천재교육 빅데이터 7기 파이썬 프로그래밍 실습과제 1

박현석

January 31, 2024

Abstract

어떻게 하면 서울시의 자치구 중 일자리가 많은 구와 주거지가 많은 구를 구분하고 피크시간대를 파악할 수 있을까? 서울시 출퇴근 시간의 지하철 혼잡도는 날이 갈수록 높아지고 있다. 또한 특정 자치구로 일자리 몰림 현상 또한 증가하고 있다. 최근에는 이런 문제들로 불편함을 겪는 사람들이 많아지고 있고, 자율출퇴근제의 도입으로 인해서 어느정도 피해갈 수 있다. 따라서 출퇴근 시간의 지하철 혼잡도를 분석해서 어느 자치구가 일자리가 많고 어느 자치구가 주거지가 많은지 파악하고 해당 구의 피크시간대를 파악하여 불편함을 줄일 수 있는 인사이트를 얻고자 한다.

Keywords— 지하철 혼잡도, 피크 시간대, 데이터 분석

1 Introduction

최근들어 서울시의 지하철 혼잡도는 매우 빠르게 증가하고 있다. 또한 특정 자치구로의 일자리 몰림, 주거지 몰림 현상으로 인해 출퇴근 하는 사람들의 목적지는 비슷하다. 더해서 와중에 몰리는 많은 사람들로 인해 최근 발생하고 있는 안전 문제(호흡곤란, 공황, 부상)에 대한 경각심도 높아지고 있다. 그러는 와중에 많은 기업들의 자율출퇴근제 도입으로 인한 사람들이 출퇴근 시간대를 정할 수 있어 사람이 몰리는 특정 시간대를 피해서 출퇴근 하는 사람들도 늘어나고 있다. 하지만 출/퇴근 하는 장소가거주지가 모여있는 자치구인지, 일자리가 모여있는 자치구인지에 따라서도 몰리는 시간대가 변한다. 또한 아무 데이터 없이 특정 시간대에 사람이 없고, 몰리는지 판단하려면 직접 경험해보면서 시간과 돈을 사용해야 알 수 있다. 따라서 이런 사회적 배경을 반영하여 서울교통공사에서 제공하는 지하철 혼잡도 정보 데이터와 자치구별 지하철 역 정보 데이터를 사용해서 일자리가 몰려있는 자치구와 거주지가 몰려있는 자치구를 분석하고, 특정 자치구나 지하철역에 사람이 몰리는 시간대를 분석하여 사람들의 불편함을 줄일 수 있게 하는 인사이트를 제공한다. 이를 통해서 이 데이터를 통해서 사람들의 시간과 돈을 절약할 수 있게 하고자 한다.

2 Method

여기에서는 이 프로젝트를 수행하기 위해서 사용되는 가설, 절차, 데이터, 개발 환경 등을 소개한다.

2.1 가설

여기에서는 데이터를 분석하기 전에 작성한 가설을 소개한다. 가설은 다음과 같다.

- 서울시의 지하철 혼잡도가 높은 시간대는 비슷할 것이다.
- 서울시의 지하철 시간대별 혼잡도는 일자리가 많은 역과 거주지가 많은 역이 다를 것이다;
- 서울시의 지하철 시간대별 혼잡도는 일자리가 많은 구와 거주지가 많은 구가 다를 것이다;

이 가설들이 데이터에서 실제로 나타나는지 확인하기 위해서 분석을 진행한다.

2.2 데이터

여기에서는 데이터 분석을 위해서 사용한 공공데이터를 소개한다. 사용한 공공데이터는 서울교통공사에서 제공하는 지하철 혼잡도 정보 데이터와 자치구별 지하철 역 정보 데이터를 사용한다. 전반적인 흐름은 Figure 1에서 보여준다.

2.2.1 지하철 혼잡도 정보 데이터

지하철 혼잡도 정보 데이터는 1659개의 행 데이터와 요일구분, 출발역, 호선, 30분 간격의 시간의 혼잡도 등의 42개의 컬럼으로 이루어져 있다. 출/퇴근을 분석하기 위해서 요일구분에서 평일을 제외한 나머지는 제거하고, 호선 또한 상관없는 데이터라서 제거한다. 각 지하철역 별로 상행선과 하행선으로 나눠져있지만 통합해서 확인하기 위해 상행선 데이터와 하행선 데이터를 합치며 이 과정에서 혼잡도는 평균으로 계산한다. 이 과정을 통해서 평일에 해당되는 지하철역의 시간대별 혼잡도 데이터를 얻는다. 또한 출/퇴근 시간대만 확인하기 위해서 07시00분부터 10시00분, 17시00분부터 20시00분 까지의 데이터만 사용하는 추가 작업을 거친다. 이를 통해 239행 39컬럼의 전처리 된 데이터를 분석에 사용한다.

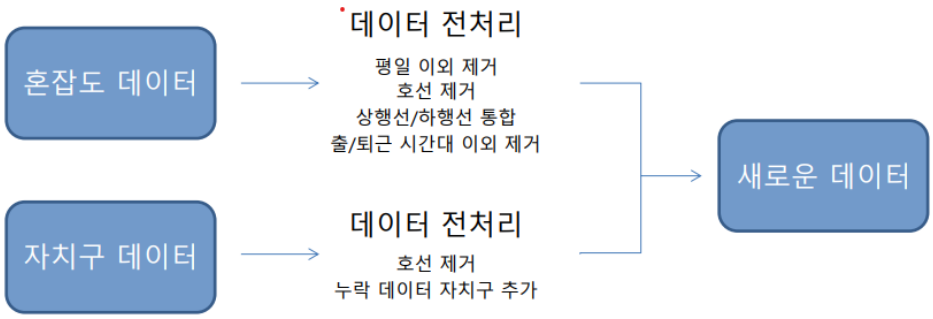

Figure 1: 데이터 전처리 과정을 보여준다.

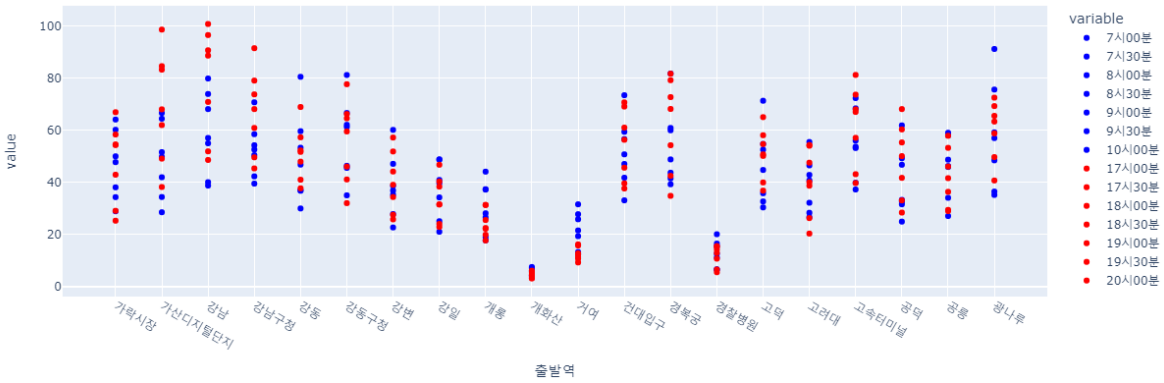

Figure 2: 시간대별 지하철역의 혼잡도를 보여준다.

2.2.2 자치구별 지하철 역 정보 데이터

지하철역의 자치구를 통해 분석하기 위해서 자치구별 지하철 역 정보 데이터를 추가적으로 사용한다. 이 데이터는 각 지하철 역 마다의 자치구가 있는 데이터이다. 하지만 모든 지하철역이 존재하지 않고 "역이름(호선)"과 같이 표현되어 있어서 역이름만 남기는 전처리 작업을 진행했고, 존재하지 않는 지하철역에 대한 자치구는 직접 검색을 통해서 추가해줬다. 또한 자치구 혼잡도를 분석할 때 같은 자치구에 있는 지하철역들의 혼잡도를 평균해서 사용한다. 전처리가 완료된 데이터는 지하철 혼잡도 정보 데이터와 추후에 결합하여 분석한다.

2.3 절차

전처리된 데이터를 사용해서 EDA(탐색적 데이터 분석)을 진행한다. 이 과정을 통해서 출/퇴근시간 지하철역 혼잡도 데이터가 어떤 분포를 보이고 있는지, 어떤 특성을 가지고 있는지 확인한다. 시간대 별 지하철역 혼잡도를 분석하기 위해서 지하철역과 혼잡도를 사용하고, 시간대 별 자치구 혼잡도를 분석하기 위해서 자치구와 혼잡도를 사용한다.

2.4 개발환경

이 모든 과정을 진행하기 위해서 Python 3.11버전으로 분석한다. 분석과 전처리를 위해서 pandas, re, plotly 패키지를 사용한다.

2.5 결과

가설의 입증을 위해서 여러가지 시각화 결과를 보여준다.

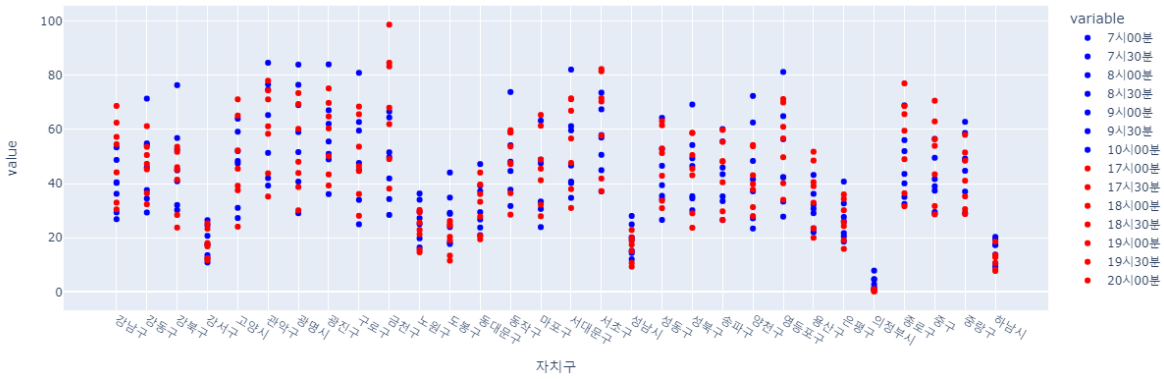

Figure 3: 시간대별 자치구의 혼잡도를 보여준다.

2.5.1 시간대 별 지하철역 혼잡도

Fig 2는 첫번째 역 부터 20번째 역까지를 보여주고 있다. 파란색 점은 출근시간, 빨간색 점은 퇴근시간대를 나타낸다. 또한 x축은 지하철역, y축은 혼잡도를 나타낸다. 강남역, 강남구청역, 가산디지털단지, 경복궁역이 대표적으로 퇴근시간대 사람이 몰리는 것을 볼 수 있고, 강동역, 개롱역, 광나루역이 출근시간대 사람이 몰리는 것을 볼 수 있다. 강남역, 강남구청역, 가산디지털단지역은 대표적인 일자리가 많은 곳이며 퇴근시간에 많은 사람들이 몰려서 혼잡도가 높다는 것을 한눈에 확인된다. 추가적으로 혼잡도가 가장 높을 때 시간대는 출근시간이면 8시부터 8시30분, 퇴근시간이면 18시부터 18시30분이 가장 높다. 이 결과로 대부분의 사람들이 8시에 출근하고 18시에 퇴근한다는 것을 확인한다.

2.5.2 시간대 별 자치구 혼잡도

Fig 3는 모든 자치구의 시간대 별 혼잡도를 보여준다. 금천구, 강남구, 서초구, 종로구, 중구와 같이 일자리가 많은 자치구는 퇴근시간에 사람이 몰려서 지하철 혼잡도가 높다. 반대로 노원구, 성북구, 도봉구, 양천구와 같이 주거지가 더 많은 구의 경우엔 출근시간에 지하철 혼잡도가 높다. 추가적으로 거리가 존재하는 하남시, 의정부시를 제외한 모든 구/시에서 출근시간은 8시에 가장 혼잡하고 퇴근시간은 18시에 가장 혼잡하다. 대부분의 구/시에서 출근시간은 9시로 맞추면 혼잡도가 약 절반으로 감소하고 퇴근시간은 19시에 맞추면 약 30%가 감소하는 경향을 보인다.

위와 같은 분석을 통해서 우리는 주거지가 많은 자치구와 일자리가 많은 자치구의 혼잡도가 출/퇴근 시간대에 따라서 다른 것을 확인했다. 자세하게 말하면, 주거지가 많다면 출근시간에 혼잡하고, 일자리가 많다면 퇴근시간에 혼잡하다는 특성을 가지는 것을 직접 확인했다. 또한 지하철역 뿐만 아니라 자치구별로도 비슷한 양상을 띄는 것을 확인했다.

2.6 토론

이 프로젝트의 결과를 기반으로 우리는 어느 구가 일자리가 많은지, 주거지가 많은지 한눈에 확인할 수 있었고, 어느 시간대에 가장 많이 몰리는지 알 수 있었다. 자율출퇴근제의 도입이 된다면 이 데이터를 통해서 자신이 가고자 하는 장소에 사람이 가장 많이 안몰리는 시간대를 파악할 수 있을 것이고 직접 경험하지 않고 파악할 수 있어서 시간과 교통비를 절약할 수 있다. 추후에 지하철 노선도 모양으로 시간대별 사람들의 이동을 나타내는 분석을 통해서 좀 더 직관적이고 정보를 알기 편하게 만들 수 있을 것이라고 생각한다.

References

서울교통공사지하철혼잡도정보데이터 : https://www.data.go.kr/data/15071311/fileData.do
서울교통공사지하철역자치구정보데이터 : https://www.data.go.kr/data/15081868/fileData.do
Pandas : https://pandas.pydata.org/
plotly : https://plotly.com/python/