기계학습기초: Team Project

기계학습을 통한 요주의 전세계약 식별 알고리즘 구현

Kang, Hyeonseung 〈강 현승, 202221498, h5k@ajou.ac.kr〉

Table of Contents

1 서론

1.1 Business Understanding

최근 '빌라왕 사건'과 같이 신축빌라를 활용한 전세사기가 사회적으로 큰 이슈가 되었다. 이는 주로 자산이 많지 않은 서민층 임차인을 대상으로 하여, 임대인이 전세보증금을 반환하지 못함으로써 발생한다. 전세보증금은 일반적으로 임차인의 전 재산이나 다름 없는 경우가 많아, 보증금 반환이 이루어지지 않을 경우 그 피해는 고스란히 서민에게 전가된다. 실제로 대항력이 있는 임차인이라 하더라도 실제 매각 절차가 완료되어 보증금을 돌려받기까지 긴 시일이 걸리며, 보증금이 실제 매각액에 비해 높아 보증금을 완전히 돌려받지 못하는 사례도 빈번히 발생하고 있다.

최근에는 이와 같은 보증금 미반환 사건이 증가함에 따라, 보증기관의 전세보증금 대위변제액도 기하급수적으로 늘고 있다. 이는 공공재정의 부담을 초래할 뿐만 아니라, 사회적 신뢰의 기반이 되는 부동산 시장의 건전성을 심각하게 훼손한다. 나아가 부동산 자산의 환금성과 평가가 하락하면서 부동산의 시장가치가 불확실해지고, 이는 투자심리 위축, 자산 유동성 저하, 금융기관의 대출담보 평가에도 부정적인 영향으로 작용한다. 따라서 본 연구에서는 공개된 정보들을 바탕으로 기계학습을 통하여 분류하는 모델을 구축함으로써 보증금 미반환 위험이 있는 전세 계약을 사전에 식별할 수 있는 가능성을 모색하고자 한다.

요주의 전세물건으로 판단하기 위해서 법원경매 데이터를 수집, 활용한다. 각 경매사건의 소재지에서 체결된 임대차계약 중 경매사건접수일 이전에 이루어진 계약을 요주의 전세계약으로 분류한다. 이후 요주의 전세계약으로 분류된 계약과 분류되지 않은 계약을 구분할 수 있도록 기계학습 방법을 사용하여 모델을 학습한다.

기존 프로젝트 프로포절에서 변경된 내용으로는 K-Means 방법을 사용하여 분류하도록 프로젝트를 설계하였지만, feature 수가 당초 예상치를 초과하여 고차원에서 성능을 기대하기 힘든 K-Means 방법에서 트리 기반 방법으로 선회하였다.

2 데이터셋

2.1 Data Understanding

연구에 필요한 데이터들을 거래데이터, 법원경매데이터, 시계열 지표, 기초자치단체 단위의 통계, 개별 물건의 입지로 분류하여 수집하였다.

2.1.1 거래 데이터

확보한 거래 데이터는 2018 년 1 월부터 2024 년 12 월 이내에 체결된 아파트, 다세대주택, 오피스텔의 임대차거래로 각각 7,073,653 건, 1,659,161 건, 1,297,112 건 확보하였다. 임대차 계약 데이터를 확보하기 위해 국토교통부 실거래가 공개시스템[1]에서 1 개월 단위로 제공하는 csv 파일을 활용하였다. 거래 데이터는 아파트, 다세대주택, 오피스텔로 구분하여 제공하고 있으며, 읍면동 수준까지의 주소, 번지, 전용면적, 계약년월일, 보증금, 월세금, 층, 건축년도, 계약기간을 담고 있다. 제공 웹페이지의 다운로드 동작을 자동화하기 위해 Playwright 라이브러리를 사용하여 브라우저를 조작, 획득하였다.

2.1.2 법원경매 데이터

[1] 국토교통부, 실거래가 공개시스템, https://rt.molit.go.kr/pt/xls/xls.do.

확보한 법원경매 데이터는 2025 년 6 월 2 일 수집 시점에 법원경매정보[2]에서 공개하고 있던 데이터로, 공판기일이 수집시점으로부터 2 주일 후 사이로 정해진 사건 및 수집시점으로부터 매각기일이 7 일 이내였던 사건 10,353 건을 수집하였다. 수집된 사건의 경매사건접수일 범위는 2014 년 1 월 20 일부터 2025 년 3 월 20 일까지이며, 주거용건물로 용도분류된 물건으로 한정하여 수집하였다. 법원경매정보는 가공된 경매사건 자료를 따로 제공하지 않고, 웹페이지를 통해 공개하고 있으므로 제공 페이지의 Origin 서버로 주고받는 JSON 본문의 HTTP 요청을 모방하여 수집하였다.

2.1.3 시계열 지표

거래가 이루어진 시점을 Classifier 에 반영하기 위해 2018 년 1 월부터 2024 년 12 월까지의 시계열 지표를 확보하였다. 시계열 지표로 한국은행 기준금리, 주택담보대출 상품의 금리, 지역별 전월세 전환율을 수집하였다.

한국은행 기준금리[3]를 한국은행 웹페이지에서 수작업으로 획득하여 csv 파일로 정리, 활용하였다.

은행연합회 웹페이지[4]에 공개되어 있는 1 개월 주기로 갱신되는 시중은행별 신규취급액 기준 분할상환방식 주택담보대출 상품의 금리 자료를 수집하였다. 해당 자료에는 시중은행과 신용평가사(KCB)의 신용점수 구간별 평균 대출금리를 공개하고 있다. 본 연구에서는 1 개월 주기를 대표하는 주택담보대출 금리 지표를 해당 1 개월 내 공개된 모든 은행과 신용점수 구간의 금리의 중앙값으로 채택하여 활용하였다.

월세 거래를 전세거래와 같이 활용하기 위해 월차임을 전세금으로 변환할 수 있도록 한국부동산원에서조사한 전국주택가격동향조사의 지역별 전월세 전환율[5] 자료를 KOSIS 에서 확보했다. 지역별 전월세 전환율은 1 개월 단위로 갱신되는 자료로, 서울 5 개권역(도심권, 동북권, 서북권, 서남권, 동남권)과 광역 시/도 별로 제공한다.

2.1.4 기초자치단체 단위의 통계

확보한 전력데이터는 2018 년 1 월부터 2024 년 12 월까지로, 거래 물건이 위치한 소재지의 인구, 산업 특성을 반영하기 위해 한국전력 전력데이터 개발포털시스템에서 계약종별 및 산업분류별 고객수와 사용량 데이터, 복지할인 혜택고객수, 가구평균전력사용량 데이터를 확보하였다.

계약종별 데이터는 주택용, 일반용, 교육용, 산업용, 농사용, 가로등, 심야로 구분된 자료로, 고객수와 사용량을 포함한다. 산업분류별 데이터는 한국표준산업분류[6]에 따라 구분된 자료로, 고객수와 사용량을 포함한다. 복지할인 혜택고객수 데이터는 장애우, 유공자, 기초수급자, 차상위, 사회복지시설, 생명유지장치, 대가족, 다자녀로 구분된 자료로, 고객수 데이터이다. 가구평균전력사용량 데이터는 지역별 가구수와 평균 전력사용량을 제공한다.

한국전력 전력데이터 개발포털시스템의 API 가이드에서는 조회년월, 시군구코드를 JSON 포맷의 HTTP 요청을 통해 데이터를 요청하도록 하고 있으며, 이에 따라 시군구 기초자치단체 단위의 1 개월 주기 데이터를 확보, 데이터베이스에 직렬화하여 저장하였다. 이 과정에서 강원도와 전라북도의 산업분류별 데이터를 API 가이드에 따라 제공하는 법정동 코드로 조회하였으나, 자료가 없다는 404 응답을 반환받는 문제가 발생하였다. 이에 전력데이터서비스 제공부서에 문의하였으나 고객 주소의 가공

---

[2] 대한민국 대법원, 대한민국 법원 법원경매정보, https://www.courtauction.go.kr.

[3] 한국은행, 통화정책방향-한국은행기준금리추이, https://www.bok.or.kr/portal/singl/baseRate/list.do?dataSeCd=01&menuNo=200643.

[4] 은행연합회, 가계대출금리 - 은행별 비교공시, https://portal.kfb.or.kr/compare/loan_household_new.php.

[5] 한국부동산원, 전국주택가격동향조사, 2025.03, 지역별 전월세전환율.

[6] 통계청, 10 차 개정 표준산업분류(KSIC), KSSC 통계청 통계분류포털.

문제로 빠른 시일 내에 제공이 어렵다는 답변을 받아 강원도, 전북도 소재지의 데이터를 연구에서 제외하였다.

2.1.5 개별 물건의 입지

개별 거래 물건 각각을 구분하는 특성으로 주변 건물과 지형데이터를 수집했다. 한국지역정보개발원에서 제공하는 위치정보요약 DB[7] 의 csv 파일을 통해 각 주소의 좌표정보를, 도로명주소 전자지도 DB[8]의 GIS 데이터파일을 통해 전국 단위의 건물 정보와 건물용도코드, Polygon 을 확보하였다.
해당 데이터에는 주소가 부여된 철도역사와는 다르게 별도의 주소가 부여되지 않은 버스정류장은 포함하고 있지 않기에 공공데이터포털에서 국토교통부_전국버스정류장 위치정보[9] 자료를 추가로 확보하였다. 국토교통부_전국버스정류장 위치정보 데이터는 버스정류장의 목록과 좌표정보를 포함한다. 고도가 높을수록 접근성이 떨어지며, 이는 임대료에 부정적으로 반영된다는 일반적인 인식을 Classifier 에 반영하고자 하였다. NASA Jet Propulsion Laboratory 에서는 EOS Terra 위성에서 확보한 전세계 30m 수준 공간해상도를 가진 ASTER GDEM(Global Digital Elevation Model)을 제공하고 있다. 해당 데이터셋의 전체 크기는 약 1.3TB 로, 이 연구에서 필요한 한반도와 부속도서를 포함하는 위도 33°N - 40°N, 경도 124°E - 132°E 범위로 데이터셋을 한정하여 활용하였다.

2.1.6 주소정보

확보한 데이터들이 출처가 파편화되어 있어 하나의 전세거래로 병합하기 위해선 공통된 인자를 필요로 하였다. 이를 위해 전국 단위의 구주소(지번주소)와 신주소(도로명주소) 자료를 확보하였다. 한국지역정보개발원에서는 PC 용 주소검색기[10]를 통해 오프라인 환경에서 정규화된 주소를 검색할 수 있게 제공하고 있다. PC 용 주소검색기는 h2 Database 기반의 주소 데이터를 담고 있다. 해당 H2 Database 는 Username-Password 인증으로 보호되고 있으며, 이를 사용하고자 사용자 환경에서 동작하는 클라이언트를 reverse engineering 하여 획득하였다. 사용자 클라이언트를 설치하였을 때, 설치 디렉토리에서 컴파일된 JAVA 소스코드를 확인할 수 있었다. 확보한 소스로부터 <Figure 1>과 같이 com.jusoro.common 패지지의 DBHandler.class 파일을 Online JAVA Decompiler[11]를 사용하여 소스코드를 확인, Username 과 Password 를 획득하여 데이터를 활용하였다.

```
try {
    Class.forName("org.h2.Driver");
    this.conn = DriverManager.getConnection("jdbc:h2:./db/pcrns;MV_STORE=FALSE;QUERY_CACHE_SIZE=16;", "pcrns", [redacted]);
    this.conn.setAutoCommit(false);
    this.conn2 = DriverManager.getConnection("jdbc:h2:./db/pcrns_cfg;MV_STORE=FALSE;QUERY_CACHE_SIZE=16;", "pcrns", [redacted]);
    this.conn2.setAutoCommit(false);
    var1 = true;
```

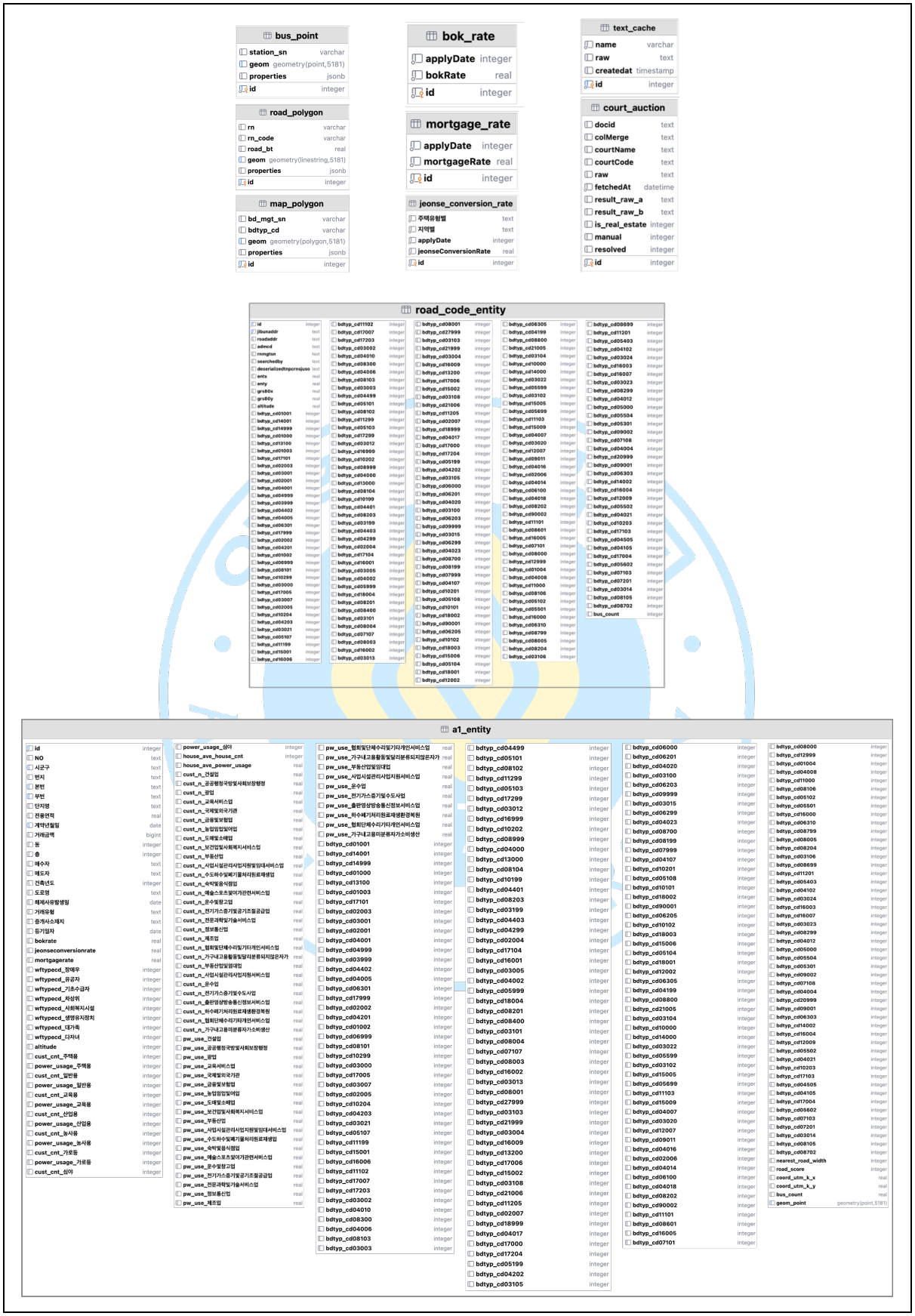
<Figure 1> Decompiled java source code

2.2 Data Preparation

확보한 데이터를 통합하여 관리하기 위해 DBMS 를 도입하였다. 이 연구에서 사용한 DBMS 는 PostgreSQL 로, 인덱싱 성능이 좋으며 GIN 과 같은 인덱싱 기법을 지원하고, 지리데이터 활용을 위한

[7] 행정안전부 한국지역정보개발원, 위치정보요약 DB, https://business.juso.go.kr.
[8] 행정안전부 한국지역정보개발원, 도로명주소 전자지도 DB, https://business.juso.go.kr.
[9] 공공데이터포털, 국토교통부_전국 버스정류장 위치정보, https://www.data.go.kr/data/15067528/fileData.do.
[10] 주소기반산업지원서비스, PC 용 주소검색기, https://business.juso.go.kr/addrlink/tchnlgySport/pcAdresFinder.do.
[11] Decompiler - Disassemble, decompile and analyze binary files online, https://www.decompiler.com.

PostGIS 익스텐션을 사용할 수 있어 선택하였다. <Figure 2>는 이 연구에서 사용한 테이블들의 관계를 도식화한 다이어그램이다. a2_entity, b2_entity, d2_entity 테이블은 각각 아파트, 다세대주택, 오피스텔의 임대거래 내역을 저장하였으며, 각 테이블의 컬럼이 같으므로 <Figure 2>에서 b2_entity 와 d2_entity 는 생략했다. bus_point 테이블은 버스정류장의 좌표데이터를 저장한 테이블이며, 테이블 road_polygon 과 map_polygon 은 각각 도로의 Polygon 과 건물정보 및 건물의 지상 Polygon 을 저장한 테이블이다. bok_rate, mortgage_rate, jeonse_conversion_rate 테이블은 각각 한국은행 기준금리, 주택담보대출 금리, 지역별 전월세 전환율을 저장한 테이블이다. court_auction 테이블은 법원경매정보를 저장한 테이블이다. 테이블 간의 관계는 없으므로 표시하지 않았다.



〈Figure 2〉 Database diagram

2.2.1 거래데이터 가공

국토교통부 실거래가 공개시스템에서 제공하는 거래데이터의 csv 파일은 <Table 1>과 같은 컬럼을 가진다.

NO	시군구	번지	본번	부번	단지명	전용면적(㎡)
계약년월	계약일	거래금액(만원)	동	층	매수자	매도자
건축년도	도로명	해제사유발생일	거래유형	중개사소재지	등기일자	

<Table 1> 거래데이터 csv 파일 컬럼 목록

이를 분석에 활용하기 위해 <Table 2>와 같이 데이터베이스에 테이블을 만들어 저장하였다.

필드이름	형태	설명	필드이름	형태	설명	필드이름	형태	설명
id	integer	pk	EMD_CD	text	읍면동수준코드	RN_CD	text	도로명코드
BD_MA_SN	varchar	건물본번	DB_SB_SN	text	건물부번	entX	real	UTM-K 위도값
entY	real	UTM-K 경도값						

<Table 2> 데이터베이스의 거래데이터 테이블 명세

2.2.2 주소정보 가공

수집한 거래데이터의 위치정보는 지번 및 도로명 주소로 구분되며, 이를 정규화된 건물관리번호 및 도로명주소 코드로 관리하기 위해 주소를 분리하여 가공을 실시하였다. 또한 많은 양의 개별 거래 엔티티마다 수집한 데이터를 업데이트하는 것보다 상대적으로 엔티티가 적은 가공된 주소 테이블에 업데이트함으로써 성능 상 이점을 얻고자 하였다. 가공된 주소를 저장하기 위해 road_code_entity 테이블을 생성하였다. road_code_entity 테이블의 개별 엔티티는 jibunaddr 필드와 roadaddr 로 구분되며, 각 필드는 거래데이터 테이블의 시군구, 본번, 부번 필드를 concatenate 및 본/부번을 포함한 도로명주소 필드를 값으로 하여 중복된 주소가 없도록 거래데이터 테이블을 순회하며 삽입하였다. 또한 PC 용 주소검색기에서 확보한 pcrns 테이블을 바탕으로 개별 주소에 대한 정보를 역직렬화하여 deserializedtnpcrnsjuso 필드에 저장하였다.
본 연구에서는 시계열 데이터를 제외한 위치정보를 바탕으로 하는 데이터들을 road_code_entity 테이블에 우선 저장하였다.

2.2.3 위치정보요약 DB 병합

road_code_entity 테이블에 삽입된 주소를 대상으로 좌표를 삽입하였다. 위치정보요약 DB 는 각각의 도로명주소에 EPSG:5179(GRS80 타원체, UTM-K 좌표계)에 따른 좌표를 명시한 자료로, 광역자치단체별 csv 파일로 제공된다.

시군구코드	출입구일련번호	법정동코드	시도명	시군구명	읍면동명	도로명코드	도로명
지하여부	건물본번	건물부번	건물명	우편번호	건물용도분류	건물군여부	관할행정동
X 좌표	Y 좌표						

<Table 3> 위치정보요약 DB csv 파일 컬럼 목록

위 정보를 road_code_entity 테이블의 grs80x, grs80y 필드에 업데이트하기 위해 road_code_entity 테이블의 엔티티를 순회하여 위치정보요약 DB 의 좌표 값을 가져오도록 했다. DBMS 를 사용하는 경우 약 640 만 건의 위치정보요약 DB 가 저장된 테이블의 SELECT 쿼리 성능이 좋지 않을 것이라 예상하였다. 이에 따라 KV 형식의 In-memory 데이터베이스인 Redis 를 사용하였다. 위치정보요약 DB

csv 파일의 각 행을 template string 으로 나타낸 “map-location-coord_${법정동코드}_${도로명코드}_${건물본번}-${건물부번}”을 Key 값, 좌표정보를 JSON 으로 직렬화하여 Value 값으로 설정하여 일괄 삽입하였다.
road_code_entity 테이블에 삽입된 엔티티를 순회하며, 각 엔티티의 deserializedtnpcrnsjuso 필드 값을 역직렬화하여 법정동코드, 도로명코드, 건물본번, 건물부번을 확보하였다. 확보한 주소별 정보를 바탕으로 위의 키 값을 사용하여 Redis 에 질의, grs80x, grs80y 필드에 좌표 값을 삽입하였다.

2.2.4 고도정보 가공

ASTER GDEM 데이터셋을 geotiff 이미지 활용에 대한 도메인 지식 없이도 좌표를 통해 해발고도를 얻을 수 있는 Open Topo Data[12]의 Elevation API 도커 이미지를 활용하여 가공된 주소정보에 대한 해발고도를 수집하였다.
앞서 road_code_entity 테이블에 삽입한 grs80x, grs80y 필드는 EPSG:5179 좌표계를 따른 좌표이므로 Elevation API 에서 지원하는 EPSG:4326(WGS84 타원체)로 변환하여야 한다. proj4 라이브러리를 사용하여 변환을 진행하였다. 변환한 좌표를 Elevation API 에 HTTP 요청으로 질의하여 획득한 m 단위의 고도를 개별 road_code_entity 테이블의 altitude 필드에 업데이트하였다.

2.2.5 시계열 지표 병합

수집한 한국은행 기준금리 자료, 주택담보대출 금리 자료, 지역별 전월세전환율 자료로부터 각 임대차 거래 인스턴스에 부합하는 값을 적용하기 위해 각각 bok_rate, mortgage_rate, jeonse_conversion_rate 테이블에 저장하였다.
한국은행 기준금리와 주택담보대출 금리는 임대차 거래 당시 유효한 기준금리를 적용하였다. 각 임대차 거래에 한국은행 기준금리 및 주택담보대출 금리 자료를 병합하기 위해 거래데이터 테이블에 bokrate, mortgagerate 컬럼을 추가하였다. 각 인스턴스별 bokrate, mortgagerate 에는 SQL 쿼리를 통해 bok_rate, mortgage_rate 테이블로부터 공표 및 기준시점인 applyDate 이 거래년월일 이전이면서 가장 최근의 금리를 추가하였다.
월세 거래를 함께 학습에 활용하기 위해 확보한 지역별 전월세 전환율을 적용하기 위해 거래데이터 테이블에 jeonseconversionrate 컬럼을 추가하여 임대차 거래 인스턴스의 시군구 값과 거래년월일 값에 따라 각 인스턴스의 jeonseconversionrate(전세전환율) 값을 추가하였다.

2.2.6 월세금전환보증금 가공

월차임을 전세보증금에 가산하여, 혹은 월세만을 명시한 전세거래를 전세보증금으로 전환하기 위해 거래데이터 테이블에 추가한 jeonseconversionrate 컬럼의 값을 활용하였다. 월세금전환보증금을 계산하기 위해 <Formula 1>을 사용하여 일괄 추가하였다.

월세금전환보증금 = 기존 전세보증금 + 월세금 * 12 / 전월세전환율(%)

<Formula 1> 월세금전환보증금 계산 공식

[12] Open Topo Data, Open Topo Data introduction, https://www.opentopodata.org.

2.2.7 전력데이터 병합

한국전력 전력데이터 개발포털시스템에서 수집한 지역, 조회년월별 전력사용통계를 text_cache 테이블에 직렬화하여 보관하였으며, 이를 순회화여 각 인스턴스에 추가하였다. 각 임대차거래에 테이블 text_cache 에 저장된 전력사용통계를 병합하기 위해 거래데이터 테이블에 계약종별, 산업분류별, 복지할인 혜택고객수, 가구평균전력사용량 컬럼을 추가하였다. 아래에서 컬럼명을 표현할 때, template string 을 사용하였다.
계약종별 데이터와 산업종별 데이터는 전기계약종별 및 산업종별에 따른 고객수와 전력사용량을 포함하고 있으므로, 각각 cust_cnt_${계약종별코드 및 계약종별코드}, power_usage_${계약종별코드 혹은 계약종별코드} 형식으로 컬럼을 추가하였다. 복지할인 혜택고객수 데이터는 혜택유형에 따른 고객수를 포함하고 있으므로, wftypecd_${혜택코드} 형식으로 컬럼을 추가하였다. 가구평균전력사용량 데이터는 지역별 가구수와 평균 전력사용량을 포함하고 있으므로, house_ave_house_cnt, house_ave_power_usage 컬럼을 추가하였다.
text_cache 테이블의 각 인스턴스를 순회하여 거래데이터 테이블에 추가한 컬럼의 값을 업데이트 하였다. 각 순회동안 직렬화하여 저장한 JSON 응답을 역직렬화, SQL 쿼리를 통해 인스턴스의 광역도시명과 기초단체명, 조회년월에 해당하는 소재와 계약년월일을 가진 임대차거래 인스턴스를 선택하여 일괄 업데이트하였다.

2.2.8 도로명주소 전자지도 DB 가공 및 병합

수집한 도로명주소 전자지도 DB 에는 이 연구에서 활용하고자 하는 전국 단위의 건물용도코드를 포함한다. 이 연구에 개별 임대차 거래 인스턴스 별 물건지 반경 1km 내에 존재하는 용도별 건물의 수를 반영하고자 한다.
수집한 도로명주소 전자지도 DB 의 GIS 파일을 파싱하기 위해 npm 의 shapefile 라이브러리를 사용하였다. 도로명주소 전자지도 DB 의 shp, dbf 파일은 Polygon 모형과 각 Polygon 모형을 설명하는 데이터베이스로 이루어져 있으며, 해당 파일을 shapefile 라이브러리를 통해 파싱하였을 때, 파일에 저장된 Polygon 모형의 꼭짓점 좌표와 각 모형별로 저장된 건물의 상세한 정보를 반환하였다. 이를 데이터베이스에 map_polygon 테이블을 만들어 일괄 삽입하였다. map_polygon 테이블의 geom 필드는 geometry(polygon, 5181) 형식이며, 빠른 SELECT 성능을 위해 GIX 인덱스를 생성하였다.
road_code_entity 테이블에 건물용도코드별 건물 수를 저장할 수 있도록 필드를 추가하였다. Template string 으로 나타낸 “bdtyp_cd$건물용도코드”를 필드들의 이름으로 하여 추가하였다.
road_code_entity 테이블의 엔티티를 순회하며 〈Formula 2〉 쿼리를 통해 엔티티 반경 1km 내의 건물을 조회하고, 건물용도코드별 갯수를 산출한 후 이를 위에서 template string 으로 표현한 필드의 값으로 반영하였다.

```
SELECT * FROM map_polygon WHERE ST_DWithin(geom,
ST_SetSRID(ST_MakePoint(${grs80x},${grs80y}), 5181), 1000);
```

〈Formula 2〉 쿼리

2.2.9 전국버스정류장 위치정보 병합

임대차거래 엔티티에 도보 약 8 분 거리인 반경 400m 내 버스정류장의 갯수를 반영하였다. 수집한 국토교통부 전국버스정류장 위치정보는 각 버스정류장의 이름과 EPSG:5179 좌표계에 따른 좌표가 수록된 csv 파일 형태로 제공된다. 이를 데이터베이스에 bus_point 테이블을 만들어 일괄 삽입하였다.

bus_point 테이블의 geom 필드는 geometry(point, 5181)로, proj4 라이브러리를 사용하여 변환 후 삽입하였다.
road_code_entity 테이블에 근방 버스정류장의 수인 bus_count 필드를 추가하였다. road_code_entity 테이블의 엔티티를 순회하며 〈Formula 3〉 쿼리를 통해 필드의 값을 업데이트하였다.

```
SELECT * FROM bus_point WHERE ST_DWithin(geom,
ST_SetSRID(ST_MakePoint(${grs80x},${grs80y}), 5181), 400);
```

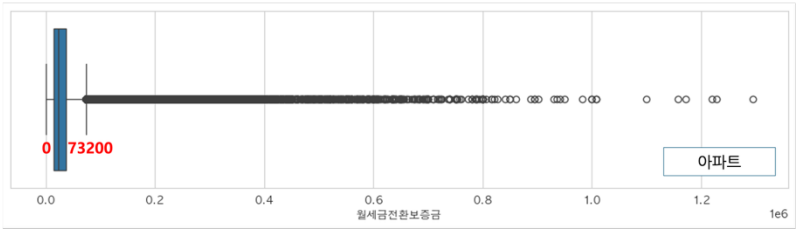
〈Formula 3〉 쿼리

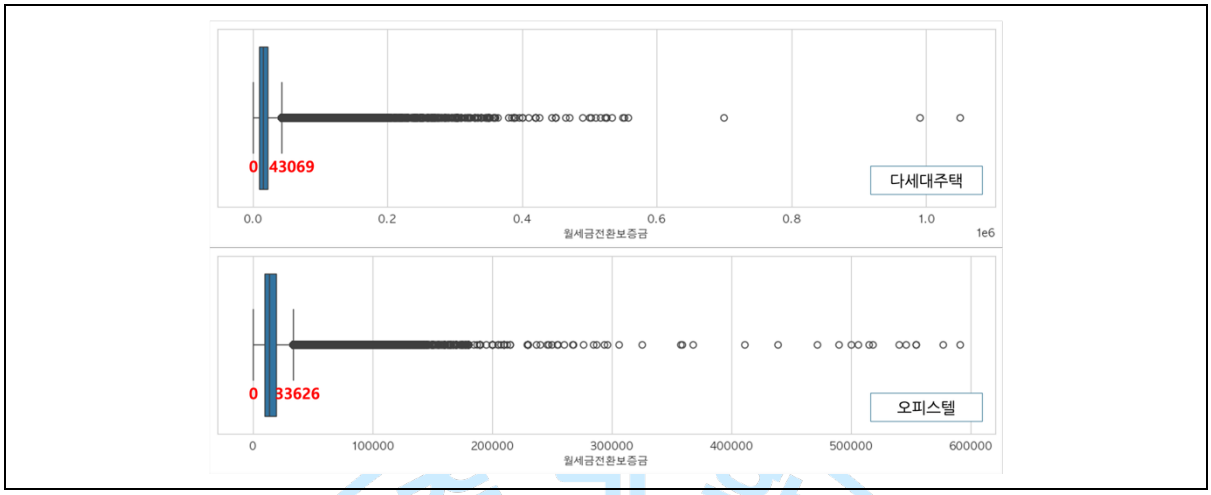
2.2.10 법원경매 데이터 반영

수집한 법원경매 데이터를 저장하기 위해 데이터베이스에 court_auction 테이블을 생성하여 저장하였다. 응답받은 JSON 본문을 직렬화하여 raw 필드에 저장하였다. 매각기일 및 경매사건접수일 등의 상세정보는 별도의 JSON 요청을 통해 수집하였기에 응답받은 JSON 본문을 직렬화하여 result_raw_a 필드에 저장하였다.
최초의 계획으로는 수집한 법원경매 데이터의 주소를 특정하기 위해 JSON 본문의 도로명 및 지번 주소를 사용하고자 하였다. 그러나 정규화되지 않은 주소를 다루고 있어 오랜 전처리 시간을 요구했다. JSON 본문에는 xCordi, yCordi 키에 KATEC(Bessel 타원체, UTM-K 좌표계) 좌표계에 의한 좌표가 저장되어 있어 이를 주소 정규화에 활용하였다. xCordi, yCordi 좌표를 KATEC 좌표계에 의한 좌표에서 EPSG:5181 좌표계에 의한 좌표로 변환 후, map_polygon 테이블의 엔티티 중 geom 필드에 저장된 Polygon 도형에 xCordi, yCordi 좌표가 접하는 엔티티를 주소정규화에 활용하였다. 좌표를 통해 얻은 map_polygon 엔티티는 bd_mgt_sn 필드를 가지고 있으며, 이 필드는 건물관리번호로, 건물관리번호의 첫 10 자리는 광역자치단체 수준부터 법정동 수준까지의 코드를 concatenate 한 것과 같다. 이와 같이 획득한 법정동 수준의 코드와 JSON 응답에서 획득한 도로명주소를 함께 사용하여 주소를 정규화하였다.
개별 임대차거래 엔티티를 요주의 전세계약임을 나타내는 bad 필드를 업데이트함으로써 구별하였다. 수집한 법원경매데이터를 순회하여 업데이트하였으며, 순회 동안 정규화된 주소와 경매사건접수일 이전 전세계약인지, 동일 층인지, 사건 당사자에 임차인이 있는지를 조건으로 하여 일치하는 계약을 일괄 업데이트하였다.

2.2.11 N/A Elimination and Filling, Outlier Handling

폐지된 주소에 대한 임대차거래는 주소를 통한 좌표 획득이 불가하여 제거하였다. 이외 결측치는 건물용도코드 갯수 등 수량 필드에서 발생한 것으로, 0 으로 채워 넣었다.
IQR 방법을 통한 이상치 검사를 수행하였다. 〈Figure 3〉은 위에서부터 차례로 아파트, 다세대주택, 오피스텔 순으로 월차임을 전세보증금에 가산한 월세금전환보증금 필드를 박스플롯을 통해 시각화한 것이다.





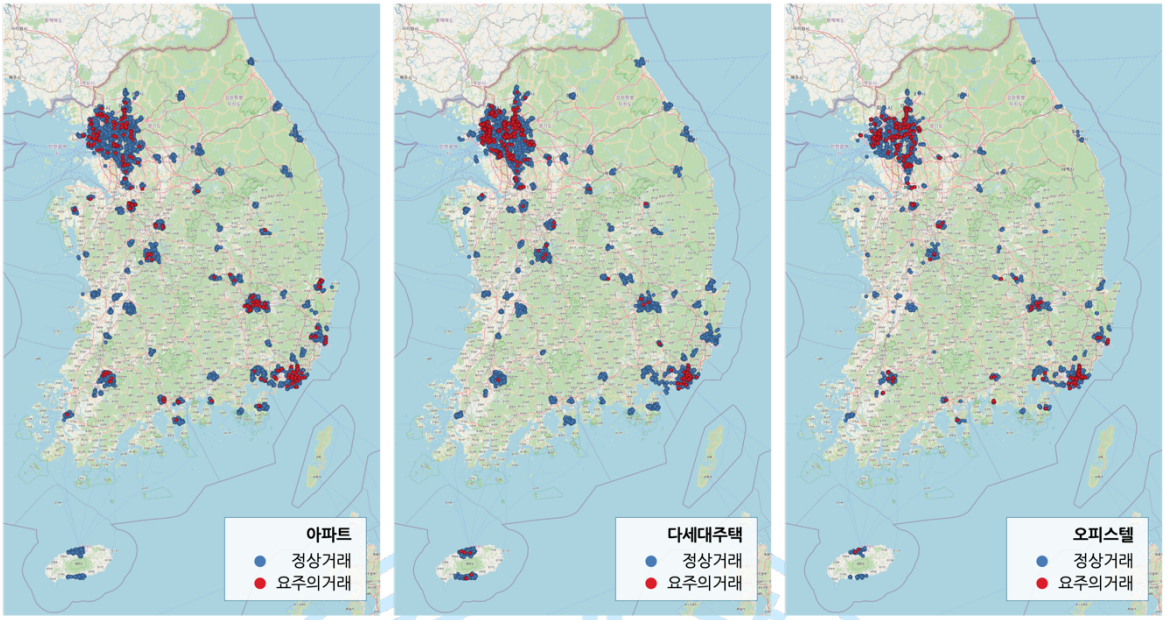
〈Figure 3〉 주택유형별 월세금전환보증금의 박스플롯

월세금전환보증금 필드에서 임계상한값을 상회하는 다수건의 이상치를 검출하였지만, 검출된 이상치도 학습에 필요한 표본으로 판단하여 배제하지 않았다. 아파트 임대차거래 테이블의 월세금전환보증금 필드의 임계상한값은 7 억 3200 만원으로, 서울지역 아파트 5 분위 전세평균가격 중 4 분위가 7 억 원, 5 분위가 11 억원 선에 형성됨[13]을 감안하였을 때, 검출된 이상치를 제거 시 얻는 성능 상의 이점보다 학습하여야 할 표본을 유지하는 것이 더 중요하다 판단하였다.
기초자치단체 수준에서 일괄 업데이트한 필드 및 좌표를 통한 건물의 입지를 계산한 필드는 별도의 이상치 검사를 수행하지 않았다. 원천이 실거래가공개시스템인 데이터에서 별도 가공하지 않은 필드인 층, 건축년도에서 이상치를 검출하였다. 다만 본 연구에서 사용하는 임대차거래 데이터셋은 제공기관인 국토교통부 부동산 소비자보호기획단[14] 및 접수 행정기관[15]에서 검증체계를 운영[16]하여 가공한 것으로 해당 필드에서 검출한 이상치는 배제하지 않았다.

2.2.12 Overview

학습에 이용할 테이블은 a2_entity, b2_entity, d2_entity, 각각 아파트, 다세대주택, 오피스텔 임대차거래 데이터이다. 결측치를 제거한 뒤 잔여한 아파트 거래 7,073,653 건 중 0 건, 다세대주택 거래 1,659,161 건 중 0 건, 오피스텔 거래 1,297,112 건 중 1,294,078 건을 학습에 이용하였다. 이 중 경매 사건을 통하여 요주의 거래로 구분한 전세계약은 아파트 거래 10,259 건, 다세대주택 거래 4,437 건, 오피스텔 8,543 건이다. 〈Figure 4〉는 학습에 이용한 정상거래와 요주의거래를 GQIS[17] 도구를 통하여 시각화한 것이다.

[13] KB 부동산, KB 부동산 데이터허브, https://data.kbland.kr/kbstats/wmh?tIdx=HT08&tsIdx=aptQteRentPrice.
[14] 국토교통부, 국토교통부 부동산소비자보호기획단, https://rt.molit.go.kr/pt/info/info.do.
[15] 부동산 거래신고 등에 관한 법률 [법률 제 20194 호, 2024 년 2 월 6 일, 시행 2024 년 5 월 17 일], 제 6 조의 2(주택 임대차계약의 신고).
[16] 부동산 거래신고 등에 관한 법률 [법률 제 20194 호, 2024 년 2 월 6 일, 시행 2024 년 5 월 17 일], 제 5 조(신고 내용의 검증).
[17] QGIS Development Team. (2025). QGIS Geographic Information System (Version 3.x) [Computer software]. Open Source Geospatial Foundation. https://qgis.org



〈Figure 4〉 주택유형별 정상거래 및 요주의거래의 분포

3 분석

3.1 Modeling

데이터셋에 정상거래와 요주의거래 간 클래스 불균형이 발생하였다. 이는 모델 성능 측정 간 요주의거래 클래스 데이터 수와 근접하게 정상거래 데이터를 나누어 다수개의 데이터프레임을 만든 뒤, 요주의 거래 데이터와 병합하여 샘플 데이터프레임을 만들어 해결하였다. 성능지표는 각 데이터프레임에서 cross validation 을 수행한 점수의 평균을 채택하였다.
임대차거래 테이블 중 엔티티 수가 제일 적은 오피스텔 임대차거래를 대상으로 여러 기계학습방법 중 효과적인 방법을 선별하여 이를 수집한 전체 거래로 확대하였다. 본 연구에서는 트리 기반 학습방법인 XGBoost 와 LightGBM 방법을 비교, 더 나은 성능을 보이는 방법을 채택하였다.

3.1.1 LightGBM

〈Table 4〉는 LightGBM 의 하이퍼파라미터 튜닝의 대상 파라미터 및 값이다.

num_leaves	31, 63, 127
max_depth	5, 7, 10
n_estimators	100, 300, 500

〈Table 4〉 LightGBM 모델의 하이퍼파라미터 튜닝 계획

LightGBM 모델 사용 시 아래의 파라미터와 사용시 가장 좋은 성능지표를 나타냈다.

```
LGBMClassifier(learning_rate=0.1, num_leaves=63, max_depth=7, n_estimators=500)
          (Avg. accuracy score = 0.970175, Avg. auc score = 0.992470)
```

〈Formula 4〉 하이퍼파라미터 튜닝된 LightGBM 모델식

3.1.2 XGBoost

<Table 5>는 XGBoost 의 하이퍼파라미터 튜닝의 대상 파라미터 및 값이다.

n_estimators	100, 300, 500
max_depth	5, 7, 10
min_child_weight	3, 5, 7

<Table 5> LightGBM 모델의 하이퍼파라미터 튜닝 계획

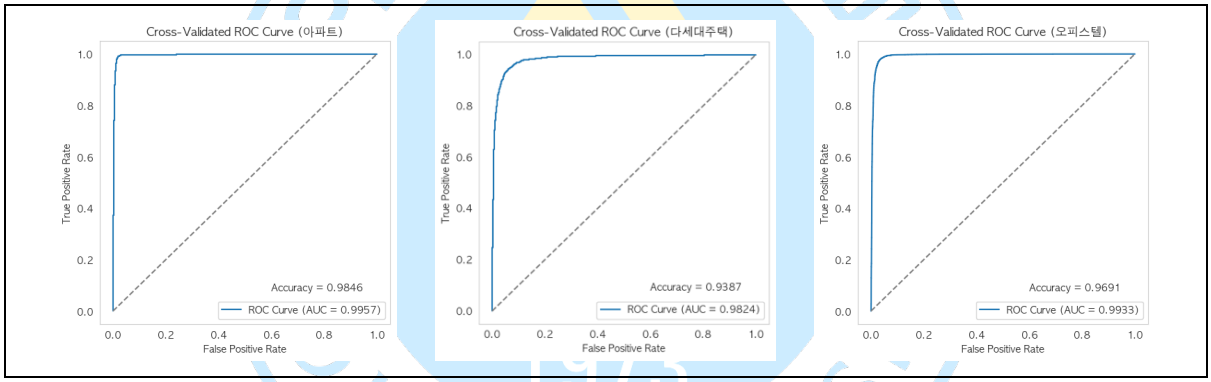
XGBoost 모델 사용 시 아래의 파라미터와 사용시 가장 좋은 성능지표를 나타냈다.

```
XGBClassifier(learning_rate=0.1, max_depth=7, n_estimators=500, min_child_weight=5)
(Avg. accuracy score = 0.959522, Avg. auc score = 0.988969)
```

<Formula 5> 하이퍼파라미터 튜닝된 XGBClassifier 모델식

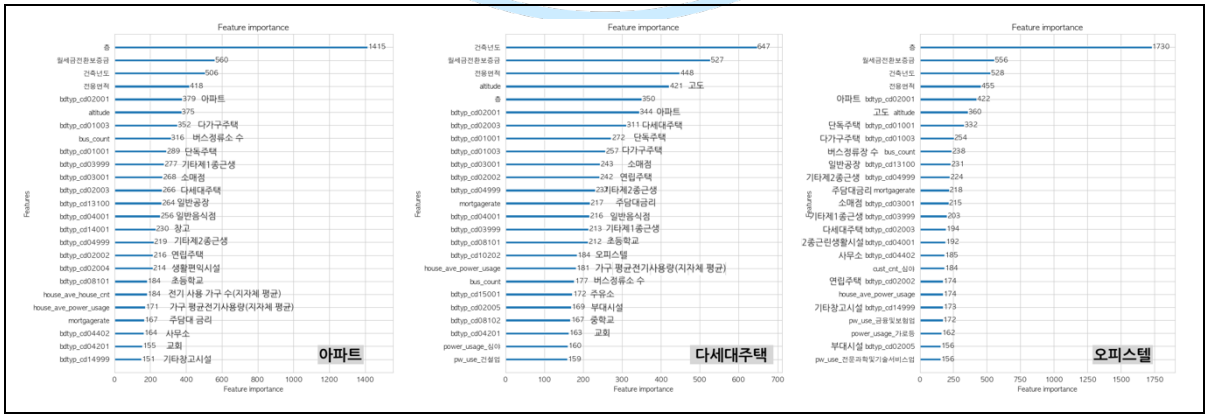
3.1.3 Best Model

앞선 성능검증결과에 따라 제일 우수했던 <Formula 4>의 LightGBM 모델을 각 주택유형별 아파트 및 다세대주택 임대차거래 테이블에 추가로 적용하였다. Feature importance 를 확인하면서 클래스 간 불균형 문제를 해결하기 위해 랜덤으로 샘플링하여 두 클래스 간 인스턴스 차이를 해소하였다. <Figure 5>는 주택유형별 classifier 의 cross-validated ROC Curve, AUC, accuracy 이다.



<Figure 5> 주택유형별 ROC Curve, AUC, Accuracy

각 주택유형의 classifier 의 accuracy 는 0.95 이상으로 우수한 성능을 보인다. AUC 점수 또한 0.95 이상으로 클래스 간 분리능력이 우수하며, 안정적인 분류 성능을 기대할 수 있다.
<Figure 6>은 주택유형별 classifier 의 feature importance 이다.



<Figure 6> 주택유형별 Feature Importance

<Figure 7>은 주택유형별 classification report 이다.

아파트				
	precision	recall	f1-score	support
False	0.99	0.97	0.98	8385
True	0.98	0.99	0.98	8385
accuracy			0.98	16770
macro avg	0.98	0.98	0.98	16770
weighted avg	0.98	0.98	0.98	16770

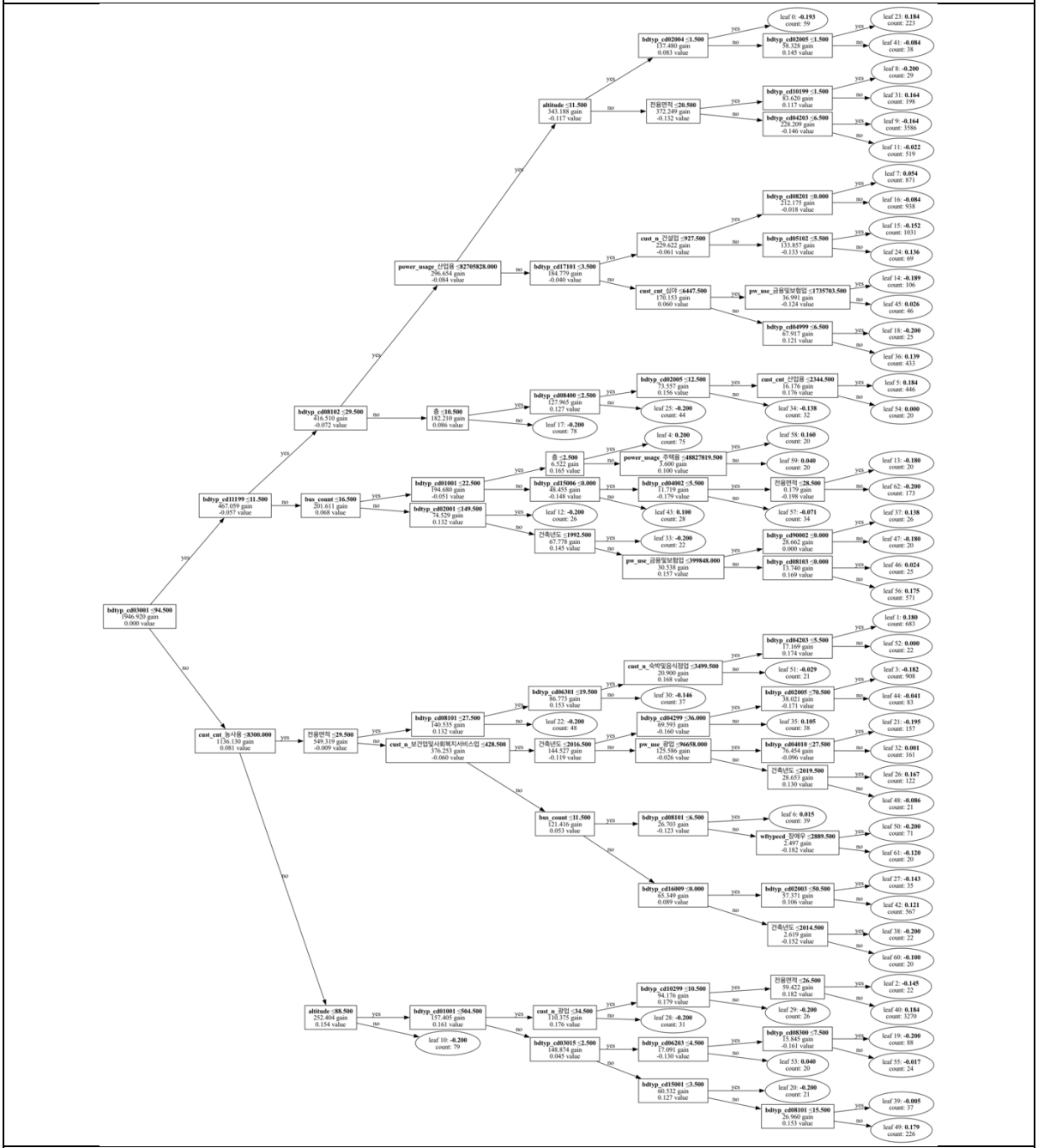
다세대주택				
	precision	recall	f1-score	support
False	0.96	0.92	0.94	3962
True	0.92	0.96	0.94	3962
accuracy			0.94	7924
macro avg	0.94	0.94	0.94	7924
weighted avg	0.94	0.94	0.94	7924

오피스텔				
	precision	recall	f1-score	support
False	0.99	0.95	0.97	8290
True	0.95	0.99	0.97	8290
accuracy			0.97	16580
macro avg	0.97	0.97	0.97	16580
weighted avg	0.97	0.97	0.97	16580

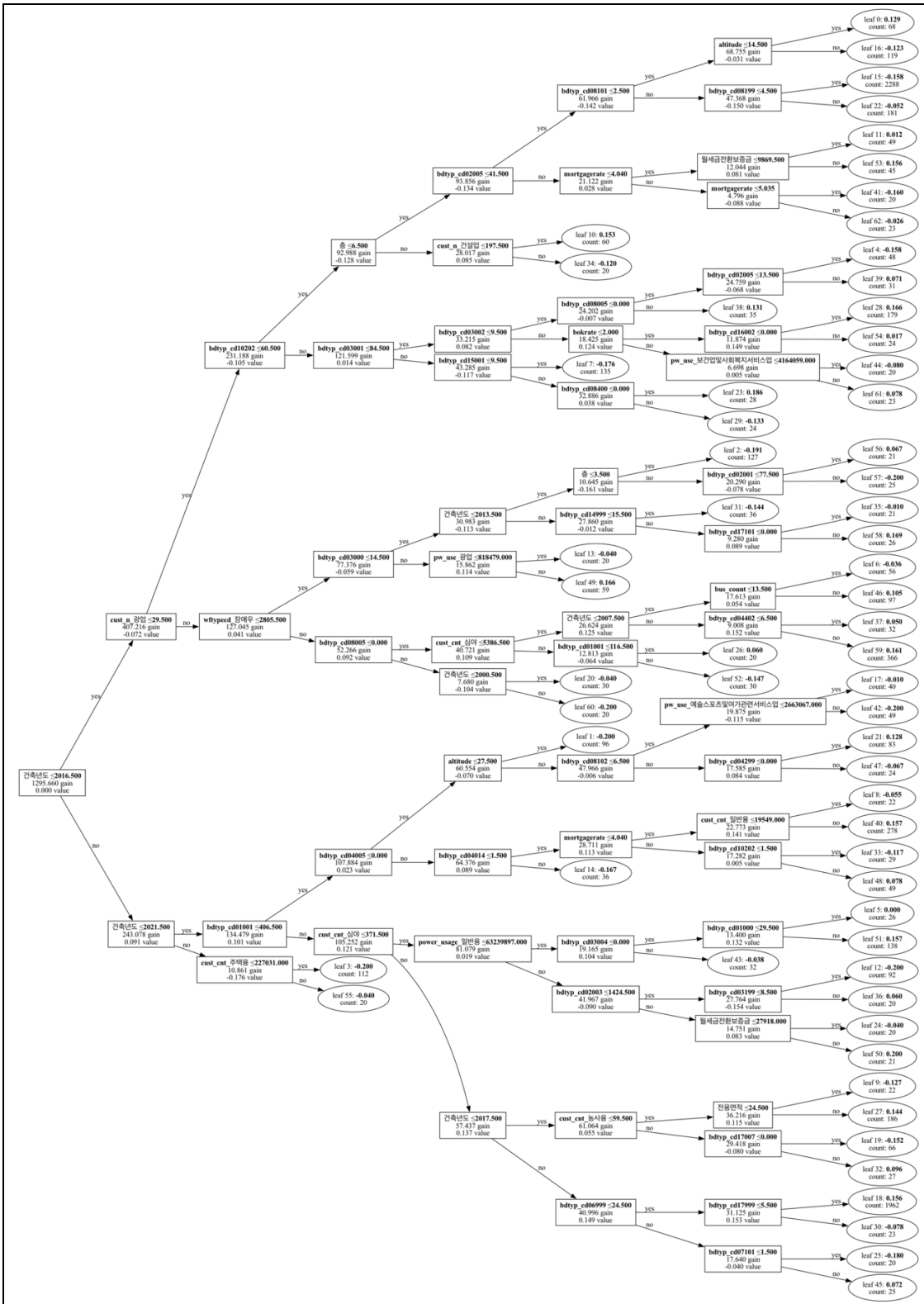
〈Figure 7〉 주택유형별 Classification Report

클래스간 불균형이 없으며, f1 score 또한 우수하여 과적합 우려가 없다.
〈Figure 8〉은 주택유형별 classifier 의 tree plot 이다.

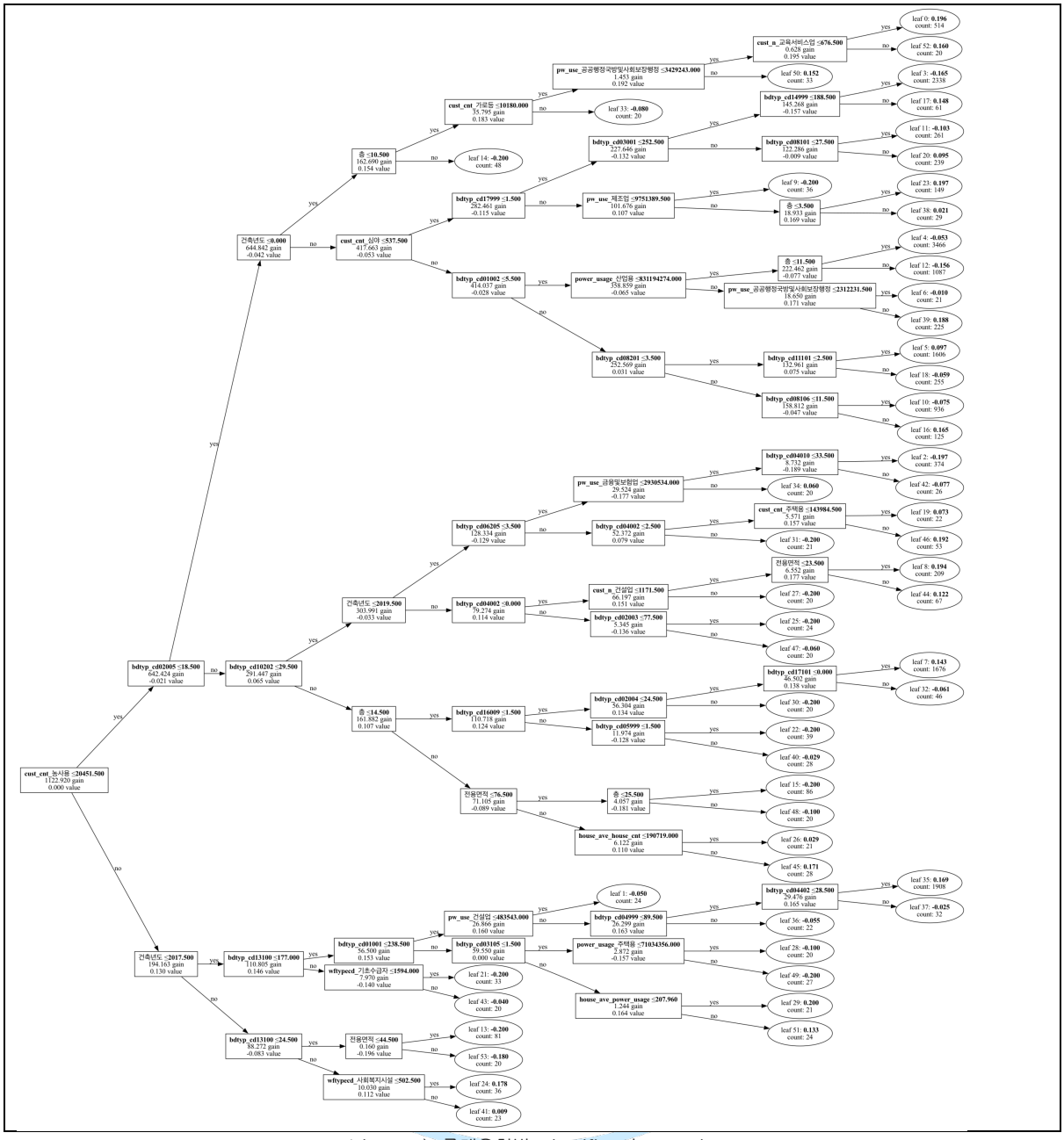
아파트



다세대주택


건축년도 ≤2016.500
1295.660 gain
0.000 value
yes
no
cust_n_광업 ≤29.500
407.216 gain
-0.072 value
wftypecd_창애우 ≤2805.500
127.045 gain
0.041 value
bdtyp_cd10202 ≤60.500
231.188 gain
-0.105 value
층 ≤-6.500
92.998 gain
-0.128 value
bdtyp_cd02005 ≤41.500
93.856 gain
-0.134 value
bdtyp_cd08101 ≤2.500
61.966 gain
-0.142 value
altitude ≤14.500
68.755 gain
-0.031 value
leaf 0: 0.129
count: 68
leaf 16: -0.123
count: 119
bdtyp_cd08199 ≤4.500
47.368 gain
-0.150 value
leaf 15: -0.158
count: 2288
leaf 22: -0.052
count: 181
mortgagerate ≤4.040
21.122 gain
0.028 value
월세금전환보증금 ≤9869.500
12.044 gain
0.081 value
leaf 11: 0.012
count: 49
leaf 53: 0.156
count: 45
mortgagerate ≤5.035
4.796 gain
-0.088 value
leaf 41: -0.160
count: 20
leaf 62: -0.026
count: 23
cust_n_건설업 ≤197.500
28.017 gain
0.085 value
leaf 10: 0.153
count: 60
leaf 34: -0.120
count: 20
bdtyp_cd03001 ≤84.500
121.599 gain
0.014 value
bdtyp_cd03002 ≤9.500
33.215 gain
0.082 value
bdtyp_cd08005 ≤0.000
24.202 gain
-0.007 value
bdtyp_cd02005 ≤13.500
24.759 gain
-0.068 value
leaf 4: -0.158
count: 48
leaf 39: 0.071
count: 31
leaf 38: 0.131
count: 35
bokrate ≤2.000
18.425 gain
0.124 value
bdtyp_cd16002 ≤0.000
11.874 gain
0.149 value
leaf 28: 0.166
count: 179
leaf 54: 0.017
count: 24
pw_use_보건업및사회복지서비스업 ≤4164059.000
6.698 gain
0.005 value
leaf 44: -0.080
count: 20
leaf 61: 0.078
count: 23
bdtyp_cd15001 ≤9.500
43.285 gain
-0.117 value
leaf 7: -0.176
count: 135
bdtyp_cd08400 ≤0.000
32.886 gain
0.038 value
leaf 23: 0.186
count: 28
leaf 29: -0.133
count: 24
wftypecd_창애우 ≤2805.500
127.045 gain
0.041 value
bdtyp_cd03000 ≤14.500
77.376 gain
-0.059 value
건축년도 ≤2013.500
30.983 gain
-0.113 value
층 ≤3.500
10.645 gain
-0.161 value
leaf 2: -0.191
count: 127
bdtyp_cd02001 ≤77.500
20.290 gain
-0.078 value
leaf 56: 0.067
count: 21
leaf 57: -0.200
count: 25
bdtyp_cd14999 ≤15.500
27.860 gain
-0.012 value
leaf 31: -0.144
count: 36
bdtyp_cd17101 ≤0.000
9.280 gain
0.089 value
leaf 35: -0.010
count: 21
leaf 58: 0.169
count: 26
pw_use_광업 ≤818479.000
15.862 gain
0.114 value
leaf 13: -0.040
count: 20
leaf 49: 0.166
count: 59
bdtyp_cd08005 ≤0.000
52.266 gain
0.092 value
cust_cnt_심야 ≤5386.500
40.721 gain
0.109 value
건축년도 ≤2007.500
26.624 gain
0.125 value
bus_count ≤13.500
17.613 gain
0.054 value
leaf 6: -0.036
count: 56
leaf 46: 0.105
count: 97
bdtyp_cd04402 ≤6.500
9.008 gain
0.152 value
leaf 37: 0.050
count: 32
leaf 59: 0.161
count: 366
bdtyp_cd01001 ≤116.500
12.813 gain
-0.064 value
leaf 26: 0.060
count: 20
leaf 52: -0.147
count: 30
건축년도 ≤2000.500
7.680 gain
-0.104 value
leaf 20: -0.040
count: 30
leaf 60: -0.200
count: 20
pw_use_예술스포츠및여가관련서비스업 ≤2663067.000
19.875 gain
-0.115 value
leaf 17: -0.010
count: 40
leaf 42: -0.200
count: 49
건축년도 ≤2021.500
243.078 gain
0.091 value
bdtyp_cd01001 ≤406.500
134.479 gain
0.101 value
bdtyp_cd04005 ≤0.000
107.884 gain
0.023 value
altitude ≤27.500
60.554 gain
-0.070 value
leaf 1: -0.200
count: 96
bdtyp_cd08102 ≤6.500
47.966 gain
-0.006 value
bdtyp_cd04299 ≤0.000
17.585 gain
0.084 value
leaf 21: 0.128
count: 83
leaf 47: -0.067
count: 24
bdtyp_cd04014 ≤1.500
64.376 gain
0.089 value
mortgagerate ≤4.040
28.711 gain
0.113 value
cust_cnt_일반용 ≤19549.000
22.773 gain
0.141 value
leaf 8: -0.055
count: 22
leaf 40: 0.157
count: 278
bdtyp_cd10202 ≤1.500
17.282 gain
0.005 value
leaf 33: -0.117
count: 29
leaf 48: 0.078
count: 49
leaf 14: -0.167
count: 36
cust_cnt_주택용 ≤227031.000
10.861 gain
-0.176 value
leaf 3: -0.200
count: 112
leaf 55: -0.040
count: 20
cust_cnt_심야 ≤371.500
105.252 gain
0.121 value
power_usage_일반용 ≤63239897.000
61.079 gain
0.019 value
bdtyp_cd03004 ≤0.000
19.165 gain
0.104 value
bdtyp_cd01000 ≤29.500
13.400 gain
0.132 value
leaf 5: 0.000
count: 26
leaf 51: 0.157
count: 138
leaf 43: -0.038
count: 32
bdtyp_cd02003 ≤1424.500
41.967 gain
-0.090 value
bdtyp_cd03199 ≤8.500
27.764 gain
-0.154 value
leaf 12: -0.200
count: 92
leaf 36: 0.060
count: 20
월세금전환보증금 ≤27918.000
14.751 gain
0.083 value
leaf 24: -0.040
count: 20
leaf 50: 0.200
count: 21
건축년도 ≤2017.500
57.437 gain
0.137 value
cust_cnt_농사용 ≤59.500
61.064 gain
0.055 value
전용면적 ≤24.500
36.216 gain
0.115 value
leaf 9: -0.127
count: 22
leaf 27: 0.144
count: 186
bdtyp_cd17007 ≤0.000
29.418 gain
-0.080 value
leaf 19: -0.152
count: 66
leaf 32: 0.096
count: 27
bdtyp_cd06999 ≤24.500
40.996 gain
0.149 value
bdtyp_cd17999 ≤5.500
31.125 gain
0.153 value
leaf 18: 0.156
count: 1962
leaf 30: -0.078
count: 23
bdtyp_cd07101 ≤1.500
17.640 gain
-0.040 value
leaf 25: -0.180
count: 20
leaf 45: 0.072
count: 25

오피스텔



〈Figure 8〉 주택유형별 Classifier 의 Tree Plot

4 결론과 토의

본 연구에서는 기계학습을 통하여 전세계약 이후 임대인의 보증금 반환이 이루어지지 않는 요주의 전세계약을 시계열 지표, 기초자치단체 단위의 통계, 개별 인스턴스의 주변 입지를 바탕으로 분류할 수 있는지 확인하였다.

주택유형과 무관히 층, 월세금전환보증금, 건축년도, 전용면적, 소재지의 인구를 반영하는 아파트 수, 다가구주택 수, 단독주택 수 변수가 분류에 유의미하게 작용했다. 아파트와 다세대주택 모델은 학교 수와 가구평균전기사용량, 교회 변수가 분류에 유의미하게 작용하였는데, 이는 전통적인 가족 단위의 정주환경의 우수함이 영향을 미쳤다. 노동자의 수요가 높은 일반공장 변수가 아파트 모델의 분류에, 금융및보험업과 전문과학및기술업 변수가 오피스텔 모델의 분류에 유의미하게 작용한 것은 각

주택유형에 따른 주택 수요층이 차별화되는 것을 의미한다. 이 밖에 버스정류장 수, 주택담보대출 금리가 공통적으로 분류에 유의미하게 작용한 것과 같이 주택유형별 분류기는 일반적으로 집값에 영향을 미치는 요인들을 충실히 설명하고 있음을 알 수 있다. 분류 모델이 사회 통념상 설명 가능한 규칙에 따라 요주의 전세계약들을 분류하고 있음을 확인하였으므로, 본 연구가 제안하는 분류모델은 공개된 정보인 실거래 정보를 기반으로 요주의 전세계약을 사전에 식별할 수 있는 잠재력을 지닌다. 이는 향후 개인이나 부동산에서 전세보증금 반환 여력이 없는 물건을 사전에 차단하거나 금융기관의 보증보험 심사 자동화 시스템 구축의 실질적인 기반이 될 수 있다.

한계로는 국토부에서 제공하는 실거래가공개시스템의 데이터셋이 완전히 법원경매의 물건과 일치하지 않은 문제가 있다. 이는 실거래가공개시스템에서 제공하는 데이터셋에 거래 물건의 동호수와 같은 상세주소가 표기되어 있지 않기 때문이다. 또한 부동산 거래가 서울 및 수도권과 영남지방에 편중되어 있어 학습에 이용한 자료의 편향성에 대한 우려가 있다. 상대적으로 적은 부동산 거래가 이루어지는 지방에서는 분류기의 성능을 장담할 수 없다.

향후 연구에서 실제 동호수 표기가 되어 있는 데이터셋을 이용하거나, 익명화 가공하여 보증보험 대위변제 혹은 법원경매 레코드와 연계된 데이터셋을 이용하여 더욱 범용적이고 정확한 모델을 만들 필요가 있다.

이번 프로젝트를 진행하면서 특히 원천 데이터의 클래스 불균형 문제를 해소하는데 방점을 두었습니다. 확보한 샘플들을 최대한 학습에 반영하면서 평가를 왜곡하지 않도록 접근하는 방법에서 많은 것을 배웠습니다. 또한 이번 프로젝트에서 처음으로 지리데이터를 다루어 보았는데, PostgreSQL 과 같은 RDBMS 에 PostGIS 와 같은 익스텐션을 설치해서 데이터베이스에서 접하지 못했던 새로운 자료형을 다루는 경험을 할 수 있었습니다. 수 백만 건의 엔티티를 처리하면서 RDBMS 인덱스 기능의 중요성을 깨달았습니다. 최초 프로젝트를 시작할 때에는 문자열을 잠깐 저장할 생각으로 SQLite 를 사용하였으나, 쿼리 성능이 미달하여 PostgreSQL 로 전환하게 되었습니다. 이후 쿼리성능 향상을 확인할 수 있었습니다. PostgreSQL 과 연동하여 QGIS 에서 시각화하는 기능을 이번 프로젝트 간 배웠습니다. 지리데이터를 시각화하는데 유용함을 확인할 수 있었습니다.

5 참고문헌

국토교통부, 실거래가 공개시스템, https://rt.molit.go.kr/pt/xls/xls.do.
대한민국 대법원, 대한민국 법원 법원경매정보, https://www.courtauction.go.kr.
한국은행, 통화정책방향-한국은행기준금리추이, https://www.bok.or.kr/portal/singl/baseRate/list.do?dataSeCd=01&menuNo=200643.
은행연합회, 가계대출금리 - 은행별 비교공시, https://portal.kfb.or.kr/compare/loan_household_new.php.
한국부동산원, 전국주택가격동향조사, 2025.03, 지역별 전월세전환율.
통계청, 10 차 개정 표준산업분류(KSIC), KSSC 통계청 통계분류포털.
행정안전부 한국지역정보개발원, 위치정보요약 DB, https://business.juso.go.kr.
행정안전부 한국지역정보개발원, 도로명주소 전자지도 DB, https://business.juso.go.kr.
공공데이터포털, 국토교통부_전국 버스정류장 위치정보, https://www.data.go.kr/data/15067528/fileData.do.
주소기반산업지원서비스, PC 용 주소검색기, https://business.juso.go.kr/addrlink/tchnlgySport/pcAdresFinder.do.
Decompiler - Disassemble, decompile and analyze binary files online, https://www.decompiler.com.
Open Topo Data, Open Topo Data introduction, https://www.opentopodata.org.
KB 부동산, KB 부동산 데이터허브, https://data.kbland.kr/kbstats/wmh?tIdx=HT08&tsIdx=aptQteRentPrice.
국토교통부, 국토교통부 부동산소비자보호기획단, https://rt.molit.go.kr/pt/info/info.do.
부동산 거래신고 등에 관한 법률 [법률 제 20194 호, 2024 년 2 월 6 일, 시행 2024 년 5 월 17 일], 제 6 조의 2(주택 임대차 계약의 신고).
부동산 거래신고 등에 관한 법률 [법률 제 20194 호, 2024 년 2 월 6 일, 시행 2024 년 5 월 17 일], 제 5 조(신고 내용의 검증).

QGIS Development Team. (2025). QGIS Geographic Information System (Version 3.x) [Computer software]. Open Source Geospatial Foundation. https://qgis.org