디지털 보물찾기 팀의 아파트 실거래가 예측 도전기

'디지털 보물찾기(Digital Treasure Quest)' 팀이 '아파트 실거래가 예측' 경연에 도전한 여정을 소개합니다. 저희 팀은 다양한 경력과 경험을 가진 5명의 멤버로 구성되어 있으며, AI 기술을 활용하여 서울시 아파트 실거래가를 예측하는 모델을 개발하는 과제에 도전했습니다. 저희 들의 도전 과정과 결과, 그리고 그 과정에서 얻은 교훈을 함께 살펴보겠습니다.

Team #3 :: 디지털 보물찾기 (Digital Treasure Quest)

박석, 백경탁, 한아름, 이승현, 이한국

팀 구성 및 특징

'디지털 보물찾기' 팀은 다양한 배경을 가진 5명의 멤버로 구성되어 있습니다. 팀의 주요 특징으로는 다양한 경력과 경험을 가진 팀원들, 평균 나이가 높다는 점, 그리고 AI Assistant에 대한 수용력이 높다는 점을 들 수 있습니다. 반면, Git을 활용한 팀 단위의 R&D 경험 수준이 낮고, Python 기반 R&D 경험 수준이 낮으며, 머신러닝/딥러닝 R&D 경험 수준이 낮다는 단점도 있었습니다. 또한 경연 주제와 관련된 도메인 지식이 낮고, 각 팀원의 Career Path에 대한 개인적인 목표가 모두 달랐습니다.

다양성

다양한 경력과 경험을 가진 팀원 구성

AI 수용성

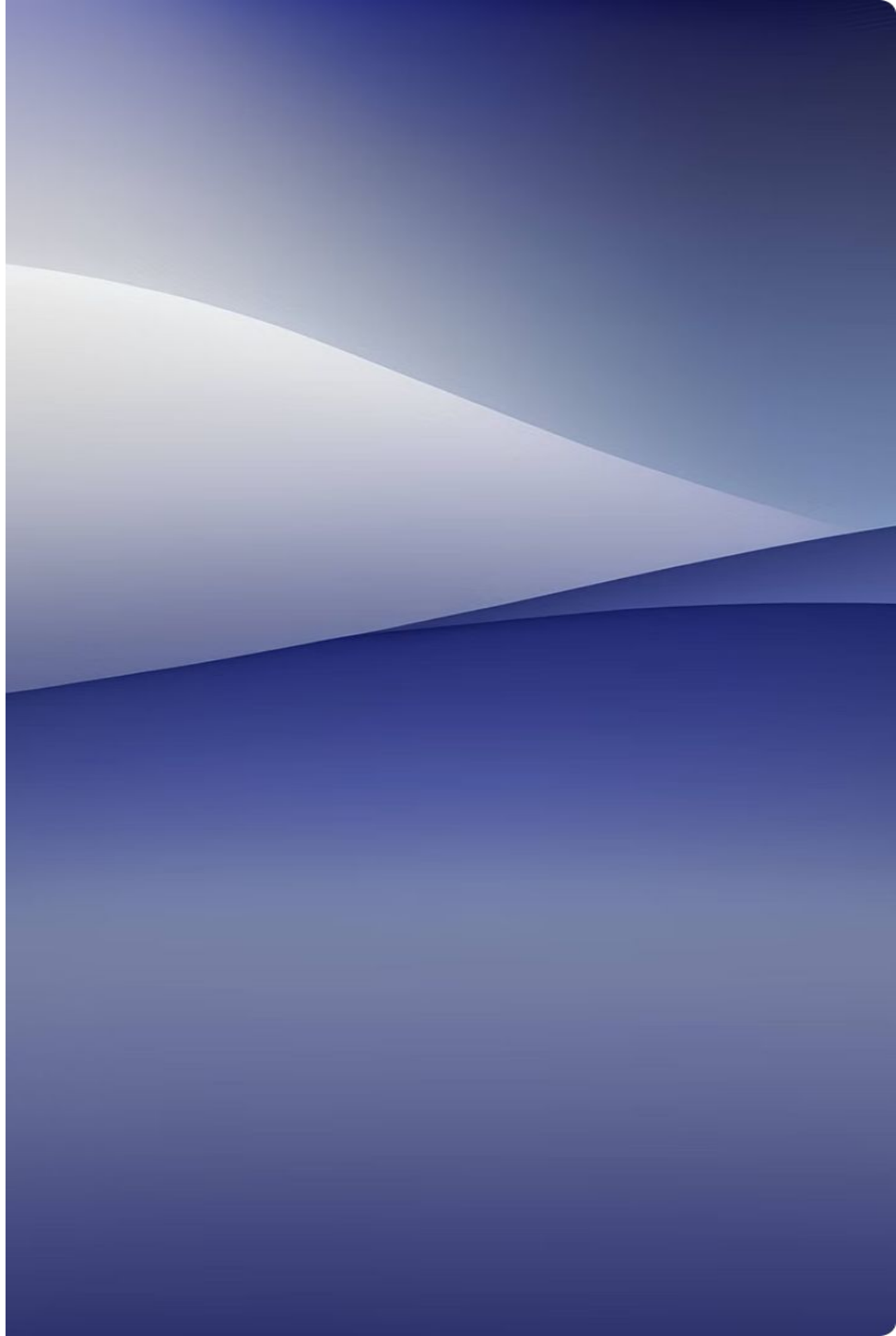
AI Assistant에 대한 높은 수용력

경험

평균 나이가 높음

도전 과제

R&D 경험과 도메인 지식 부족



팀의 전략적 접근

'디지털 보물찾기' 팀은 자신들의 장단점을 고려하여 두 가지 주요 전략을 세웠습니다. 첫째, DataRobot과 같은 AutoML 도구를 적극 활용하여 Feature Engineering과 Model Selection의 방향성을 잡는 것입니다. 이는 팀의 머신러닝 경험 부족을 보완하고 효율적인 모델 개발을 가능하게 하는 전략이었습니다. 둘째, 팀원별로 서로 다른 머신러닝 모델링을 각자의 수준에 맞게 진행하는 것입니다. 이 접근 방식은 팀원 각자의 강점을 살리고 다양한 모델을 실험해볼 수 있는 기회를 제공했습니다.

1

AutoML 활용

DataRobot을 사용하여 Feature Engineering과 Model Selection 방향성 설정

2

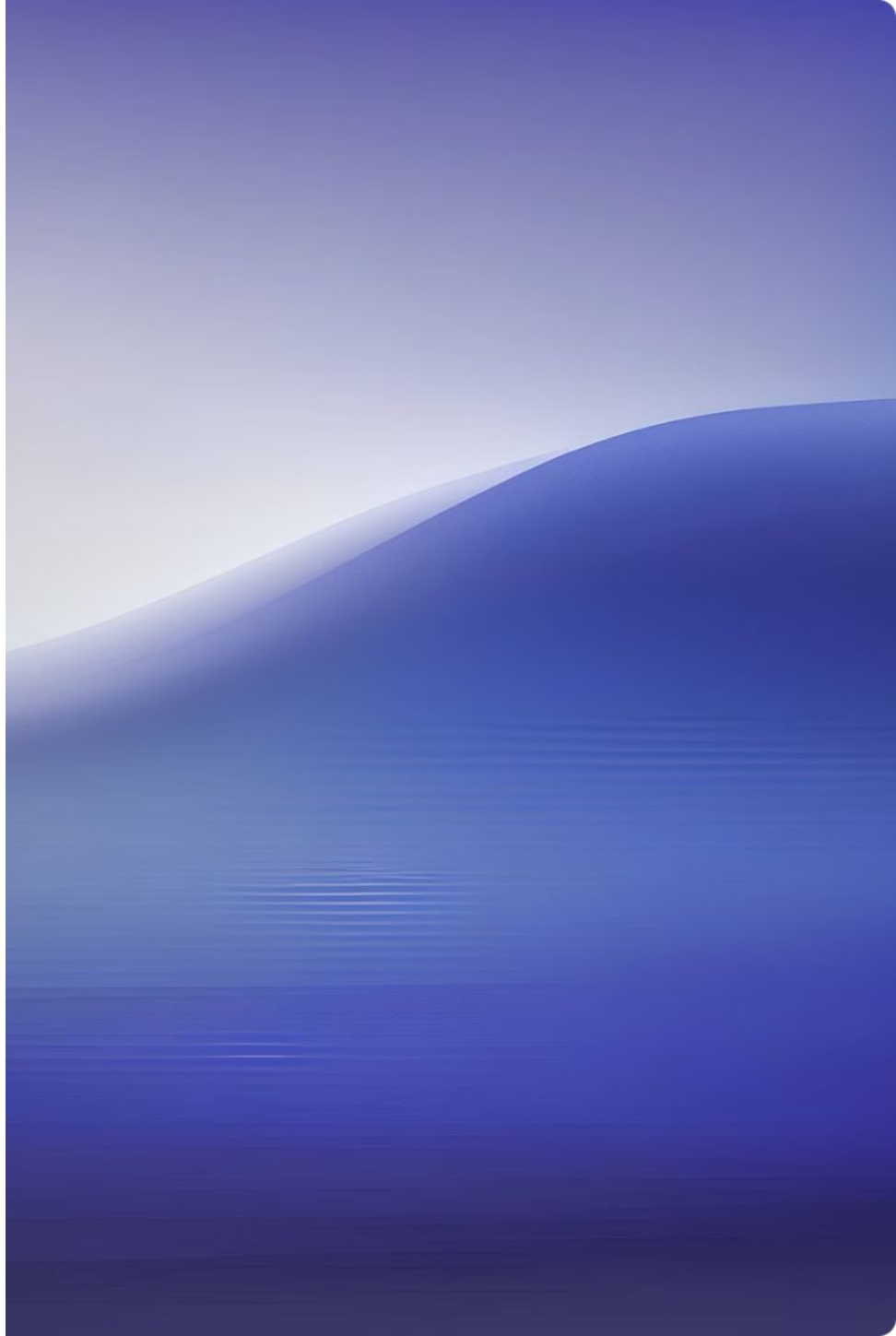
개별 모델링

팀원별 수준에 맞는 서로 다른 머신러닝 모델링 진행

3

결과 분석

다양한 모델의 성능 비교 및 최적 모델 선정



팀 문화와 정신

'디지털 보물찾기' 팀은 경연 참가의 목적을 개인별 학습을 통해 머신러닝 R&D에 필요한 지식과 경험을 얻는 것에 두었습니다. 팀원 각자가 처한 상황을 서로 이해하고 인정하며 존중하는 문화를 만들었습니다. AI Assistant를 적극적으로 활용하여 개인별 생산성을 극대화하는 것도 중요한 원칙이었습니다. 팀 전체 목표를 위해 팀원 개개인의 스케줄이나 리소스를 희생하지 않도록 했으며, 각

학습 중심

개인별 학습을 통한 머신러닝 R&D 지식과 경험 획득 목표

상호 존중

팀원 각자의 상황을 이해하고 인정하는 문화

AI 활용

AI Assistant를 적극 활용한 개인별 생산성 극대화

균형 잡힌 참여

팀원 각자 최소 1회 제출 목표, 개인 리소스 존중

대회 개요

대회의 목표는 서울시 아파트 실거래가 예측 모델 개발이었습니다. 참가자들은 다양한 요인을 고려해 정확한 시세를 예측해야 했습니다.

1 **목표**

서울시 아파트 실거래가 예측 모델 개발

2 **데이터**

아파트 실거래가, 지하철역, 버스정류장 정보 제공

3 **평가**

RMSE(Root Mean Squared Error) 사용

대회 일정

대회는 7월 9일부터 19일까지 진행되었습니다. 팀 병합, 개발, 최종 모델 제출 등 주요 일정이 있었습니다.

1

대회 시작

7월 9일 (화) 10:00

2

팀 병합 마감

7월 10일 (수) 10:00

3

개발 및 테스트

7월 9일 ~ 7월 18일

4

최종 모델 제출

7월 19일 (금) 19:00

평가 방법

RMSE(Root Mean Squared Error)를 평가지표로 사용했습니다. 예측된 값과 실제 값 간의 평균편차를 측정합니다.

RMSE 의미

예측 오차의 제곱 평균의 제곱근

낮은 RMSE

모델의 예측 성능이 우수함을 의미

맥락

실제 거래 가격과의 일치도를 정량적으로 나타냄

프로젝트 구조

프로젝트는 code, docs, images 디렉토리로 구성되었습니다. 각 팀원의 실험 코드와 문서가 포함되었습니다.

code

팀원별 실험 소스 코드 및 관련 문서

docs

팀 문서(발표자료, 참고자료 등)

images

첨부 이미지

README.md

프로젝트 개요 및 설명

데이터셋 개요

학습데이터와 예측데이터로 구성되었습니다. 각각 다양한 특성을 가진 데이터셋이었습니다.

구분	학습데이터	예측데이터
파일명	train.csv	test.csv
행 수	1,118,822	9,272
특성 수	52	51
크기	244 MB	2.46 MB

데이터 전처리 과정

데이터 전처리 과정에서는 여러 가지 기법이 사용되었습니다. 먼저, 결측치 처리를 위해 수치형 데이터는 중앙값으로, 범주형 데이터는 최빈값으로 대체했습니다. 이상치 제거를 위해 Z-점수를 사용했으며, 특정 기간 내에서 이상치를 판단했습니다. 텍스트 데이터 처리를 위해 TfidfVectorizer를 사용하여 '시군구'와 '아파트명' 컬럼을 벡터화했습니다. 또한, 계약 날짜를 연도와 월로 분리하는 등의 추가적인 피처 엔지니어링을 수행했습니다.

1

결측치 처리

수치형 데이터는 중앙값, 범주형 데이터는 최빈값으로 대체

2

이상치 제거

Z-점수를 사용하여 특정 기간 내 이상치 판단 및 제거

3

텍스트 데이터 처리

TfidfVectorizer를 사용하여 '시군구'와 '아파트명' 컬럼 벡터화

4

추가 피처 엔지니어링

계약 날짜를 연도와 월로 분리 등

EDA - 피쳐 설명

데이터셋의 주요 피쳐들을 분석했습니다. 각 피쳐의 유형, 고유값 수, 결측치 등을 확인했습니다.

특성명	유형	고유값	결측치
target	Numeric	12,875	0
도로명	Text	9,195	973
아파트명	Text	6,522	1,712
전용면적(㎡)	Numeric	14,218	0

EDA - Excess zeros

5개 변수에서 과도한 0값이 발견되어 제거되었습니다. 이는 데이터 품질 향상에 도움이 되었습니다.

1 **제거된 변수**

k_85㎡135㎡이하, 건축면적, k_전용면적별세대현황(60㎡85㎡이하), k_전용면적별세대현황(60㎡이하), 부번

2 **목적**

데이터 품질 향상 및 모델 성능 개선

3 **영향**

불필요한 노이즈 제거로 모델 정확도 향상 기대

EDA - Outlier

15개 변수에서 이상치가 발견되어 제거되었습니다. 이는 모델의 안정성을 높이는 데 기여했습니다.

1

대상 변수

target, 전용면적(㎡),
k_연면적, k_주거전용면적
등 15개 변수

2

처리 방법

통계적 방법으로 이상치
식별 후 제거

3

기대 효과

모델의 안정성 향상 및 예측
정확도 개선

EDA - 위장된 결측치

위장된 결측치는 발견되지 않았습니다. 이는 데이터의 품질이 양호함을 시사합니다.

위장된 결측치

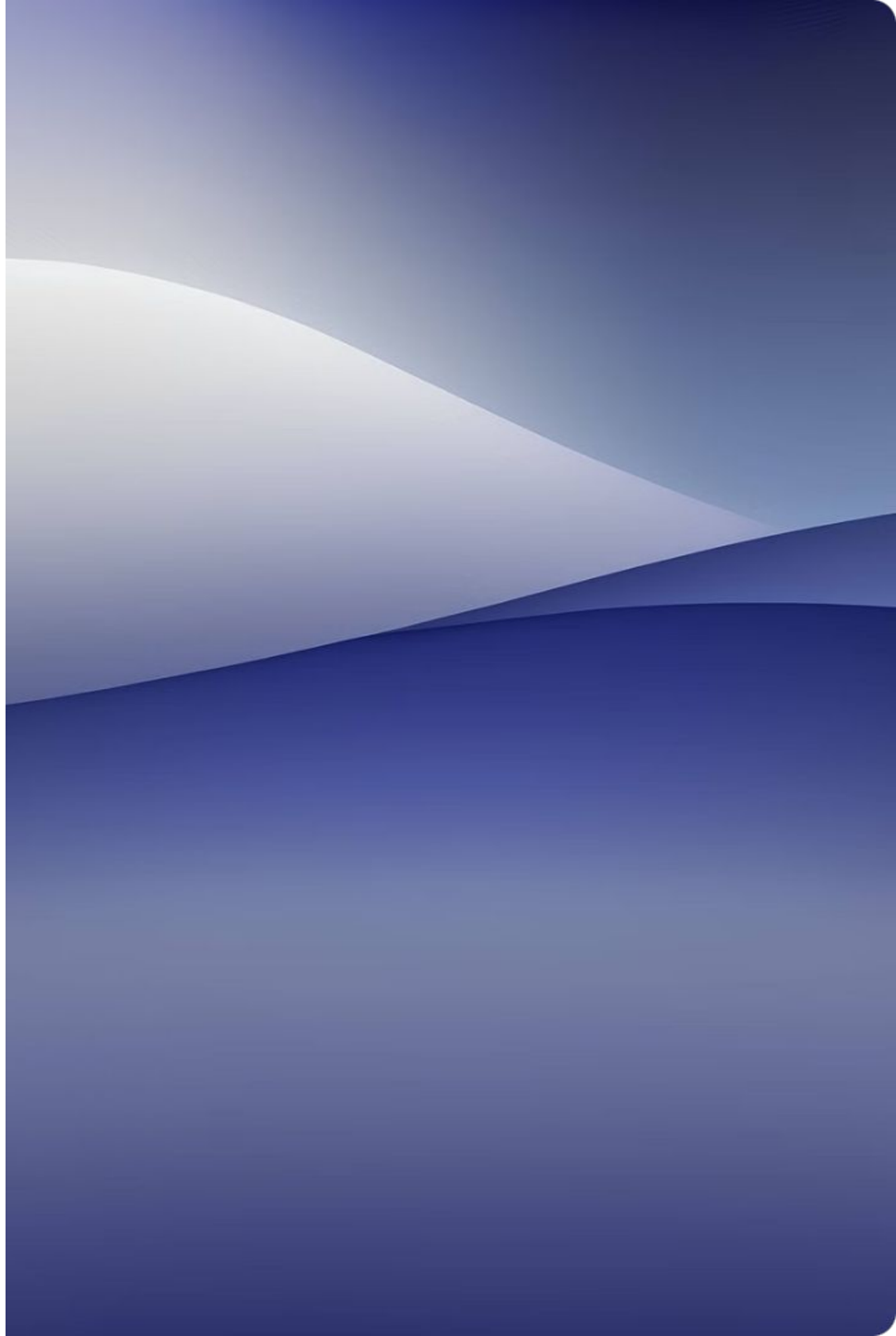
발견되지 않음

의미

데이터 품질이 양호함을 시사

영향

추가적인 데이터 정제 작업 불필요



EDA - Inlier

Inlier 발생 여부는 도메인 지식 부족으로 판단하지 못했습니다.

이는 향후 개선이 필요한 부분입니다.

1

내부값 판단

도메인 지식 부족으로 판단 불가

2

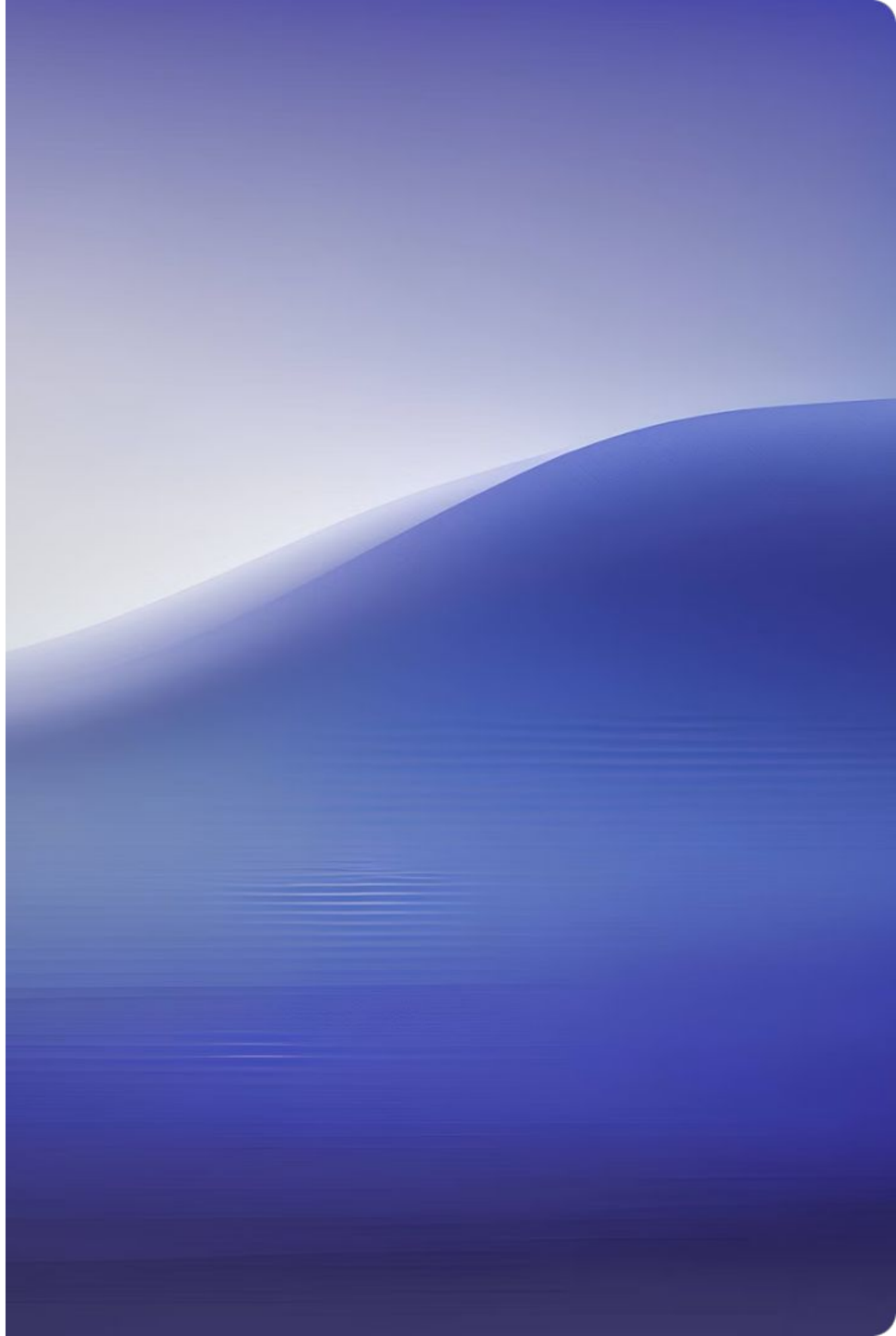
한계점

데이터의 깊이 있는 이해 부족

3

개선 방향

도메인 전문가와의 협업 필요성 인식



EDA - 타겟 유출

첫 번째 EDA 과정에서 타겟 유출 현상은 발견되지 않았습니다. 이는 모델 학습의 공정성을 보장합니다.

타겟 유출

발견되지 않음

의의

모델 학습의 공정성 보장

영향

신뢰할 수 있는 모델 개발 가능

피쳐 공학 - 방법론

5가지 주요 방법을 고려했습니다. 각 방법은 데이터의 특성에 맞게 적용되었습니다.

1 **One-Hot Encoding**

범주형 데이터를 이진 벡터로 변환

2 **Missing Values Imputed**

결측치를 중앙값으로 대체

3 **Smooth Ridit Transform**

연속형 변수를 순위 기반 점수로 변환

4 **Binning**

연속형 변수를 범주형으로 변환

5 **TF-IDF**

텍스트 데이터를 수치형 피쳐로 변환

피쳐 선택 - 방법론

특성 선택 시 합리성을 기준으로 했습니다. 정보가 낮은 특성들을 제외했습니다.

1

제외 기준

대부분 1 또는 0인 특성, 중복된 특성, 결측치가 많은 특성

2

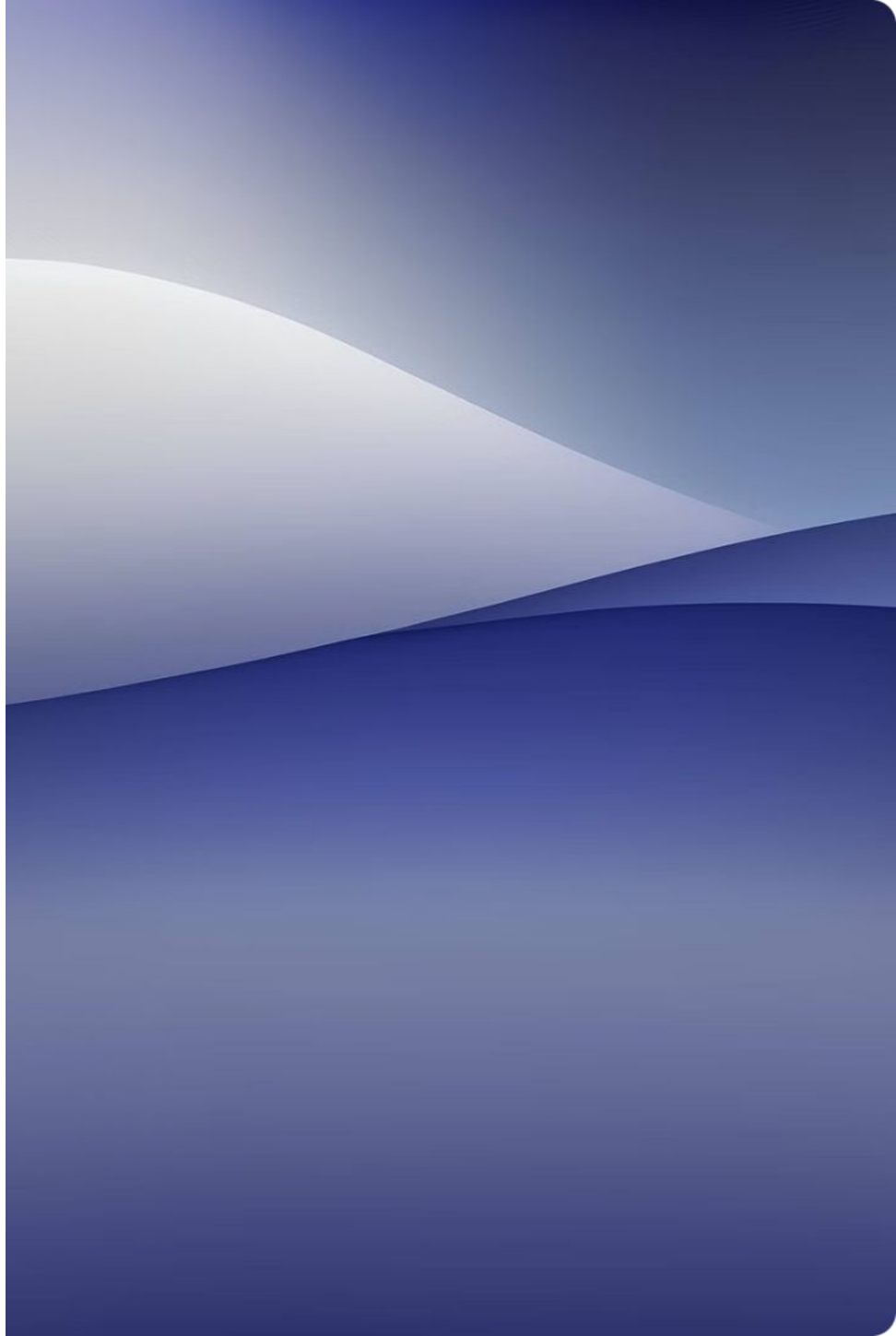
선택 도구

Feature Association Matrix 활용

3

고려 사항

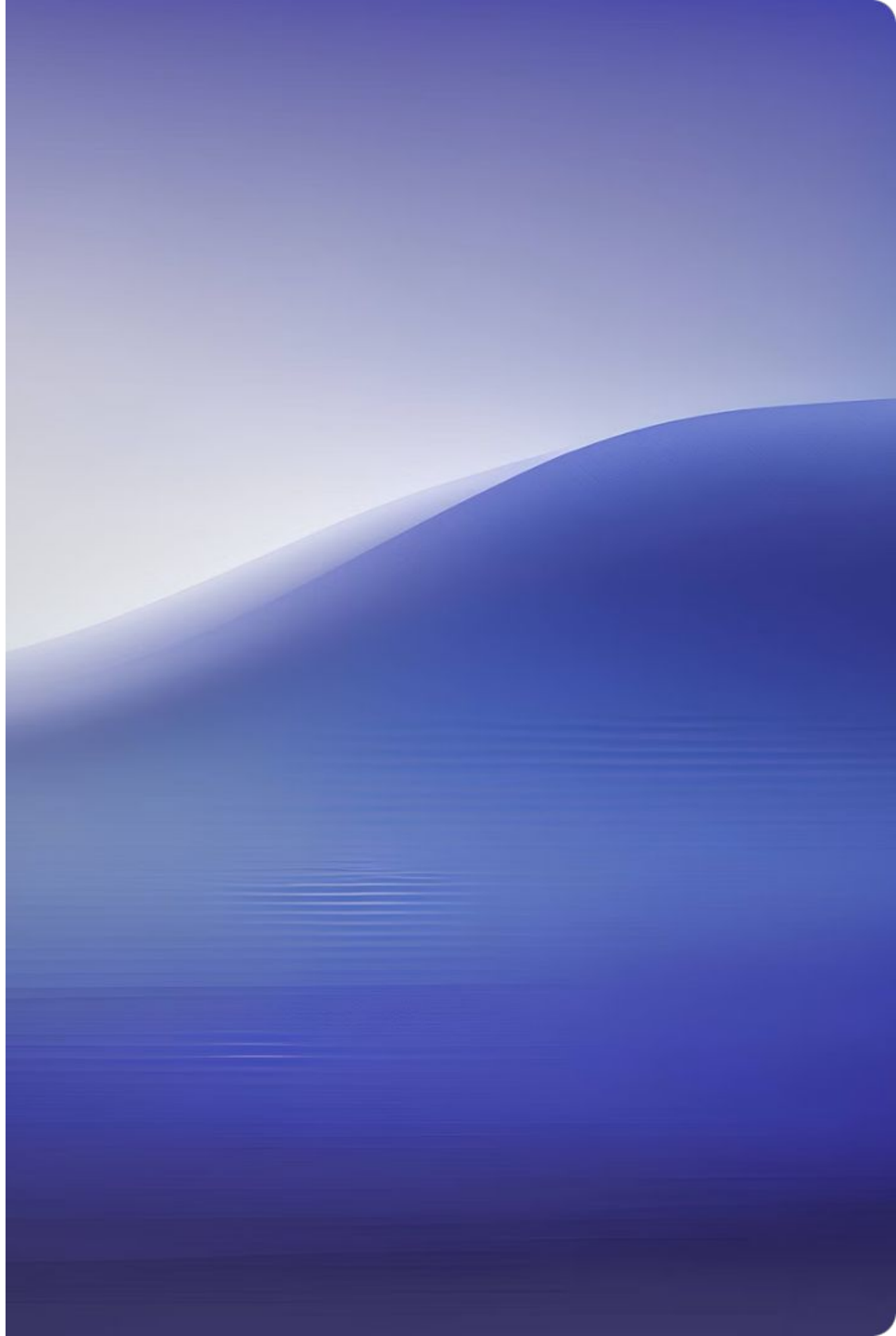
타겟 변수와의 상관관계, Feature Importance



피쳐 선택 - 결과

최종적으로 7개의 입력 특성이 선택되었습니다. 이들은 모델 학습에 중요한 역할을 했습니다.

Feature Name	Var Type	Unique	Missing
시군구	Text	338	0
번지	Categorical	6,555	184
아파트명	Text	6,522	1,712
전용면적(㎡)	Numeric	14,218	0
계약년월	Numeric	198	0
건축년도	Numeric	60	0
target	Numeric	12,875	0



타겟 특성 분석

타겟 특성인 'target'은 편향된 분포를 보였습니다. 이를 위해 정규화 방법을 고려했습니다.

타겟 특성

'target' - 아파트 실거래가

분포 특성

오른쪽으로 편향된 분포

처리 방법

로그 변환 또는 정규화 고려

피처 선택 과정

피처 선택 과정에서는 합리성을 기준으로 중요한 피처들을 선별했습니다. 대부분의 값이 1이나 0으로 구성된 피처, 중복된 피처, 결측치가 많은 피처 등 정보가 낮다고 판단되는 피처들을 제외했습니다. Feature Association Matrix를 활용하여 피처 간의 연관성을 시각화하고 분석했으며, 이를 통해 피처 클러스터를 식별했습니다. 또한, 타겟 변수와의 상관관계를 고려하여 Feature Importance를 분석했습니다. 이러한 과정을 통해 최종적으로 선택된 입력 피처들은 모델링에 사용되었습니다.

1 **합리성 기준 선별**

정보가 낮은 피처 제외 (대부분 1 또는 0인 피처, 중복 피처, 결측치 많은 피처 등)

2 **Feature Association Matrix 활용**

피처 간 연관성 시각화 및 분석, 피처 클러스터 식별

3 **Feature Importance 분석**

타겟 변수와의 상관관계 고려하여 중요 피처 선정

4 **최종 입력 피처 선택**

분석 결과를 바탕으로 모델링에 사용할 최종 피처 선정

모델 선택 과정

모델 선택 과정에서는 DataRobot을 활용하여 다양한 모델을 비교 분석했습니다. 이 과정에서 모델의 과적합을 방지하기 위해 표준 모델링 기술을 사용했으며, k-겹 교차 검증 프레임워크를 통해 모델 성능의 샘플 외 안정성을 테스트했습니다. 또한, 홀드아웃 샘플을 사용하여 추가적인 샘플 외 모델 성능 테스트를 진행했습니다. 데이터 분할 방법으로는 무작위 샘플링과 시계열적 경향을 고려한 분할 방법을 모두 시도했습니다. 최종적으로 DataRobot의 Leaderboard 상위권에 속한 3개의 머신러닝 모델을 선택했습니다.

모델 비교 분석

DataRobot을 활용한 다양한 모델 성능 비교

과적합 방지

k-겹 교차 검증 및 홀드아웃 샘플 사용

데이터 분할 방법

무작위 샘플링 및 시계열 고려 분할 시도

모델 선택 - 검증 안정성

과적합 방지를 위해 k-겹 교차 검증을 사용했습니다. 홀드아웃 샘플로 추가 테스트를 진행했습니다.

1

교차 검증

5-겹 교차 검증 사용

2

홀드아웃

20% 데이터를 홀드아웃 샘플로 사용

3

목적

모델의 일반화 능력 평가 및 과적합 방지

모델 선택 - 결과

DataRobot을 활용해 모델 선택을 수행했습니다. 상위권 모델 3개를 최종 선택했습니다.

1

선택 모델 1

eXtreme Gradient Boosted Trees Regressor

2

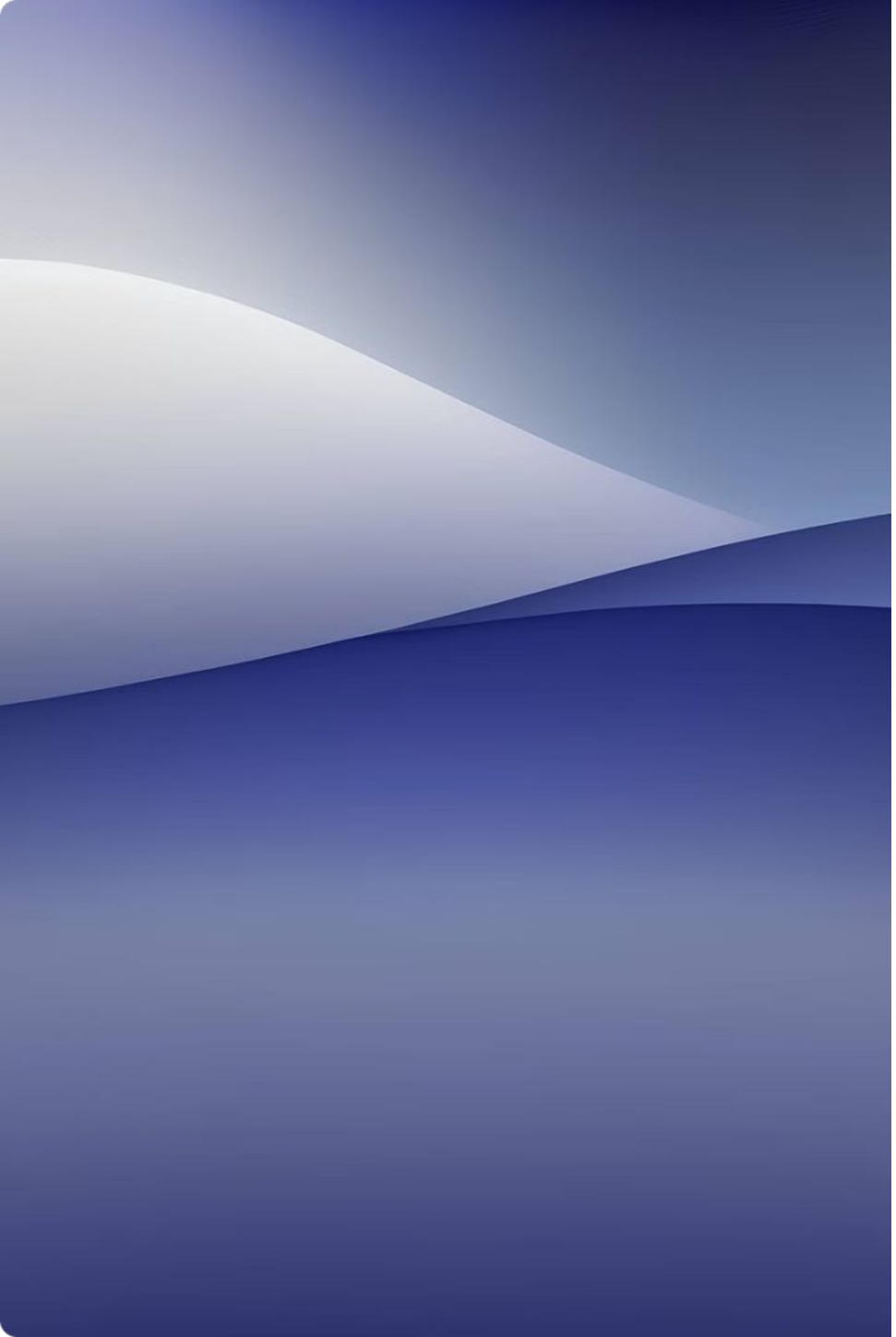
선택 모델 2

Keras Slim Residual Network Regressor

3

선택 모델 3

Light Gradient Boosted Trees Regressor



eXtreme Gradient Boosted Trees Regressor(DataRobot)

eXtreme Gradient Boosted Trees Regressor(XGBoost)는 DataRobot에서 선택된 주요 모델 중 하나입니다. 이 모델은 Gradient Boosting Machines의 매우 효율적인 병렬 버전으로, 높은 예측 정확도를 제공합니다. XGBoost는 최소 제곱 손실을 기본으로 사용하지만, 다양한 손실 함수를 지원합니다. 그리드 검색을 통해 최적의 파라미터를 찾고, Early Stopping을 사용하여 과적합을 방지합니다. 이 모델은 Feature Impact 분석을 통해 각 특성의 중요도를 시각화하여 모델의 예측 과정을 이해하는 데 도움을 줍니다.

1 **효율적인 병렬 처리**

XGBoost는 GBM의 병렬 버전으로 빠른 학습과 높은 정확도 제공

2 **다양한 손실 함수**

최소 제곱 손실 외에도 다양한 손실 함수 지원

3 **그리드 검색**

최적의 XGBoost 파라미터 값을 찾기 위한 그리드 검색 수행

4 **Early Stopping**

과적합 방지를 위한 Early Stopping 기능 제공

Keras Slim Residual Network Regressor(DataRobot)

Keras Slim Residual Network Regressor는 DataRobot에서 선택된 또 다른 주요 모델입니다. 이 모델은 딥러닝 기반의 회귀 모델로, Residual Connection을 활용하여 깊은 네트워크의 학습을 용이하게 합니다. Keras 프레임워크를 사용하여 구현되었으며, 자체 정규화 신경망 기술을 적용하여 배치 정규화 없이도 기울기 소실 문제를 해결합니다. 이 모델은 복잡한 비선형 관계를 학습할 수 있어 아파트 가격 예측과 같은 복잡한 문제에 적합합니다.

1. **Residual Connection**

   깊은 네트워크의 학습을 용이하게 하는 Residual Connection 활용

2. **자체 정규화 신경망**

   배치 정규화 없이 기울기 소실 문제 해결

3. **비선형 관계 학습**

   복잡한 비선형 관계를 학습할 수 있는 능력

4. **Keras 프레임워크**

   유연하고 효율적인 Keras 프레임워크를 사용한 구현

Light Gradient Boosted Trees Regressor(DataRobot)

Light Gradient Boosted Trees Regressor(LightGBM)는 DataRobot에서 선택된 세 번째 주요 모델입니다. 이 모델은 효율적이고 빠른 그래디언트 부스팅 프레임워크로, 대규모 데이터셋에서도 우수한 성능을 보입니다. LightGBM은 리프 중심 트리 성장 전략을 사용하여 더 균형 잡힌 트리를 생성하고, 카테고리 특성을 효율적으로 처리합니다. 또한, 그리드 검색과 Early Stopping을 지원하여 모델의 성능을 최적화하고 과적합을 방지합니다.

1. **효율적인 그래디언트 부스팅**
   빠른 학습 속도와 우수한 성능을 제공하는 LightGBM 프레임워크

2. **리프 중심 트리 성장**
   더 균형 잡힌 트리 생성을 위한 리프 중심 전략 사용

3. **카테고리 특성 처리**
   카테고리 특성을 효율적으로 처리하는 기능 제공

4. **최적화 도구**
   그리드 검색과 Early Stopping을 통한 모델 성능 최적화

PyTorch Residual Network Regressor(박석)

PyTorch Residual Network Regressor는 팀원 박석이 개발한 모델입니다. 이 모델은 PyTorch 프레임워크를 사용하여 구현된 딥러닝 기반의 회귀 모델로, Residual Network 구조를 활용합니다. 모델 구현 과정은 라이브러리 임포트, 데이터 전처리, 커스텀 데이터셋 클래스 정의, Residual Network 모델 정의, 학습 및 추론 함수 구현 등의 단계로 이루어졌습니다. 이 모델은 복잡한 비선형 관계를 학습할 수 있어 아파트 가격 예측에 적합합니다.

1 라이브러리 임포트

필요한 PyTorch 라이브러리 및 기타 모듈 임포트

2 데이터 전처리

결측치 처리, 이상치 제거, 텍스트 데이터 벡터화 등

3 모델 정의

Residual Network 구조를 가진 PyTorch 모델 정의

4 학습 및 추론

모델 학습 및 예측을 위한 함수 구현

Random Forest 모델링 과정(백경탁)

팀원 백경탁이 진행한 XGBoost 모델링 과정은 여러 차례의 시도와 개선을 거쳤습니다. 초기에는 Baseline 코드를 그대로 사용하여 시작했으며, 점진적으로 피처 선택과 모델 파라미터 조정을 통해 성능을 향상시켰습니다. 주요 피처를 18개에서 시작하여 점차 줄여가며 최적의 조합을 찾았고, 시계열 k-Fold 교차 검증을 적용하여 모델의 안정성을 높였습니다. 또한, 결측치 처리와 원-핫 인코딩 등의 데이터 전처리 기법을 적용하여 모델의 성능을 개선했습니다.

1

Baseline 코드 실행

초기 모델링 시작

2

피처 선택 최적화

18개에서 시작하여 점진적으로 주요 피처 선별

3

시계열 k-Fold 적용

모델 안정성 향상을 위한 교차 검증 적용

4

데이터 전처리 개선

결측치 처리, 원-핫 인코딩 등 적용

Light GBM 모델링 과정(한아름)

팀원 한아름이 진행한 Light GBM 모델링 과정은 다양한 모델 비교와 데이터 분할 방법 실험을 포함했습니다. 초기에는 Baseline 코드를 사용하여 RandomForest, XGBoost, LightGBM, CatBoost 등 여러 모델의 성능을 비교했습니다. 이후 Light GBM 모델을 선택하여 Time Series K-Fold 데이터 분할 방법을 적용했습니다. 5개 fold의 평균, Top3 fold의 평균 등 다양한 방식으로 성능을 측정했으며, Feature Importance를 고려하여 주요 피처를 선별하는 과정을 거쳤습니다.

모델 비교

RandomForest, XGBoost, LightGBM, CatBoost 등 다양한 모델 성능 비교

Time Series K-Fold

시계열 특성을 고려한 데이터 분할 방법 적용

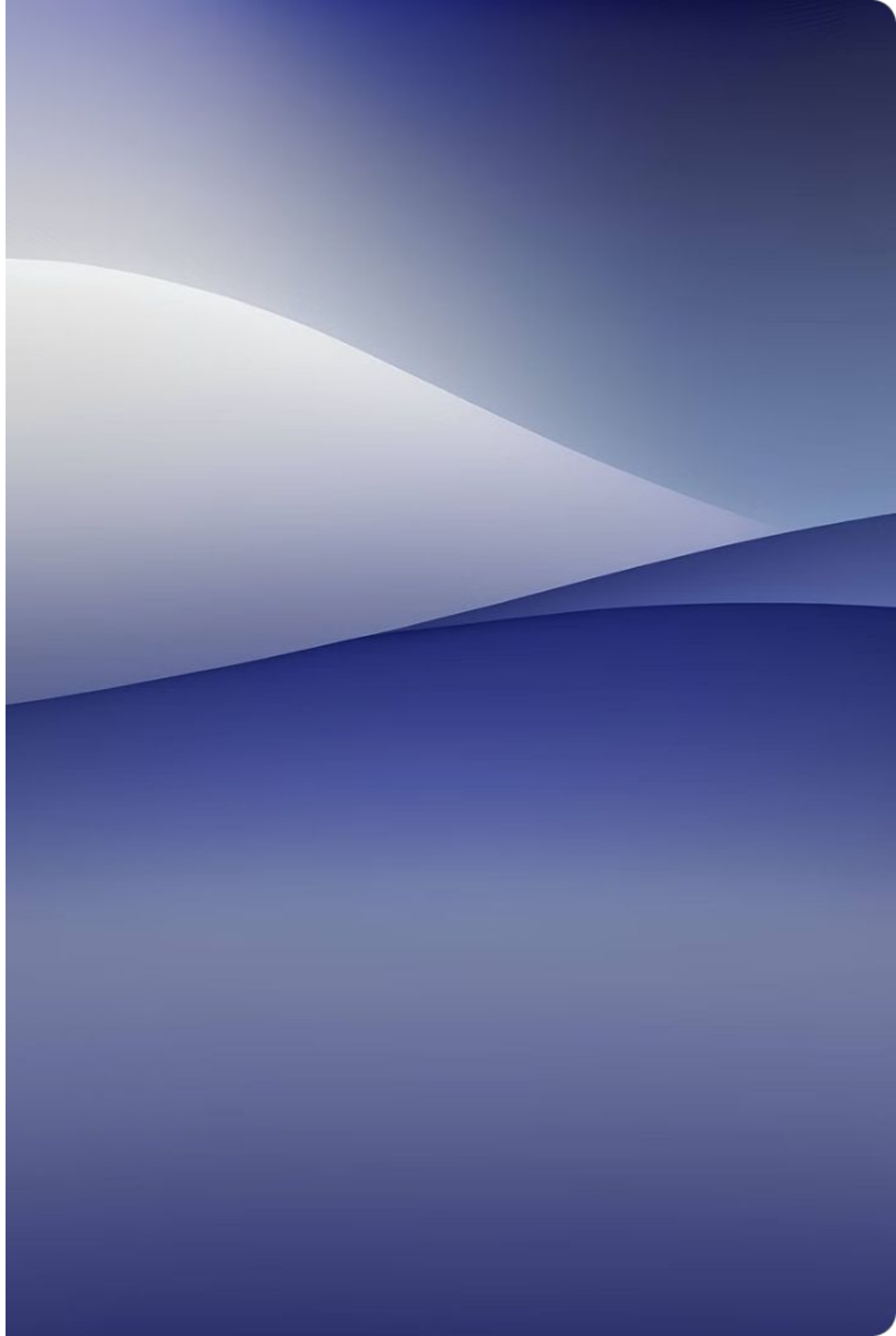
성능 측정

다양한 fold 조합을 통한 모델 성능 평가

모델 성능 테스트 비교

4개 모델의 성능을 RMSE로 비교했습니다. Keras Slim ResNet(DataRobot) 이 가장 우수한 성능을 보였습니다.

모델	RMSE
Random Forest (백경탁)	14779.9300
Keras Slim ResNet (DataRobot)	11943.8758
Light GBM (한아름)	30844.8901
PyTorch ResNet (박석)	553395.0503



최종 리더보드 결과

팀은 최종적으로 3위를 달성했습니다. Keras Slim ResNet (DataRobot) 모델이 가장 좋은 성능을 보였습니다.

최종 순위

3위

최고 성능 모델

Keras Slim ResNet (DataRobot)

최종 RMSE

11943.8758

학습 및 개선점

팀은 다양한 학습과 개선점을 발견했습니다. 도메인 지식 강화와 협업 능력 향상이 필요했습니다.

1

도메인 지식

부동산 시장에 대한 이해 필요

2

협업 능력

Git 활용 및 팀 R&D 경험 강화 필요

3

기술 역량

Python, 머신러닝/딥러닝 역량 향상 필요

향후 계획

팀은 이번 경험을 바탕으로 지속적인 학습과 개선을 계획했습니다. 다음 대회에서의 더 나은 성과를 기대합니다.

1

역량 강화

개인별 기술 역량 향상 계획 수립

2

협업 개선

Git 활용 및 팀 프로젝트 경험 축적

3

도메인 학습

부동산 시장 관련 도메인 지식 습득

4

다음 도전

새로운 데이터 분석 대회 참가 계획