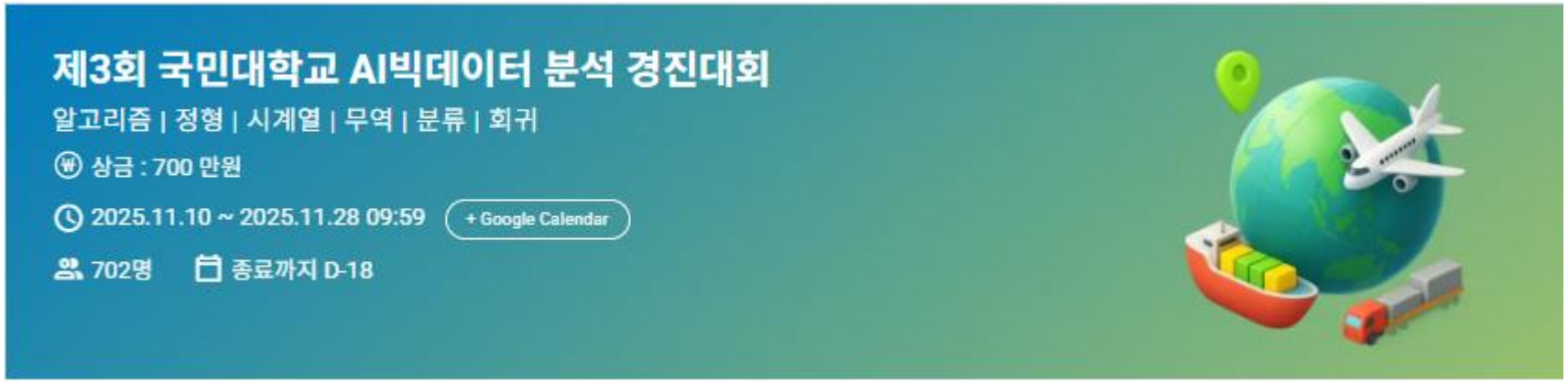
ACD II Week12-1

건양대학교 인공지능학과 김준화

AI빅데이터 분석 경진대회



대회 링크
: https://dacon.io/competitions/official/236619/overview/description

대회 목적
: 무역 품목 간 공행성 쌍 판별 및 후행 품목 무역량 예측 AI 모델 개발

AI빅데이터 분석 경진대회

[배경]

국민대학교 경영대학원 AI빅데이터/디지털마케팅전공과 경영대학에서 '제3회 국민대학교 AI빅데이터 분석 경진대회'를 개최합니다.

이번 대회는 국민대학교 경영대학원과 한국기계산업진흥회(KOAMI)이 공동으로 주최하여 품목 간 무역 연동성과 미래 예측 가능성에 대한 AI 기술의 응용을 주제로 진행됩니다.

참가자는 100개 수입 품목의 월별 무역 데이터를 분석하여 공행성이 있는 품목 쌍을 판별하고, 선행 품목의 흐름을 바탕으로 후행 품목의 다음 달 무역량을 예측하는 알고리즘을 개발하게 됩니다.

이를 통해 무역 데이터에서의 품목 간 구조적 관계를 탐색하고, 실무에 적용 가능한 예측 기반 의사결정 지원 도구로 활용될 수 있는 AI 모델의 가능성을 확인할 수 있을 것으로 기대합니다.

[대회 방식]

본 대회는 예선, 본선 두 단계로 진행됩니다.

◆예선: 예선 Private 리더보드 기준 상위 20팀이 본선 진출자로 선발됩니다.

◆본선: 본선 진출 20팀은 추가 학습 데이터(8월 원시 무역 데이터)를 바탕으로 모델을 추가 개선하여 제출합니다.

※ 본선 모델 개선을 위한 리더보드 제출은 별도 대회 페이지에서 제공되며, 추후 안내 예정입니다.

[설명]

원시 무역 수입 데이터(2022년 1월 ~ 2025년 7월)를 기반으로, 품목 간 공행성(comovement)이 존재하는 선후행 쌍을 예측하고, 공행성이 있다고 판단된 경우에는 후행 품목의 다음 달(2025년 8월) 총 무역량(value)을 예측하는 AI 모델을 개발합니다.

참가자는 주어진 원시 무역 데이터를 분석하여 품목 간 선후행 관계가 존재하는 공행성 쌍(A → B)을 찾아야 하며, 이후에는 선행 품목(A)의 흐름을 활용해 후행 품목(B)의 다음달의 **총 무역량(value)**을 예측해야 합니다.

1. 개인 또는 팀 참여 규칙

- 개인 또는 팀을 이루어 참여할 수 있으며, 동일인이 개인 또는 복수팀에 중복하여 등록 불가
- 개인 참가 방법 : 팀 신청 없이, 자유롭게 제출탭에서 제출 가능
- 팀 참가 방법 : 팀 탭에서 가능, 상세 내용은 팀 탭에서 팀 병합 정책 확인
- 팀 구성 방법: 팀 페이지에서 팀 구성 안내 확인
- 팀 최대 인원: **5인**

2. 대회 규칙

1) 사전 학습 모델 사용 가능 범위

- 2025년 11월 10일 전(~2025.11.09)에 공식적으로 가중치가 공개되었으며, **최소 상업적 이용이 허용**된 오픈소스 라이선스 (예: MIT, Apache 2.0 등)로 배포된 사전 학습 모델만 사용 가능합니다. 해당 조건을 충족하지 않는 모델은 사용할 수 없습니다.

2) API 사용 제한

- 원격 서버를 통해서만 접근 가능한 API 형태의 모델(예: OpenAI API, Gemini API 등)은 사용이 불가능합니다. 모든 모델은 로컬 환경에서 직접 실행 가능해야 하며, 외부 서버에 의존하는 방식은 허용되지 않습니다.

3) 외부 데이터 사용 금지

- 본 경진대회에서 제공한 데이터 외의 모든 외부 데이터는 사용이 금지됩니다. 단, 제공된 학습 데이터를 바탕으로 사전 학습 모델 또는 허용된 도구를 활용해 데이터 증강 또는 생성하는 것은 가능합니다.

4) 평가 데이터 누수(Data Leakage)

- 일반적인 AI 경진대회 원칙과 동일하게 테스트 데이터에 대한 사전 접근·활용 그로 인한 Data Leakage 행위는 일체 금지됩니다. 참가자는 학습 데이터 기반으로만 모델을 개발해야 합니다.

AI빅데이터 분석 경진대회

대회 주요 일정

[세부일정]

- 참가 신청 기간 : 2025년 11월 03일(월) 10:00 ~ 2025년 11월 28일(금) 10:00
- 예선 기간 : 2025년 11월 10일(월) 10:00 ~ 2025년 11월 28일(금) 10:00
  - 팀 병합 마감 : 2025년 11월 21일(금) 23:59
- 예선 종료 : 2025년 11월 28일(금) 10:00
- 본선 기간 : 2025년 11월 28일(금) 12:00 ~ 2025년 12월 08일(월) 10:00
  - 본선 리더보드 제출 마감 : 2025년 12월 05일(금) 10:00
  - 코드 및 솔루션 자료 제출 마감 : 2025년 12월 08일(월) 10:00
- 코드 검증 : 2025년 12월 08일(월) ~ 2025년 12월 12일(금)
- 최종 수상자 발표 : 2025년 12월 15일(월) 10:00
- 오프라인 시상식 : 추후 안내

※ 세부 일정은 대회 운영 상황에 따라 변동될 수 있습니다.

평가 방법

1. 예선 리더보드

• **평가 산식 :** Score = 0.6 × F1 + 0.4 × (1 − NMAE) [코드]

1) F1 = (2 × Precision × Recall) ÷ (Precision + Recall)

• Precision = TP ÷ (TP + FP)
• Recall = TP ÷ (TP + FN)

여기서

• TP(True Positive): 정답과 예측 모두에 포함된 공행성쌍
• FP(False Positive): 예측에는 있으나 정답에는 없는 쌍
• FN(False Negative): 정답에는 있으나 예측에 없는 쌍

2) NMAE = (1 / |U|) × Σ[min(1, |y_true - y_pred| ÷ (|y_true| + ε))]

• U = 정답 쌍(G)과 예측 쌍(P)의 합집합
• y_true: 정답의 다음달 무역량 (정수 변환)
• y_pred: 예측 무역량 (정수 반올림)
• FN 또는 FP에 해당하는 경우 오차 1.0(100%, 최하점)로 처리
• 오차가 100%를 초과하는 경우에도 1.0(100%, 최하점)로 처리

• Public score : 전체 테스트 데이터 100%
• Private score : 예선 종료 시점의 Public score

2. 평가 방식

• 예선 평가 : 예선 리더보드 Private 상위 20팀 선발
• 본선 평가 : 예선 Private 리더보드 점수 50% + 본선 Private 리더보드 점수 50%
  • 모델 성능 항목 환산식 : 50 × ((팀의 Private 리더보드 점수) / (최고 점수)) ^ N

※ '최고 점수'는 최종 평가 대상자 중 Private 리더보드 순위가 가장 높은 팀의 점수를 기준으로 하며, N은 1~5 사이의 비공개 조정 계수로 설정

※ 본선 리더보드는 본선 진출자 대상으로 별도의 페이지에서 제공 예정

[총 상금 700만 원]

🥇 대상(1팀, 한국기계산업지능회장상) - 300만 원

🥈 최우수상(1팀, 국민대학교경영대학원장상) - 200만 원

🥉 우수상(2팀) - 각 100만 원

수업 평가

- 데이콘 성적 순(리더보드 private 기준)으로 아래 표와 같이 차등 부여 (25점)

수상(4등 이내)	25점 + 발표 면제
5% 이내	23점
5% ~ 10%	20점
10% ~ 20%	16점
20% ~ 30%	12점
30% 이하	10점

- 팀 X, 개별 참여

Ex) 최종 등록 팀 210팀 중 10등 → 5% 이내 → 22점 획득

- 최종 발표 점수 (5점) – 대회 종료 후 "12월 5일 예정"

1. Import ¶

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from tqdm import tqdm
```

2. 데이터 전처리

```
train = pd.read_csv('./dataset/train.csv')
train.head()
```

	item_id	year	month	seq	type	hs4	weight	quantity	value
0	DEWLVASR	2022	1	1.0	1	3038	14858.0	0.0	32688.0
1	ELQGMQWE	2022	1	1.0	1	2002	62195.0	0.0	110617.0
2	AHMDUILJ	2022	1	1.0	1	2102	18426.0	0.0	72766.0
3	XIPPENFQ	2022	1	1.0	1	2501	20426.0	0.0	11172.0
4	FTSVTTSR	2022	1	1.0	1	2529	248000.0	0.0	143004.0

- 필요 라이브러리 Import

- Feature 확인

- Item id : 무역품 식별 ID
- Year : 년
- Month : 월
- Seq : 동일-년-월 내 일련번호
- Type : 유형 분류 코드
- Hs4 : HS4 코드
- Weight : 중량
- Quantity : 수량
- Value : 무역량

```
# year, month, item_id 기준으로 value 합산 (seq만 다르다면 value 합산)
monthly = (
    train
    .groupby(["item_id", "year", "month"], as_index=False)["value"]
    .sum()
)
```

```
monthly.head()
```

	item_id	year	month	value
0	AANGBULD	2022	1	14276.0
1	AANGBULD	2022	2	52347.0
2	AANGBULD	2022	3	53549.0
3	AANGBULD	2022	5	26997.0
4	AANGBULD	2022	6	84489.0

- 같은 item_id, year, month를 가지는 row(열)을 묶어서 그 안의 value 값들을 전부 더함(sum)
- 같은 월의 여러 기록들을 한 번에 월 총합으로 묶음

→ 품목 – 연도 – 월 단위로 묶어서 월별 집계가 된 시계열을 만드는 단계

```
# year, month를 하나의 키(ym)로 묶기
monthly["ym"] = pd.to_datetime(
    monthly["year"].astype(str) + "-" + monthly["month"].astype(str).str.zfill(2)
)
```

```
monthly.head()
```

	item_id	year	month	value	ym
0	AANGBULD	2022	1	14276.0	2022-01-01
1	AANGBULD	2022	2	52347.0	2022-02-01
2	AANGBULD	2022	3	53549.0	2022-03-01
3	AANGBULD	2022	5	26997.0	2022-05-01
4	AANGBULD	2022	6	84489.0	2022-06-01

- Year와 month를 문자열로 바꾼 다음, "YYYY-MM" 형태로 합치고, 날짜 타입으로 변환해서 ym 이라는 새로운 feature를 만듬
- str.zfill(2) → 두자리 월 형식으로 맞춤

→ 연도/월이 따로 있으면 불편하니, 시계열 인덱스로 쓰기 편한 하나의 날짜형 컬럼으로 묶음

```
# item_id × ym 피벗 (월별 총 무역량 매트릭스 생성)
pivot = (
    monthly
    .pivot(index="item_id", columns="ym", values="value")
    .fillna(0.0)
)

pivot.head()
```

ym	2022-01-01	2022-02-01	2022-03-01	2022-04-01	2022-05-01	2022-06-01	2022-07-01
item_id							
AANGBULD	14276.0	52347.0	53549.0	0.0	26997.0	84489.0	0.0
AHMDUILJ	242705.0	120847.0	197317.0	126142.0	71730.0	149138.0	186617.0
ANWUJOKX	0.0	0.0	0.0	63580.0	81670.0	26424.0	8470.0
APQGTRMF	383999.0	512813.0	217064.0	470398.0	539873.0	582317.0	759980.0
ATLDMDBO	143097177.0	103568323.0	118403737.0	121873741.0	115024617.0	65716075.0	146216818.0

- 피벗(pivot)을 이용해서, 데이터를 재구성

- Index : item_id
- Columns : ym
- Values : value(해당 품목의 그 달 무역량)

- Fillna(0,0) : 어떤 품목은 특정 달에 데이터가 없으면 NaN이 생기는데, 0으로 결측치 처리

- 피벗(pivot) : 표의 모양을 바꾸는 작업
  ex) 세로로 나열된 데이터를, 내가 원하는 축으로 가로x세로 매트릭스로 재배치하는 것

- 데이터의 의미를 변경하는 것이 아닌, 데이터의 모양(구조)를 변경하는 작업

Item_id	Year	Month	Value
A	2020	1	10
A	2020	2	15
B	2020	1	5
B	2020	2	7

Item_id	2020-01	2020-02
A	10	15
B	5	7