다빈치 SW 공모전

20180378 응용통계학과 정현희
20184698 응용통계학과 김민지

0. abstract

명절 연휴 귀향 시 코로나로부터 안전하게 휴게소를 이용할 수 있도록 돕는 어플리케이션을 제작했다. 귀향 시 사람들이 많이 밀집되어 집단감염될 가능성이 높은 장소를 휴게소로 잡고, 휴게소를 타겟으로 하여 사람들을 분산시키는 방법을 제시하고자 ai를 적용하여 어플리케이션을 제작하였다. 이는 출발지와 도착지를 입력하면 경로 추천과 함께 해당 경로에서 이용가능한 drive-thru 휴게소를 안내하는 기능을 한다.

가장 먼저 공공데이터로 제공하는 코로나 확진자 관련 데이터를 시각화하여 전반적인 현황에 대한 분석을 진행했다. 이에 이어서 지역별로 확진자수를 분석하여 clustering 으로 위험군을 선정하였고, 시계열 모델인 ARIMA, Facebook prophet, RNN모델을 사용하여 확진자 수를 예측한 결과를 통해 위험지역으로 갈 경우, 알림을 주어 경각심을 주도록 했다.

두 번째로는 2019 명절 고속도로 톨게이트 입출구 교통량 데이터를 이용하여 밀집도가 높을 것으로 예상되는 휴게소를 추려내고, 예상 밀집도 상위 5개 휴게소 근방 졸음쉼터를 명절 기간 동안 Drive-thru 휴게소로 운영할 예정이다. 이에 대한 자세한 내용은 함께 제출한 notebook 파일 혹은 pdf 파일에서 확인 가능하다.

※ pdf파일에서는 folium을 이용해서 지도에 시각화한 것이 제대로 보이지 않는 것 같아(다른 컬러로 표현한 icon들이 같은 색상으로 표현됨) notebook 파일도 같이 제출하였습니다!

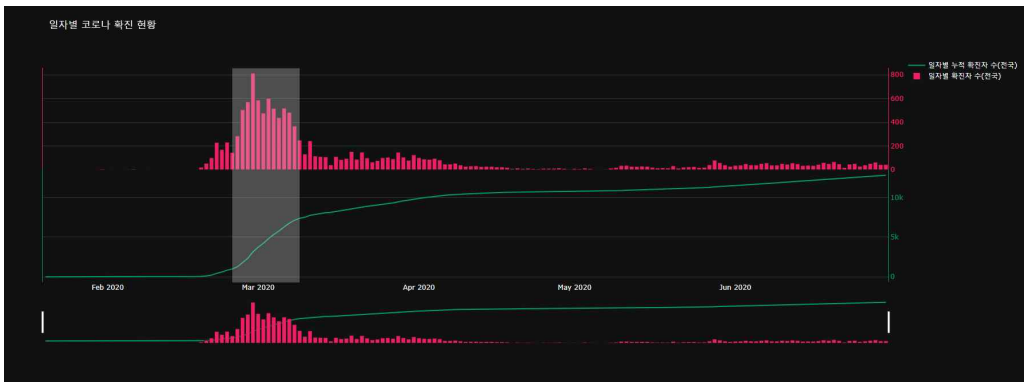
1. 코로나 데이터 분석

코로나 데이터를 바탕으로 탐색적 분석을 통하여 전반적인 코로나 감염 현황을 확인한 후 군집분석(clustering)을 사용하여 위험군을 선정한다. 이후에 시계열 모델과 순차(sequential) 모델인 RNN을 사용하여 확진자를 예측했다.

1-1. 일자별 신규 & 누적 확진자수 확인

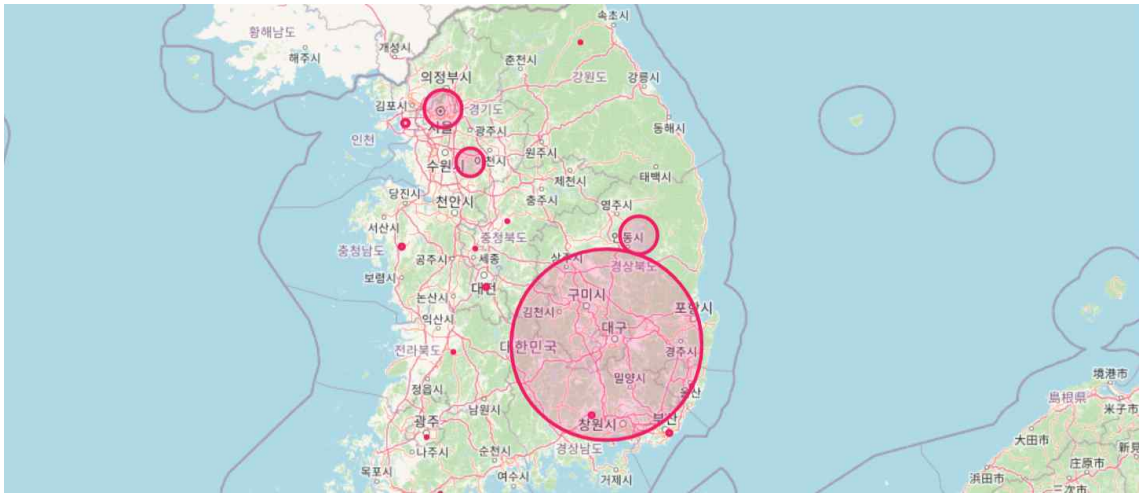
1-1-1. 일별 확진자수를 기존 데이터에 추가

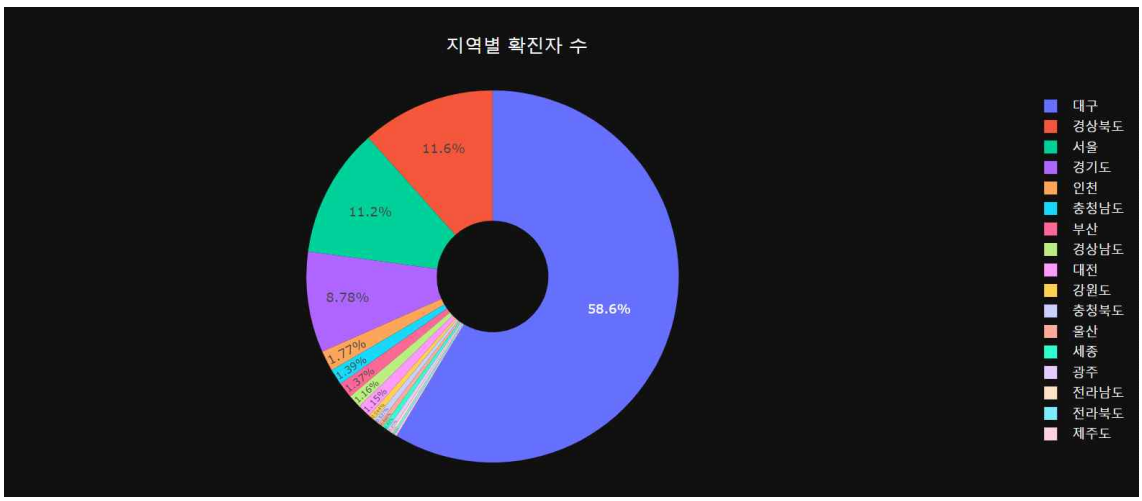
가장 먼저, 기존 데이터에 일별 확진자 데이터가 없음을 확인한 수에 일별 확진자수를 기존 데이터에 추가하도록 하는 함수를 작성하여 데이터를 구축했습니다. 이를 바탕으로, 일자별 누적 확진자수와 일자별 확진자수를 시각화한 결과, 3월에 가장 많은 확진자가 나왔음을 확인할 수 있고, 다시 잠잠하다 2020년 6월부터 재확산 되었음을 확인할 수 있다.



1-1-2. 코로나 19 확진자 map

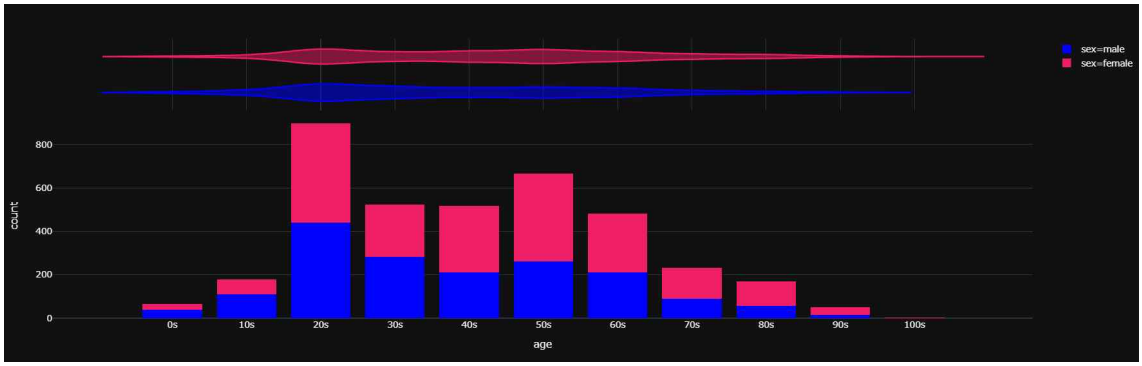
각 지역의 위,경도 데이터를 활용하여 지역별 감염 추세를 살펴보았다.





위의 pie chart를 확인하면 대구, 경북, 경기 순으로 코로나 확진자가 있음을 확인할 수 있다. 대구는 신천지 집단 감염 사태로 인해 확진자가 폭증했는데 이와 인접한 결과이다.
최근 확진세에 의해 대구,경북을 제외하면 수도권 지역 확진자수가 가장 많다.

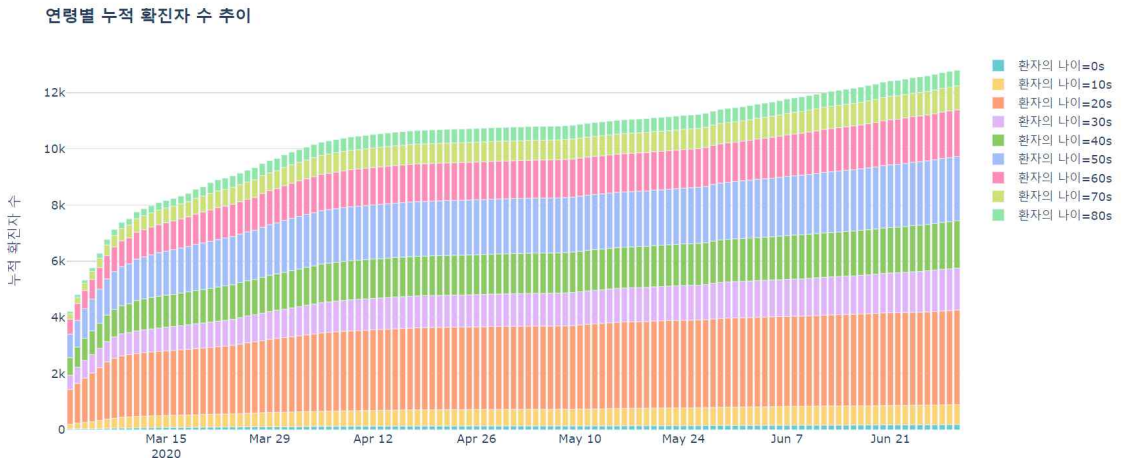
1-1-3. 성별 / 나이대별 확진자수

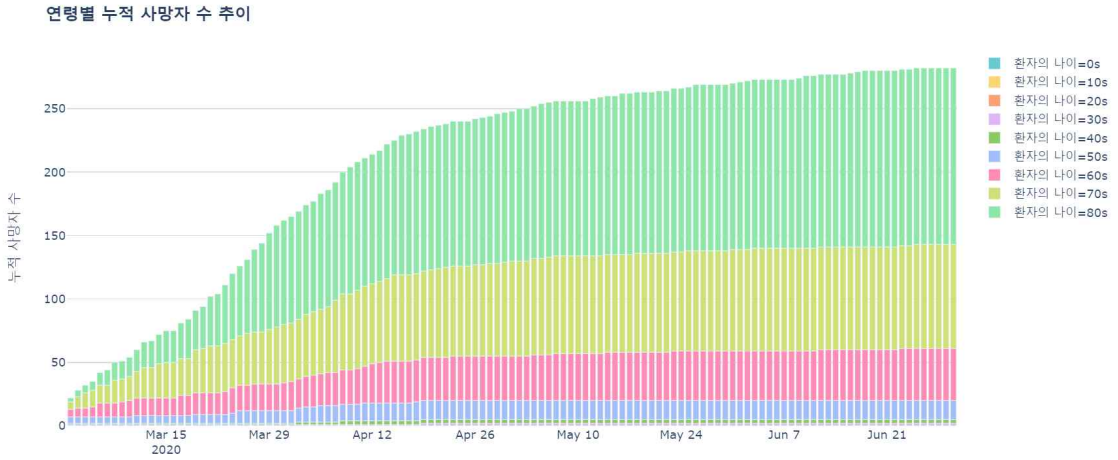


위의 barplot을 확인하면 20대의 확진자 비율이 가장 높음을 확인해 볼 수 있다. 이는 20대가 10대보다 비교적 활동하는 반경도 없고, 30대보다 비교적 여유로운 시간을 가진다는 점에서 나온 결과이다.

1-2. 연령별 누적 확진자 및 사망자 분석

확진자 데이터에서 환자의 나이를 연령대별로 카테고리를 나누어 범주형 변수로 바꿔줬다. 이를 바탕으로 연령별 누적 확진자수와 사망자수 추이를 살펴본 결과는 아래와 같다.

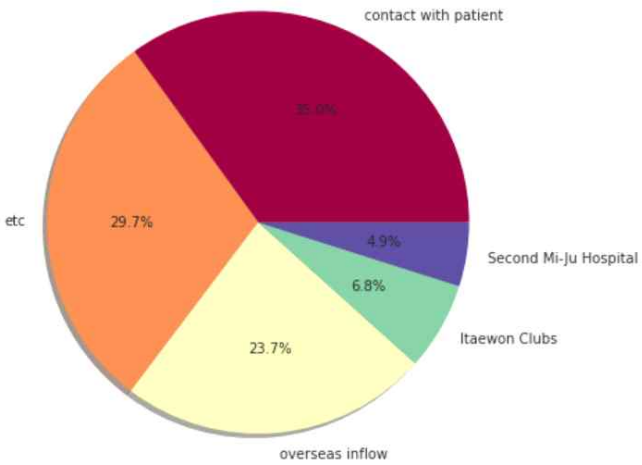




위의 결과를 확인하면, 연령별 확진자수는 청년층 38%, 중년층 31%, 고령층 24%, 미성년자 7% 수준이다. 반면 사망자수를 확인하면 고령층이 압도적으로 많음을 확인할 수가 있다. 즉 고령층은 다른 연령대보다 코로나가 생사에 더 치명적이라고 할 수가 있다. 특히나 추석은 고령층이 청,장년층과 만나는 비율이 높은 시기이므로 가장 조심해야 할 때라고 판단할 수가 있다.

1-3. 감염 이유 분석

코로나 감염 이유 데이터를 사용하여 짐단 감염 사례 중 상위 6개를 확인했다. 6개를 확인한 결과 신천지 집단 감염 사례는 독보적이므로 그를 제외한 나머지 다섯가지 사례에 대한 pie chart를 생성하여 시각화 했다.

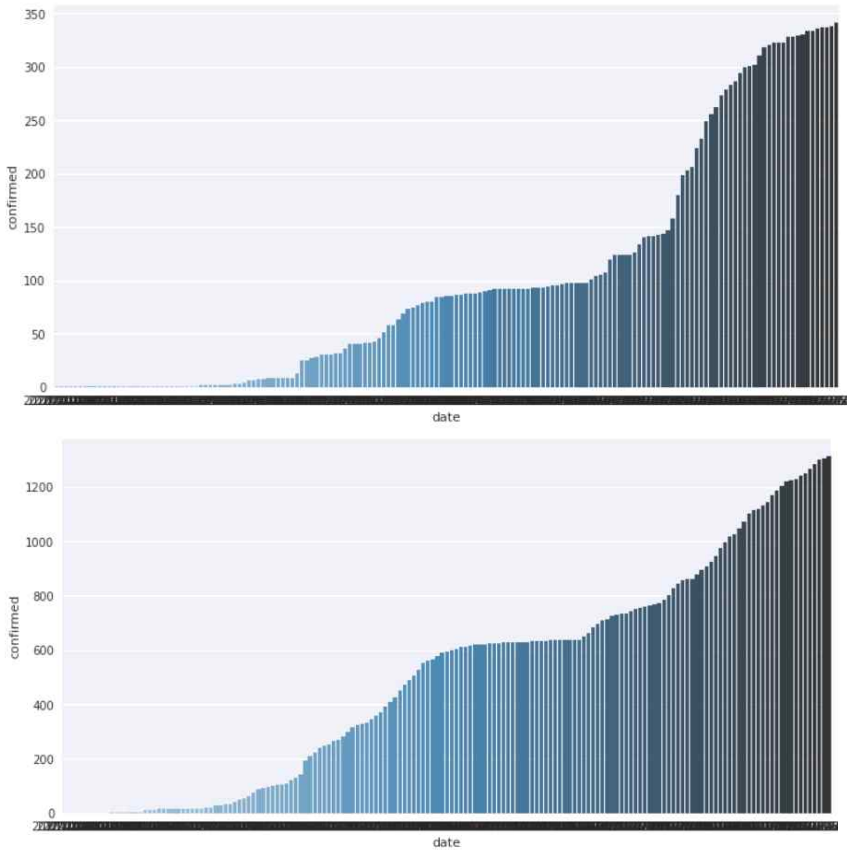


위의 감염 이유를 확인하면, 해외유입을 제외하면 대부분 집단감염이 가장 큰 원인이었음을 확인할 수가 있다. 따라서 다가오는 추석이 집단감염으로 COVID-19가 전파될 위험이 아주 큼을 확인할 수가 있다.

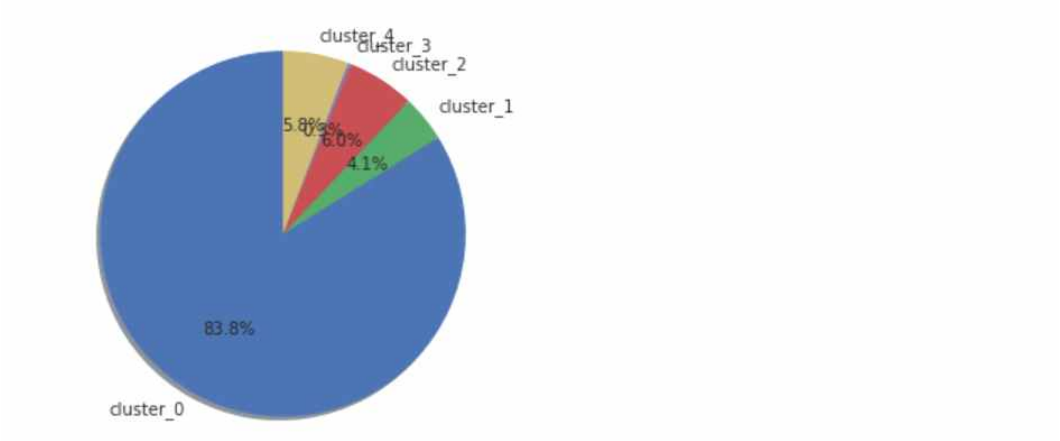
1-4. 확진자수를 기준으로 위험군 선정

코로나로 인한 사회적 거리두기 기간이 길어짐에 따라 사람들이 코로나의 위험성에 대한 경각심이 사라지고 있는 상황이다. 따라서 이동하는 지역이 확진자 수가 높은 위험 지역으로 분류된다면 알림을 주도록 하는 기능을 추가하고자, 군집분석(clustering)을 사용하여 위험군을 선정했다.

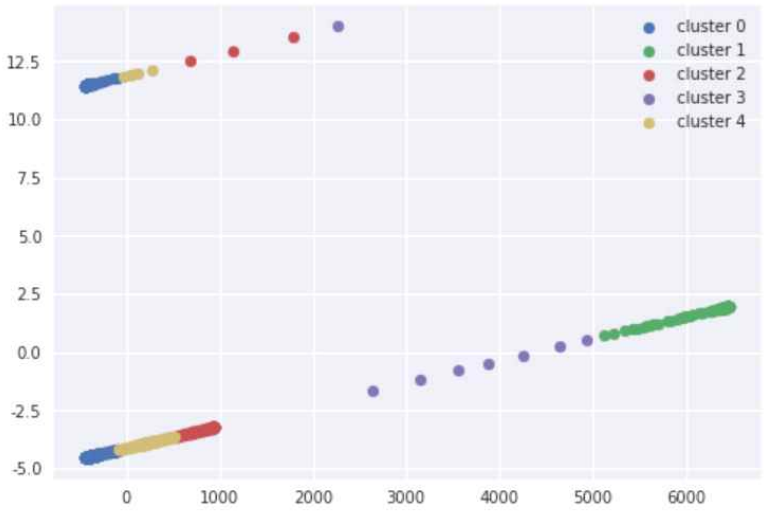
위험군을 선정하기 전, 몇 지역에 대해 확진자수 추이를 살펴본 결과는 아래와 같다. 대표적으로 인천, 서울지역 확진자에 대한 시각화를 순서대로 첨부한다.



지역별 확진자수 데이터를 바탕으로 k-means clustering을 적용하여 총 5개의 군집을 만들다. 차례대로 최고위험군, 고위험군, 보통, 저위험군, 최저위험군으로 구분했다. 클러스터링 결과는 아래와 같다.

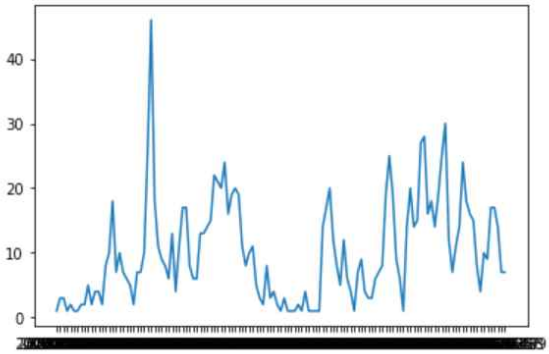


클러스터링이 잘 되었는지를 확인하기 위해 PCA를 적용한 결과는 아래와 같고, 클러스터링이 잘 이루어 졌음을 확인할 수 있다.



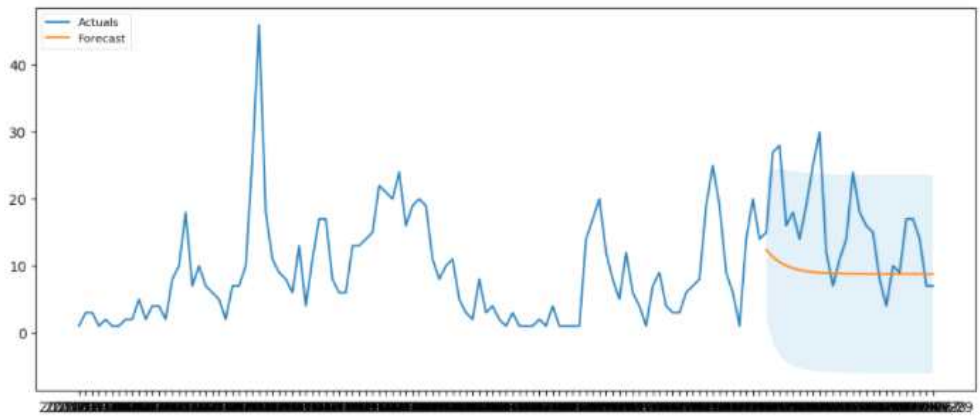
1-5. 지역별 확진자수 예측 모델링

먼저 전체 확진자 데이터를 지역별로 분할하여 따로 데이터를 저장했다. 이를 바탕으로 시간대별 확진자 추이를 지역별로 확인했다. 총 17개의 지역이 있었으며, 이 보고서에서는 서울지역에 대한 시각화와 모델링 결과를 첨부한다.



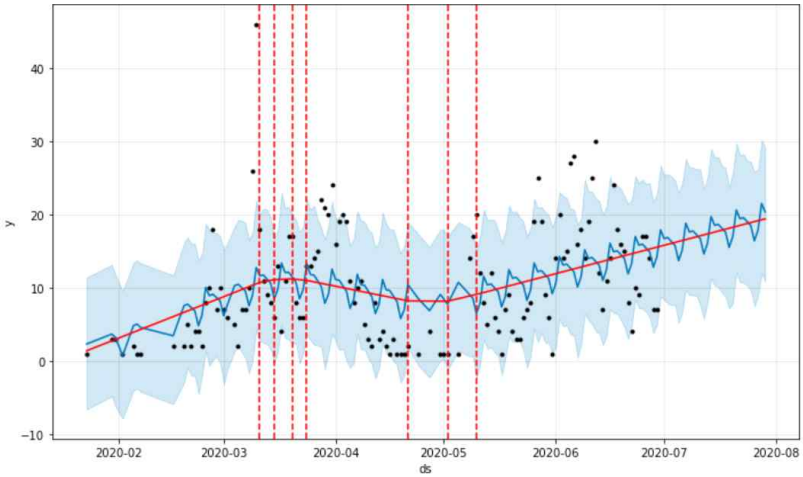
날짜에 따른 서울지역 확진자 추이는 위와 같다. 위의 그래프를 확인하면 중간중간 (3월, 6월) 등에 이벤트가 생기면서 확진자가 급증하는 형태를 확인해볼 수가 있다. 따라서 그런 불규칙한 이벤트를 고려하는 시계열 모델을 사용해야 한다고 판단했다.

사용한 모델은 총 3가지로, ARIMA, facebook prophet, RNN모델이다. ARIMA의 경우는 기초적인 시계열 분석 통계모델로 사용되는 AR, MA, ARMA모델들이 분석하는 데이터의 추세와 계절성 뿐 아니라 데이터의 자기상관성까지 고려하는 모델이므로 사용했다, 아래의 결과는 ARIMA 모델을 적용하여 예측한 결과를 시각화한 형태이다.



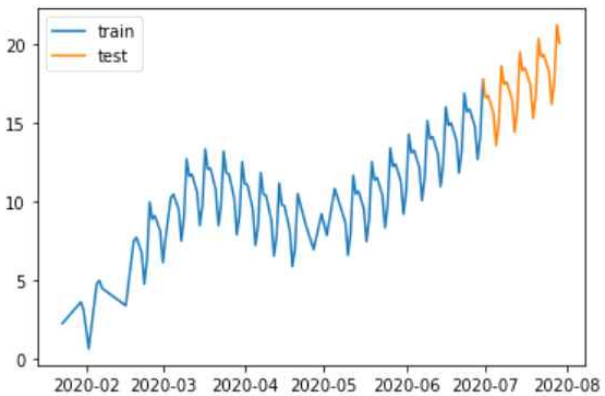
(* 파란색 선은 실제 결과이고 주황색 선은 예측한 결과이다)
위의 결과를 확인하면 예측결과가 그다지 좋지 않음을 확인할 수 있다. 예측이 잘 되지 않은 이유를 불규칙한 이벤트 발생이라고 보고, non-linear growth과 불규칙 이벤트를 고려하는 모델인 facebook prophet을 사용했다.

facebook prophet는 growth, seasonality, holidays의 세가지 요소를 고려한다. 먼저 growth의 경우는 linear growth, non-liear growth를 고려한다. seasonality의 경우는 사용자들의 행동양식으로 주기적으로 나타나는 패턴을 나타내며, 푸리에 급수를 사용해 패턴의 근사치까지 찾아준다. 마지막으로 holidays는 주기성을 가지진 않지만 전체 추이에 큰 영향을 주는 이벤트를 분석해 준다.



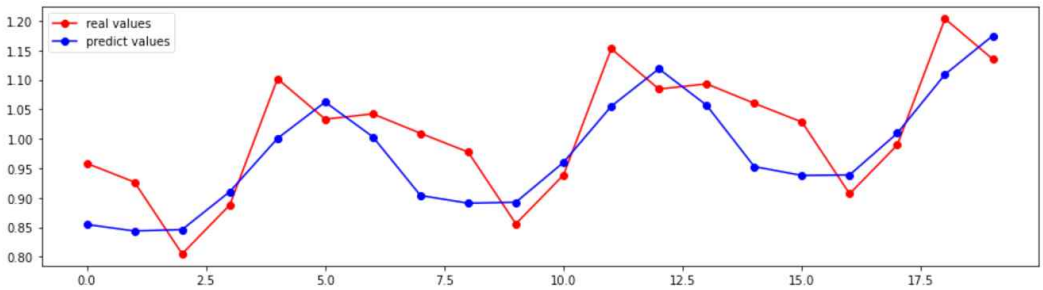
위의 결과는 facebook prophet 모델을 사용하여 확진자수를 예측한 결과를 시각화한 형태이다. 검은색 점은 실제 확진자를 의미하여, 파란색 선은 예측한 결과를 나타낸다. 비교적 ARIMA에 비해 이벤트 예측이 잘 되며, 빨간색 점선으로 change point를 그려본 결과, 포인트를 잘 잡아냄을 확인했다.

마지막 RNN 모델을 사용하여 예측한 결과는 아래와 같다. 먼저 train, test데이터를 만들기 위해 데이터를 섞지 않고 특정 시검을 기준으로 학습용, 검증용 데이터를 구분했다. min-max scaler를 사용하여 스케일링을 진행했으며, 모델에 적용이 가능한 데이터 형태로 가공했습니다. RNN 모델의 경우는 이전 시점의 결과를 다음 시점에 반영하므로 시계열 모델에 최적화되어 사용하게 되었다. 아래는 RNN을 사용하여 예측한 결과를 시각화한 형태이다.



```
Epoch 195/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0064
Epoch 196/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0064
Epoch 197/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0066
Epoch 198/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0064
Epoch 199/200
4/4 [==============================] - 0s 3ms/step - loss: 0.0061
Epoch 200/200
4/4 [==============================] - 0s 4ms/step - loss: 0.0062
```

학습 결과는 위와 같다. loss값은 0.0062로 학습이 잘 되었음을 확인할 수가 있다.



위의 결과는 RNN 모델을 사용하여 predict한 결과를 시각화한 형태이다.

마지막으로 세가지 모델을 사용한 결과, 학습 결과가 좋았던 Facebook prophet, RNN 모델을 앙상블 하여 예측 결과를 얻었다.