딥러닝을 이용한

역별 지하철 승하차인원

예측 모델 구축

박순혁(Soon-Hyeok Park)

1 . 서 론

본 논문에서는 다양한 시계열 분석 모델과 딥러닝 알고리즘을 이용하여 역별 지하철 승하차인원을 예측하는 모델을 구축하고자 한다. 여기서 지하철의 승하차인원 외 다양한 변수들을 활용하여 좀 더 설명력있는 예측 모델을 구축할 것이다. 그 다양한 변수 중 하나가 바로 날씨이다. 사람들은 날씨가 맑으면 밖에 나가서 외부활동을 하고 싶어한다. 반대로 날씨가 흐리거나 비가 오면 나가기가 꺼려진다. 실제로 사람의 보행에 가장 많은 영향을 미치는 날씨는 바로 비가 내릴 때이다. 비가 오는 날씨는 맑은 날씨에 비하여 통계적으로 유의미하게 보행량을 감소시키는 효과가 나타났다. 이 점을 활용하여 지하철 승하차인원에 어떠한 날씨 변수들이 얼마만큼의 영향을 주는 지 파악하고 예측 모델을 구축할 때 날씨에 대한 변수들을 이용할 것이다. 또 다른 변수는 주말과 공휴일에 대한 변수이다. 일반국도에서 평일보다 주말과 공휴일의 교통량이 평균적으로 더 많은 것으로 드러났다. 이 실험결과를 바탕으로 주말과 공휴일의 변수가 지하철 승하차인원에 영향을 줄 수 있을 것이라는 추론을 할 수 있다. 마지막으로 코로나19 바이러스(COVID-19) 확진자 변수를 활용할 것이다. 최근 코로나19 바이러스 감영병으로 인해 전세계적으로 많은 확진자와 사망자가 나오면서 여객 부분 수송실적이 큰 폭으로 감소하고 있다. 한국의 경우 육상교통 부문에서는 2020년 2월 1주차부터 3주차까지 감소 폭이 비슷하게 유지되다가 신천지발 집단감염 사태로 대구지역의 확진자가 크게 증가함에 따라 2월 4주차부터 고속버스, 시외버스, 전철, 버스, 택시 등 모든 부문에서 감소 폭이 급격하게 증가하고 있다. 이것은 매우 유의미한 결과이며 지하철 승하차인원에 큰 영향을 끼칠 것으로 보인다. 한국의 코로나19 확진자 변수가 지하철 승하차인원 예측 모델에 더 많은 설명력을 부여해줄 것으로 기대하고 있다. 그러나 최근의 데이터라는 점에서 train과 test 데이터로 나뉘어 모델 검증을 할 때 한계가 있을 것으로 예측된다.

1. . 이론적 배경

2. 1 날씨와 지하철 이용 간의 관계

‘도시공간에서 날씨와 계절이 보행량에 미치는 영향’ 이라는 논문에서 실험결과 지하철역의 보행량은 하절기에는 증가에 강한 정(+)의 영향을 미치지만 동절기에는 그 효과가 작거나 부(-)의 영향을 미쳤다. 이것을 통해 온도가 지하철역의 승하차인원에 영향을 줄 수 있다는 추론을 할 수 있다. 또한, 보행에 가장 많은 영향을 미치는 날씨는 “비” 인 것으로 나타났다. 비가 오는 날씨는 대부분의 맑은 날씨에 비하여 통계적으로 유의미하게 보행량을 감소시키는 효과가 있었다. 비 오는 날씨의 보행량 감소효과는 주거지역보다 상업지역에서 크게 나타났다. 이것은 비가 오면 상업지역에 위치한 지하철역의 이동도 줄어든다는 것을 나타낸다. 따라서 비가 오는 날에는 지하철 승하차인원도 감소할 것이라는 추론을 할 수 있다.

2. 2 휴일이 지하철 승하차인원에 미치는 영향

요일별 일반국도 평균 교통량을 분석한 결과 평일보다 공휴일과 주말에 높은 교통량이 나타났다. 또한 요일별 지난 6년동안의 평균 교통량(AADT)과 공휴일의 교통량의 비에 대한 분석결과 주말과 공휴일의 교통량이 지난 6년동안의 요일별 교통량 평균보다 높은 것으로 나타났고 평일은 모두 낮은 것으로 나타났다. 이 비에 대한 유의성을 검증하기 위해 쌍체비교검정을 실시하였고 t-검정 결과 평일과 공휴일의 AADT에 대한 비의 평균에 차이가 있다는 것이 밝혀졌다. 이 점을 통해 평일과 휴일의 여부에 따라 지하철의 승하차인원이 영향을 받을 수 있다는 것을 추측할 수 있다.

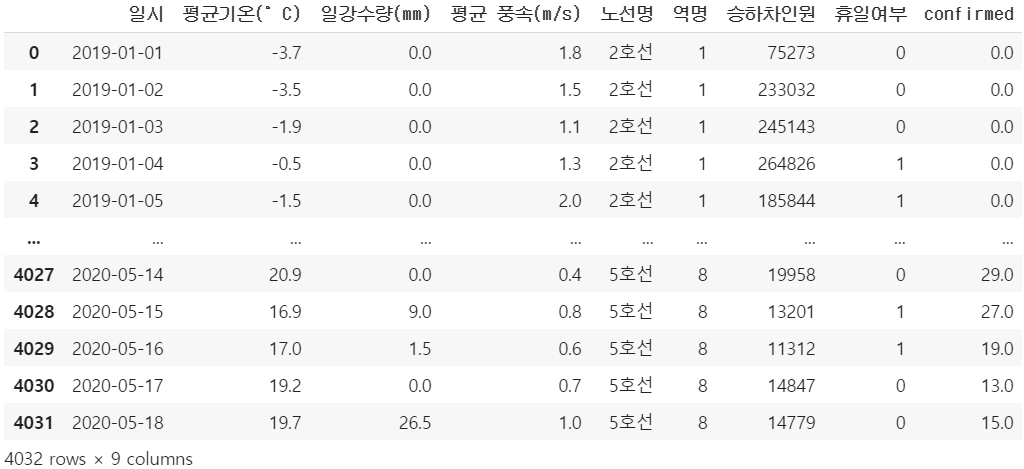
2. 3 코로나19가 지하철 이용에 미치는 영향

‘코로나19의 육상교통부문 영향분석 및 대응방향’ 이라는 논문에서 코로나 바이러스가 지하철 이용에 미치는 영향을 분석하였다. 코로나 19가 미치는 영향을 진단하기 위하여 국내에서 첫 확진자가 발생하기 전인 1월 3주차부터 3월 첫주까지 주별로 측정 비교하였다. 측정결과 2월 1주차부터 3주차까지 감소 폭이 비슷하게 유지되다가 신천지발 집단감염 사태로 대구지역의 확진자가 크게 증가하여 3월 1주차에는 1월 3주차에 비하여 38%나 감소하였다. 그러나 주로 지역 내 교통수단에 비하여 전국 지역 간 교통이 큰 영향을 받고 있는 것을 보여 코로나 19 변수가 지하철 이용에 큰 영향을 미치는지는 아직 미지수이다. 이번 논문을 통해 코로나19가 지하철 승하차인원과 어떠한 상관관계가 있는지 파악하고 예측 모델에 활용할 것이다.

3. 자료의 구축

3. 1 데이터 전처리

여러 다중변수들로 지하철 승하차인원을 예측하기 위해서는 변수들에 대한 자료가 필요하다. 종속변수인 일별, 역별 지하철 승하차인원은 서울 열린데이터광장([data.seoul.go.kr](http://data.seoul.go.kr/))이라는 웹사이트에서 CSV데이터를 얻을 수 있었다. 원래는 수도권 지하철의 역별 승하차인원 예측모델을 구축하고자 하였으나 raw데이터에 수도권의 모든 역이 포함되어있지 않아 총 8개의 역을 선정하여 예측모델을 구축하고자 한다. 역 선정기준은 임의의 날짜를 정해 승하차인원이 제일 많은 순으로 4개의 역(강남,잠실(송파구청),고속터미,홍대입구)을 선정하였고 2개의 역(건대입구, 여의나루)은 날씨, 휴일변수에 영향을 많이 받을 것으로 추정되어 선정하였고 나머지 2개의 역(이태원, 명동)은 코로나19 확진자 변수에 많은 영향을 받을 것이라 생각되어 선정하였다. raw데이터에는 승차인원과 하차인원이 나뉘어져 있어서 파이썬(python)이라는 오픈소스 프로그래밍 언어를 활용하여 두 변수를 더하여 승하차인원이라는 변수를 생성하였다. 독립변수에는 일별, 역별 평균온도, 강수량, 평균풍속, 휴일여부, 코로나19 확진자가 있다. 평균온도와 강수량, 평균풍속은 기상자료개방포털(<https://data.kma.go.kr/>)이라는 웹사이트에서 데이터를 얻었다. 지하철역별 주소 데이터를 얻어 파이썬 코딩작업을 통해 역 주소에 해당하는 지역구 별 기상데이터를 병합하였다. 휴일여부는 월~목요일을 0으로 금~일요일과 공휴일을 1로 나타내어 요일의 범주형 변수를 이진 변수(binary variable)로 변환하였다. 금요일 같은 경우는 지하철 승하차인원이 평균적으로 높게 나타나서 휴일여부에 포함시킬 수 있게 1로 변환하였다. 코로나19 확진자는 2020년 5월 18일까지 일별 국내 코로나19 확진자 수(누적 확진자 수 아님)를 나타낸다. 아래 그림1은 데이터 전처리를 한 후의 결과데이터를 파이썬을 통해 보여주고 있다.

그림 1. Data preprocessing

3. 2 EDA(탐색적 데이터 분석)

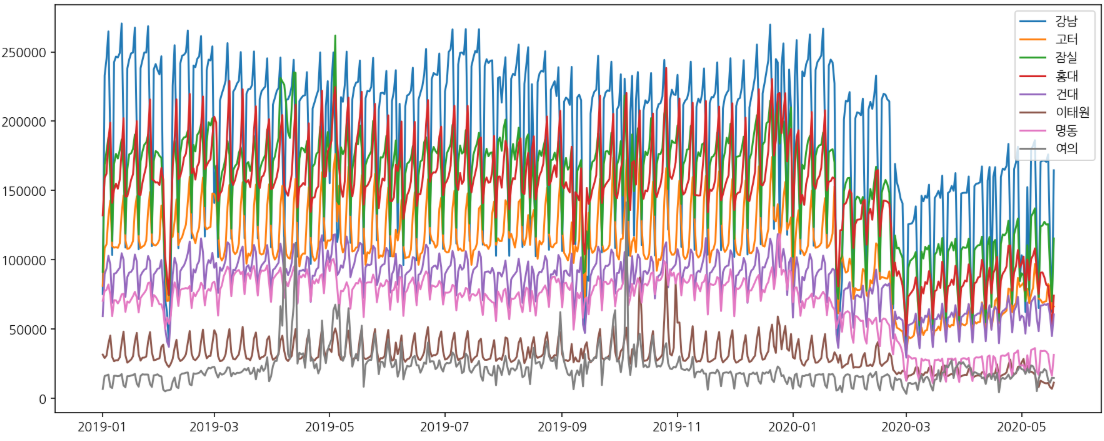


그림 2. 역 별 승하차인원 시계열 그래프

그림 2 를 통해 2019-01-01 ~ 2020-05-18 까지의 지하철 승하차인원을 역 별로 추이 및 스케일을 확인할 수 있다. 승하차인원은 평균적으로 강남역(2호선)이 제일 많고 여의나루역이 제일 적다. 그래프를 통해 알 수 있듯이 코로나19가 국내로 확산되기 시작한 1월 중순 시점에서부터 승하차인원은 줄어드는 추세를 띄고 있다. 특히 주로 중국인들이 많이 찾아오는 명동 같은 경우는 그 감소폭이 더 크다. 이것을 통해 코로나 19 확진자 변수가 지하철 승하차인원에 많은 영향을 끼칠 것이라는 추론을 할 수 있을 것이다.

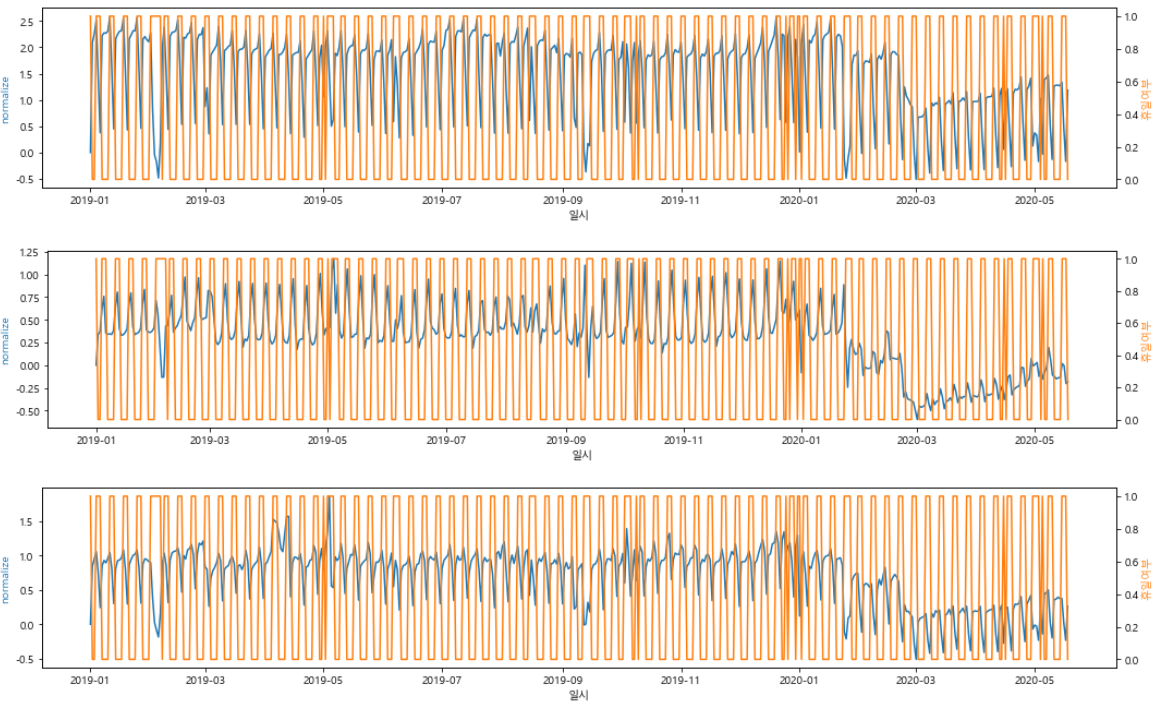


그림 3. 승하차인원과 휴일여부 시계열 그래프

그림 3 은 대표적으로 강남, 고속터미널, 잠실역의 승하차인원과 휴일여부의 시계열 그래프를 동시에 보여주고 있다. 휴일여부가 0과 1의 이진변수((binary variable)이기 때문에 그래프에서 승하차인원과 비교가 가능하게 나타내기 위해 정규화 과정을 거쳐 보기 쉽게 시계열 그래프로 나타냈다. 파란색 선이 승하차인원이고 오렌지색 선이 휴일여부를 나타낸다. 그림 3의 맨 상단에 있는 첫번째 그래프는 강남역 그래프로 휴일여부가 0일 때는 승하차인원이 높고 1일 때 낮아지는 추세를 보이고 있다. 이것은 강남역에 회사가 많아 주로 평일에 출퇴근하는 회사원들이 많아 휴일보다 평일에 더 많은 승하차인원을 보여주고 있는 것으로 추측된다. 나머지 그 밑에 있는 두번째, 세번째 그래프는 고속터미널역과 잠실역으로서 강남역과는 다른 추세를 보인다. 휴일여부가 0일 때 승하차인원이 낮고 1일 때 높아지는 추세를 보인다. 이것은 평일보다 휴일에 승하차인원이 더 많다고 해석할 수 있다.



그림 4. 승하차인원과 코로나 확진자 시계열 그래프

그림 4는 대표적으로 강남, 고속터미널, 잠실역의 승하차인원과 국내 코로나19 확진자의 시계열 그래프를 동시에 보여주고 있다. 파란색 선이 승하차인원이고 오렌지색 선이 코로나19 확진자 추이이다. 그래프를 통해 알 수 있듯이 세개의 역 모두 승하차인원이 국내에 코로나19 확진자가 발생하기 시작한 1월 중순 시점부터 감소하기 시작하였다. 그러다 신천지발 집단감염사태가 일어난 2월말부터 엄청난 감소폭을 보이며 최저점을 찍었다가 3월 중순부터 점차 증가하고 있는 추세를 보이고 있다. 시계열 그래프가 거의 끝나가는 시점에서는 이태원 클럽발 집단감염사태로 인해 확진자가 다시 늘어나기 시작했고 그에 따라 승하차인원 또한 살짝 감소하는 추세 보이고 있다. 이것을 통해 코로나19 확진자 변수가 지하철 승하차인원에 막대한 영향을 끼치고 있다는 것을 확인 할 수 있다.



그림 5. MinMaxScaler correlation 그림 6. 강남역 제외한 상관관계

그림 1 을 통해 알 수 있듯이 데이터셋의 변수별로 값의 범위나 척도가 차이가 많이 나기 때문에 파이썬의 머신러닝 라이브러리 중 하나인 Scikit-learn의 MinMaxScaler 메소드를 사용하여 0과 1사이의 데이터로 정규화시켜주었다. 그림 5 는 정규화시킨 후 변수들 간의 상관관계를 보여주고 있다. 승하차인원과 나머지 변수들 간의 상관관계를 분석해 보았을 때 눈에 띄는 상관관계를 갖는 변수는 없어보이지만 그나마 코로나19 확진자 변수가 -0.17의 상관계수를 보이고 있다. 예측모델 구축에 있어 주목할 만한 변수이다. 한편 휴일여부는 0.07이라는 생각보다 낮은 상관계수를 보이고 있다. 이것은 그림 3 에서 강남역의 승하차인원에 휴일여부 변수가 영향을 끼치지 못하는 결과를 보이고 있고 예측모델로 선정한 역들 중에서 강남역이 승하차인원이 제일 많기 때문인 것으로 해석된다. 그래서 그림 6 에서 처럼 강남역을 제외한 변수들 간의 상관관계를 진행해 보았다. 휴일여부 변수의 상관계수는 기존 0.07에서 0.12로 증가하였다. 또한 코로나19 확진자 변수도 기존 0.17에서 0.18로 증가한 모습을 보였다. 여기서 주목할 만한 점은 평균풍속 변수의 상관계수가 기존 0.08에서 0.19로 급증하였다는 것이다. 강남역의 승하차인원이 나머지 역들과 달리 변수들의 영향이 적을 것이라고 추측된다. 하지만 본 논문에서 구축하고자 하는 예측모델은 역 별로 구축이 되기 때문에 분석을 계속 진행하고자 한다.

4. 분석방법 및 결과

4. 1 코로나19 시점 이후 독립표본 t검정

독립표본 t-검정이란 독립된 두 집단의 평균 차이를 검정하는 기법이다. 쌍체검정과는 달리 서로 무관한 독립된 두 집단을 사용해야한다. 이 검정은 등분산 여부에 따라 결과값이 달라지기 때문에 독립표본 t-검정을 시행하기 앞서 등분산 검정을 시행한 후 그 결과에 따라서 독립표본 t-검정을 시행해야 한다. 여기서 등분산 검정이란 두 집단들의 분산이 같은 지 다른 지를 검증하는 것이다. 등분산검증에서 사용되는 귀무가설은 두 집단의 분산이 서로 같다는 것이며, 대립가설은 적어도 어느 하나는 다른 것과는 같지 않다는 것이다. 각 집단의 분산이 동질적인지를 따지는 방법으로는 Hartley 검증, Bartlett 검증, Cochran 검증, Levene 검증, Scheffe 검증 등이 있다.

본 논문에서는 코로나19 확진자 변수와 역 별 승하차인원의 영향을 확인하기 위해 독립표본 t검정을 하고자 한다. 코로나19 확진자가 폭발적으로 증가하기 시작한 신천지발 집단감염사태(2020-02-20)를 기준으로 시점을 나눠 평균의 차이를 분석할 것이다. 우선 귀무가설()은 ‘신천지발 집단감염사태 이전 지하철 승하차인원과 이후 지하철 승하차인원에 평균 차이는 없다.’ 로 설정한다. 대립가설()은 ‘두 집단 평균 간의 차이가 있다.’ 로 설정하였다.

신천지발 집단감염사태 이전과 이후의 평균 승하차인원에 차이가 없다.

신천지발 집단감염사태 이전과 이후의 평균 승하차인원에 차이가 있다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 전 체 |
| Levene-F | 0.638 | 8.270 | 0.576 | 14.831 | 4.995 | 13.161 | 8.940 | 9.276 | 2.925 |
| p-value | 0.425 | 0.004 | 0.448 | 0.000 | 0.026 | 0.000 | 0.003 | 0.002 | 0.088 |

표 1. 신천지 등분산 검정

집단의 등분산 검정을 하기 위해 여러 검증 방법 중 Levene 검증을 이용했다. 표 1은 통계에 적합한 파이썬의 SciPy 라이브러리를 활용하여 Levene 검증을 실행한 결과표이다. Levene-F값은 등분산 검정 통계량으로서 집단 간 분산과 집단 내 분산의 비율이다. F값이 크면 클수록 두 표본의 분산은 동일하지 않다. 표 1에서 ‘전체’ 항목은 모든 역의 승하차인원 평균 값을 신천지 이전과 이후로 나눠 등분산 검정을 진행한 결과이다. 강남역과 잠실역, 전체의 경우 p-value 값이 0.05보다 크기 때문에 등분산을 가정하여 t-검정을 해야한다. 나머지 역들은 p-value 값이 0.05보다 작기에 등분산을 가정하지 않고 t-검정을 진행한다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 전 체 |
| t-statistic | -11.497 | -33.028 | -20.719 | -37.611 | -26.433 | -23.795 | -47.113 | -7.913 | -24.753 |
| p-value | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

표 2. 신천지 t-검정

표 2 는 등분산 검정 후 그에 맞게 t-검정을 진행한 결과이다. p-value 값이 모두 유의수준 0.05보다 작게 나타났으므로 귀무가설은 기각하고 대립가설이 통계적으로 유의하다고 판단한다. 즉, 모든 역의 신천지발 집단감염사태 이전과 이후의 승하차인원에 차이가 있다고 할 수 있다.

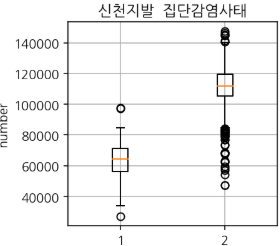


그림 7. 전체 역 평균 승하차인원(신천지)

그림 7 은 전체 8개 역의 신천지발 집단감염사태 이전 이후의 승하차인원 boxplot이다. 1이 신천지 사태 이후의 승하차인원이고 2가 신천지 사태 이전의 승하차인원이다. 그래프를 통해서도 값의 분포의 차이가 많이 난다는 것을 확인할 수 있다. 1의 중앙값과 2의 중앙값의 차이 또한 매우 크다는 것을 알 수 있다.

독립표본 t검정을 통해 신천지발 집단감염사태가 승하차인원에 영향을 끼친 것을 확인하였다. 이번에는 이태원 클럽발 집단감염사태(2020-05-07) 이전과 이후의 승하차인원 평균의 차이를 검증할 것이다. 귀무가설()은 ‘이태원 클럽발 집단감염사태 이전 지하철 승하차인원과 이후 지하철 승하차인원에 평균 차이가 없다.’ 로 설정한다. 대립가설()은 ‘두 집단 평균 간의 차이가 있다.’ 로 설정하였다.

이태원 클럽발 집단감염사태 이전과 이후의 평균 승하차인원에 차이가 없다.

이태원 클럽발 집단감염사태 이전과 이후의 평균 승하차인원에 차이가 있다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 전 체 |
| Levene-F | 0.823 | 5.523 | 1.476 | 4.123 | 3.044 | 5.139 | 3.395 | 1.433 | 2.260 |
| p-value | 0.365 | 0.019 | 0.225 | 0.043 | 0.082 | 0.024 | 0.066 | 0.232 | 0.133 |

표 3. 이태원 등분산 검정

표 3은 이태원 클럽발 집단감염사태(이하 이태원 사태)의 독립표본 t검정을 하기 전에 등분산 검정을 하기 위해 표 1과 마찬가지의 방법으로 Levene 검증을 시행한 결과이다. 표 3에서 ‘전체’ 항목은 모든 역의 승하차인원 평균 값을 이태원 사태 이전과 이후로 나눠 등분산 검정을 진행한 결과이다. 등분산 검증 결과 고속터미널역과 홍대입구역, 이태원역은 p-value 값이 0.05보다 작기에 등분산을 가정하지 않고 t-검정을 진행한다. 나머지 역들은 p-value 값이 0.05보다 크기 때문에 등분산을 가정하여 t-검정을 해야한다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 전 체 |
| t-statistic | -2.702 | -10.271 | -4.118 | -14.580 | -4.422 | -22.022 | -6.389 | -1.623 | -5.014 |
| p-value | 0.007 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.105 | 0.000 |

표 4. 이태원 t-검정

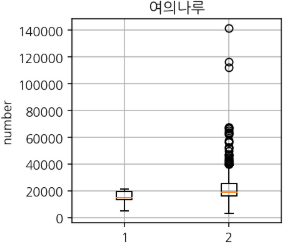
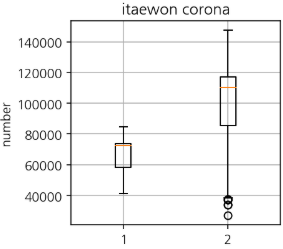


그림 8. 전체 역 평균 승하차인원(이태원) 그림 9. 여의나루역(이태원)

표 4는 등분산 검정 후 그에 맞게 t-검정을 진행한 결과이다. 여기서 주목할 점은 여의나루역의 p-value 값이다. 0.105로 유의수준 0.05보다 커 귀무가설을 채택한다. 두 집단간의 평균에 차이가 없다는 것이다. 그림 9에서 1이 이태원 사태 이후의 승하차인원이고 2가 그 이전의 승하차인원이다. 1의 중앙값과 2의 중앙값의 차이가 얼마나지 않은 것을 확인할 수 있다. 여의나루역의 이태원 사태가 지하철 승하차인원에 큰 영향을 주지 못하고 있다는 것을 알 수 있다. 그 외 나머지 역들은 p-value값이 0.05보다 작기 때문에 귀무가설을 기각하고 대립가설이 통계적으로 유의하다고 판단한다. 그림 8을 통해서는 이태원 클럽발 집단감염이 승하차인원에 전체적으로 어느정도 영향을 준다는 것을 알 수 있다.

4. 2 시계열 분석

4. 2-1 Simple Moving Average

단순이동평균은 특정 기간 동안의 개의 데이터를 더하여 으로 나눈 값이다. 즉, 특정 기간의 데이터의 평균을 계산한다는 것이다. 단순이동평균은 특정 기간 동안의 데이터의 중요도를 동일하다고 간주한다. 그래서 특정 기간 동안의 모든 데이터는 동일한 가중치가 적용된다. 이동평균의 특징인 지연(lag)이 발생하며 수학적으로 시간 만큼의 지연이 발생한다. 단순이동평균의 공식은 다음의 식 1과 같다.

식 1. 단순이동평균

본 논문에서는 이 공식을 통해 파이썬의 StatsModel 라이브러리를 활용하여 특정 기간 동안의 평균으로 역 별 승하차인원의 단순이동평균을 구하여 그 다음 날을 예측할 것이다. 또한 특정 기간을 2일, 3일, 7일로 설정하여 각각의 잔차를 확인할 것이다. 여기서 주의할 점은 rolling의 경우, window 3을 예로 들면 t시점에 대한 값으로 t시점, t-1시점, t-2시점의 평균을 사용한다. 그러나 본 논문에서는 t시점을 위해 t-1시점, t-2시점, t-3시점의 평균을 구해야 하므로 shift함수를 사용하여 예측값을 하루씩 밀어준다.

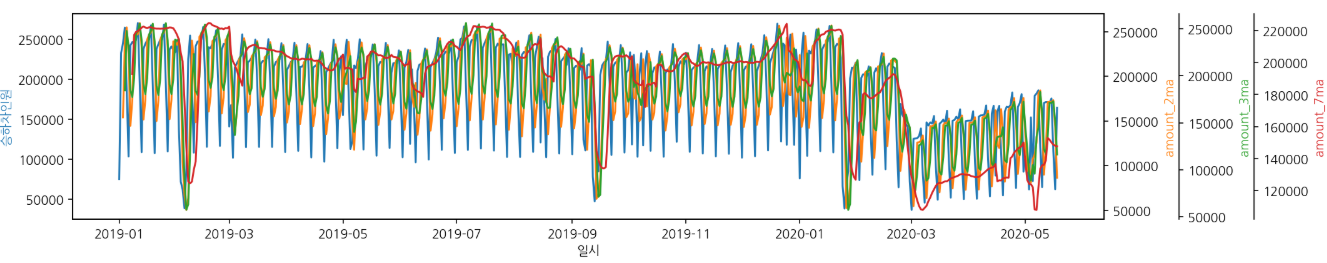


그림 10. 고속터미널역 단순평균이동 예측

그림 10은 고속터미널역 승하차인원의 실제값과 2일, 3일, 7일의 단순이동평균을 시계열 그래프로 나타내고 있다. 7일(빨간색 선)의 단순이동평균 추세가 실제값(파란색 선)과 비슷한 추세를 보이고 있다. 그러나 단순히 육안으로 어느 모델이 실제값과 비슷한 지는 판단할 수 없다. 어떠한 모델이 더 적합할지 수학적으로 증명하기 위해 MAE(Mean Absolute Error) 방법을 사용하고자 한다. MAE란 평균절대오차로서 쉽게 설명하면 예측값()에서 실제값()을 빼 절대값을 씌어 평균한 값이다. 식은 다음과 같다.

식 2. MAE 공식

잔차들의 절대값의 평균이기 때문에 단순히 잔차들의 평균(ME)을 통해 비교하는 경우 음수로 인해 발생하는 평균의 왜곡을 피할 수 있어 MAE를 적절한 예측모델 평가 방법으로 선정하였다. MAE를 통해 전체 8개 역의 2일, 3일, 7일 단순이동평균 모델을 비교해보았다. 2일 모델은 MAE 값이 18138로 나왔고 3일 모델은 18421, 7일 모델은 15901로 7일 모델이 MAE값이 제일 낮게 나왔다. 단순이동평균에서는 7일 모델이 제일 성능이 좋다고 해석할 수 있다.

4. 2-2 Exponential Moving Average

Exponential Moving Average(지수이동평균)은 가중이동평균 중의 하나로 단순이동평균보다 최근의데이터에 높은 가중치를 부여하는 방법이다. 단순이동평균과는 대조적으로 평활상수(Smoothing Constant)를 조절하여 반응속도와 각 데이터의 가중치를 결정할 수 있다. 과거의 데이터를 계산에서 제외하는 것이 아니라 서서히 그 영향력을 사라지게 하기 때문에 스파이크의 영향을 줄일 수 있는 좋은 방법이다. 지수이동평균의 공식은 식 3과 같다.

식 3. 지수이동평균 공식

식 4. 평활상수

식 3을 통해 알 수 있듯이 평활상수() 값에 따라 가중치의 감소추이가 변화하며, 이로 인해 실효데이터 개수와 반응속도가 결정된다. 다음 식 4는 지수이동평균의 평활상수()와 실효 데이터()개수의 상관관계를 나타낸다. 예를 들어 값을 0.2로 정의할 경우 실효데이터 개수는 9개로 한정할 수 있다는 것을 뜻한다.

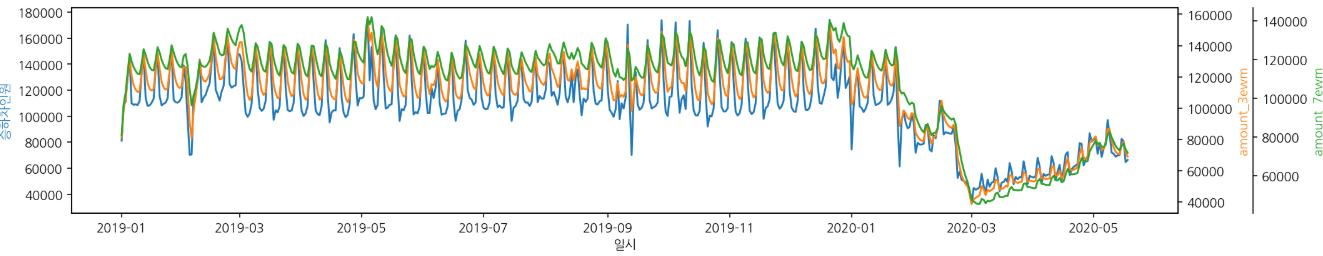


그림 10. 고속터미널 지수이동평균 예측

그림 10 은 고속터미널역 승하차인원의 실제값과 실효데이터()의 개수를 3과 7로 지정했을 때의 예측 시계열 그래프를 보여준다. 즉, 실제값과 3일, 7일 예측 모델 그래프이다. 그림 9(단순이동평균)의 예측 그래프와 비교하였을 때 그림 10(지수이동평균)이 예측값과 더 비슷한 추세를 보이는 것처럼 나타난다. 육안으로 어느 모델이 더 적합한지 판단할 수 없기에 MAE(식 2 참조)방법으로 판단하도록 하겠다. 3일치 예측 모델의 MAE값은 8229이고 7일치 예측 모델의 MAE값은 11961이다. 3일치 예측 모델이 더 적합하다는 것을 확인할 수 있다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 구 분 | SMA(단순이동평균) | | | EMA(지수이동평균) | |
| 2일 | 3일 | 7일 | 3일 | 7일 |
| MAE | 18138 | 18421 | 15901 | 8229 | 11961 |

표 5. 이동평균 MAE값

표 5는 단순이동평균과 지수이동평균의 일별 예측 모델 MAE값을 나타낸다. 단순이동평균의 예측모델들의 MAE값이 지수이동평균의 MAE값 보다 전체적으로 높아 지수이동평균의 예측 모델들이 단순이동평균의 모델들보다 성능이 뛰어나다고 해석할 수 있다. 또한 EMA의 3일 예측모델이 이동평균 모델 중 가장 MAE값이 적어 가장 뛰어난 성능의 예측모델이라고 평가할 수 있다.

4. 2-3 Holt’s Exponential Smoothing

지수평활법에는 추세 및 계절변동의 고려여부에 따라서 대략 3가지의 방법으로 구분된다. 본 논문에서는 그 중 홀트지수평활법을 활용하여 예측모델을 구축하고자 한다. 홀트지수평활법이란, 단순지수평활법에 추세변동을 반영하는 방법이다. 예측을 위한 식 외에 수준(level) 평활을 위한 식과 추세(trend) 평활을 위한 식이 포함된다. 그렇게 구축된 예측모형은 선형적으로 나타나기 때문에 예측 범위가 멀어질수록 over-forecast 되는 경향이 있다. 홀트지수평활법을 수식으로 나타내면 다음 식 5와 같다.

단,

식 5. 홀트지수평활법

식 5에서 는 t시점에서의 수준(level)의 추정치를 나타낸다. 수준에 대한 식인 는 t시간에 대한관측치 와 훈련예측 의 가중 평균을 나타낸다. 는 t시점에서의 기울기(추세)의 추정치를 나타낸다. 추세에 대한 식 는 에 근거한 t시간에 대한 추정치와 이전 추정치인 의 가중 평균을 나타낸다.

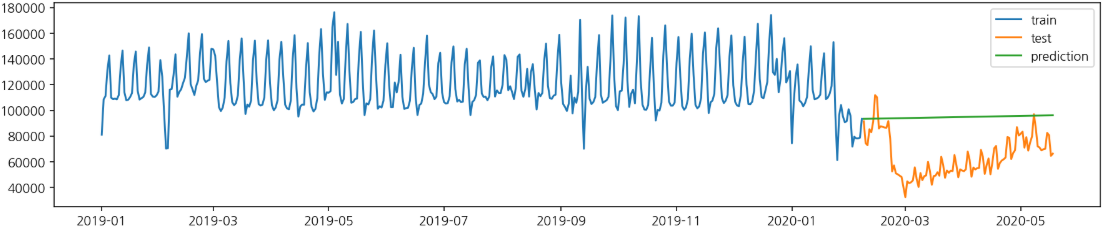


그림 11. 고속터미널역 홀트지수평활 예측

그림 11은 고속터미널역 승하차인원을 홀트지수평활법을 활용하여 예측한 시계열 그래프이다. 승하차인원의 추세를 반영하면 더 좋은 모델을 구축할 수 있을 것 같아 추세를 반영하기 위해 홀트지수평활법을 사용했다. train data는 원래 처음에 전체 데이터셋의 70%로 설정하였으나 그렇게 설정하게 되면 2019년 12월 중순까지의 데이터만 train data로 되서 코로나19로 인해 감소하는 승하차인원의 추세를 반영하지 못하게 된다. 그래서 그 추세를 반영해주기 위해 train data를 전체 데이터셋의 80%로 설정하였고 나머지 20%는 test data로 설정하였다. 그래프로 보면 2020년 2월초까지의 데이터가 train data로 설정되어 추세가 반영된 것을 확인할 수 있다. 그러나 실제 test값과 예측값의 차이가 많이 나는 것으로 보여진다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 평 균 |
| MAE | 62637 | 31299 | 37360 | 45339 | 15460 | 9787 | 25962 | 4339 | 29022 |

표 6. 홀트지수평활 역 별 MAE값

표 6은 모델의 예측성을 수학적으로 검증하기 위해 MAE 방법을 활용하여 역 별로 비교해보았다. 단순이동평균과 지수이동평균은 train과 test로 데이터를 구분하여 모델을 학습시키지 않았기 때문에 직접적인 비교가 불가하지만 홀트지수평활법의 MAE값이 전체적으로 크다는 것을 확인할 수 있다. 특히 강남역의 경우 MAE값이 제일 높았고 여의나루역이 제일 낮은 MAE값을 보였다. 전체 역의 평균 MAE값은 29022이다.

4. 2-4 prophet

Prophet은 페이스북에서 개발한 시계열 예측 패키지이다. ARIMA 같은 시계열 모델은 시간에 종속적인 구조를 가지지만 Prophet은 시간에 종속적이지 않다. 또한 ARIMA는 확률론적이고 이론적인 모형이지만 Prophet은 몇가지 경험적 규칙(heuristic rule)을 사용하는 단순 회귀 모형이다. 정확도가 높고 학습 속도가 빠르며 직관적으로 이해할 수 있는 하이퍼 파라미터(Hyper parameter)로 모델을 쉽게 수정할 수 있다는 장점을 가지고 있다. Prophet 모델은 Growth, Seasonality, Holidays로 구성되어 있다. 이 세가지를 결합하면 식 6과 같은 공식으로 나타낸다.

식 6. Prophet 공식

식 6에서 는 주기적이지 않은 변화인 Growth(추세)를 나타낸다. 이것은 Linear Growth(선형)와 Logistic Growth(로지스틱)로 나뉜다. Linear Growth 같은 경우 구간 구분점인 change point를 자동으로 탐지한다. 는 Seasonality로서 주기성, 계절성 등 주기적으로 나타나는 패턴들을 나타낸다. 푸리에 급수(Fourier Series)를 이용하여 패턴의 근사치를 찾는다. 는 Holidays로서 계절성을 가지진 않지만 전체 추이에 영향을 주는 이벤트를 의미하며 이벤트의 효과는 독립적이라고 가정한다. 휴일과 같이 불규칙한 이벤트들을 나타내고 특정기간에 값이 비정상적으로 증가하거나 감소하면 Holidays를 통해 모델에 반영할 수 있다. 마지막으로 은 정규분포라고 가정한 오차이다.

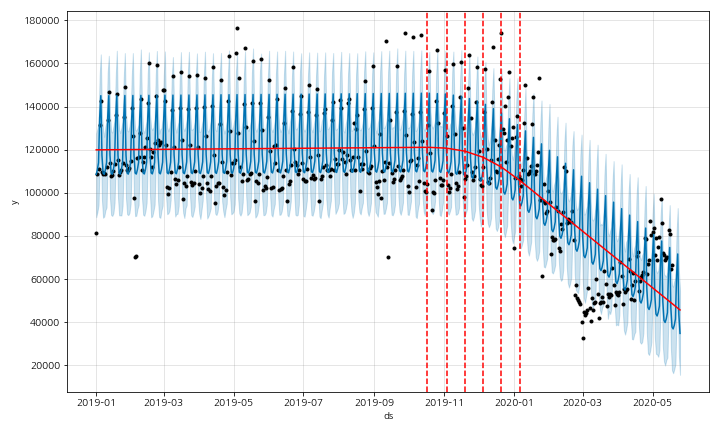
그림 12. 전체 역 평균 Prophet

그림 12는 전체 역 승하차인원의 일자별 평균 Prophet 예측모델이다. 역 별로 모델을 구축하기 전에 전체적인 추세를 확인하기 위하여 전체 8개역 승하차인원의 일자별 평균 Prophet 모델을 시계열 그래프로 나타냈다. 검은색 점이 실제값이고 파란색 선이 Prophet으로 예측한 추세선이다. 추세선과 점이 비슷한 추세를 보이고 있다는 것을 확인 할 수 있다. 빨간색 점선은 추세가 변하는 지점인 change point을 나타낸다. Prophet 같은 경우는 기본적으로 트렌드가 변경되는 지점을 자동을 감지하여 트렌드를 예측한다. 또한 감지하는 지점을 임의로 설정해줄 수 있다. 그러나 그림 12의 Prophet 예측모델에서는 따로 임의의 지점을 설정해주지 않았다. 따로 설정해주지 않아도 줄어드는 지점을 잘 포착했다는 것을 빨간색 점선을 통해 확인할 수 있다. 여기서 주목할 점은 마지막 빨간색 점선 시점 이후로 국내에 코로나19가 확산되기 시작하면서 승하차인원이 줄어들고 있는 추세로 변경되었다는 것이다.

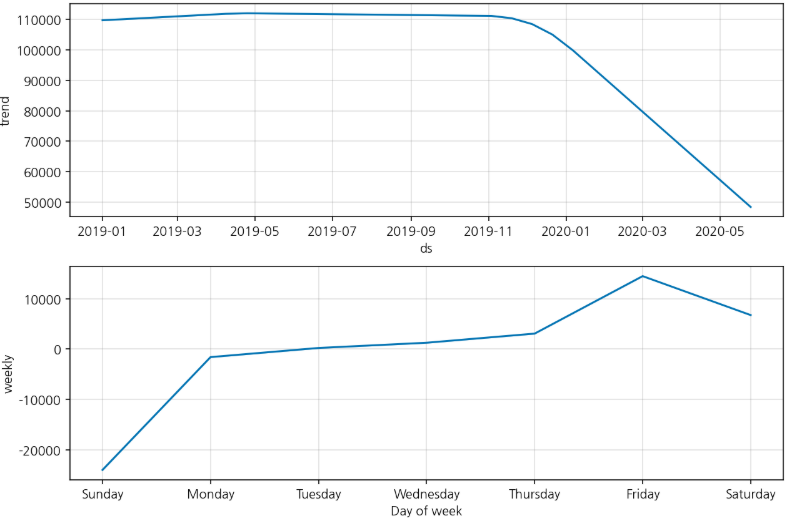


그림 13. 전체 역 평균 Prophet[trend, weekly]

그림 13은 전체 8개역의 승하차인원 일자별 평균 Prophet 모델을 추세(trend)와 주별(weekly)로 나타낸 그래프이다. 먼저 추세는 2019년 12월 시점 이후부터 쭉 감소하고 있는 추세로 예측된다. 그리고 주별로는 금요일이 승하차인원이 제일 많고 그 다음이 토요일로 나타나는 패턴을 보인다. 여기서 주목할 점은 의외로 일요일이 제일 낮은 승하차인원의 패턴을 보이고 있다는 점이다. 따라서 이후에 구축할 다른 모델들의 예측성을 높이기 위해 기존에 데이터 변수 중 휴일여부를 수정하고자 한다. 기존 월~목요일을 0으로 설정하고 금~일요일, 공휴일을 1로 설정한 것을 일요일, 월~목요일을 0으로 수정하고 금요일, 토요일과 공휴일을 1로 수정하였다. 즉, 일요일만 0으로 변경하고 나머지 값은 그대로인 것이다.

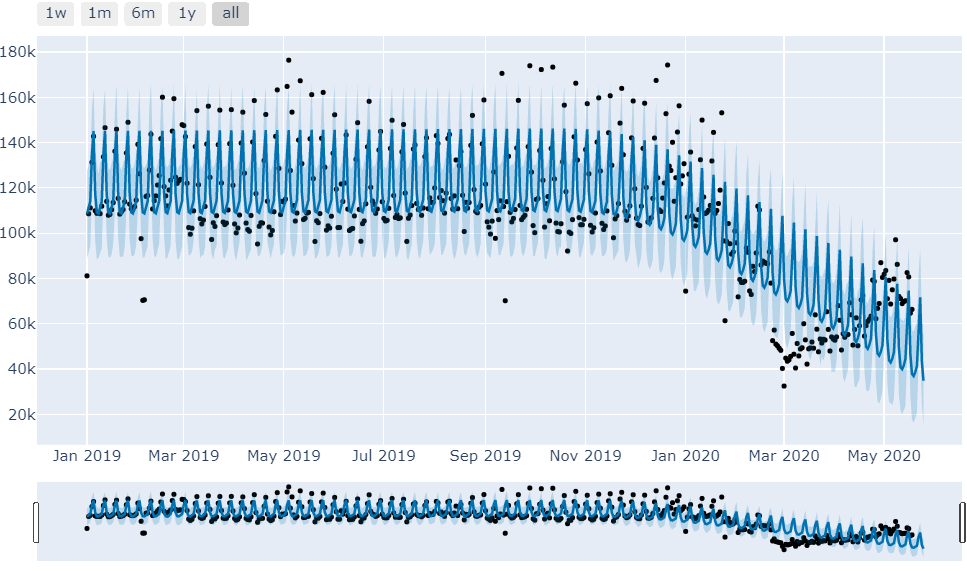


그림 14. 고속터미널역 Prophet

그림 14는 고속터미널역의 승하차인원 Prophet 예측 모델이다. Growth는 change point를 자동으로 탐지하기 위해 Logistic Growth(로지스틱)가 아닌 Linear Growth(선형) 방법을 선택하였다. 검은색 점이 승하차인원의 실제값이고 파란색선이 예측값이다. 그래프 마지막 시점 부분에 파란색선은 있고 검은색 점이 없는 것은 이 모델로 7일 간의 승하차인원을 예측하였기 때문이다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 2020.05.19 | 2020.05.20 | 2020.05.21 | 2020.05.22 | 2020.05.23 | 2020.05.24 | 2020.05.25 |
| 실제값 | 66045 | 70309 | 72250 | 87898 | 91203 | 66165 | 71969 |
| 예측값 | 36889 | 38591 | 41481 | 61854 | 71650 | 43532 | 34814 |
| 잔 차 | 29156 | 31718 | 30769 | 26044 | 19553 | 22633 | 37155 |

표 7. 고속터미널역 Prophet 잔차

7일간 예측한 결과값과 실제값을 비교해 본 결과가 표 7에 나와있다. 전체적으로 잔차가 크다는 것을 확인할 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| train size  역 명 | 0.7 | 0.8 | 0.8(holiday=20,  additive) |
| 강 남 | 48275 | 54040 | 72128 |
| 고속터미널 | 45175 | 41090 | 51305 |
| 잠실(송파구청) | 63047 | 49549 | 60552 |
| 홍대입구 | 56772 | 61468 | 74441 |
| 건대입구 | 28493 | 19862 | 27415 |
| 이태원 | 13665 | 9235 | 11753 |
| 명 동 | 46437 | 24875 | 24988 |
| 여의나루 | 6428 | 9966 | 8357 |
| 평 균 | 38,536.50 | 33,760.63 | 41,367.38 |

표 8. Prophet MAE

파이썬의 Prophet 라이브러리와 for 반복문을 통해 모든 역의 승하차인원을 예측하는 역 별 Prophet 모델을 구축하여 보았다. 이 과정에서 최적의 모델을 구축하기 위해 여러가지 설정을 조정하며 MAE값을 비교해보았다. 먼저 train data의 크기를 얼마로 설정하는 것이 더 좋은 예측값을 보이는 지 비교하기 위하여 train data를 전체 데이터의 각각 70%와 80%로 설정하여 비교하였다. 표 8이 MAE값을 모델별로 비교해놓은 표이다. train data를 0.8로 설정한 것이 0.7로 설정한 것보다 MAE값이 더 작다. 이것은 train data를 0.8로 설정한 것이 더 좋은 성능의 모델이라고 해석할 수 있다. 표 8에서 0.8(holiday=20, additive) 모델은 train data를 0.8로 설정하고 holiday 관련 하이퍼 파라미터(Hyper parameter)의 반영 강도를 20으로 설정하고 seasonality mode의 하이퍼 파라미터를 Additive로 구분한다는 뜻이다. Holiday는 데이터에 영향을 미치는 휴일이나 이벤트를 알고 있다면, 모델에 반영하여 정확도를 높일 수 있는 하이퍼 파라미터이다. 이전에 그림 13에서 weekly 그래프를 보면 금요일과 토요일이 높은 패턴을 보이고 있기 때문에 holiday 하이퍼 파라미터로 설정해준다. 또한 코로나19라는 이벤트로 인해 승하차인원이 감소하고 있기 때문에 신천지발 집단감염사태 시점 이후부터 마지막 시점까지의 데이터를 holiday 하이퍼 파라미터로 설정한다. Additive는 승하차인원 데이터의 진폭이 거의 일정하기 때문에 seasonality mode 하이퍼 파라미터를 additive로 설정해주었다는 뜻이다. 이렇게 최적의 모델을 찾기 위해 여러 종류의 하이퍼 파라미터를 조정한 뒤 MAE값을 구해보았다. 그러나 임의로 하이퍼 파라미터를 설정한 0.8(holiday=20, additive) 모델이 앞서 두 개의 모델보다 더 높은 MAE값을 보이고 있다. 최종적으로 Prophet 예측 모델 중에서는 0.8 모델이 제일 최적의 성능을 가진 예측 모델이다.

4. 3 다중선형회귀모형(Multiple Linear Regression Model)

다중선형회귀모형은 선형회귀 중 하나로서 두 개 이상의 독립변수와 하나의 종속변수를 다루는 다중회귀분석으로 예측하는 모형이다. 다중선형회귀식은 다음 식 7과 같다.

식 7. 다중선형회귀식

종속변수 와 개의 독립변수 와의 관계식이다. 는 각 독립변수 의 회귀계수이며 는 오차 변수로서 관찰되지 않는 종속변수와 독립변수 사이에 오차를 뜻한다. 본 논문에서는 파이썬의 Sklearn 라이브러리의 LinearRegression 메소드와 statsmodel의 OLS 메소드를 활용하여 다중선형회귀모형을 구축하고자 한다. LinearRegression 메소드로 예측값을 구하고 OLS 메소드를 통해 각 변수들의 요약을 보고서 형태로 확인할 것이다. 데이터셋은 train data는 전체데이터의 80%, test data는 전체데이터의 20%로 분리하여 설정하였다. 다음에 보이는 표 9는 역 별 다중회

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 평 균 |
| MAE | 872878 | 1002411 | 1150641 | 1096095 | 682690 | 281965 | 755806 | 155804 | 749786 |

표 9. LinearRegression MAE값

귀모형을 구축한 뒤 test데이터와 예측값의 MAE값이다. MAE값이 다른 이전의 모델들과 비교하였을 때 매우 크다는 것을 확인할 수 있다. 다중선형회귀모형은 최적의 승하차인원 예측 모델이 아니라고 해석할 수 있다.

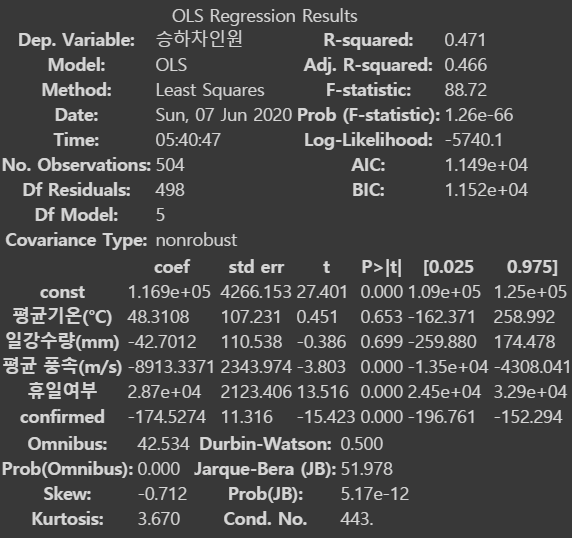


그림 15. 고속터미널역 OLS summary

그림 15는 고속터미널역 다중선형회귀모형의 OLS 메소드를 활용한 보고서이다. 여기서 R-squared값은 보통 설명력이라고 하는데 주어진 데이터가 현재 모형을 얼마나 잘 설명하고 있는 지를 나타내는 값이다. 대개 0.7이상이 나와야 설명력이 높다고 하지만 이 모형에서는 0.471이라는 낮은 설명력을 보이고 있다. 변수들의 P> 값은 변수의 계수에 대한 p-value로 해당 독립변수가 유의미한지 판단한다. 평균기온과 일강수량의 변수가 0.05 이상으로 통계적으로 유의미하지 않은 변수라는 것을 확인할 수 있다. 변수들의 coef는 각 변수들의 회귀계수를 뜻한다. 변수들 중에서는 평균풍속의 회귀계수가 -8913으로 종속변수인 승하차인원에 제일 큰 영향력(가중치)을 미치고 있다고 유추할 수 있다. 이 보고서를 토대로 평균기온과 일강수량 변수를 빼고 OLS를 돌려봤지만 별 차이가 없는 결과가 나와 다중선형회귀모형은 더이상 추가적인 모형을 구축하지 않기로 결정하였다.

4. 4 LSTM 모델

LSTM은 RNN의 장기의존성 문제를 해결하기 위해 제안된 기법으로서 RNN의 일종이다. RNN은 학습을 할 때 입력데이터 뿐만 아니라 이전 단계에서 출력된 데이터를 동시에 이용하여 학습이 가능하기 때문에 시계열 데이터처리에 특화된 알고리즘으로 알려져있다. 그러나 RNN은 데이터가 길어져 이를 표현하는 신경망이 깊어져야 할 경우 문제가 발생한다. RNN은 역전파(Backpropagation)라는 방법을 통해 학습을 하는데 그래디언트가 너무 작아져 학습이 잘 안되는 가중치 소실(vanishing gradient) 문제가 발생한다. 이 문제를 방지하기 위해 LSTM 기법이 제안되었다. LSTM의 핵심인 셀 스테이트(cell state)라는 개념을 도입하여 그 내부에 있는 게이트(gate)들을 통해 어떤 정보를 기억하고 어떤 정보를 버릴지 추가적인 학습을 가능하게 한다. 이를 통해 RNN이 가진 가중치 소실 문제를 해결할 수 있다. Horchreiter & Schmidhuber(1997)에 의해 처음 제안되었고, 많이 개선되고 대중화되면서 다양한 문제에 적용되기 시작했다.

LSTM 모델을 구현하기 위해 ‘텐서플로(Tensorflow)’ 라는 딥러닝 라이브러리를 이용하고자 한다. 텐서플로는 구글(Google)에서 개발하여 공개한 딥러닝과 머신러닝(기계학습)을 위한 오픈소스 라이브러리이다. 기본적으로 파이썬 환경에 최적화되어 있어 본 논문에서 파이썬으로 구축하고자 하는 lstm 예측 모델을 쉽게 구축할 수 있다. 먼저 신경망을 빠르고 수월하게 학습시키려면 데이터를 정규화시켜야 하기 때문에 다음의 식 8을 활용하여 각 변수별로 데이터의 범위가 0에서 1사이가 되도록 정규화를 시킨다.

식 8. 정규화

그 후에 lstm 모델의 하이퍼 파라미터를 설정해준다. 총 10개의 하이퍼 파라미터를 설정하고자 한다. 먼저 tf.set\_random\_seed(777)을 통해 난수를 설정해준다. Input dimension은 입력데이터의 변수 개수로서 데이터의 변수가 평균기온, 일강수량, 평균풍속, 휴일여부, 코로나19 확진자 수, 승하차인원으로 총 6개이기 때문에 6으로 설정해준다. Output dimension은 결과데이터의 개수가 1개이기 때문에 1로 설정한다. Sequence length는 이전 7일치의 데이터를 통해 예측하는 것이기 때문에 1개의 시퀀스의 길이를 7로 설정한다. Hidden dim은 임의로 20으로 설정해준다. Forget bias(망각편향)은 기본값인 1.0으로 설정한다. Stacked layers는 layer의 개수가 1개이기 때문에 1로 설정한다. Keep\_prop 값은 dropout과 관련하여 주어진 유닛을 유지할 확률, 즉 drop을 하지 않을 확률이다. 본 논문에서는 실제 값을 측정해야 하기 때문에 keep\_prob을 1.0으로 설정하여 모든 유닛을 유지하여 모델의 성능을 극대화시킬 것이다. Epoch 횟수는 전체학습데이터를 반복해서 학습하는 횟수이다. 처음에는 1000번으로 설정하여 100번씩마다 RMSE(Root Mean Square Error)값이 출력되도록 하여 비교하였는데 100번의 Epoch가 RMSE값이 제일 낮게 나와 Epoch횟수를 100으로 설정하였다. 마지막으로 Learning rate는 학습률로서 0.01로 설정하여 lstm 모델을 구축하였다. 이 lstm 모델은 과거 특정기간의 승하차인원을 이용해서 다음날의 승하차인원을 예측하기 때문에 MANY-TO-ONE 형태의 모델이다. 데이터셋은 train data는 전체데이터의 80%, test data는 전체데이터의 20%로 분리하여 설정하였다. 손실함수로 평균제곱오차를 사용하였고 최적화함수로 AdamOptimizer를 사용하여 손실함수를 최소화시키도록 학습하였다. Epoch횟수에 따른 모델의 오차를 검증하는 방법에는 식 9의 RMSE(Root Mean Square Error)를 활용하였다. RMSE는 표준편차를 기준으로 하기 때문에 큰 에러 값을 효율적으로 줄일 수 있다. RMSE의 문제점은 scale(크기)에 의존적이라는 것인데 이 모델에서는 이미 scale을 0과 1사이의 범위로 정규화 시켜주었기 때문에 scale의 문제점은 발생하지 않는다. 따라서 이 모델의 오차를 검증하는 최적의 방법으로 RMSE를 이용했다.

식 9. RMSE(Root Mean Square Error)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 평 균 |
| epoch(100) | 0.2007 | 0.2305 | 0.1805 | 0.2272 | 0.2150 | 0.2083 | 0.2402 | 0.1685 | 0.21 |
| epoch(200) | 0.2095 | 0.2772 | 0.2942 | 0.2661 | 0.3580 | 0.3071 | 0.2898 | 0.2317 | 0.28 |

표 10. Epoch횟수에 따른 역 별 RMSE값

위의 표 10을 통해 Epoch 횟수가 100, 200일때의 역 별 test data에 대한 RMSE값을 비교해보았다. Epoch 횟수가 100번일 때가 200번일 때 보다 전체적으로 RMSE값이 낮고 평균 RMSE값 또한 낮은 것을 확인할 수 있다. 단순히 이것만 보았을 때는 Epoch 횟수가 100번일 때의 모델이 200번일 때의 모델보다 더 좋은 성능의 모델이라고 추론할 수 있다.

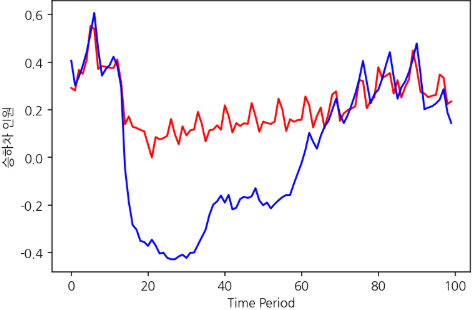
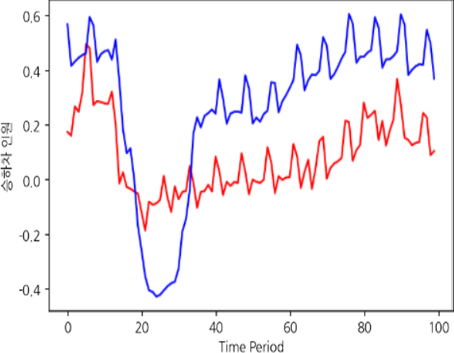


그림 16. lstm 고속터미널역 Epoch(100) 그림 17. lstm 고속터미널역 Epoch(200)

그림 16과 그림 17은 고속터미널역 승하차인원 lstm 모델의 Epoch가 각각 100, 200일때의 test data에 대한 예측 그래프이다. 파란색 선이 예측값이고 빨간색 선이 실제값이다. 두 그래프를 비교해보았을 때 최근의 데이터에서는 Epoch가 200일때가 100일때보다 더 비슷한 추세를 보이고 있다. 앞서 표 10에서 Epoch 횟수에 따른 RMSE값을 비교했을 때는 100이 더 좋은 성능의 모델이라고 추론하였다. 그러나 그래프를 통해 최근의 데이터만 확인하였을 때는 Epoch 200이 더 좋은 예측값을 보일 것이라고 추측할 수 있다. 그리하여 최종적으로 본 lstm 모델에서는 Epoch 횟수가 200번일 때의 모델을 최적의 성능을 가진 모델로 선택하고자 한다

5. 결 론

본 논문에서는 딥러닝 알고리즘을 이용하여 총 여섯 종류의 역 별 승하차인원 예측 모델을 구축해보았다. MAE값을 통해 오차검증을 시행한 모델은 단순이동평균, 지수이동평균, 홀트지수평활, Prophet, 다중선형회귀모형이다. 이 중 단순히 MAE값만 비교해봤을 때 좋은 성능의 예측모델은 3일 지수이동평균(EMA) 모델이다. 3일 지수이동평균은 MAE값이 8229로 위의 모델들 중 제일 낮은 값을 보였다. 그러나 여기서 주의할 점은 단순히 MAE값만 비교하였을 때의 결과라는 것이다. 지수이동평균과 단순이동평균은 train data와 test data로 데이터셋을 분리하지 않았기 때문에 모델에 대한 과적합(overfitting)의 문제가 발생할 수 있다. 또한 홀트지수평활, Prophet과 다중선형회귀모형은 모두 데이터셋을 분리시켜서 학습시킨 예측모델이고 test data에 대한 MAE값이라 직접적인 비교를 통해 최적의 성능을 가진 예측 모델을 구하는 것은 한계가 있을 수 있다. 여섯 종류의 예측모델 중 오차검증을 통해 최적의 성능을 지닌 모델을 구하고 싶으나 위와 같은 한계와 함께 LSTM 모델의 경우 정규화된 값으로 RMSE를 통해 오차를 검증하였기 때문에 직접적인 비교가 불가하다. 때문에 같은 오차검증의 방법(MAE)을 통해 비교하고자 한다. 표 11은 여섯 종류의 예측 모델을 활용하여 실제 마지막 데이터의 다음날인 2020년 5월 19일의 승하차인원을 역 별로 예측하여 그 예측값과 실제값의 MAE값을 비교한 표이다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구 분 | 강 남 | 고ㆍ터 | 잠 실 | 홍 대 | 건 대 | 이태원 | 명 동 | 여의나루 | 평 균 |
| S.M.A | 22605 | 5819 | 8541 | 1919 | 2063 | 1371 | 3585 | 847 | 5,843.75 |
| E.M.A | 33124 | 5845 | 14580 | 4115 | 4573 | 1135 | 5423 | 45 | 8,605.00 |
| Holt | 32788 | 298 | 12930 | 1190 | 3921 | 2021 | 5486 | 141 | 7,346.88 |
| Prophet | 27677 | 29156 | 30219 | 23466 | 13155 | 3269 | 19963 | 1840 | 18,593.13 |
| LinearReg | 18739 | 34148 | 30637 | 58016 | 16631 | 14031 | 31482 | 53264 | 32,118.50 |
| LSTM | 18171 | 2852 | 6207 | 9853 | 3644 | 2439 | 4980 | 433 | 6,072.38 |

표 11. 2020년 5월 19일자 예측 모델 MAE

표 11에서 단순히 MAE값을 통해 비교해보았을 때, S.M.A(단순이동평균)의 MAE값이 5843으로 제일 낮고 그 다음이 LSTM 모델이었다. 다중선형회귀모형(LinearReg)은 32118이라는 제일 높은 MAE값을 보였다. MAE값만 봤을 때, 의외로 상대적으로 단순한 모형인 S.M.A 모델이 제일 최적의 성능을 지닌 모델로 나타났다. 데이터의 수가 적고 승하차인원의 뚜렷한 seasonality가 없어서 복잡한 모형보다 단순한 모형이 더 좋은 성능을 보인 것으로 추측된다. 그 다음으로 좋은 성능을 지닌 모델은 LSTM 모델이다. 굉장히 복잡한 모형이지만 S.M.A 모형과 MAE값이 얼마 차이 나지 않는다. 데이터의 수가 많거나 더 다양한 변수들을 사용했다면 더 좋은 성능의 모델로 향상되었을 것이라 추측된다.

본 논문을 통해 LSTM 모델은 굳이 변수들의 상관관계가 높지 않더라도 다양한 변수들을 활용하면 LSTM 모델로 충분히 좋은 예측을 할 수 있다는 점을 알게 되었다. 향후 연구과제로서 더 좋은 성능을 갖춘 LSTM 모델을 구축하기 위해 승하차인원과 높은 상관관계를 갖는 다양한 변수들을 찾는 연구가 요구된다. 예를 들면 미세먼지, 초미세먼지 농도가 승하차인원에 미치는 영향에 대한 분석이 필요하다. 또한 데이터의 수가 적고 코로나19 확진자 변수의 데이터가 다른 변수들에 비해 부족하기 때문에 추후 코로나19 확진자 변수 데이터가 어느정도 쌓이면 새로운 예측 모델에 대한 연구가 필요할 것으로 판단된다.

참 고 문 헌

이수미, 홍성조(2019). 도시공간에서 날씨와 계절이 보행량에 미치는 영향. 한국산학기술학회

오주삼, 정진혁, 김연복(2002). 일반국도에서 법정 공휴일에 따른 교통량 변동패턴 분석. 대한토목학회

장동익, 임서현, 성낙문(2020). 코로나 19의 육상교통부문 영향분석 및 대응방향. 한국교통연구원

이기종(2004). 교육평가용어사전. 학지사

지성민(2019). 데이콘 상점 신용카드 매출 예측 경진대회. DACON.site.

Retrieved from <https://dacon.io/competitions/official/140472/codeshare/419/>

윤기훈, 김건욱, 최재훈, 박수준(2010). 지수이동평균을 이용한 RSSI 기반 근거리 사용자 탐지 시스템

임은순(2007). 레스토랑 매출액 예측 지수평활법과 ARIMA모형을 중심으로. 한국호텔외식관광경영학회

정성호, 이태업, 이경상(2018). 딥러닝 오픈 라이브러리를 이용한 하천수위 예측, 한국방재학회