

입원일 수 예측 모델

이준석 | 2022.01.11

1. 데이터 합병 및 전처리

저는concept.csv파일을 제외한 모든 데이터 테이블 합병하여서 사용하였습니다.

그리고 합병할시에 visit\_occurrence을 index로 합병하였습니다. 그리고 duplicates를 사용하여서 중복되는 부분을 drop했습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 모니터, 화면, 플랫이(가) 표시된 사진

자동 생성된 설명

Visit\_occurrence 의 입원(9201)환자만 추출

텍스트, 모니터, 텔레비전, 화면이(가) 표시된 사진

자동 생성된 설명

Condition\_occurrence와 합병 결과

텍스트이(가) 표시된 사진

자동 생성된 설명입원일수계산

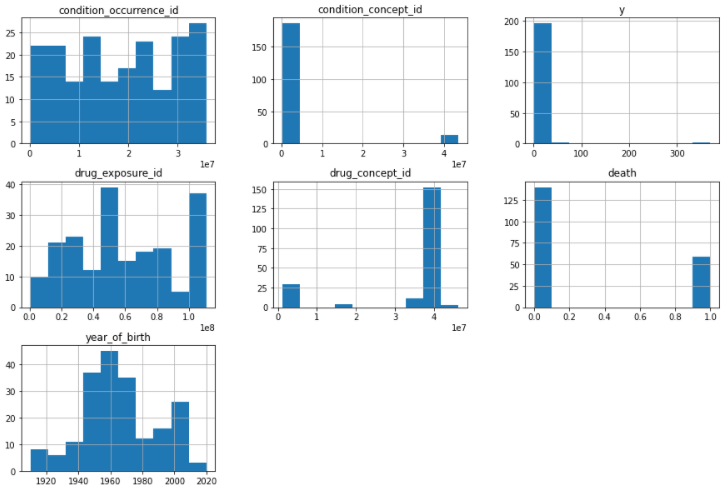
텍스트, 모니터, 실내, 검은색이(가) 표시된 사진

자동 생성된 설명

나머지 table들을 합병한후의 최종 데이터셋

테이블 합병 후 총 데이터 개수는 199개로 작은 데이터셋으로 구성했습니다. 데이터셋이 작은 이유는 유의미한 결과를 얻기 위해서 결측치가 있는 데이터들을 최대한 배제했기 때문입니다. 그래서 합병시에 최대한 inner join을 사용하였습니다. 아무리 많은 데이터가 있어도 dummy 데이터가 많으면 오히려 model 훈련에 방해가 될것이라고 생각했기 때문입니다.

2. 데이터 분석

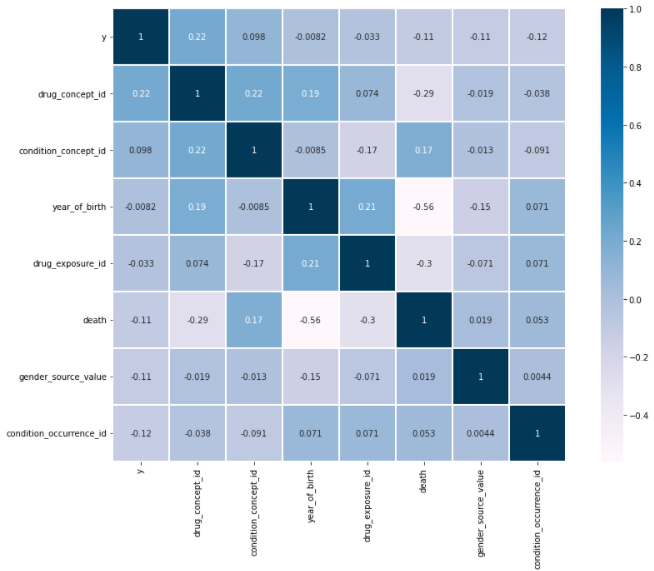


Hist 로 데이터의 분포를 확인해 보았습니다.

텍스트, 모니터, 화면, 검은색이(가) 표시된 사진

자동 생성된 설명

Outlier확인을 위해 describe를 이용했습니다.



Heatmap을 이용하여서 입원일 수와의 상관관계를 확인해본 결과 처방의약품(drug\_concept\_id), 진단명(condition\_concept\_id), 출생년도 순으로 중요한 Column인 것을 확인할 수 있었습니다. 그리고 오히려 훈련에 방해가 될것 같은 'ethnicity\_source\_value', 'condition\_occurrence\_id'는 drop했습니다.

3.데이터 전처리

데이터 전처리는 sklearn의 LabelEncoder와 death를 확인하기 위해서 Binary로 만들어주는 개인적인 함수를 사용하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명 binary

텍스트이(가) 표시된 사진

자동 생성된 설명LabelEncoder

텍스트, 모니터, 검은색, 화면이(가) 표시된 사진

자동 생성된 설명

최종 데이터셋

텍스트이(가) 표시된 사진

자동 생성된 설명

Train,Test split

텍스트이(가) 표시된 사진

자동 생성된 설명train\_dataset

텍스트이(가) 표시된 사진

자동 생성된 설명test\_dataset

4.검증 방법(Loss function)

Loss는 MAPE(Mean\_abolsute\_percentage\_error)를 사용하였습니다.

MAE나 MSE,RMSE등은 어느정도 성능인지 가시적으로 확인이 안되어서 MAPE Loss fuction을 사용하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명MAPE Loss function

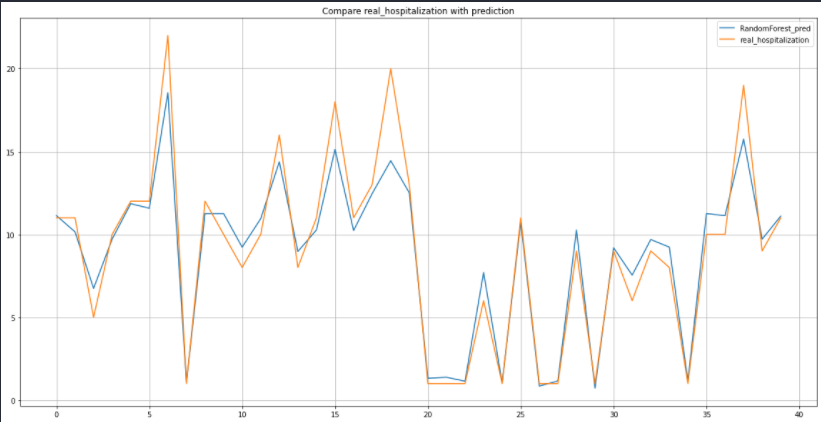
5. 모델 선택

텍스트이(가) 표시된 사진

자동 생성된 설명

처음엔 VotingRegressor를 사용하여서 결과를 내보았습니다.

모델은 Catboost,XGB,LGB,RandomForest를 사용해보았습니다.



Voting Regressor Result plotting

텍스트이(가) 표시된 사진

자동 생성된 설명 Result

MAPE 의 오차가 12.5% 정도의 나쁘지 않은 성능을 냈습니다.

하지만 성능을 좀더 끌어 올리기 위해서 개별의 모델들을 사용해본 결과 RandomForestRegressor가 가장 성능이 좋았습니다.

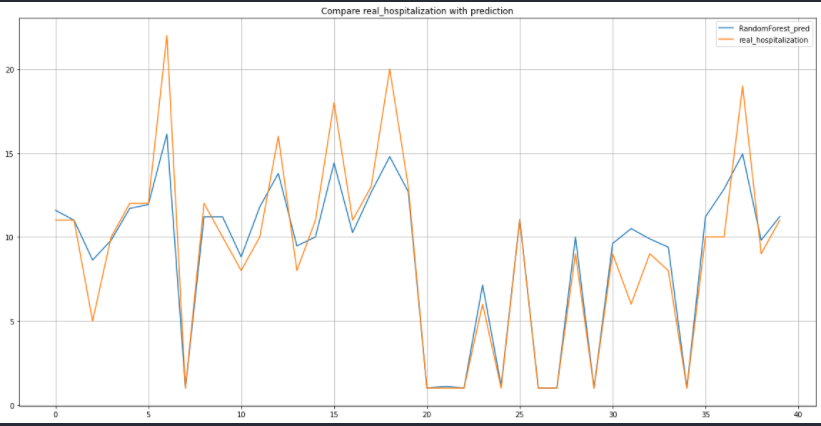
텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

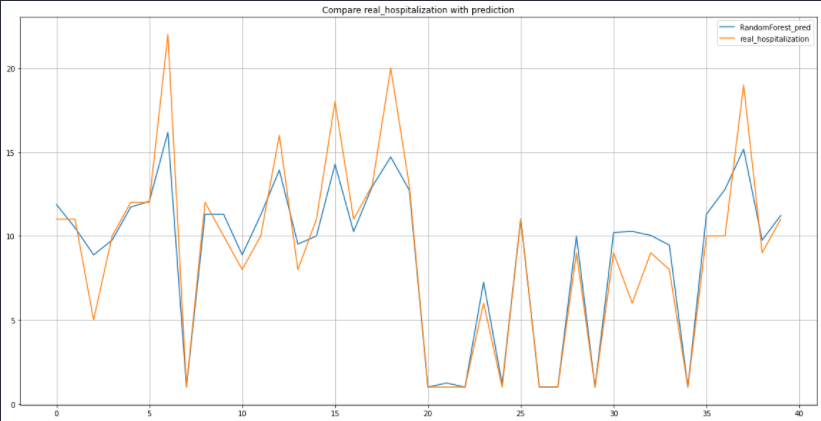
텍스트이(가) 표시된 사진

자동 생성된 설명

n\_estimator 와 n\_jobs를 파라미터로 적용하였으며, n\_estimator에 차이를 주어보았습니다.



n\_estimator:300 result



n\_estimator: 1000 result

텍스트이(가) 표시된 사진

자동 생성된 설명

KFold validation n\_split=10으로 결과를 내본 결과

n\_estimator:300 -> 11.9%

n\_estimator:100 -> 11.3%

n\_estimator가 1000일 때 Voting Regressor보다 1.2%정도 성능이 향상되었으며,

n\_estimator가 300일 때보다 0.6%정도의 성능이 향상되었습니다.

또한 개별로 LGBRegressor 를 적용해보았습니다.

LGBRegressor에서는 RandomizedSearchCV를 사용하여서 best parameter를 찾으려 했습니다.

텍스트, 화면, 실내이(가) 표시된 사진

자동 생성된 설명

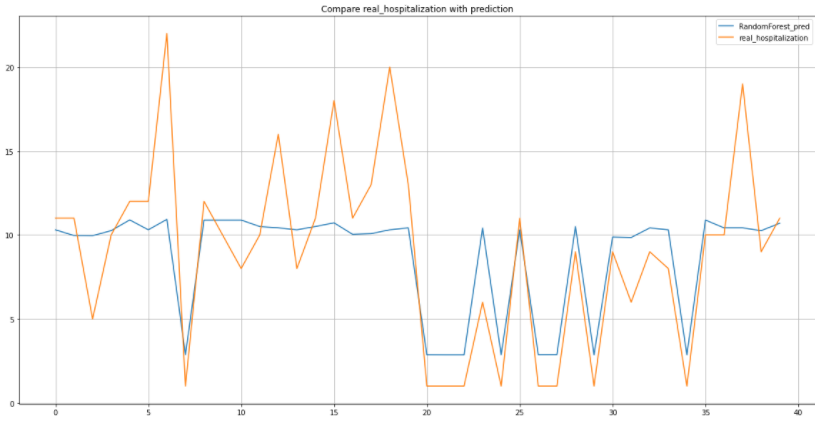
텍스트이(가) 표시된 사진

자동 생성된 설명

총 5번의 검증을 거쳐서 best\_param을 얻으려 했습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명best params



Result plotting



결과가 그다지 좋지는 않았습니다.

Best Model : RandomForestRegressor(n\_estimator:1000,n\_jobs=8)