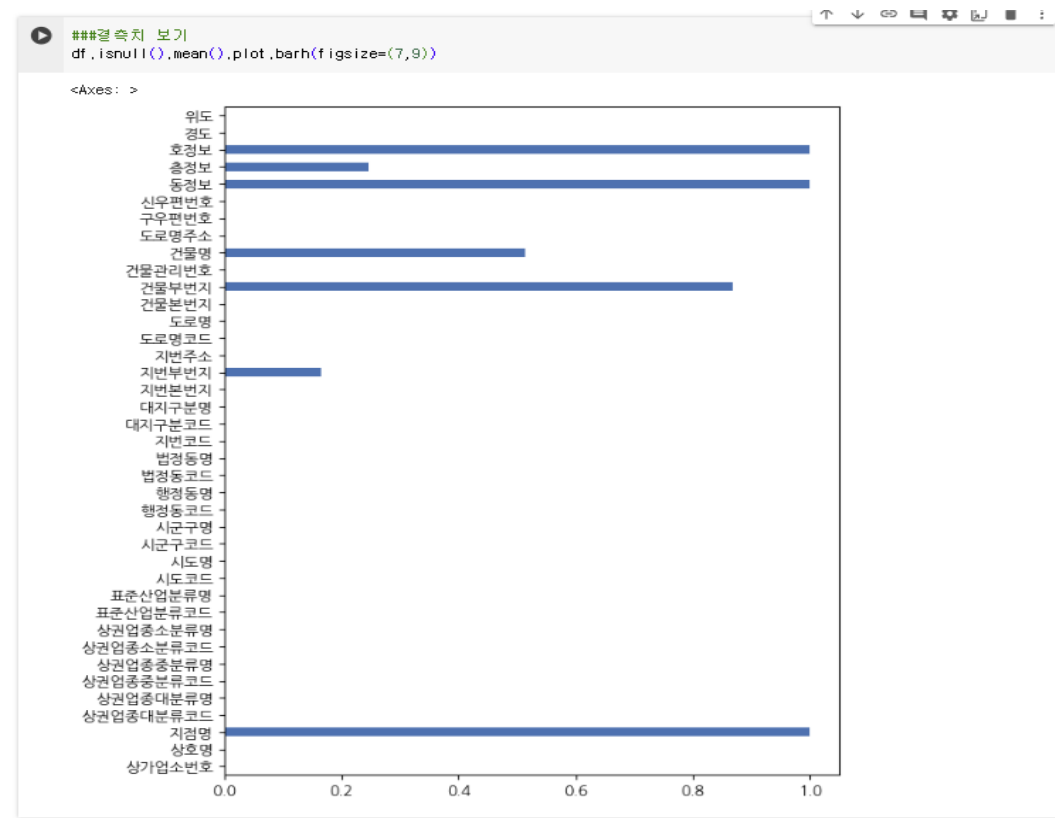
4-3. 분석 결과

먼저, 던킨도너츠와 배스킨라빈스의 상호 수를 확인하고 지도에서 분포 수를 확인할 수 있도록 하였다. 분포도를 통해 매장 간 위치가 얼마나 가깝고 먼저 시각적으로 확인할 수 있다.

그 후 매장을 입점시키기 위한 위치를 선정하고자 하였다. 선형회귀분석을 통해 경도와 위도 간의 관계 모델링하여 새로운 매장 위치를 예측하고자 하였으나, 위도 -500이라는 존재하지 않는 결과가 나왔다. 매장의 최적의 입지 분석을 위해서는 단순히 분포도 뿐만 아니라 지역 거주자, 매출 등의 정보가 필요하다는 생각을 하게 되었다.

5. 데이터 시각화

1) 데이터 전처리



```
df.loc[df['상호명_소문자'].str.contains('(배|베)스킨라빈스|baskinrobbins'),fillna(False), '상호명_소문자']

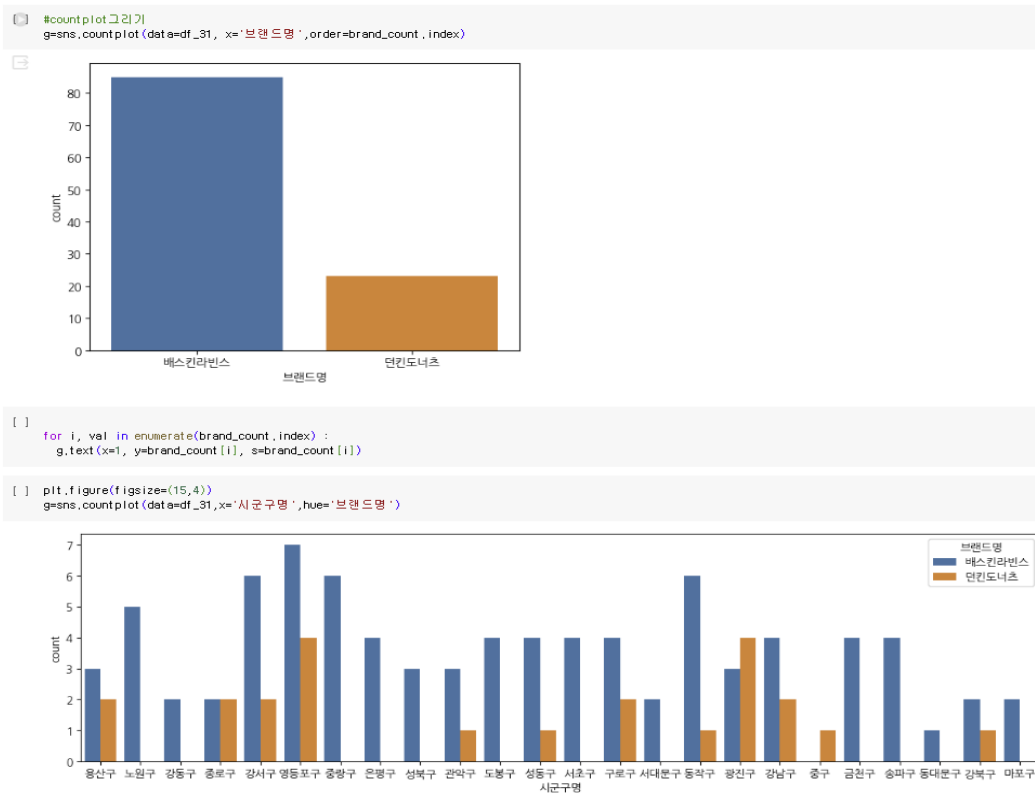
<ipython-input-81-f56bcca68cf6>:1: UserWarning: This pattern is interpreted as a regular expression, and has match groups. To actually get the groups, use
df.loc[df['상호명_소문자'].str.contains('(배|베)스킨라빈스|baskinrobbins'),fillna(False), '상호명_소문자']
2110    비알베스킨라빈스하날hive
5209    비알베스킨라빈스동부미송점
9630    베스킨라빈스강계역점
10223    베스킨라빈스강동역점
10559    비알베스킨라빈스서충대병원
...
122918    베스킨라빈스방학
123082    베스킨라빈스금천
123546    베스킨라빈스신영화역점
126219    베스킨라빈스플로우도곡
129556    베스킨라빈스코유터미널경부선점
Name: 상호명_소문자, Length: 85, dtype: object

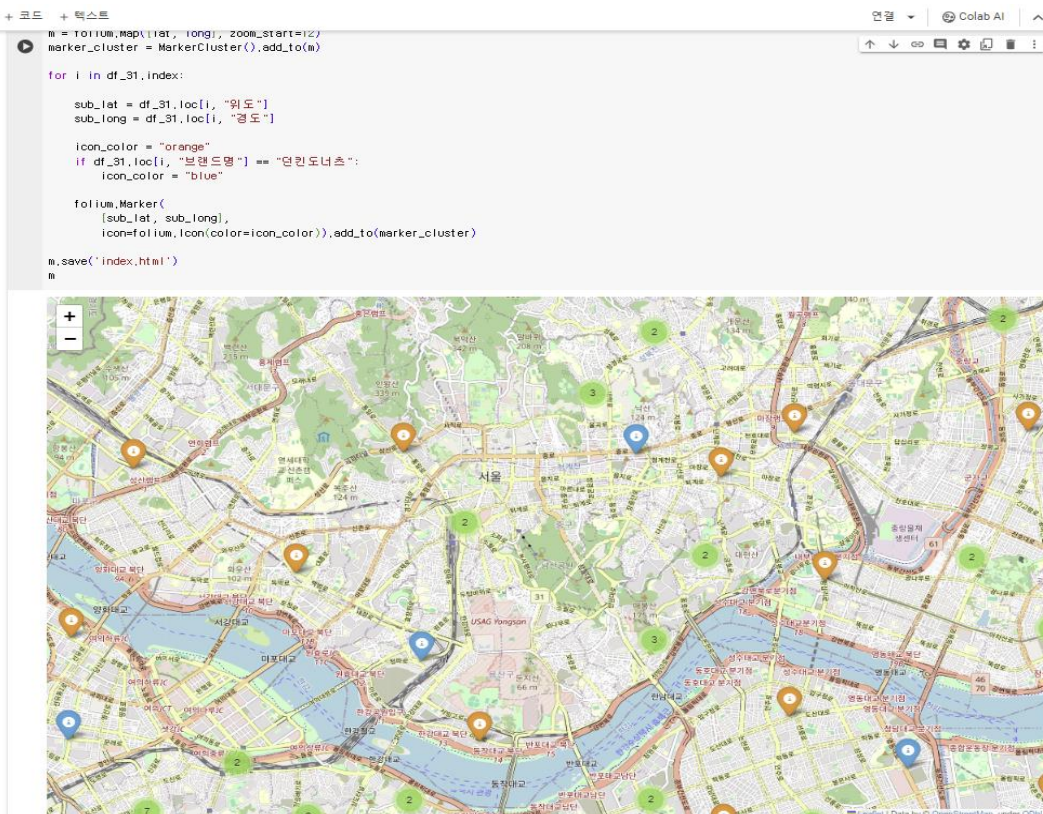
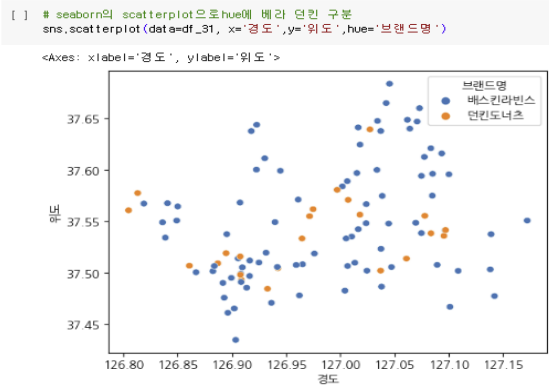
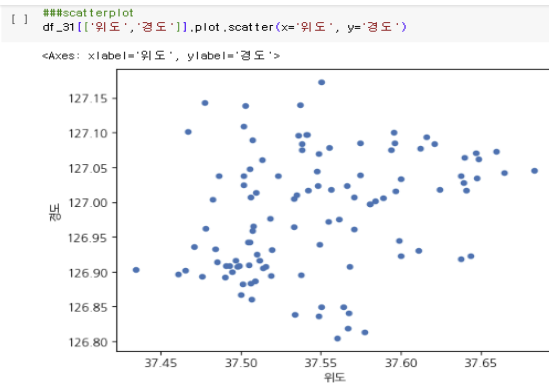
[ ] #연킨 | dunkin
df.loc[df['상호명_소문자'].str.contains('연킨|dunkin'),fillna(False), '상호명_소문자']
16461    비알연킨도너츠산림양지병원
23657    연킨도너츠충대문
24991    연킨도너츠목동남포점
28661    연킨도너츠
29717    연킨현대아이파크
32156    연킨상도점
32688    명성인터내셔널연킨
32712    베스킨라빈스/연킨축석dt점
36641    연킨도너츠신도림
59346    연킨금천
76440    연킨도너츠
78156    연킨도너츠
78706    연킨도너츠용산하마로마트점
81895    비알연킨도너츠강북구청사거리
83127    연킨도너츠한림대강남성심병원
90130    비알연킨도너츠공평힐도서출역사점
94919    연킨도너츠역삼gs
97751    연킨군자역점
106290    비알연킨도너츠서충대병원
106992    연킨도너츠태크노마트
113859    연킨신พล역점
115884    비알연킨도너츠김포공항국제선
116673    연킨
118332    연킨도너츠
121169    연킨도너츠영도표역사
Name: 상호명_소문자, dtype: object

[ ] #연킨+베스킨라빈스
df_31 = df[df['상호명_소문자'].str.contains('(배|베)스킨라빈스|baskinrobbins|연킨|dunkin'),fillna(False)].copy()

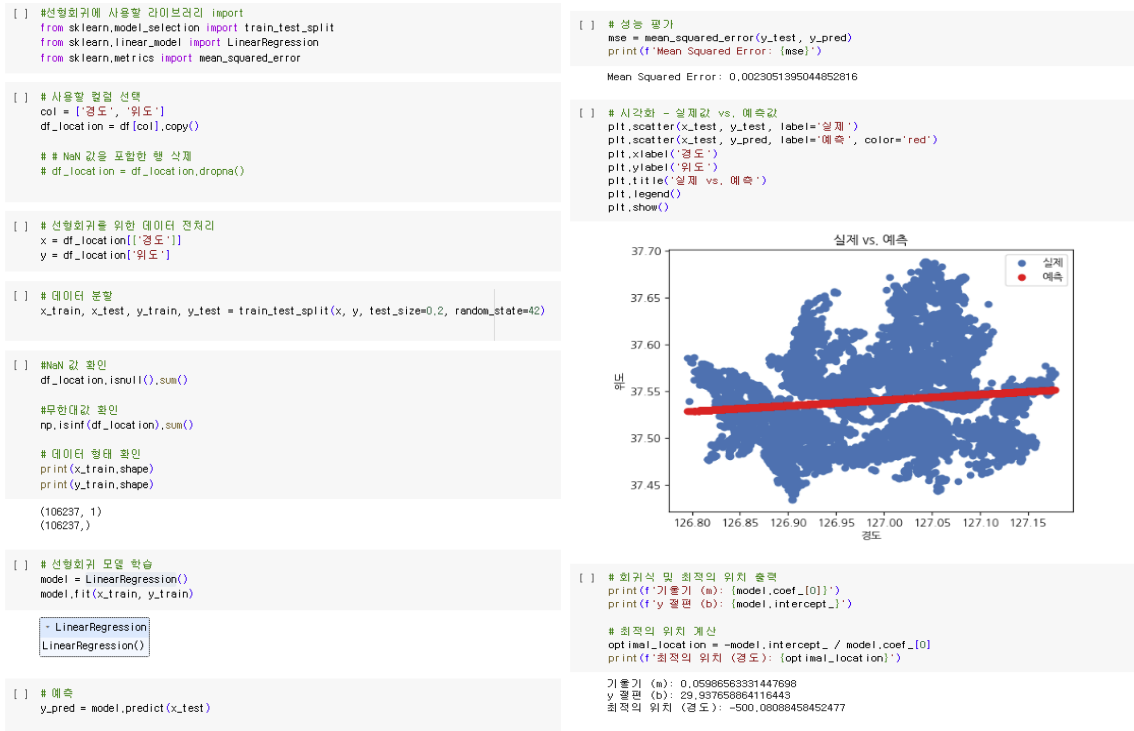
<ipython-input-83-d5898536417c>:2: UserWarning: This pattern is interpreted as a regular expression, and has match groups. To actually get the groups, use
```

2) 데이터 시각화





3) 경도와 위도를 활용한 선형회귀분석 단계



6. 결론 AND/OR 제언

브랜드명별 프랜차이즈 매장 수를 비교하고 시군구별 브랜드 분포도를 시각화하였다. 이를 통해 매장 간 위치 및 거리를 시각적으로 확인할 수 있으며, 배스킨라빈스와 던킨도너츠의 서울시 매장 분포와 브랜드별 특징을 확인할 수 있었다. 그 후 선형회귀분석을 통해 경도와 위도 간의 관계 모델링하여 새로운 매장 위치를 예측하고자 하였으나, 위도 -500이라는 존재하지 않는 결과가 나왔다.

이에 대한 제언 사항은 아래와 같다. 선형회귀는 종속 변수와 하나 이상의 독립 변수 간의 선형 관계를 모델링하는 통계적 기법이고 위치분석은 새로운 매장을 어느 곳에 위치시킬지에 대한 의사결정사항이다. 전국적으로 분포된 데이터에서 단일 최적의 위치를 찾고자 하여 결과가 예상대로 나오지 않았다. 지역별로 나누어 선형회귀모델을 적용하거나 매장별 매출, 브랜드 입지 전략, 지역별 인구수 등의 다양한 변수를 고려하여 복잡한 모델에서 최적의 위치를 찾아내는 방법을 고려해야 할 것 같다.

7. 출처

<https://www.hankookilbo.com/News/Read/A2023121914250002242?did=GO>

<https://www.data.go.kr/dataset/15012005/fileData.do>

누구나 파이썬 너도 데이터 가지고 놀 수 있어! (민형기)