

포스코 미래창조아카데미

골목상권 매출 분석 및 예측 서비스

팩맨(Factory Man)

박민우 윤다빈 윤종관

Contents

① 프로젝트 개요

② 프로젝트 수행 절차 및 방법

③ 프로젝트 수행 경과

④ Q&A

골목상권 매출 예측 서비스

- 자영업자 폐업률 급증 이슈
- 서울시 상권 분석 서비스의 한계
- 매출 예측 서비스 구현
- 서울시 골목상권 분석 서비스

골목상권의 등대, 그 불빛을 다시 밝히자

[Special Report | 어쩌다...폐업 자영업자 100만 시대] 최저임금, 내수 부진, 고물가·고금리로 곳곳 '비명' 차별화, 저비용, 현장 맞춤 형 지원으로 재기 기회 제공해야

차남수 소상공인연합회 정책홍보본부장 입력 2024-08-26 09:00:01

가 0 0

최저임금, 내수 부진, 고물가·고금리로 곳곳 '비명'

50대 창업자 두 명 중 한 명이 자영업자

배달 앱, 안 쓰면 망하고 쓰면 적자

숙박·음식점업 10개 중 8개가 5년 내 폐업

무너진 자영업 생태계 바로 세워야

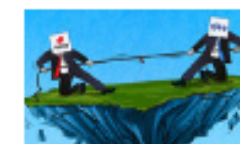


소상공인·자영업자를 일컬어 골목상권의 등대 구실을 한다고 한다. 골목 곳곳을 밝히고 우리나라 경제의 근간인 지역경제의 지킴이 역할을 해왔기 때문이다. 하지만 언제부터인지 이 등대의 불빛이 꺼지기 시작했다. 24시간 골목을 환하게 비추던 등대가 사라지면서 골목 곳곳의 온기는 싸늘해지고 어두워지면서 대한민국 경제가 무너지고 있다. 이들은 왜 점점 벼랑 끝으로 더 내몰리는 것일까.

삼성생명

기획·연재

최병천, 겹눈으로 보다 구독



보수는 자본주의를, 진보는 복지국가를 만들었다



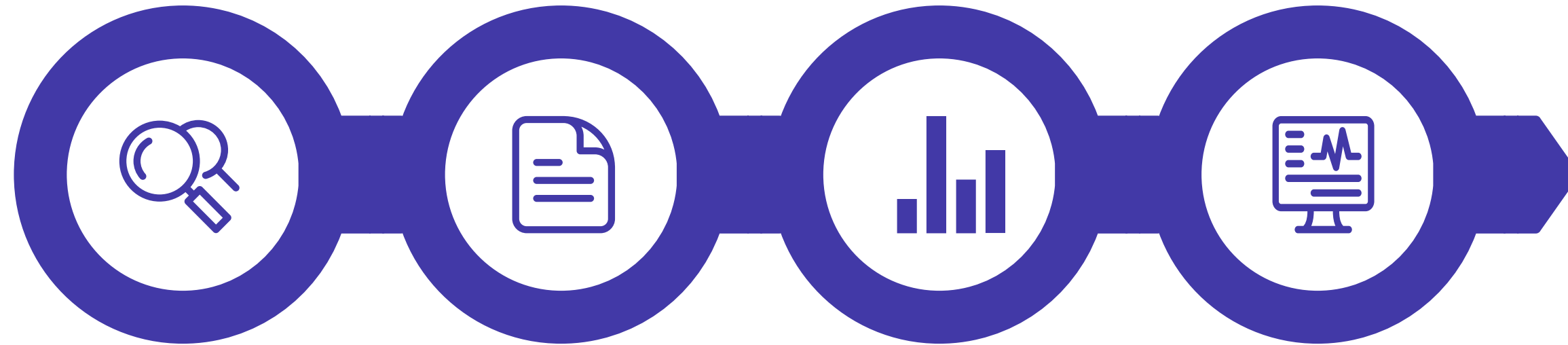
한동훈, '당신은 윤석열과 무엇이 다른가'에 답해야



박정희 다음으로 노무현 좋아하는 새로운 보수의 출현

1 2 3 4

프로젝트 구조



데이터 수집

데이터 전처리

데이터 모델링

서비스 구현

프로젝트 팀 구성 및 역할

박민우	팀장	<ul style="list-style-type: none">- 데이터 전처리 및 시각화- 서비스 시스템 설계- 데이터 모델링
윤다빈	팀원	<ul style="list-style-type: none">- 데이터 전처리 및 시각화- 통계 분석- 데이터 모델링
윤종관	팀원	<ul style="list-style-type: none">- 데이터 전처리 및 시각화- 서비스 웹 구현- 데이터 모델링

프로젝트 수행 절차 및 방법

구분	기간	활동
사전 기획	2024.08.26	<ul style="list-style-type: none">- 프로젝트 주제 선정 및 로드맵 구성- 기획안 구성
데이터 수집	2024.08.27 ~ 2024.08.28	<ul style="list-style-type: none">- 필요 데이터 및 수집 절차 정의- 외부 데이터 수집
데이터 전처리 및 모델링	2024.08.28 ~ 2024.08.30	<ul style="list-style-type: none">- 데이터 정제 및 정규화- 데이터 시각화- 통계 분석 및 예측
서비스 구축	2024.08.31	<ul style="list-style-type: none">- 웹 서비스 설계- 서비스 구현
보고서 및 발표 준 비	2024.09.01 ~ 2024.09.03	<ul style="list-style-type: none">- 프로젝트 보고서 작성- 발표 준비

골목상권 매출 데이터

- 서울 열린데이터 광장
- 파일명
 - 서울시_상권분석서비스(추정매출-상권배후지)_2019~2023년.csv (5개)
 - 서울시 상권분석서비스(영역-자치구).csv
- 데이터 정보
 - object: 기준년분기, 상권 구분, 상권코드, 업종코드,
 - int: 당월, 주중/주말, 요일별, 시간별, 성별, 연령별(매출, 건수)
 - int64(50), object(5)
 - column 55 x row 148,000

The screenshot shows the Seoul Open Data Platform interface. At the top, there's a navigation bar with links: 서울 열린데이터 광장, 공공데이터, 통계, 서울빅데이터, 소식&참여, and 이용안내. Below this is a blue header with '데이터셋' on the left and a breadcrumb trail 'Home > 공공데이터 > 공공데이터' on the right. A search bar is present with the placeholder text '찾고 싶은 데이터를 입력해 주세요.' and buttons for '상세 검색' and '통합 검색'. Below the search bar is a checkbox for '결과 내 재검색'. The main content area features a card for '서울시 상권분석서비스(추정매출-상권배후지)'. The card has a blue diagonal label '공공데이터' and an icon of a bar chart with an upward arrow. It includes buttons for '활용사례(갤러리) 등록', 'URL 복사', and '목록 이동'. The title of the dataset is '서울시 상권분석서비스(추정매출-상권배후지)', and the description is '서울시 상권배후지 영역 내의 점포들의 추정 매출 정보를 제공합니다. ※단위: 원'. Below the description is the label '일반행정'.

컬럼 제거

기준_년분기_코드	상권_구분_코드	상권_구분_코드_명	상권_코드	상권_코드_명	서비스_업종_코드	서비스_업종_코드_명	당월_매출_금액
20221	A	골목상권	3110389	한일병원	CS200005	스포츠 강습	8571429
20221	A	골목상권	3110535	영천시장입구	CS200002	외국어학원	6033530
20221	A	골목상권	3110989	대치동아우편취	CS300027	섬유제품	29573670
20221	A	골목상권	3110775	대동초등학교	CS200037	노래방	105541754
20221	A	골목상권	3110775	대동초등학교	CS300021	문구	2193308277
20221	A	골목상권	3110992	도곡초등학교	CS200019	PC방	708271102
20221	A	골목상권	3110866	난우중학교	CS300007	육류판매	384812252
20221	A	골목상권	3110288	동구여중	CS100002	중식음식점	63228778
20221	A	골목상권	3110976	언북초등학교	CS100005	제과점	159923119
20221	A	골목상권	3110179	광나루역 1번	CS200016	당구장	1790311
20221	A	골목상권	3110954	국기원	CS200002	외국어학원	1245879104
20221	A	골목상권	3110397	도봉산입구	CS300008	수산물판매	18555925
20221	A	골목상권	3110022	경신고등학교	CS100004	양식음식점	407109692
20221	A	골목상권	3110884	봉천역 4번(봉천	CS100002	중식음식점	285418061
20221	A	골목상권	3110919	내방역 8번	CS100003	일식음식점	206743678
20221	A	골목상권	3111027	문정초등학교	CS100006	패스트푸드점	607384245
20221	A	골목상권	3110884	봉천역 4번(봉천	CS200026	자동차미용	38248980
20221	A	골목상권	3111027	문정초등학교	CS200019	PC방	356209694
20221	A	골목상권	3111027	문정초등학교	CS100001	한식음식점	2589261285

상권 코드 -> 지역구명 변경

기준_년분기_코드	상권_구분_코드	상권_구분_코드_명	지역구	상권_코드_명	서비스_업종_코드	서비스_업종_코드_명	당월_매출_금액
20221	A	골목상권	도봉구	한일병원	CS200005	스포츠 강습	8571429
20221	A	골목상권	서대문구	영천시장입구	CS200002	외국어학원	6033530
20221	A	골목상권	강남구	대치동아우편취급	CS300027	섬유제품	29573670
20221	A	골목상권	영등포구	대동초등학교	CS200037	노래방	105541754
20221	A	골목상권	영등포구	대동초등학교	CS300021	문구	2193308277
20221	A	골목상권	강남구	도곡초등학교	CS200019	PC방	708271102
20221	A	골목상권	관악구	난우중학교	CS300007	육류판매	384812252
20221	A	골목상권	성북구	동구여중	CS100002	중식음식점	63228778
20221	A	골목상권	강남구	언북초등학교	CS100005	제과점	159923119
20221	A	골목상권	광진구	광나루역 1번	CS200016	당구장	1790311
20221	A	골목상권	강남구	국기원	CS200002	외국어학원	1245879104
20221	A	골목상권	도봉구	도봉산입구	CS300008	수산물판매	18555925
20221	A	골목상권	종로구	경신고등학교	CS100004	양식음식점	407109692
20221	A	골목상권	관악구	봉천역 4번(봉천	CS100002	중식음식점	285418061
20221	A	골목상권	서초구	내방역 8번	CS100003	일식음식점	206743678
20221	A	골목상권	송파구	문정초등학교	CS100006	패스트푸드점	607384245
20221	A	골목상권	관악구	봉천역 4번(봉천	CS200026	자동차미용	38248980
20221	A	골목상권	송파구	문정초등학교	CS200019	PC방	356209694
20221	A	골목상권	송파구	문정초등학교	CS100001	한식음식점	2589261285

상권 코드 -> 지역구명 변경

```
import pandas as pd

df = pd.read_csv(r"C:/Users/alfm0/sf6-data/data_csv/sub.csv", encoding="EUC-KR")
df1 = pd.read_csv(r"C:/Users/alfm0/sf6-data/data_csv/main.csv", encoding="EUC-KR")

df2 = df.groupby('자치구_코드_명')['상권_코드'].unique().apply(list).reset_index()
df2.columns = ['자치구_코드_명', '상권_코드']
# print(df2.head())

mapping_dict = {}
for index, row in df2.iterrows():
    for code in row['상권_코드']:
        mapping_dict[code] = row['자치구_코드_명']

# 상권 코드 대체
df1['상권_코드'] = df1['상권_코드'].map(mapping_dict).fillna(df1['상권_코드'])

df1.to_csv("C:/Users/alfm0/sf6-data/datas/main(수정).csv", index=False, encoding='utf-8-sig')
```

```
# 데이터 로드
df1 = pd.read_csv(r"C:/Users/yjk23/sf6-data/datas/main.csv", encoding="EUC-KR")
df1['상권배후지_코드'] = df1['상권배후지_코드'].astype(str)
for index, value in enumerate(df1['상권배후지_코드_명']):
    global count
    global fail_count
    result = search_map(value)

    # 주소 추출
    address = result['items'][0]['address'] if 'items' in result and result['items'] else None
    if '서울특별시' not in qwer:
        fail_count += 1
        df1.loc[index, '다른지역'] = address
    if address: # 주소가 유효한 경우
        count += 1
        district = address.split(" ")[1] # 구 정보 추출
        df1.loc[index, '상권배후지_코드'] = district # 차례대로 할당
        df1.loc[index, '상권_구'] = district

df1.to_csv(r"C:/Users/yjk23/sf6-data/datas/real_main.csv", encoding="EUC-KR", index=False)

# 결과 출력
print("성공: ", count)
print("실패: ", fail_count)
# print(df1)
```

매출 합계 전처리(분기별)

```
# 각 년도에 대해 파일을 읽고, 각 분기별 매출을 계산
for i, year in enumerate(years):
    # 해당 년도의 CSV 파일 읽기
    df = pd.read_csv(r"C:/Users/yjk23/sf6-data/data_csv/main_" + f"{year}.csv", encoding="utf-8-sig")

    # 각 분기별 매출 합계 계산
    for quarter in quarters[i]:
        # 분기 및 업종 필터링
        filtered_df = df[df['기준_년분기_코드'] == quarter]

        # 총 매출 금액 계산
        total_sales = filtered_df['당월_매출_금액'].sum()

        # 결과를 리스트에 추가
        sales_data.append({
            '기준_년분기_코드': quarter,
            '당월_매출_금액': total_sales
        })

# 데이터프레임 생성
result_df = pd.DataFrame(sales_data)
```

기준_년분기_코드	당월_매출_금액
2019-01-01	₩23,684,900,000,000
2019-04-01	₩24,685,700,000,000
2019-07-01	₩24,987,600,000,000
2019-10-01	₩26,802,400,000,000
2020-01-01	₩24,302,500,000,000
2020-04-01	₩26,332,500,000,000
2020-07-01	₩25,827,700,000,000
2020-10-01	₩25,591,700,000,000
2021-01-01	₩24,412,100,000,000
2021-04-01	₩26,888,700,000,000
2021-07-01	₩26,930,100,000,000
2021-10-01	₩29,591,300,000,000
2022-01-01	₩27,490,100,000,000
2022-04-01	₩31,155,500,000,000
2022-07-01	₩31,774,900,000,000
2022-10-01	₩33,039,000,000,000
2023-01-01	₩32,053,000,000,000
2023-04-01	₩33,418,600,000,000
2023-07-01	₩33,059,200,000,000
2023-10-01	₩32,987,000,000,000

업종별 전처리(분기별)

```
# 업종 코드에 따른 전처리 작업
for index, code in enumerate(df['서비스_업종_코드']):
    if 'CS1' in code:
        df.loc[index, '업종_전처리'] = '요식업'
    elif 'CS2' in code:
        df.loc[index, '업종_전처리'] = '서비스업'
    elif 'CS3' in code:
        df.loc[index, '업종_전처리'] = '판매업'
```

업종_전처리	서비스_업종_코드	서비스_업종_코드_명
서비스업	CS200005	스포츠 강습
서비스업	CS200002	외국어학원
판매업	CS300027	섬유제품
서비스업	CS200037	노래방
판매업	CS300021	문구
서비스업	CS200019	PC방
판매업	CS300007	육류판매
요식업	CS100002	중식음식점
요식업	CS100005	제과점
서비스업	CS200016	당구장
서비스업	CS200002	외국어학원
판매업	CS300008	수산물판매
요식업	CS100004	양식음식점
요식업	CS100002	중식음식점
요식업	CS100003	일식음식점
요식업	CS100006	패스트푸드점
서비스업	CS200026	자동차미용
서비스업	CS200019	PC방
요식업	CS100001	한식음식점

업종별 전처리(분기별)

```
# 각 년도에 대해 파일을 읽고, 각 분기별 매출을 계산
for i, year in enumerate(years):
    # 해당 년도의 CSV 파일 읽기
    df = pd.read_csv(r"C:/Users/yjk23/sf6-data/data_csv/main_" + f"{year}.csv", encoding="utf-8-sig", low_memory=False)

    # 현재 년도에 맞는 분기를 반복
    for quarter in quarters[i]:
        for cook in cooks:
            # 분기 및 업종 필터링
            filtered_df = df[(df['기준_년분기_코드'] == quarter) & (df['업종_전처리'] == cook)]
            # 총 매출 금액 계산
            total_sales = filtered_df['당월_매출_금액'].sum()

            # 결과를 리스트에 추가
            sales_data.append({
                '기준_년분기_코드': quarter,
                '업종': cook,
                '총 매출 금액': total_sales
            })

# 데이터프레임 생성
df1 = pd.DataFrame(sales_data)
```

기준_년분기_코드	업종	총 매출 금액
20191	요식업	₩ 6,008,555,047,787.00
20191	서비스업	₩ 4,737,147,319,034.00
20191	판매업	₩ 12,939,195,472,899.00
20192	요식업	₩ 6,481,667,488,743.00
20192	서비스업	₩ 4,977,837,935,842.00
20192	판매업	₩ 13,226,181,792,895.00
20193	요식업	₩ 6,580,376,420,742.00
20193	서비스업	₩ 5,006,864,977,886.00
20193	판매업	₩ 13,400,362,563,624.00
20194	요식업	₩ 7,202,289,198,367.00
20194	서비스업	₩ 5,345,261,861,572.00
20194	판매업	₩ 14,254,865,074,284.00
20201	요식업	₩ 6,126,806,278,029.00
20201	서비스업	₩ 4,739,492,759,974.00
20201	판매업	₩ 13,436,182,368,443.00
20202	요식업	₩ 6,714,069,815,393.00
20202	서비스업	₩ 5,048,492,573,792.00
20202	판매업	₩ 14,569,892,672,764.00
20203	요식업	₩ 6,483,857,401,021.00
20203	서비스업	₩ 5,068,200,607,520.00
20203	판매업	₩ 14,275,653,089,711.00
20204	요식업	₩ 6,067,085,417,198.00
20204	서비스업	₩ 5,022,485,787,020.00
20204	판매업	₩ 14,502,114,599,268.00
20211	요식업	₩ 5,544,484,616,429.00
20211	서비스업	₩ 4,981,236,335,685.00
20211	판매업	₩ 13,886,340,280,235.00
20212	요식업	₩ 6,767,955,749,096.00
20212	서비스업	₩ 5,496,922,304,273.00
20212	판매업	₩ 14,623,847,077,632.00

지역구별 전처리(분기별)

```
for i, y in enumerate(year):
    df = pd.read_csv(r"C:/Users/yjk23/sf6-data/data_csv/main_" + f"{y}.csv", encoding="utf-8-sig", low_memory=False)

    for quarter in quarters[i]:
        quarter_data = {'기준_년분기_코드': quarter}

        for seoul in seoul_map:

            filtered_df = df[(df['기준_년분기_코드'] == quarter) & (df['상권_코드'] == seoul)]
            total_sales = filtered_df['당월_매출_금액'].sum()

            quarter_data[seoul] = total_sales

        sales_data.append(quarter_data)

df1 = pd.DataFrame(sales_data)
```

기준_년분기_코드	금천구	중랑구	관악구
20191	₩ 1,164,181,482,463.00	₩ 1,051,898,317,704.00	₩ 1,235,641,548,842.00
20192	₩ 1,048,919,430,611.00	₩ 1,124,780,752,634.00	₩ 1,324,817,076,019.00
20193	₩ 1,191,525,933,764.00	₩ 1,147,311,163,813.00	₩ 1,303,121,436,025.00
20194	₩ 1,180,045,819,333.00	₩ 1,227,903,665,133.00	₩ 1,406,399,186,970.00
20201	₩ 1,214,332,238,624.00	₩ 1,134,853,439,607.00	₩ 1,242,392,322,530.00
20202	₩ 1,198,994,279,240.00	₩ 1,253,556,897,266.00	₩ 1,377,807,827,048.00
20203	₩ 1,287,238,138,300.00	₩ 1,223,148,378,012.00	₩ 1,313,422,379,925.00
20204	₩ 1,194,700,386,690.00	₩ 1,194,068,821,608.00	₩ 1,281,185,086,219.00
20211	₩ 1,289,505,729,709.00	₩ 1,134,816,776,124.00	₩ 1,194,520,734,974.00
20212	₩ 1,229,160,890,713.00	₩ 1,279,270,995,897.00	₩ 1,373,182,968,124.00
20213	₩ 1,364,172,019,192.00	₩ 1,304,034,724,221.00	₩ 1,326,535,321,787.00
20214	₩ 1,420,262,454,214.00	₩ 1,372,339,001,424.00	₩ 1,482,224,597,681.00

시계열 분석

ADF 검증

데이터의 정상성을 판단하는 검증

차분

정상성을 확보하는 기법

ARIMA 모델 예측

시계열 데이터 예측 모델

시계열 분석

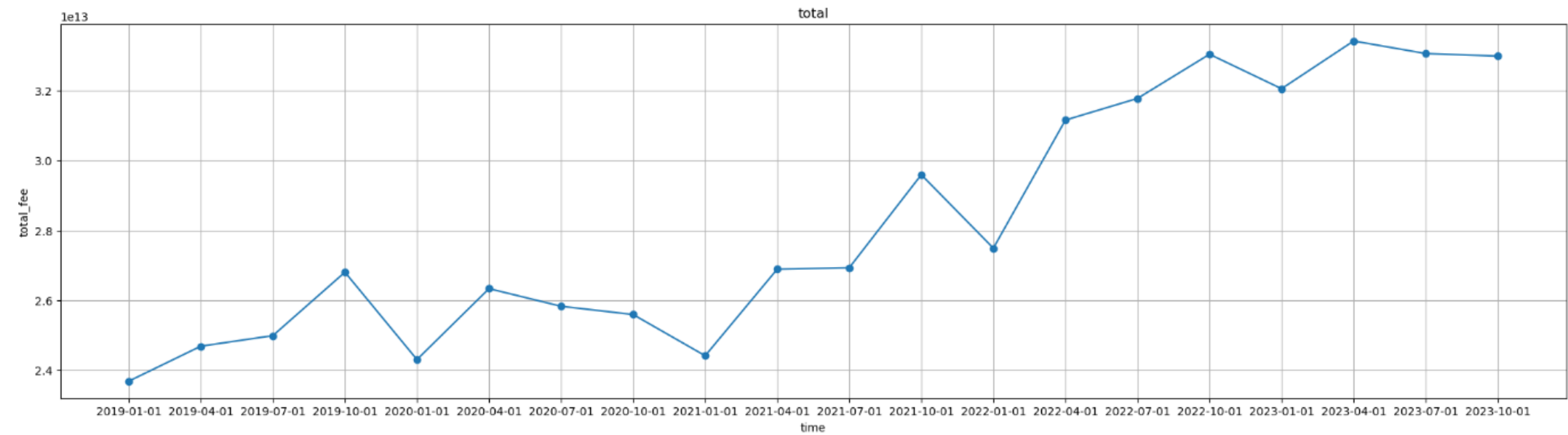
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima.model import ARIMA
import warnings
```

```
# ADF 검정으로 정상성 확인
adf_result = adfuller(quarterly_sales['총매출금액'])
print('ADF Statistic:', adf_result[0])
print('p-value:', adf_result[1])
```

```
# 비정상성을 보정하기 위해 차분 수행
quarterly_sales['차분된_총매출금액'] = quarterly_sales['총매출금액'].diff().dropna()

# 차분된 데이터에 대해 ADF 검정 수행
adf_result_diff = adfuller(quarterly_sales['차분된_총매출금액'].dropna())
print('차분 후 ADF Statistic:', adf_result_diff[0])
print('차분 후 p-value:', adf_result_diff[1])
```

```
# ARIMA 모델 적합 (차분된 데이터 사용)
model = ARIMA(quarterly_sales['차분된_총매출금액'].dropna(), order=(1, 0, 1))
model_fit = model.fit()
```



```
ADF Statistic: -0.49974465975789517
p-value: 0.8920663148929234
```

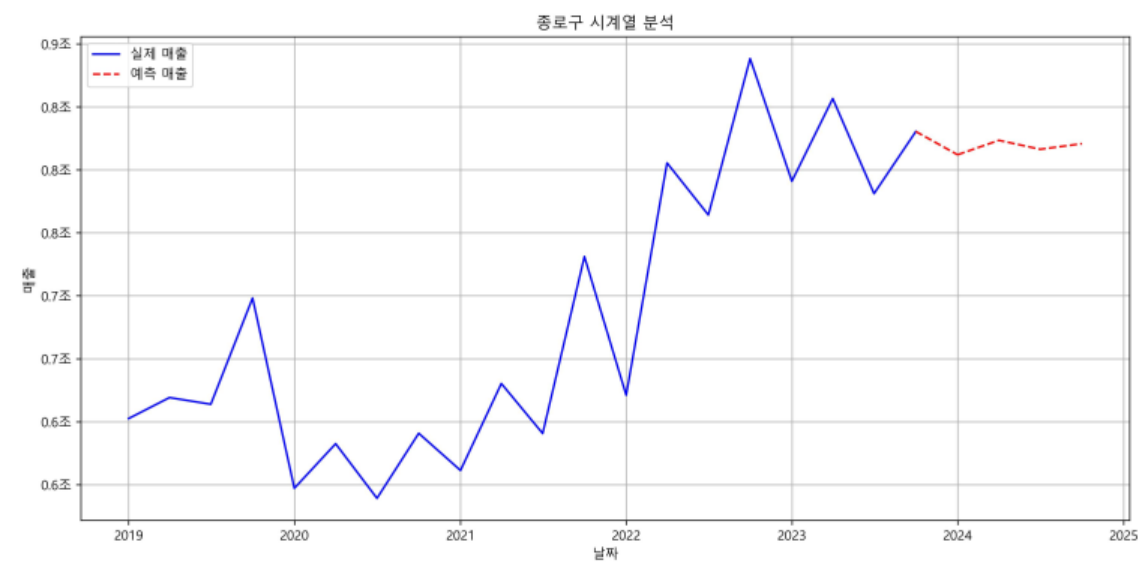
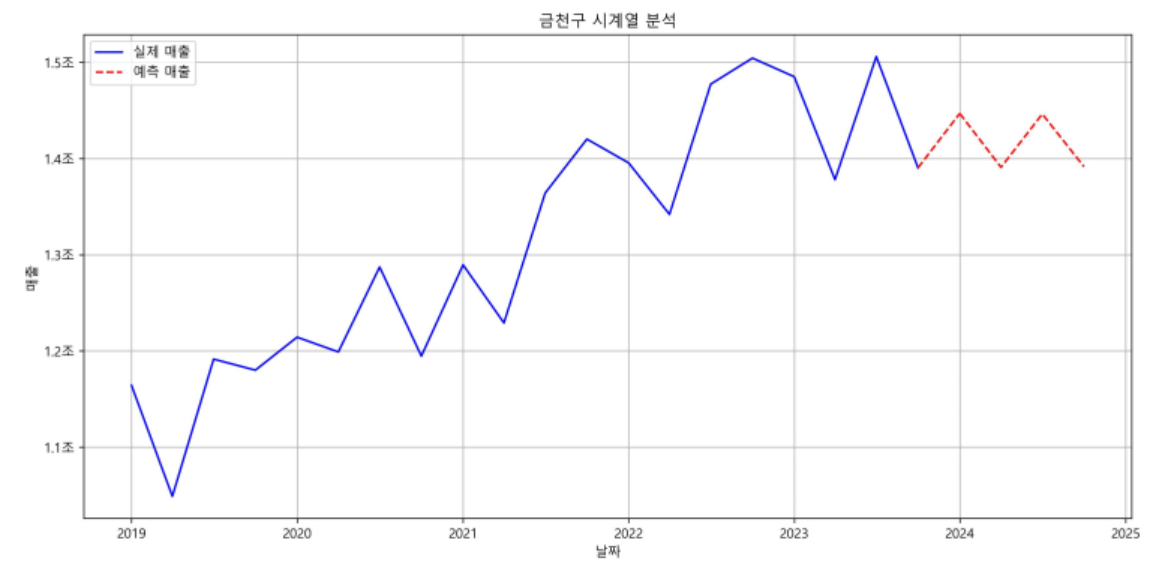
```
차분 후 ADF Statistic: -2.9735013155494205
차분 후 p-value: 0.03744424471279066
```

```
다음 4분기 총매출 예측: 2024-01-01    3.247767e+13
2024-04-01    3.396130e+13
2024-07-01    3.346169e+13
2024-10-01    3.493564e+13
```


업종별 시계열 분석



지역별 시계열 분석



골목상권 분기별 예측 서비스

Q & A

골목상권의 등대, 그 불빛을 다시 밝히자 기사

<https://shindonga.donga.com/economy/article/all/13/5128018/1>

서울시 열린데이터 광장

<https://data.seoul.go.kr/>

서울시 상권 분석 서비스

<https://golmok.seoul.go.kr/commercialArea/commercialArea.do>