| 무역 데이터를 이용한 품목별 무역수지 예측 모델 생성

김 란 한유진

목 차

1 연구 배경 및 목적 연구 방법 6 분석 시각화 2 연구 절차 주요 코드 7 결론 및 고찰

연구 배경 및 목적

배경

- 최근 2년간 지속되는 무역수지 적자
- 96.8%로 높은 무역의존도 (수출의존도 + 수입의존도)
- 외부 충격에 매우 취약한 경제 구조

목적

- 무역의존도가 높은 상품 파악
- 무역수지가 높은 품목 활성화
- 무역수지 낮은 품목 개선
- 품목별 무역수지 흐름 파악

연 구 절 차

자료 수집	자료 전처리	군집 분석(표본추출)	모델 생성	결론 및 시각화	
● 나라별 경제지표를 나타낼 수 있는 무역 데이터 수집	● Null값 처리 ● 형변환 • 풍목코드 (숫자->객체) • 그 외 변수 (콤마제거,객체->숫자) ● 표준화	● Best-Cluster ● K-means ● 상위 백분위 샘플링	 • 각 군집별 주요 무역 품목 추출 • SARIMA 모델 생성 	● 예측 모델 검증 ● 결과 해석	

연구방법

데이터 수집

품목별 국가별 수출입실적

(출처: 수출입무역통계 https://unipass.customs.go.kr/ets/index.do)

GDP, 대출금리, 환율

(출처: https://kosis.kr/statHtml/statHtml.do?orgld=101&tblld=DT 2KAA811)

표본 추출

조사 방법: 상위 백분위 샘플링을 통해

군집분석 후 각 군집의 상위 25%를 추출

데이터 수집 시기: 2022년

주요코드

```
# best cluster 본기
size = range(1, 11) # k값 범위
inertia = [] # 응답도 (중심점과 포인트 긴 거리 제급함)

for k in size:
    obj = KMeans(n_clusters = k)
    model = obj.fit(merged2)
    inertia.append(model.inertia_)

print(inertia)

# 시작화
plt.plot(size, inertia, '-o')
plt.xticks(size)
plt.show()

# 클러스터링
kmeans = KMeans(n_clusters=3, random_state=10)
clusters = kmeans.fit(merged2)
```

주요코드

주요코드

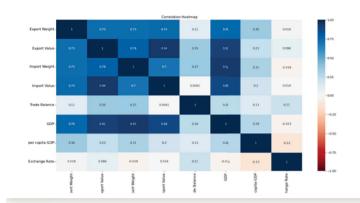
```
# 훈련, 테스트 세트로 스플릿
                                                                                                                                                                                    # 모델과 전치 확인
model_aic = model.fit()
print(model_aic.sunmary().tables[1])
y_train = y[:len(y)-11]
                                                                          for on in pdg:
                                                                               for pm_seasonal in seasonal_idq :
                                                                                                                                                                                     model_aic.plot_diagnostics(figsize=(16,8))
y_test = y[(len(y)-11):]
y_test
                                                                                       model - sm.tsa.statespace.SARIMAX(y_train,
                                                                                                                             order = pm,
seasonal_order = pm_seasonal,
enforce_stationarity=False,
enforce_invertibility=False
                                                                                                                                                                                    plt.show()
                                                                                                                                                                                    # 모델로 무역수지 예측
forecast = model_aic.get_prediction(
v test.nlot()
                                                                                                                                                                                     start = pd.to_datetime('2022-07-01'),
end = pd.to_datetime('2023-08-01'),dynamic = False)
prediction = forecast.predicted_mean
y_train.plot(figsize = (20,10))
                                                                                       print("ARINA()xSEASONAL()12 - AIC: ()".format(om.om seasonal, model aic.aic)"
                                                                                       metric_aic_dict.update({(pm,pm_seasonal): model_aic.aic})
# AD Fullter test 를 통한 시계원 데이터 정상성 확인
                                                                                                                                                                                     actual = y_test['2021-07-01':]
result = adfuller(y_train)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
                                                                          model aic.aic
                                                                                                                                                                                    forecast = model_aic.get_forecast(steps=12)
                                                                                                                                                                                    # 예측 신뢰구간
predictions = forecast.predicted_mean
                                                                          metric_aic_dict.items()
 print('Critival Values : ')
                                                                          min_aic = min(metric_aic_dict.it;ms(), key=lambda x: x[1])
for key, value in result[4].items():
                                                                                                                                                                                      ci = forecast.conf_int()
                                                                          print(min_sic)
# 최적화 모델
print('\t%s: %.3f' %(key,value))
len(y_train) #37
                                                                                                                                                                                     # 예측 vs 실제 데이터 그래프 시각화 + 미래 무역수지 예측
#ACF and PACF
                                                                          #(((0, 1, 1), (0, 1, 1, 12)), 1031.8627749588624)
                                                                                                                                                                                     fig = y.plot(label='observed', figsize-(15,7))
fig.set_vlabel('Date')
fig.set_vlabel('Trade Balance')
                                                                       (k: v for k,v in sorten,m-

# 阿耳爾 四度 MEM 性理 四度 例如

model * M. Tas thirtespee, SetTMM(v, Iredin,
order - (0,1,1),

sessional_order - (0,1,1,12)
                                                                          {k: v for k,v in sorted(metric_aic_dict.items(), key = lambda x: x[1])}
p = d = q = range(\theta, 2)
                                                                                                                                                                                     predictions.plot(ax-fig, label-'predicted', color-'r')
pdq = list(itertools.product(p,d,q))
                                                                                                                                                                                     seasonal_pdq = [(x[0], x[1], x[2], 12) for x in pdq]
 seasonal odg
```

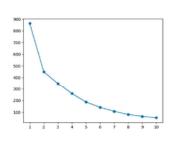
분석 시각화 (상관 관계)



군집 분석 전 변수 간의 상관 관계 확인 및 시각화

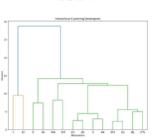
분석 시각화 (군집 분석)





elbow method를 통한 최적의 갯수 찾기

군집 분석



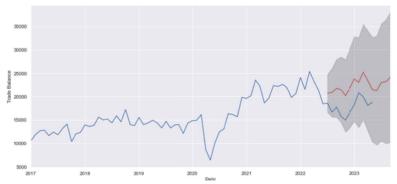
시각화를 위한 계층적 군집 분석

분석 시각화 (Word Cloud)



각 군집별 무역수지가 높은 HScode 시각화

분석 결과 (SARIMA 모델)



생성된 SARIMA 모델을 사용하여 cluster2의 HScode 39류를 예측 (파란색 선은 실제 데이터 결과이며, 붉은색 선은 모델로 예측된 결과)

결론 및 고찰

결론 - 1 (군집분석)	결론 - 2 (시계열 분석)	결론 - 3 (기대효과)	고찰
cluster0 : 베트남, 이란, 인도 시장성장률이 높음 cluster1 : 미국,중국 현재 시장 점유율이 높음 cluster2 : 그 외의 나라	두개의 군집에서 공통적으로 나타난 주요 품목 세 가지(87, 85, 39)에서 전반적인 상승 트렌드를 볼 수 있었음	러스터링을 통해 얻은 객관적인 데이터를 바탕으로 성장 가능성 이 큰 수출 시장(현 분석에서는 이란,베트남,인도)을 발굴함으 로써 수출 다변화를 추진하여 특정 시장에 대한 수출 의존도 를 줄일 수 있을거라 기대	- 나라의 경제적 특징이 강하여 군집 안의 나라가 고르게 나눠질 거라는 예상과는 다른 결과가 나옴 - 무역 데이터의 특징(오랜 시간이지난 데이터는 무의미함) 때문에 6년치의 데이터를 사용하였으나 모델을 생성하기에 충분치않음