Wrap-Up Report

1. Team Wrap Up Report

1) 프로젝트 개요

프로젝트 주제

본 프로젝트는 **수도권 아파트 전세 실거래가를 예측**하기 위해 AI와 머신러닝 기술을 활용하여 가격 예측 모델을 개발하는 것을 목표로 한다. 한국 부동산 시장에서 전세가는 매매가와 밀접하게 연관되어 있어, 전세가 예측은 부동산 시장의 정보 비대칭성을 해소하는 데 중요한 역할을 한다.

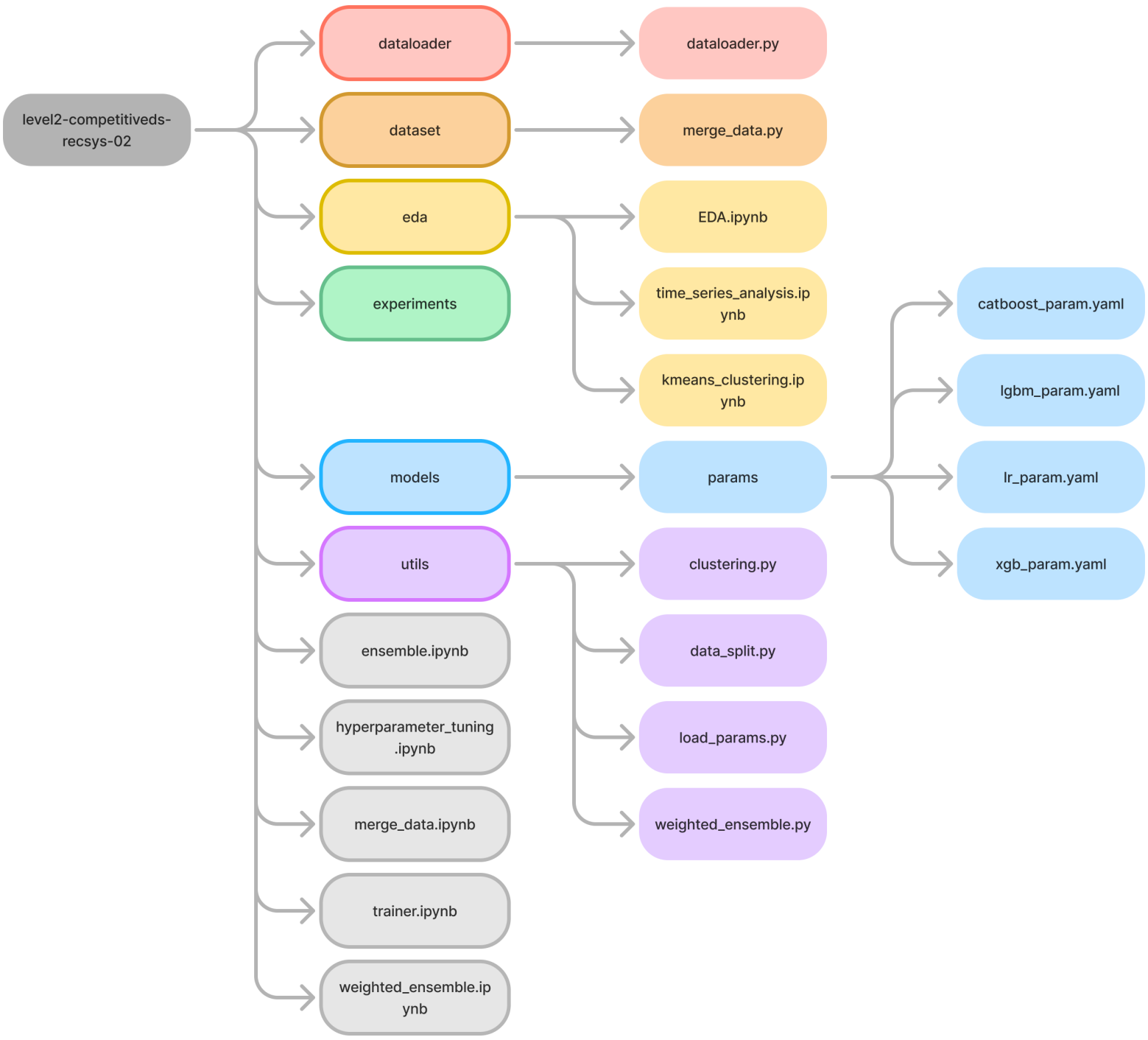
이를 위해 **약 120만 건의 수도권 아파트 전세 실거래 데이터**를 활용하여 모델을 훈련하였으며, 다양한 특성을 반영하여 전세가를 예측하고자 하였다. **Mean Absolute Error (MAE)**를 성능 평가 지표로 사용하였으며, 이 지표를 통해 모델의 예측 정확도를 개선하는 데 집중하였다.

특히, 여러 머신러닝 알고리즘과 특성 엔지니어링 기법을 적용하여 성능을 비교하고 최적의 방법론을 도출하는 과정을 거쳤다. 이 프로젝트는 부동산 데이터를 다루며 실무적인 인사이트를 도출하는 동시에, AI 모델링을 통해 예측 정확도를 높이는 것을 중점으로 삼았다.

활용 장비 및 재료

- 서버 스펙 : AI Stage GPU (Tesla V100)
- 협업 툴 : Github / Zoom / Slack / Notion / Google Drive
- 기술 스택 : Python / Scikit-Learn / Scikit-Optimize / Pandas / Numpy / PyTorch

프로젝트 구조도

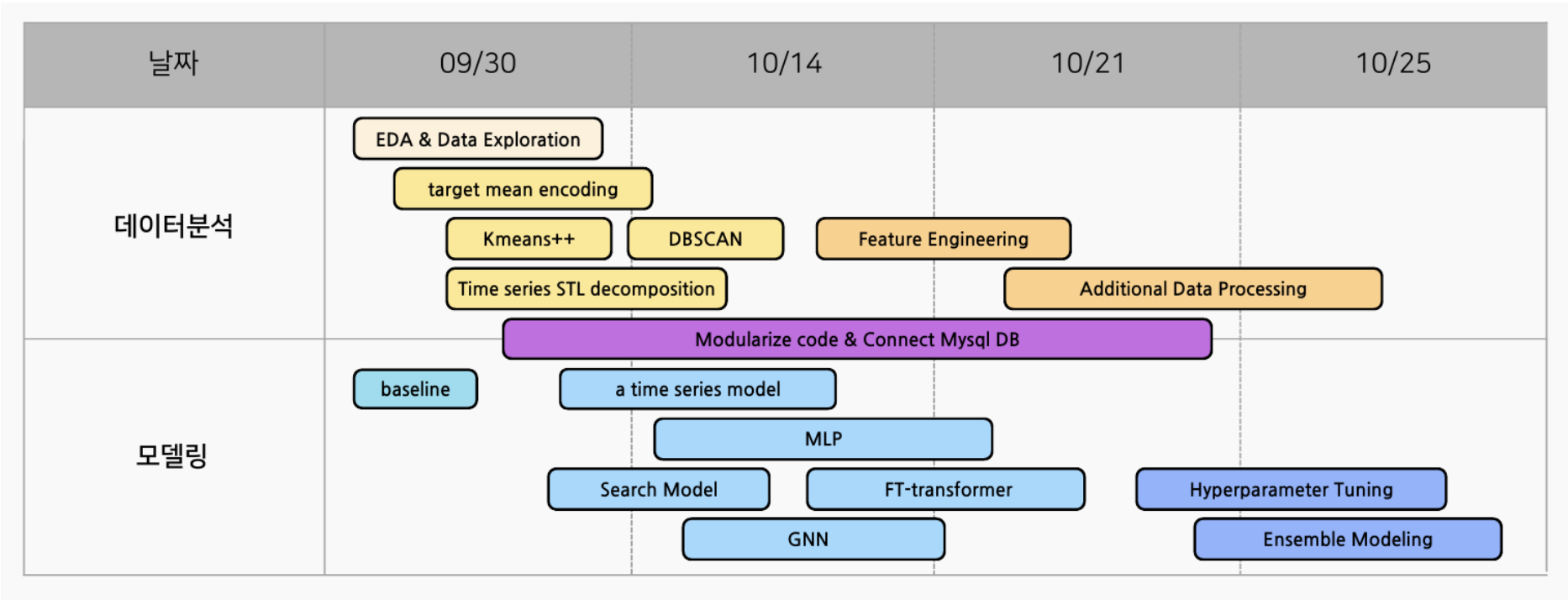


2) 프로젝트 팀 구성 및 역할

이름	역할
강현구	Data EDA, Hyperparameter tuning, KNN modeling, ensemble
서동준	data merging, feature creation, FT-transformer, retrieval modeling, Hyperparameter tuning, stacking
이도걸	Code modularization, Mysql DB connection, LGBM, XGB, RF modeling
이수미	Time series analysis, LSTM modeling, Feature selection work, clustering, data merging, ensemble
최윤혜	Deep learning modeling, MLP, GNN modeling, Added features to the dataset

3) 프로젝트 수행 절차 및 방법

수행 과정



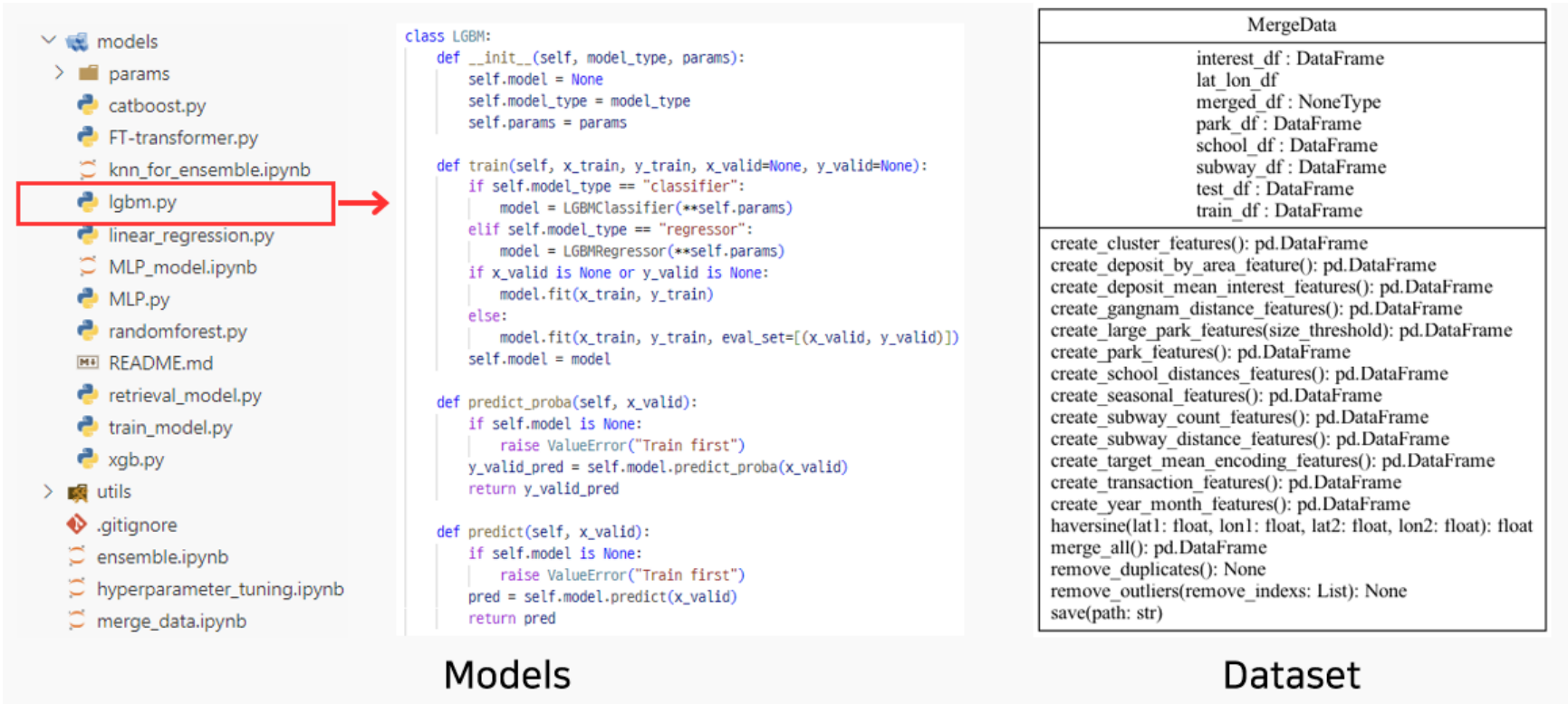
협업 문화

1. Github을 최대한 활용하자.
   - Discussion에 회의록 작성
   - 작성된 회의록에서 발생한 이슈나 시도 해볼 만 한 아이디어들을 Issue로 생성
   - Issue 번호와 이름을 기반으로 한 Branch 생성
   - Pull requests를 통한 코드 리뷰 및 진행 사항 파악

2. DB를 활용하자.
   - 과거 csv파일을 통한 실험 기록으로 팀원의 실험 결과를 확인하기가 어려웠다.
   - Main 서버에 MySQL DB를 생성하여 팀원의 실험을 실시간으로 저장, 확인이 가능하게 했다.

* date datetime	* user varchar(1	* save_name varchar(100)	MAE float	MSE float	RMSE float	R2 float	MSLE float	MAPE float	EVS float	leade float	params varchar(500)
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
2024-10-23 11:20:27	soomi	Catboost_real_final_df_time_	3794.07	45668500	6757.85	0.94	0.0224696	0.11	0.94	0	{'bagging_temperature': 0.0, 'bc
2024-10-23 11:22:06	soomi	RF_real_final_df_time_102311	4464.53	61938700	7870.11	0.92	0.0325038	0.13	0.92	0	{'n_estimators': 500, 'criterion': '
2024-10-23 11:56:43	soomi	LR_real_final_df_time_102311	4589.11	59486500	7712.75	0.93	0.0315697	0.13	0.93	0	{'fit_intercept': True, 'copy_X': Tr
2024-10-23 11:57:57	soomi	LR_real_final_df_time_102311	4402.69	56620700	7524.67	0.93	0.0294778	0.12	0.93	0	{'fit_intercept': True, 'copy_X': Tr
2024-10-23 12:20:07	hyeongu	Catboost_pure_time_102312	384.86	6470320	2543.68	0.99	0	0.01	0.99	0	{'bagging_temperature': 0.0, 'bc
2024-10-23 12:35:16	hyeongu	XGB_pure_time_10231235_cs	3884.26	46171700	6794.98	0.94	(NULL)	0.11	0.94	0	{'device': 'cuda', 'booster': 'gbtr
2024-10-23 13:13:53	hyeongu	LGBM_pure_time_10231313_.	3902.47	45114700	6716.75	0.94	(NULL)	0.11	0.95	0	{'colsample_bytree': 0.7562946
2024-10-23 13:15:07	hyeongu	Catboost_pure_time_102313'	3988.45	47426300	6886.68	0.94	(NULL)	0.11	0.94	0	{'bagging_temperature': 1.0, 'bc
2024-10-23 14:10:08	hyeongu	XGB_pure_time_10231410_cs	3785.47	45011400	6709.06	0.94	0.02	0.1	0.94	0	{'device': 'cuda', 'booster': 'gbtr
2024-10-23 14:20:13	soomi	LGBM_real_final_df_log_time_	3936.55	47088100	6862.08	0.94	0.024448	0.11	0.94	0	{'colsample_bytree': 0.6546472
2024-10-23 14:20:48	soomi	XGB_real_final_df_log_time_1	3795.97	45436200	6740.64	0.94	0.0220948	0.1	0.94	0	{'device': 'cuda', 'booster': 'gbtr
2024-10-23 14:21:45	soomi	Catboost_real_final_df_log_tir	3906.41	46513900	6820.11	0.94	0.0239625	0.11	0.94	0	{'bagging_temperature': 0.0, 'bc
2024-10-23 14:22:58	soomi	LR_real_final_df_log_time_102	4404.12	56657800	7527.14	0.93	0.0294632	0.12	0.93	0	{'fit_intercept': True, 'copy_X': Tr

3. 코드 모듈화
    - 코드 모듈화를 통해 객체의 결합도를 낮추고 응집도를 높게 만들자.
    - 가독성을 늘리고, 재사용이 가능하도록 하자.
    - 함수 명세서를 작성하여 코드를 이해하기 쉽게 하자.



4. Zoom을 활용한 커뮤니케이션
    - 정기적인 아이디어 회의를 진행하자.
    - 팀원 간 원활한 소통 및 의견 교환 촉진

4) 프로젝트 수행 결과

탐색적 데이터 분석 (EDA)

- **데이터 구성**
    - 거래 날짜, 위도, 경도, 전세가 등을 포함한 **수도권 전세 거래** 데이터와 **학교**, **지하철**, **공원**의 위도, 경도 데이터, **금리** 데이터로 구성
    - `train.csv`, `test.csv`, `subwayInfo.csv`, `schoolInfo.csv`, `parkInfo.csv`, `interestRate.csv`
    - 학습 데이터: 1,801,228 rows × 11 columns (2019-04-01 ~ 2023-12-31)
    - 테스트 데이터: 150,172 rows × 10 columns (2024-01-01 ~ 2024-06-31)

- **feature engineering**
    - 인프라 관련

        1. 공원
        - 반경 이내 공원 개수, 면적
        - 반경 이내 대형공원 개수, 면적

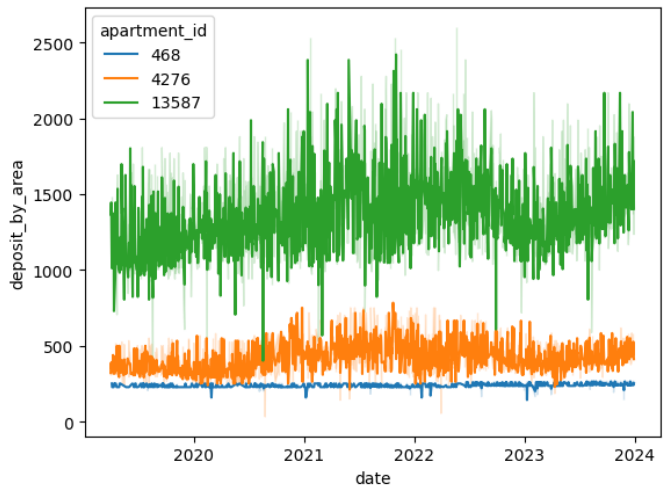
        2. 학교
        - 반경 이내 학교의 개수
        - 가장 가까운 학교까지의 거리

        3. 지하철
        - 반경 이내 지하철 역 개수
        - 가장 가까운 지하철 역까지의 거리

    - target 관련
        - target mean encoding

          아파트별로 전세가의 차이가 크고 동일 아파트 안에서는 전세가의 변동이 적은 편이다.

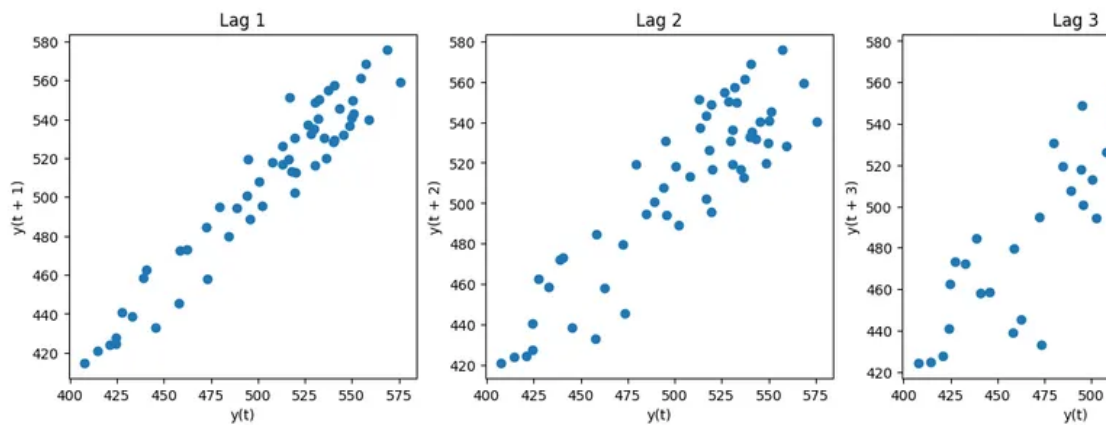
          → target mean encoding을 통해 아파트 특성 반영



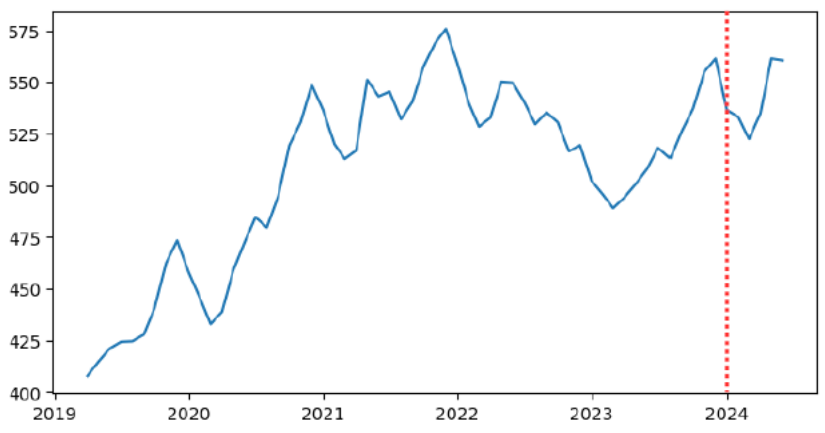
- 직전 거래가 변수

  각 아파트별로 직전 거래가와 현재 거래가의 상관관계가 높다.

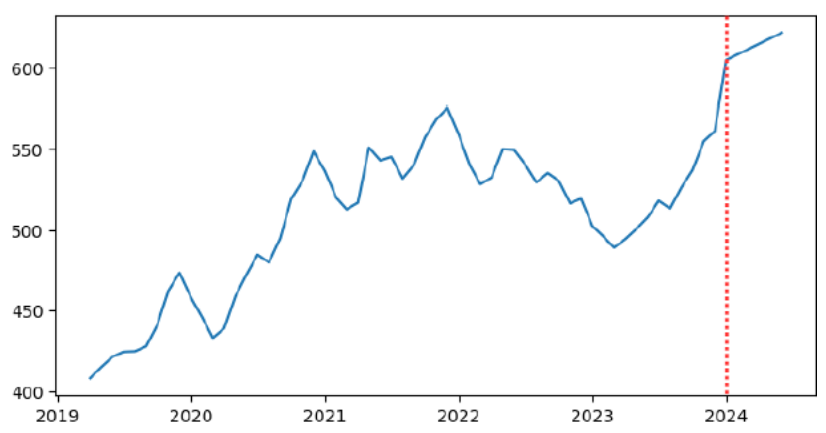
  → 직전 거래가 변수를 추가하여 시계열적 특성 반영



- deposit 경향성 변수
  - 월별 금리 데이터를 활용하여 prophet 모델로 월 평균 전세가 예측
  - 시각화를 통해 y_hat 값 대신 trend 값을 새로운 feature로 추가

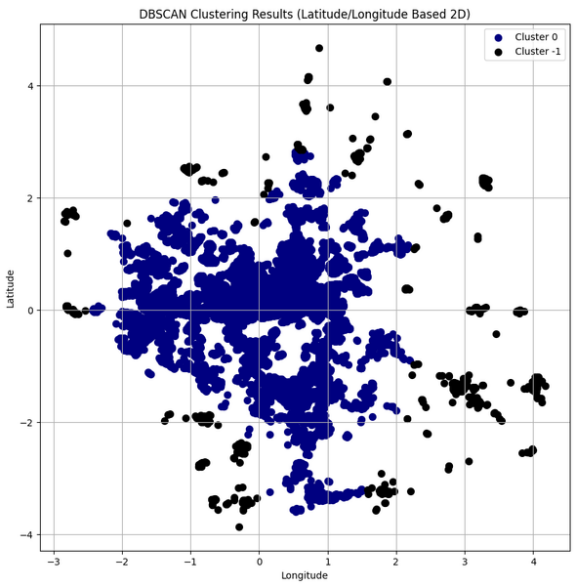


y_hat 그래프

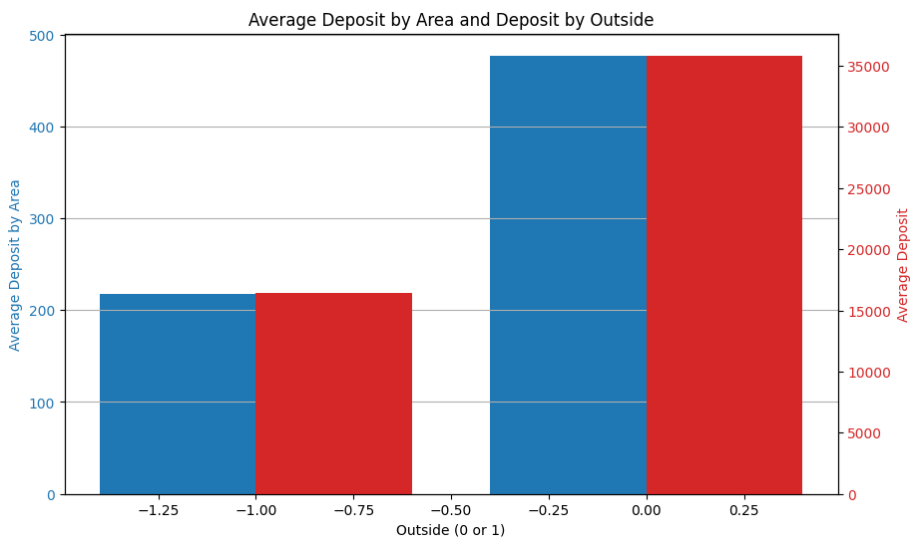


trend 그래프

  - 클러스터링
    - 외곽 여부
      - 위도, 경도를 기준으로 DBSCAN을 적용하여 이상치 (외곽) 분류
      - 이상치 (외곽)으로 분류된 경우 단위면적당 전세가가 낮게 나타남

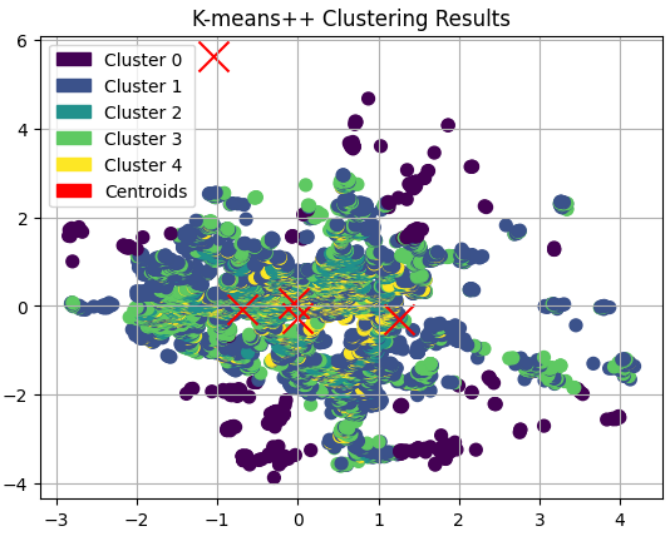


DBSCAN 클러스터링 결과 시각화

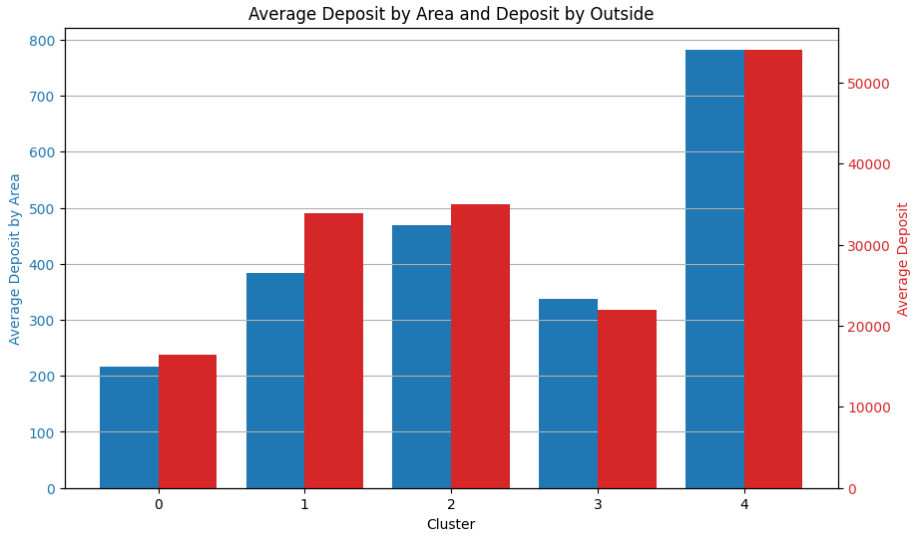


클러스터별 전세가 막대그래프

- 지역 클러스터
  - 일부 데이터 (단위면적당 전세가, 가장 가까운 지하철까지의 거리, 면적, 건축연도, 1km 내 학교 수)를 활용하여 Kmeans++ 클러스터링
  - 각 클러스터별로 전세가 차이가 극명하게 나타남



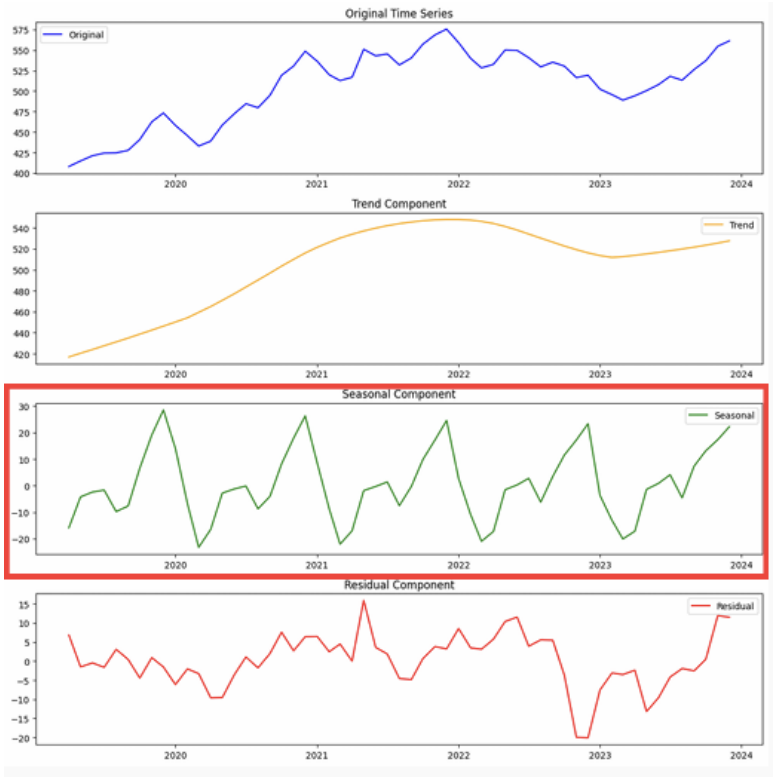
Kmeans++ 클러스터링 결과 시각화



클러스터별 전세가 막대그래프

- 시계열성
  - 시계열 STL 분해를 통해 계절성을 확인
  - sin/cos 변수로 주기성 반영

```
# 주기 : T
sin_value = sin(2 * pi * x / T)
cos_value = cos(2 * pi * x / T)
```



- **최종 데이터셋**

종류	column_name
Train features	'area_m2', 'contract_year_month', 'contract_day','contract_type', 'floor', 'built_year', 'latitude', 'longitude', 'age'
공원	'nearest_park_distance', 'park_count_500m', 'total_park_area_500m', 'park_count_1000m','total_park_area_1000m', 'park_count_2000m', 'total_park_area_2000m', 'weighted_park_score', 'avg_distance_5_parks', 'park_distance_skewness', 'park_distance_kurtosis', 'nearest_large_park_distance', 'large_park_count_3km', 'large_park_count_5km', 'large_park_count_10km', 'total_large_park_area_10km',
지하철	nearest_subway_distance_km', 'subways_within_500m', 'subways_within_1km', 'subways_within_2km'
학교	'school_count_within_1km', 'closest_elementary_distance', 'closest_middle_distance', 'closest_high_distance'
금리	'deposit_mean', 'interest_rate', 'interest_rate_diff'
클러스터링	'Is_Outside', 'cluster_kmeans'
주기성	'month_sin', 'month_cos'
기타	distance_from_gangnam'(강남역까지의 거리), 'apt_deposit_mean'(각 위도, 경도 별 직전 거래가)

모델 개요

- 기본 모델

모델명	특징
Boosting Model	LightGBM, XGBoost, CatBoost 모델
Retrieval Model	FAISS 라이브러리를 활용한 L2 거리 기반 유사도 검색 모델(prophet 을 통해 구한 월평균 값으로 가중 평균)
LSTM	시계열 모델
MLP	딥러닝 기반의 기본 모델로 baseline으로 사용
FT-Transformer	feature embedding과 Transformer를 활용한 딥러닝 모델
GNN	그래프 구조를 활용하여 공간적 정보와 네트워크 상의 관계를 학습하는 모델

- 앙상블 모델
  - Mean Ensemble : 각 모델의 예측값의 평균으로 최종 예측
  - Stacking : validation 으로 학습한 meta model로 stacking
  - 날짜별 가중치를 계산하여 최종 예측
    - 일별 평균 기반 : 각 모델의 일별 평균 예측값을 전체 모델 평균의 합으로 나누어 가중치 계산 $w_{ij} = \frac{m_i\text{의 j일자 예측값 평균}}{\sum_k m_k\text{의 j일자 예측값 평균}}$

    - 일별 변동성 기반 : 각 모델의 일별 변동성을 전체 모델 변동성의 합으로 나누어 가중치 계산 $w_{ij} = \frac{m_i\text{의 j일자 예측값 표준편차}}{\sum_k m_k\text{의 j일자 예측값 표준편차}}$

모델 선정

- 모델 선정 기준
    - bayesian optimization, Optuna를 사용하여 하이퍼파라미터 튜닝 진행
    - Validation MAE 3900 이하 모델에서 앙상블 실험
    - Mean Ensemble, Stacking, 일 평균 값으로 가중평균 앙상블, 일 표준편차 값으로 가중평균 앙상블 진행
    - public 리더보드 MAE가 가장 높은 앙상블 모델을 최종 모델로 선정

- 성능 평가

모델명	Validation MAE	public Leaderboard
Retrieval	4371.88	3966.46
MLP	4201.45	4787.51
FT-transformer	3805.89	3729.3
CatBoost	3696.32	3750.51
LightGBM	3688.53	4197.07
XGBoost	3676.56	3695.37
Mean Ensemble	3596.57	3579.91
Stacking	3594.51	**3586.08**
Weighted Ensemble	-	**3569.65**

- 최종 선정 모델: Stacking, Weighted Ensemble
    - 리더보드 결과: **Public Score: 4373.6757(6위), Private Score: 3572.0638(6위)**

자체 평가 의견

- **잘했던 점**
    - 깃허브 활용 능력이 이전 프로젝트에 비해 발전했음
    - DB를 도입하는 등 프로젝트를 보다 체계적으로 진행함
- **시도하였으나 잘 되지 않았던 점**
    - GNN 모델을 활용해보려 했으나 메모리의 한계로 중단함
- **아쉬운 점**
    - 하이퍼 파라미터 튜닝 범위를 프로젝트 마지막날 조정하여, 최적의 하이퍼 파라미터를 찾지 못함

2. 개인 회고

1) 강현구_T7502

초기 목표와 달성한 정도

전반적인 프로세스를 이해하고자 하였고, 다양한 부분을 건드릴 수 있었다.

내가 가장 신경 썼던 점

하이퍼 파라미터 튜닝 부분을 가장 많이 함. yaml 파일로 저장해서 모듈화 코드에서 쉽게 사용할 수 있게 하였다.

공원 데이터를 디테일하게 뜯어봤던 기억이 있다.

구현하지 못한 아이디어

KNN 을 활용해 예측하고자 하였으나 오래 걸림

한계와 아쉬웠던 점

EDA와 하이퍼파라미터 튜닝 부분에 집중을 하다보니, 딥러닝 모델을 사용해보지 못했다.

다음에 시도해볼 것

더 많은 방법 시도해보기

마지막 한마디

파이팅~

2) 서동준_T7527

초기 목표와 달성한 정도

- 협업한 내용을 github를 활용해 최대한 기록하고 공유하는 것을 목표로 했다. Discussion을 활용해 매일 회의 내용을 기록하고 공유했다. 또한 진행한 실험 내용은 ISSUE의 댓글을 활용해 공유했다.
- 데이터 분석을 더 치밀하게 하는 것을 목표로 했다. 이전 프로젝트보다 데이터 분석에 시간을 더 투자하기는 했지만, 최종 모델 학습을 위한 feature 선택에 대한 충분한 근거를 만들지 못해 아쉬웠다.

내가 가장 신경 썼던 점

- 이미 검증된 부스팅 모델 외의, 다른 모델들을 시도해 보는 것이 주안점이었다. 검색모델의 기본적인 원리와 FT-Transformer의 구조를 이해할 수 있었다. 하이퍼 파라미터 튜닝 이전에, 모델의 구조적인 부분에서 성능을 개선하려고 노력했다.
- 팀원간 지식 공유를 중요하게 생각했다. 팀원의 실험에서 이해가 되지 않았던 부분은 질문을 통해 이해하려고 노력했다. 또한 내가 짠 코드와, 진행한 실험들을 팀원에게 자세하게 설명하고 문서화했다. 설명 가능한 수준의 이해를 통해, 깊이 있는 학습이 가능했다.

구현하지 못한 아이디어

- 전세가 예측 Task에 적합한 딥러닝 모델 구조를 직접 설계해보는 방법을 시도했다. 시간, 공간 데이터가 특징인 이번 프로젝트에서, Transformer 구조가 시간적 자기상관과 공간적 자기상관 데이터를 학습하기에 유리하다고 판단했다. 그리고 시간에 따른 전체적인 가격 트렌드를 학습하는 구조를 자기상관 데이터로 학습한 Trend 레이어와 층수, 평수 등으로 학습한 residual 레이어로 분할하는 방법을 통해 적용해보았다. 하지만 낮은 성능과 시간적인 한계로 중단하고 FT-Transformer를 활용하는 방향으로 틀었다.

한계와 아쉬웠던 점

- 시간 관리 측면에서 아쉬움이 있다. 다양한 모델을 시도하면서 개선 가능한 아이디어가 떠오를때마다 시도를 했지만, 이 과정에서 우선순위 없이 좁은 시야로 프로젝트를 진행한 것 같다. 결과적으로 마지막 파라미터 튜닝에 많은 시간을 할애하지 못했다. 일정 관리에 좀 더 신경써서, 시야를 넓힐 필요가 있을 것 같다.
- 딥러닝 모델의 학습시간을 줄이기 위해 서버의 4개의 자원을 효율적으로 쓸 수 있는 방법에 대해 연구해봤지만, 배경지식이 부족해 시도에서 그쳤다. 딥러닝 모델을 필수적으로 도입해야되는 프로젝트를 진행하게 된다면, 꼭 다시 시도해보고 싶다.

다음에 시도해볼 것

- 2번의 프로젝트로 다진 기본기를 바탕으로, 논문도 적절히 활용해보고 싶다.
- 프로젝트 구조에 대해 한번 더 고민하는 시간을 가졌으면 한다. 실험에 적합한 디자인 패턴과, 모델 서빙에 적합한 디자인 패턴을 적절히 고려해서, 프로젝트 구조를 재정비하면 좋을 것 같다.
- Wandb나 mlfow같은 실험관리 툴을 활용해보고 싶다.

마지막 한마디

- 이전 프로젝트보다 더 친근한 도메인에서 진행된 프로젝트라, 다양한 가설을 세워볼 수 있어서 재밌었다.

3) 이도걸_T7540

초기 목표와 달성한 정도

- 지난 번 보다 발전한 프로젝트를 만들자.
  - 지난 번 프로젝트도 만족스러웠지만, 아쉬웠던 부분들도 많았다.
  - 보다 올라간 팀워크로 더 개선된 프로젝트를 만들기 위해 노력했다.
- 최종 모델 선택에서 가장 Best인 모델을 선택할 수 있도록하자.
  - 지난 프로젝트에서 public score에 overfitting된 모델을 최종 제출하여 실제 score가 많이 떨어졌던 경험이 있었다.
  - 이번 프로젝트에서는 다양한 평가 지표를 사용하여 조금 더 객관적으로 성능이 좋은 모델을 최종 제출했고, 결과도 public score와 비슷하게 나왔다.

내가 가장 신경 썼던 점

- 지난 프로젝트의 피드백을 반영한 코드 모듈화
  - 조금 더 재사용성이 높은 코드를 작성하기 위해 노력했다.
- 시간 배분을 잘 하여 프로젝트의 완성도를 높이자.
  - 지난 프로젝트에서 하이퍼 파라미터 튜닝과 앙상블에 투자할 시간이 모자랐던 것이 아쉬웠다.
  - 이번 프로젝트에서는 시간 배분을 잘하여 튜닝과 앙상블을 통해 성능을 높일 수 있었다.

구현하지 못한 아이디어

- 외부 데이터 중 각 행정구역을 나눠주는 데이터를 쓴다면 결과가 얼마나 개선될까?
  - 대회의 룰에는 외부 데이터 사용을 금지하지만 실제 서비스로 개발한다고 생각했을 때, 어떤 방법과 데이터를 활용해야 좋은 결과를 얻을 수 있을지 궁금했다.
  - 실제 점수는 얼마나 영향을 받을지 궁금했다.

한계와 아쉬웠던 점

- 생성된 Feature로 다양한 조합을 시도해보지 못한 점이 아쉽다.

다음에 시도해볼 것

- 최종 프로젝트를 위한 초석 쌓기
- 새로운 모델 사용해보기

마지막 한마디

내 집 마련은 언제쯤.... 계속 오르기만 하네... ㅠㅠ

4) 이수미_T7541

초기 목표와 달성한 정도

- 항상 근거를 가지고 모든 단계를 나아가고자 하였으나 마음처럼 되지는 않았다... 그래도 가설 세우고 검증을 했던 부분이 있기 때문에 더 발전해나갈 수 있을 것 같다.

내가 가장 신경 썼던 점

- 팀 회의에서 나왔던 이야기나 코드, 그리고 내가 실험했던 내용들을 모두 GitHub의 Discussion이나 Issue에 기록하고자 노력해왔다.

구현하지 못한 아이디어

- 시계열 모델과 데이터가 부합하지 않는다고 생각하여 사용하지 않았는데, 다른 방식으로 시계열 데이터임을 강조할 수 있는 방법이 있다면 구현해보고 싶다.

한계와 아쉬웠던 점

- 어떤 변수가 얼만큼 중요한가? 라는 질문에 대답을 하지 못하였다. 분석이 부족했다고 생각한다.
- 군집화를 할 때도 모두 모델에 넣어보고 판단해봤으면 더 나은 결과가 나왔을 수 있을 것 같아 아쉽다.

- 앙상블 할 때 명확한 근거를 가지고 다양한 시도를 해봤다면 더 만족스러운 결과가 나왔을 것 같다.

다음에 시도해볼 것

- 앙상블과 파라미터 튜닝 시간이 생각보다 부족했다. 다른 틀을 다 짜고 하는 것도 좋지만, 구현과 실험 등으로 역할을 나눠서 미리미리 시작해야할 것 같다. 또한, 지금까지의 프로젝트는 일단 하자~라는 생각으로 진행해왔지만 초기 단계부터 일정을 잡아놓고 가도 좋을 것 같다.
- 이번 프로젝트에서 물론 시계열 데이터 분석을 해보긴 했지만, 내가 이미 알고 있는 내용만을 활용한 느낌이 들어 다음 프로젝트에서는 적어도 한가지 이상의 새로운 시도를 해보겠다!

마지막 한마디

내 집 마련의 꿈... 이뤄지길...

5) 최윤혜_T7549

초기 목표와 달성한 정도

- 새로운 모델 시도해보자.
  - 지난번 프로젝트와 다르게 이미 해봤던 모델들 말고 새로운 모델들을 시도해보고자 했다. 그래서 딥러닝 모델 중 MLP와 GNN 모델을 구현해봤다. 강의에서 배웠던 내용을 기반으로 MLP 모델을 구현하였고, GNN의 경우 따로 모델에 대해 공부하면서 현재 데이터에 맞게 구현하려고 노력했다. 두 모델 모두 최종 성능은 좋지 못했지만 대충 모델을 구현해보고 성능이 별로라고 넘어가기 보다는 좋아질 수 있는 방법들을 열심히 탐색하고 적용해봤다는 점에서 나름대로 목표를 달성했다고 생각한다.
- github 최대한 활용해보자.
  - 카톡과 노션으로 내용을 공유하던 기존 방식에서 모든 기록을 github를 활용하는 방식으로 바꾸고자 했다. 덕분에 이번 프로젝트에서는 discussion, issue, pull request 등 github의 기능을 최대한 잘 활용하여 협업한 것 같다.

내가 가장 신경 썼던 점

- 딥러닝 모델 구현
  - MLP의 경우 모델 구조를 자유롭게 설정할 수 있는 만큼 높은 성능을 내기 위해 튜닝해야 하는 부분이 많았다. layer 개수나 과적합 방지 방법 등 어떤 것을 선택하는지에 따라 결과가 크게 달라졌다. 학습 결과들을 보며 원인을 파악하고 적합한 해결 방안을 적용해보며 어느 정도 성능을 올릴 수 있었다. 이 과정에서 딥러닝 모델에 대한 이해도가 높아진 것을 느꼈다.
  - 아파트의 공간적인 상관관계를 반영하기 위해 GNN을 시도해봤다. 아쉽게도 데이터가 너무 큰 탓에 메모리 부족으로 제대로 된 실험은 해보지 못했다. 그래도 메모리가 터지지 않고 돌아갈 수 있도록 랜덤 샘플링, 클러스터별 적합, 아파트별 데이터 생성 등 데이터를 줄일 수 있는 방법들을 계속 시도해봤고, 최종적으로 모델 학습까지는 성공할 수 있었다. 가장 시간을 많이 쏟은 모델이었으나 성능이 좋지 않아 아쉬웠다. 그래도 끝까지 도전해봤다는 점에 의의를 둔다.

구현하지 못한 아이디어

- 시계열 클러스터
  - 전세가를 기준으로 클러스터링 해서 전세가의 상승 하락 경향성이 유사한 아파트 그룹을 묶고 각 클러스터별로 모델 학습을 했다면 어땠을까? 시간이 없어서 시계열 클러스터는 따로 시도해보지 못했지만 kmeans 같은 일반적인 클러스터보다는 시계열적인 특성을 잘 반영했을 것 같다.

한계와 아쉬웠던 점

- 변수 먼저 만들고 모델 돌려볼걸
  - 모델 구현에 집중하다 보니 머릿속으로 적용해봐야지 생각했던 변수들을 뒤늦게 반영하게 되었다. 초반에 만들 수 있는 변수들을 최대한 만들어 놓고 모델 실험을 했다면 더 결과가 좋지 않았을까 하는 아쉬움이 남는다. 이번에도 유의한 변수 한 두개가 성능을 높이는 중요한 역할이었던 만큼 변수 생성을 가장 먼저 시도해봐야겠다.
- 컨디션 관리 실패
  - 개인적인 일정과 프로젝트 기간이 너무 겹치는 바람에 거의 쉬지 못한 것 같다. 프로젝트 후반부로 갈수록 집중력도 떨어지고 에너지도 고갈되는 느낌이었다..

다음에 시도해볼 것

- 앙상블, 튜닝 과정에 적극 참여하기
  - optuna나 wandb 처럼 튜닝 관련된 툴을 거의 써보지 못했다. 다음에는 하이퍼 파라미터 튜닝에도 적극 참여해야겠다.
  - 앙상블에도 생각보다 다양한 방식이 있고, 효과가 좋다는 것을 이번에 깨닫게 됐다. 앙상블에도 좀 더 시간을 투자해봐야겠다.

마지막 한마디

프로젝트 할 때마다 로제 APT. 들었더니 중독됐다. 이제 노래 들을 때마다 이 프로젝트 생각날 것 같다.