學 士 學 位 論 文

이미지 처리와 머신러닝을 이용한 도면 분석과 전용 면적 예측 시스템 개발

충남대학교

공과대학 컴퓨터공학과

성낙현
이재원
장수훈

지도교수 **임 성 수**

2021 年 2 月

이미지 처리와 머신러닝을 이용한
도면 분석과 전용 면적 예측 시스템 개발

지도교수 임 성 수

이 논문을 공학사학위
청구논문으로 제출함

2020 年 11 月

충 남 대 학 교

공과대학 컴퓨터공학과

201402363 성 낙 현
201402401 이 재 원
201402414 장 수 훈

목 차

그림목차

Ⅰ. 서 론

1. 연구 목적과 방법론

1.1) 연구의 배경 및 목적

부동산 시장의 활성화에 따라 많은 부동산 매물이 등록되지만 이 중에는 허위 매물이 섞여 있어 정확한 판단에 어려움을 겪고 있는 이들이 많은 실정이다. 단적인 예시로 매일경제의 한 기사의 자료[1]를 보면 국토교통부에 따르면 2020년 8월21일부터 9월20일 까지 한 달간 부동산 광고시장 감시센터에 신고 된 허위매물은 총 1천507건으로 집계 되었다고 한다. 이중 정보 명시 의무 위반이 50.1%(755건)이었다. 정보 명시 의무 위반의 경우에서도 특히 전용 면적의 경우 허위로 기재되어 있거나 기재가 누락 되었을 때 일반적인 경우 방이나 화장실의 개수 등 직접 방을 둘러보면 손쉽게 알 수 있는 다른 정보에 비해 확인이 어려운 점이 있다. 따라서 면적의 정보가 허위기재 혹은 누락 되었을 경우 면적을 높은 정확도로 예측할 수 있는 방안을 모색 할 필요가 있다.

이에 본 논문은 최근 많은 분야에서 연구, 활용되고 있는 기계학습(machine learning)을 활용하여, 부동산 매물의 도면과 간단히 확인 할 수 있는 정보들만을 사용하여 면적을 예측 할 수 있는 인공지능 모델을 만든다. 전용면적 예측에 필요한 정보와 적합한 알고리즘을 찾아 전용면적 예측의 오차를 최소화 하여 정확한 면적 예측 모델을 만드는 것을 목적으로 한다. 이 연구의 결과물을 사용한다면 조사 기관에서 사용한다면 부동산 허위 매물을 조금 더 쉽게 적발 할 수 있으며 일반적인 사용자가 사용한다면 부동산 허위 매물에 속아 피해를 입는 사례를 줄일 수 있을 것으로 기대할 수 있다.

1.2) 연구의 범위 및 방법

본 연구는 수집 할 데이터 선정, 크롤링, 데이터 전처리, 기계학습, 결과도출의 과정으로 이루어졌다.

먼저 분석을 할 데이터 수집을 위하여 여러 부동산 사이트를 조회한 결과 비교적 데이터의 정형화가 잘 이루어져 있으며 신뢰할 수 있을 만한 사이트를 선정하였다. 해당 사이트의 허가를 구하고 대전 유성구, 서구, 대덕구의 부동산 매물을 각 300개가량 크롤링 하였다. 이 과정에서 Selenium, ChromDriver, BeautifuSoup을 사용하였다. 크롤링 후 OpenCV라이브러리를 사용하여 도면 이미지에서 예측 모델에 필요한 정보를 도출 해 냈다. 이미지에서 도출한 정보와 웹페이지에서 수집한 정보를 합쳐 단일 csv파일로 만들어 데이터 전처리를 마친다.

기계학습을 통해 전용면적을 예측해본 결과 지역별 특성에 따른 매물정보의 편차가 존재하여 유성구, 대덕구, 서구 세 지역의 정보를 각각 나누어 학습 시켜 연구를 진행하여 정확도를 도출한다.

논문 구성의 구성은 다음과 같다. 먼저 데이터 수집 대상을 선정하며 수집 방법, 수집 결과에 대해 설명한 뒤, 데이터의 정제 방법을 소개한다. 기계학습의 방법과 과정으로 정확도 측정 방법을 설명하며, 실험 결과에서 오류 확인 및 수정하며 고찰하여 마무리한다.

<그림 1>Selenium

1.3) 연구 환경

Operating System : Windows 10
Processor : 6-core Intel i5 9400 (6thread)
Memory : 2 × 8 GB DDR4 (16GB)
Flash Storage : 1 × 1TB PCIe
SSD : 500GB SATA
Graphics Card : NVIDIA GTX 1660ti 6GB
NVIDIA Driver : version 457.51

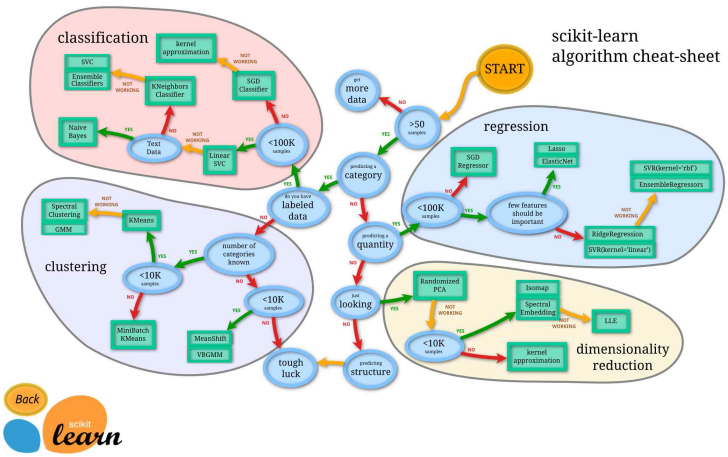
1.3.1) Anaconda를 사용한 개발환경

아나콘다(Anaconda)는 패키지 관리와 디플로이를 단순케 할 목적으로 과학 계산(데이터 과학, 기계 학습 애플리케이션, 대규모 데이터 처리, 예측 분석 등)을 위해 Python과 R 프로그래밍 언어의 자유-오픈소스를 모아 놓은 플랫폼이다. 패키지 버전들은 패키지 관리 시스템 conda를 통해 관리되며, 아나콘다는 윈도우, 리눅스, macOS에 적합한 1,400개 이상의 유명 데이터 과학 패키지가 포함되어 있다. 본 연구는 python 3.7.4 vression 과 scikit-learn 0.21.3 version을 사용 하여 개발하였다.

<그림 2> 아나콘다

1.3.2) scikit-learn을 사용한 기계학습

scikit-learn(sklearn)은 Python 프로그래밍 언어의 머신러닝 라이브러리의 일종이다. scikit-learn의 장점은 오픈소스 이며 현재도 개발이 이루어지고 있어서 정보를 찾기도 쉽다. 또한 라이브러리 외적으로는 scikit 스택을 사용하고 있기 때문에 다른 라이브러리와의 호환성이 좋다. 내적으로는 통일된 인터페이스를 가지고 있기 때문에 매우 간단하게 여러 기법을 적용할 수 있어 쉽고 빠르게 최상의 결과를 얻을 수 있다.



<그림 3> scikit-learn cheat-sheet

1.3.3) OpenCV를 이용한 이미지 처리

OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리이다. 또한 컴퓨터 비전 애플리케이션을 위한 공통 인프라를 제공하고 상용 제품에서 기계 인식의 사용을 가속화하기 위해 구축되었다. 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 인텔 CPU에서 사용되는 경우 속도의 향상을 볼 수 있는 IPP(Intel Performance Primitives)를 지원한다. 이 라이브러리는 윈도, 리눅스 등에서 사용 가능한 크로스 플랫폼이며 오픈소스 BSD 허가서 하에서 무료로 사용할 수 있다. OpenCV는 TensorFlow , Torch / PyTorch 및 Caffe의 딥러닝 프레임워크를 지원한다.

1.3.4) Jupyter Notebook, Pycharm

본 프로젝트의 데이터 수집과 기계학습 과정에서는 Jupyter Notebook을 쓰고, 이미지 처리 과정에서는 Pycharm을 활용한다. 각 개발 툴에 대한 특징을 설명하고 왜 분리해서 사용하는지 알아본다.

Jupyter notebook은 데이터 분석을 절차대로 실행하면서 확인이 가능하다. 때문에 특정 단계에서의 결과를 봐가면서 순차적인 확인이 가능하다는 뜻이다. 즉, 상호 대화식으로 동작한다. 중간에 그래프를 그리거나 이미지를 출력하는 등 다양한 시각화 효과까지 곁들일 수 있다. 무엇보다 쉽다. 파이썬 노트북의 문서 작성 마크업 언어인 마크다운 언어는 복잡하지 않으므로 금방 배울 수 있으며, 노트북의 편집기도 특별히 학습해야만 사용가능한 복잡한 기능이 없으므로 쉽게 익숙해질 수 있다.

Pycharm으로 프로젝트에서 활용된 것은 이미지 처리 과정이다. 함수 관리의 용이성에서 코드가 길어지면 함수나 클래스 여러 개가 생기면서 구조를 파악하기 어렵다. Pycharm은 이러한 용이성 면에서 이미지 처리 과정에 적합하다고 판단했다.

Ⅱ. 본 론

1. 데이터 수집

오늘날 스마트폰의 보급과 SNS 및 웹 사이트의 발달로 정형/비정형 빅데이터는 기하급수적으로 증가하였다. 이러한 빅데이터를 잘 분석한다면 미래 예측도 가능할 만큼 훌륭한 정보를 얻을 수 있다. 빅데이터를 분석하기 위해서는 먼저 대용량의 데이터 수집이 필요하다. 이러한 데이터가 가장 많이 저장되어 있는 곳은 바로 웹페이지다. 하지만 데이터의 양이 방대하기 때문에 유용한 정보를 가진 데이터가 많은 만큼 필요하지 않은 정보를 가진 데이터도 많이 존재한다. 그렇기 때문에 필요하지 않은 정보를 가진 데이터는 거르고 유용한 정보를 가진 데이터만을 수집하는 효율적인 데이터 수집의 중요성이 대두되었다. 웹 크롤러는 네트워크 대역폭, 시간적인 문제, 하드웨어적인 저장소 등의 제약으로 인해 모든 페이지를 다운로드 할 수 없다. 그렇기 때문에 원하는 내용과 관련 없는 많은 페이지들의 방문은 피하며 가능한 빠른 시간 내에 중요한 페이지만을 다운로드해야한다.[2]

본 프로젝트에서 필요한 DB, 사진 등 모든 콘텐츠에 대한 저작권이 있고, 저작권법에 보호를 받기 때문에 부동산 사이트 저작권 관련 담당자에게 크롤링 허가를 받은 데이터들을 활용하였다.

1.1) 데이터 수집 방법 및 목적

Selenium은 주로 웹앱을 테스트하는데 이용하는 프레임워크이지만, webdriver라는 API를 통해 운영체제에 설치된 Chrome등의 브라우저를 제어할 수 있다. Selenium을 통해서 렌더링이 완료된 후의 웹 DOM 결과물에 접근이 가능하다. html 태그 요소에 접근하는 방법으로 Beautiful Soup 라이브러리를 같이 사용하면 html 태그에 쉽게 접근할 수 있다.

왜 Beautiful Soup를 사용하는가?

- 매번 웹사이트를 방문해서 데이터를 수집하고, html5lib를 사용해서 크롤링을 해왔는데, html5lib가 나무구조여서 차원이 너무 많아서 원하는 요소를 뽑아오기 쉽지 않았다.
- Beautiful Soup를 사용하면 원하는 요소를 쉽게 가져올 수 있다.

webdriver를 통해서 크롬 환경에서 부동산114의 매물페이지에 접근한다. 부동산114의 매물페이지에서는 페이지마다 50개의 매물 리스트가 존재한다. 따라서 각 페이지마다 50개의 매물 정보의 url을 자동적으로 접근하여 총 1500개의 매물을 자동 크롤링 하도록 하였다.

```
str123 = ""
for i in range(len(detail_url)):
    str123 = "https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=" + list_test[i]
    detail_page.append(str123)
```

```
for i in range(len(detail_page)):
        print(i+1)
        print(detail_page[i])
        print(" ")
```

```
45
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=00D39928DD15DF

46
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=007B23F848F2CA

47
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=00D6A4DD7741DF

48
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=0003A94B1A09A2

49
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=00059228B3C23F

50
https://www.r114.com/?_c=memul&_m=HouseDetail&mulcode=00CF2AAAC506F8
```

〈그림 4〉 매물정보 url 주소로 접근하여 페이지 당 50개에 해당하는 매물 정보 크롤링

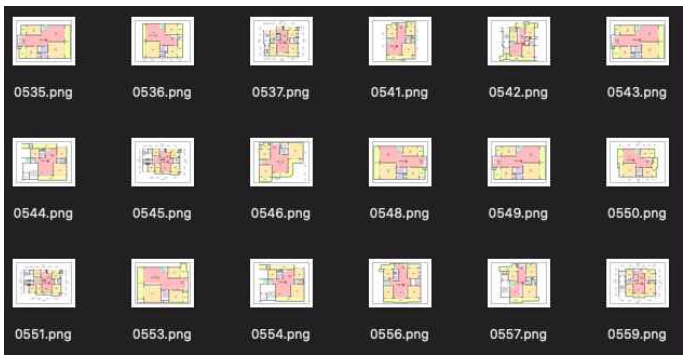
1.2) 데이터 수집 결과

본 프로젝트에서 크롤링한 데이터들은 매물 상세 정보와 평면도(plane map) 이미지이다. 매물 상세 정보로는 전용면적, 전용률, 방 개수, 욕실 개수, 복도식/계단식, 월 관리비이다. 평면도 이미지는 index 번호를 부여해서 저장해준다.

본 프로젝트의 목적은 이미지를 이용해서 전용 면적을 예측하는 것이다. 수집된 매물 정보들 중에서 전용 면적은 라벨로 사용되고 나머지 데이터들은 입력 값으로 받게 된다. 매물 상세 정보에 포함되지 않은 정보들은 저장된 이미지에서 특징 값을 추출하여 새로운 입력 값으로 함께 활용되도록 한다.

```
1
https://image.r114.co.kr/imgdata/planemap/07/01/a0701306040000300032z.gif
['84.9㎡', '78%', '3개 / 2개 ', '계단식', '160,000원', './image/0890.jpg']

2
https://image.r114.co.kr/imgdata/planemap/07/01/a0701306040000200032z.gif
['84.45㎡', '80%', '3개 / 2개 ', '계단식', '160,000원', './image/0891.jpg']

3
https://image.r114.co.kr/imgdata/planemap/07/01/a070130604000030024B.gif
['59.94㎡', '75%', '3개 / 1개 ', '계단식', '140,000원', './image/0892.jpg']

4
슬라이드

5
https://image.r114.co.kr/imgdata/planemap/07/01/a0701306040000200032z.gif
['84.45㎡', '80%', '3개 / 2개 ', '계단식', '180,000원', './image/0893.jpg']
```

<그림 5> 크롤링으로 html tag 접근을 통해 얻은 매물 상세 정보와 이미지 index



<그림 6> 해당 매물정보에 대한 plane map 이미지 저장

2. 데이터 전처리

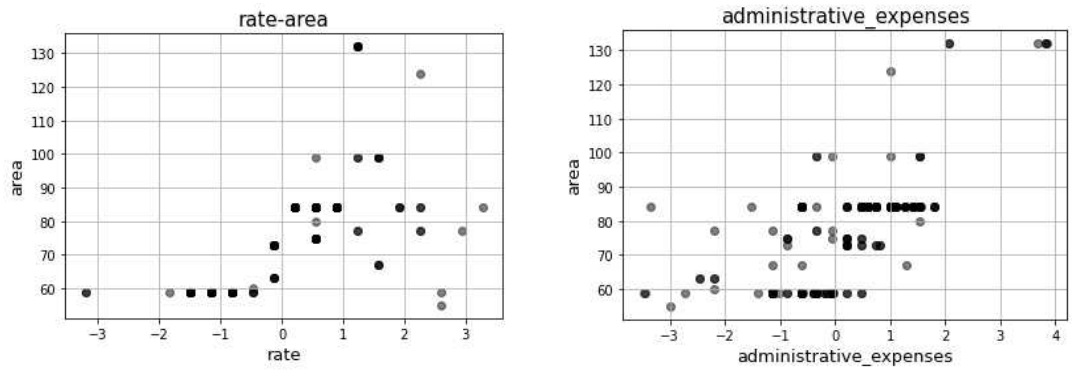
데이터 처리(preprocessing)는 일반적으로 컴퓨터에서 자동으로 실행된다. 자료는 올바르게 표현되면 편리하고 실용적인 정보가 되기 때문에, 데이터 처리 시스템은 실용성을 강조하기 위해 정보 시스템이라고도 일컬었다. 이러한 용어는 거의 같은 뜻이며, 데이터 처리 시스템이 자료를 조작해 정보를 만드는 데 반해, 정보 시스템은 자료를 입력하여 정보를 출력한다.
일반적으로 말해, 데이터 처리는 데이터를 기존의 형식으로부터 다른 형식으로 변환하는 과정이라고 정의할 수 있다. 그러나 그 뜻에는 데이터 변환이라는 용어가 적절하다. 이 관점에서 데이터 처리는 정보를 데이터로 변환하는 과정과 데이터를 정보로 변환하는 과정을 가리킨다. 데이터 처리와 데이터 변환의 차이는, 데이터 변환에서는 응답해야 할 쿼리를 필요하지 않다는 점에 있다. 이를테면, 영어의 문장을 형성하는 문자열(string) 형식의 정보는, 키보드의 키 입력으로부터 인코딩을 받아 하드웨어 방식의 코드가 되고, 또 아스키 코드로 바뀐 다음 글꼴로 변환되어 디스플레이에 보여 준다. 이것은 최종적으로 인간이 이해할 수 있는 의미가 있는 정보가 되는 예이다.
다만 이러한 예는 데이터 처리보다는 전송 시스템이나 운영 체제에 의한 하드웨어 제어의 관점에서 자주 언급된다. 일반적으로 데이터 처리라는 용어가 쓰이게 된 것은 업무를 위한 다수의 데이터를 모아서 그것들을 정보 이용자들이 쓰기 쉽도록 가치 있는 정보로 제시하는 과정에서 비롯한 것이다.

2.1) 문자와 숫자 데이터 전처리

<그림 5>에서 추출한 데이터 정보들은 그대로 기계학습에 적용할 수 없다. 데이터를 다룬다는 것은 사실 문자(string)와 숫자(number)를 다루는 것이다. 그리고 분석을 위한 데이터를 만드는 것은 문자를 숫자화 하는 과정이 대부분이다. 이 때 정규화, 표준화, 비율, 요약 등 다양한 방법이 적용될 수 있다. csv형식으로 데이터들을 저장하되, 저장하기 전에 알맞은 데이터 형식으로 바꾸어 주고 pandas dataframe으로 만들어진 테이블을 저장한다.

	con_rate	bedroom	bath	porch_structure	administrative_expenses	main_area	img_url
0	73.0	3.0	1.0	1	150000.0	59	./image/0890.jpg
1	85.0	4.0	2.0	1	260000.0	101	./image/0891.jpg
3	72.0	2.0	1.0	0	50000.0	41	./image/0893.jpg
4	68.0	2.0	1.0	0	95000.0	39	./image/0894.jpg
5	83.0	3.0	2.0	1	266579.0	84	./image/0895.jpg
6	78.0	3.0	2.0	1	150000.0	84	./image/0896.jpg
8	79.0	3.0	1.0	1	221560.0	70	./image/0898.jpg
10	72.0	2.0	1.0	0	50000.0	41	./image/0900.jpg
11	79.0	3.0	1.0	1	219522.0	70	./image/0901.jpg
13	84.0	3.0	2.0	1	170000.0	84	./image/0903.jpg
14	82.0	3.0	2.0	1	186024.0	84	./image/0904.jpg
15	83.0	3.0	2.0	1	140000.0	84	./image/0905.jpg
16	78.0	3.0	2.0	2	200000.0	84	./image/0906.jpg
17	86.0	4.0	2.0	1	200000.0	134	./image/0907.jpg

<그림 7> 문자형 데이터를 숫자형으로 전처리 한 dataframe



<그림 8> 각 변수들과 전용면적과의 상관관계

2.2) 이미지 전처리

영상 처리(Image processing) 또는 화상 처리는 넓게는 입출력이 영상인 모든 형태의 정보 처리를 가리키며, 사진이나 동영상을 처리하는 것이 대표적인 예이다. 대부분의 영상 처리 기법은 화상을 2차원 신호로 보고 여기에 표준적인 신호 처리 기법을 적용하는 방법을 쓴다.
20세기 중반까지 영상 처리는 아날로그로 이루어졌으며, 대부분 광학과 연관된 방법이었다. 이런 영상 처리는 현재까지도 홀로그래피 등에 사용되지만, 컴퓨터 처리 속도의 향상으로 인해 이런 기법들은 디지털 영상 처리 기법으로 많이 대체되었다. 일반적으로 디지털 영상 처리는 다양한 방법으로 쓰일 수 있으며 정확하다는 장점이 있고, 아날로그보다 구현하기 쉽기도 하다. 더 빠른 처리를 위해서 파이프라인과 같은 컴퓨터 기술들이 쓰이기도 한다.

2.2.1) Contour Detection

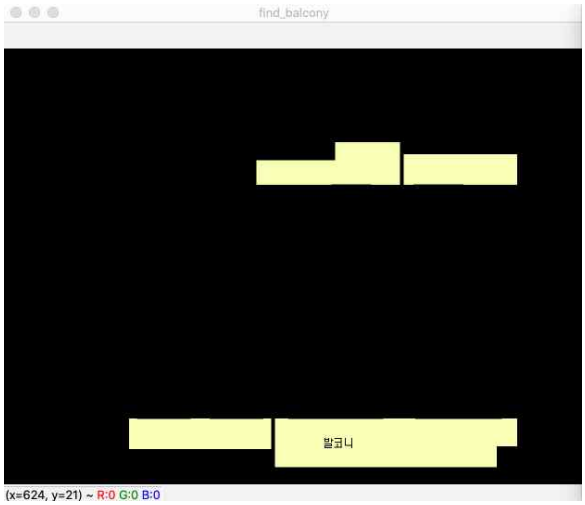
Contour란 같은 값을 가진 곳을 연결한 선이라고 생각하면 된다. 우리 주위에 자주 접할 수 있는 Contour의 예로 지도에서 같은 높이를 가진 지점을 연결한 등고선, 기상도 같은 연결선이 있다. 이미지 Contour란 동일한 색 또는 동일한 색상 강도(Color Intensity)를 가진 부분의 가장 자리 경계를 연결한 선이다. 보다 정확한 이미지 Contour를 확보하기 위해 바이너리 이미지를 사용한다. 즉, 이미지에서 contour를 찾기 전에 threshold를 사용한 컬러 경계를 구분하는 것이다.

<그림 9> 평면도 Original Image

2.2.2) 이미지 전처리 결과

본 프로젝트에서는 매물 세부 정보에 기재되지 않은 발코니의 수를 contour detection을 통해서 csv파일에 발코니의 수를 추가 입력하려고 한다. contour detection에 관한 함수들은 opencv 라이브러리를 활용한다.

녹색에 대한 threshold 탐지 색의 BGR 범위를 (30, 30, 30)에서 (70, 255, 255)로 설정하였다. 필터를 이용하여 각 픽셀을 탐색하면서 해당 threshold에 해당되는 픽셀들만 추출하여 이미지로 출력을 하면 다음과 같은 이미지를 얻을 수 있다.



<그림 10> 추출한 balcony 이미지

Opencv의 cv2.findContours() 함수는 Suzuki85라는 알고리즘을 이용해서 이미지에서 Contour를 찾는 함수이다. 이 함수는 원본 이미지를 변경시키기 때문에 원본이미지의 복사본을 가지고 contour하도록 한다. opencv에서 contour찾기는 검정색 배경에서 흰색 물체를 찾는 것과 같다. 따라서, Contour를 찾고자 하는 대상은 특정 색깔로 지정하고 배경은 검정색으로 변경해야 한다.

contour 근사법을 이용하기 위하여 opencv의 CHAIN_APPROX_NONE 인자를 사용한다. 이 인자는 contour를 구성하는 모든 점을 저장한다. drawContour()를 수행하면 우리가 찾는 contour를 실제로 그려주고, 완성된 경계선으로 다음과 같은 이미지를 얻을 수 있다.

<그림 11> contour detection으로 얻어진 발코니 이미지

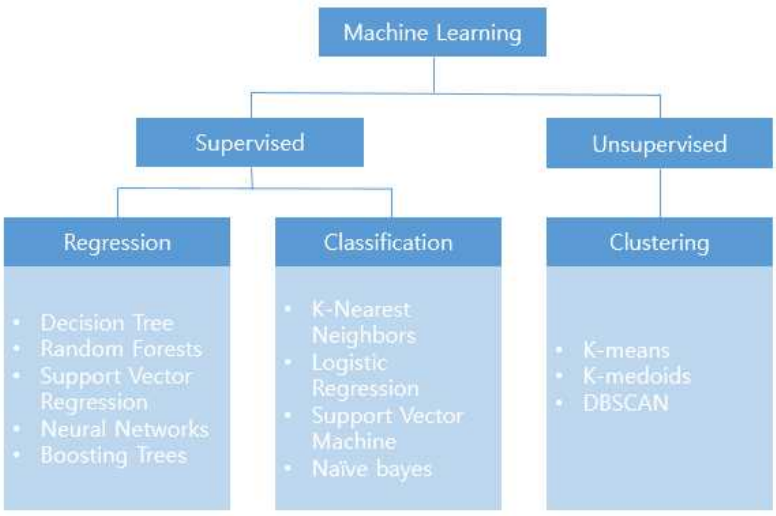
contour detection을 통해서 얻어진 발코니의 개수를 세고 csv 파일에 추가해서 저장해준다. <그림 11>에서 해당되는 발코니 라벨은 총 4개로 csv 파일 column에 4개의 발코니가 추가된다. 따라서 기계학습에서 사용될 csv 테이블은 다음 형식으로 나온다. 순서대로 전용률, 방, 화장실, 현관구조, 관리비, 전용면적, 발코니 이다.

80	3	2	1	84	./image/0002.png	4
75	3	1	1	59	./image/0003.png	3
78	3	2	1	84	./image/0004.png	4
80	3	2	1	84	./image/0005.png	3
79	3	2	1	84	./image/0006.png	3
79	3	2	1	75	./image/0007.png	3
75	3	1	1	59	./image/0008.png	3
74	3	2	1	59	./image/0009.png	4
80	3	2	1	84	./image/0010.png	4
75	3	1	1	59	./image/0011.png	3
80	3	2	1	84	./image/0012.png	3
74	3	2	1	59	./image/0013.png	3
80	3	2	1	84	./image/0014.png	4
75	3	1	1	59	./image/0015.png	3
74	3	2	1	59	./image/0016.png	3
75	3	1	1	59	./image/0017.png	4
74	3	2	1	59	./image/0018.png	3
78	3	2	1	84	./image/0019.png	4
80	3	2	1	84	./image/0020.png	3
74	3	2	1	59	./image/0021.png	3
80	3	2	1	84	./image/0022.png	4
78	3	2	1	84	./image/0023.png	3
74	2	1	1	59	./image/0024.png	3
79	3	2	1	84	./image/0025.png	2

<그림 12> 최종 csv 파일 데이터

3. 기계학습

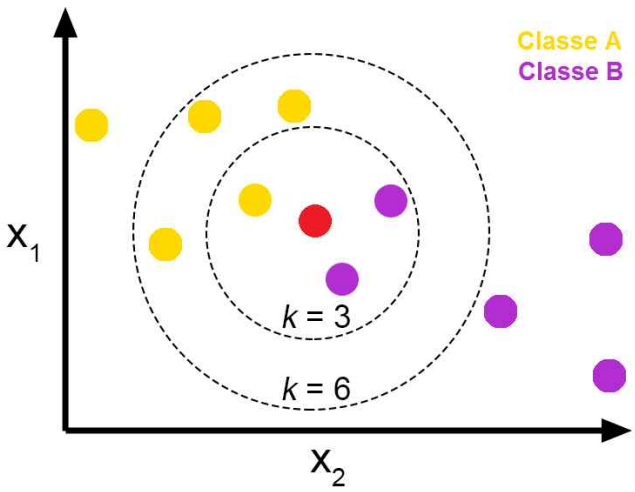
기계학습이란 컴퓨터가 프로그램 없이도 데이터를 기반으로 스스로 학습할 수 있는 능력을 제공하는 학문이다. 컴퓨터 과학자인 Arthur Samuel은 기계학습을 명시적으로 프로그램을 작성하지 않고 컴퓨터에 학습할 수 있는 능력을 부여하기 위한 연구 분야라고 정의하기도 하였다. 기계학습은 학습 방법에 따라 지도학습, 비 지도학습 그리고 강화학습으로 분류된다. 지도학습은 말 그대로 정답이 있는 데이터를 활용하여 학습을 시키며 대표적으로 분류, 회귀 문제가 있다. 비 지도학습은 지도 학습과는 달리 정답이 없는 데이터를 특징으로 분류하여 새로운 데이터에 대한 결과를 예측하는 방법이다. 마지막으로 강화학습은 지도학습, 비 지도 학습이과는 조금 다른 개념으로 시행착오를 통해 학습하는 방법이다. 본 연구에 수집한 데이터는 input data와 label data가 존재하고 데이터를 정해진 라벨에 따라 예측결과를 내야하기 때문에 지도학습을 사용하였다. 전용면적은 연속적인 정수이기에 회귀를 사용하여 예측하는 것이 일반적이지만 건물 시공사가 전용전적에 따라 구비해야하는 설비, 전용 면적 구간에 따른 세금문제 등 다양한 이유들 때문에 일반적인 전용면적이 정해져있는 것을 파악하고 분류 학습을 사용하기로 하였다. 분류학습에는 K-Nearest Neighbors, Naive Bayes, Logistic Regression, Support vector Machine 등 많은 알고리즘이 존재하지만 k개의 가장 가까운 요소를 찾아, 더 많이 일치하는 것으로 분류하는 알고리즘인 K-Nearest Neighbors 을 사용하였다.



<그림 13> 머신러닝 분류

3.1) K-Nearest Neighbors

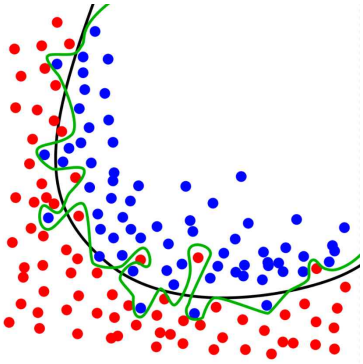
머신러닝의 많은 분류학습 중 K-Nearest Neighbors는 가장 고전적이면서 직관적인 방법이며 단순하다. 또한 기본적인 성능이 좋고, 모델 훈련 시간이 필요 없기 때문에 이 알고리즘을 선택하게 되었다. K-Nearest Neighbors의 기본적인 알고리즘은 새로운 샘플이 들어왔을 때 기존의 데이터 중 가까운 거리에 있는 몇 가지 Label을 확인하고 가장 빈도가 높은 것을 통해 분류한다. 즉 입력 값과 제일 근접한 k개의 요소를 찾아, 더 많이 일치하는 것으로 분류하는 알고리즘인 것이다. 가장 가까운 거리라 함은 유클리드 거리를 사용하여 측정을 하게 되고, 빈도가 높은 것은 인접한 k개의 개수를 참고하게 된다. 여기서 k는 기본적으로 홀수를 쓰게 된다. 이러한 이유는 k가 짝수 개를 참고하게 된다면 2:2 같은 애매한 상황이 발생하기 때문이다. 따라서 K-Nearest Neighbors 알고리즘을 사용할 때는 분류될 그룹의 종류와 특성을 파악하여 적절한 k를 설정해주는 것이 매우 중요하다. 본 연구는 Python을 이용하여 개발하였기 때문에, 기계학습 라이브러리인 Scikit-learn을 이용하여 KNeighborsClassifier를 import하여 사용하였다.



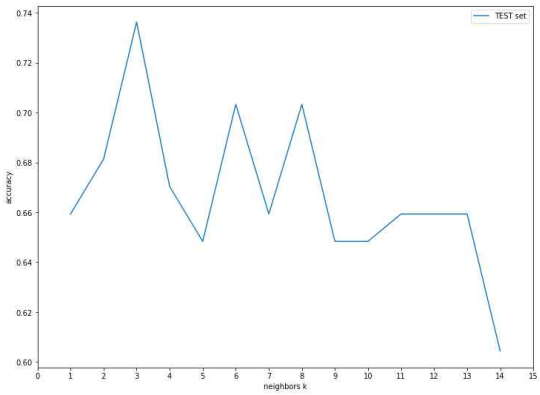
<그림 14> K-Nearest Neighbors

3.2) 최적화된 k 검출

K-Nearest Neighbors 알고리즘에서 참고하는 k의 개수는 정확도에 많은 영향을 미치게 된다. k가 1일 때 거리가 가까운 요소만 보고 신규 데이터를 분류하게 되는데, 이는 분류 정확도가 상당히 낮다고 볼 수 있다. 참고 할 수 있는 주변 요소를 볼 수 있는 범위가 매우 좁아지는 것이고, 참고하는 하나의 요소에 따라 결과가 극단적으로 바뀐다는 것이다. 이것은 과적합(OverFitting)이랑 연결된다. 과적합이란 기계학습에서 학습 데이터를 과하게 학습하는 것을 의미한다. 즉 학습데이터에 대해서는 오차가 감소하지만 실제 데이터에 대해서는 오차가 증가한다는 것이다. k를 몇으로 정할지는 실제 학습시킬 때 k의 범위를 두고 테스트 해보면서 정확도를 계산하면 쉽게 알 수 있다. 본 연구에서 학습을 시켜본 결과, 유성구의 최적화된 k가 1이라는 결론이 나왔는데 이는 학습 데이터 셋의 양과 질에 영향을 받았다고 예상한다. 유성구가 아닌 대덕구, 서구는 각각 k가 5, 3 일 때 정확도가 제일 높게 나왔다.



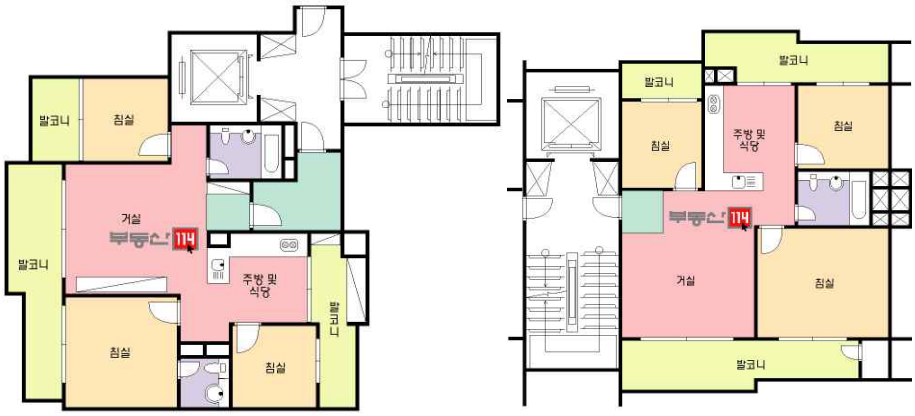
〈그림 15〉 초록색 선은 과적합



〈그림 16〉 서구 k의 범위에 따른 정확도

3.3) 지역별, 평수별 특징

기계학습을 시킬 때 정확도는 데이터의 질과 양에 영향을 받는다. 데이터의 질이라 함은 다양한 종류의 학습데이터, 확보된 데이터의 학습이 가능한 여부 등이 있을 수 있다. 본 연구에서 웹 크롤링을 통해 학습 데이터를 확보하는 과정에서, 공개되어 있는 데이터가 거래 가능한 매물들이였기 때문에 데이터의 다양성이 고르지 않았다. 크롤링한 1000개의 데이터 중에서도 한 아파트 단지의 같은 평면도, 매물 정보가 다수 존재하였다. 또한 평도면과, 매물 정보는 같으나 전용면적만 다른 경우도 존재하였으며 아파트의 연식에 따라서도 전용면적은 같으나 평면도에서 이미지 처리를 통해 추출한 데이터가 차이나기도 하였다. 이러한 이유들 이외에도 건설 회사와 건축가에 따라서 발코니면적을 줄이고 화장실 개수를 늘리는 등 많은 변수가 존재 하였다. 이러한 변수들을 전부 고려하며 지도학습을 할 수 없었기에 1차적으로는 지역구별로 나눠서 학습을 진행하였다.



〈그림 17〉 두 평면도 모두 전용면적이 59㎡지만 아파트 연식에 따라 화장실 개수가 다르다

3.4) 정확도

분류알고리즘을 사용하여 예측을 하였으면 그 분류기의 성능을 평가해봐야 한다. 분류알고리즘은 회귀분석과 다르게 다양한 성능 평가 기준이 필요하다. 성능 평가 지표는 많은 사람들이 알고 있는 정확도(Accuracy)외에도 재현율(Recall), 정밀도(Precision), F1 Score등 많은 평가지표가 존재하지만 본 연구에서는 일반적으로는 학습에서 최적화 목적함수로 사용되고 있는 정확도를 선택하였다. 정확도를 높일 수 있는 방법도 여러 가지가 존재한다. 데이터의 양, 적정한 k의 값, 표준화, 정규화 등 많은 방법이 있다. 지역별, 평수별 특징에서 언급했듯이 대전광역시의 서구, 유성구, 대덕구로 따로 분류하고 적정한 k의 값만 적용한 체, 데이터를 정규화하지 않고 학습을 시킨 결과 각각 73.33%, 93.10%, 90.17%가 나왔다. 이후 정확도를 높이기 위하여 데이터들의 평균과 표준편차를 구하고, 평균대비 몇 표준편차만큼 데이터가 떨어져 있는지를 점수화하는 방식인 z-score로 normalization하여 적용하였더니 적장한 k의 값이 변동하였고 서구, 대덕구의 정확도는 각각 76.92%, 95.53%로 증가하였지만 유성구는 93.10%로 동일하였다. 향후 이미지처리를 통해 전용면적을 예측할 수 있는 특징을 잡아낼 수 있다면 더욱 높은 정확도를 기대해 볼 수 있을 것이다.

```
def min_max_normalize(lst):
    normalized = []

    for value in lst:
        normalized_num = (value - min(lst)) / (max(lst) - min(lst))
        normalized.append(normalized_num)

    return normalized
```

```
def z_score_normalize(lst):
    normalized = []
    for value in lst:
        normalized_num = (value - np.mean(lst)) / np.std(lst)
        normalized.append(normalized_num)
    return normalized
```

<그림 18> 최소-최대 정규화와 Z-점수 정규화

```
clf = KNeighborsClassifier(n_neighbors = 2)
clf.fit(x_train, y_train)
prediction = clf.predict(x_test)
cnt = 0
for i in range(len(y_test)):
    if int(y_test.ravel()[i]) == int(prediction[i]):
        cnt+=1

print(len(y_test),'개 예측 결과 = ', clf.score(x_test, y_test))
print('예측성공 개수 : ',cnt , '예측실패 개수 :', len(y_test)-cnt )
```

```
112 개 예측 결과 =  0.9553571428571429
예측성공 개수 :  107 예측실패 개수 : 5
```

<그림 19> 정규화후의 변경된 k의 값으로 확인해본 대덕구의 모델정확도

Ⅲ. 결 론

본 연구에서는 부동산 매물 중 전용면적 누락, 허위기재에 대한 허위매물의 도면을 통한 전용 면적 예측 방법에 대하여 연구하였다. 이미지와 데이터를 수집하고, 수집한 이미지를 통하여 기계학습에 필요한 정보들을 추출하고 정보를 토대로 분류모델인 K-Nearest Neighbors을 사용하여 예측모델을 만들었다. 지역별 매물의 특성을 인지하고 지역별로 나누어 전용 면적을 예측한 결과 서구, 대덕구, 유성구 순으로 76.92%, 95.53%, 93.10%의 정확도로 평균 88.52% 정확도의 결과를 산출할 수 있었다. 과적합이 나온 모델도 있었지만 이러한 문제는 앞으로 본 연구의 활용 및 개선 방향에서도 같은 문제를 방지하기 위해 충분한 데이터의 확보와 모델의 세부 수치 조절이 필요할 것으로 보인다. 본 연구에서는 미처 다루지 못했지만 충분한 학습 데이터가 확보되어 전용면적의 다양성만 확보 할 수 있다면, 회귀모델을 사용하여 학습시켜 더욱 좋은 정확도를 가진 모델을 개발할 수 있을 것이다. 또한 지역별 특성을 고려하여 타 지역 간 공통적인 매물의 정보를 찾아 데이터 학습 모델에 적용 시킨다면 지역을 나누지 않고 통합적인 전용면적 예측 모델을 만들 수 있을 것으로 보인다.

저자가 생각하는 본 연구의 발전 방향은 다음과 같다. 이미지 처리 방식을 개선하여 동일한 형식의 도면 뿐 아니라 다른 스타일의 도면 또한 분석하여 동일한 정보를 도출한다. 현재의 방식인 도면과 매물의 세부 정보도 사용하여 면적을 예측하는 것이 아닌 오로지 도면만을 사용하여 그 안에서 세부 정보를 찾아내고 면적을 예측하는 모델을 만든다.

정확도를 더욱 높이고 앞서 언급한 개선 방향을 적용한다면 실제 부동산 시장에서 허위매물을 색출함에 편리함을 더할 수 있다는 것을 본 연구에서 보였다.

참고문헌

[1] 매일경제, “부동산 허위매물 신고 10건 중 7건은 ‘네이버’ 매물” https://www.mk.co.kr/news/realestate/view/2020/09/1007932/

[2] 최신 웹 크롤링 알고리즘 분석 및 선제적인 크롤링 기법 제안 (2019, 나철원)

[3] 기계학습 및 기본 알고리즘 연구 (2018, 김동현)