**주제**

머신러닝을 이용하여 뇌졸중을 진단하는 모델 학습하기

**데이터 정제과정**

뇌졸증 데이터셋을 다운받아 stroke로 정의하고 파일을 읽어보면 다음과 같이 나온다

테이블이(가) 표시된 사진

자동 생성된 설명

각각 특성은 id, gender(성별), age(나이), hypertension(고혈압), heart\_disease(심장질환), ever\_married(결혼 여부), work\_type(직업 유형), residence\_type(거주 유형), bmi,

avg\_glucose\_level(평균 포도당수치)에 해당한다

**stroke.info()**를 보면 다음과 같다

텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명

뇌졸중 데이터셋에서 id는 뇌졸중과 전혀 연관이 없어 보이므로 drop을 이용하여 삭제하였다

그 후 gender, ever\_married, work\_type, Residence\_type,과 같은 범주형 데이터들을 ordinalEncoder를 이용하여 변환해주었다

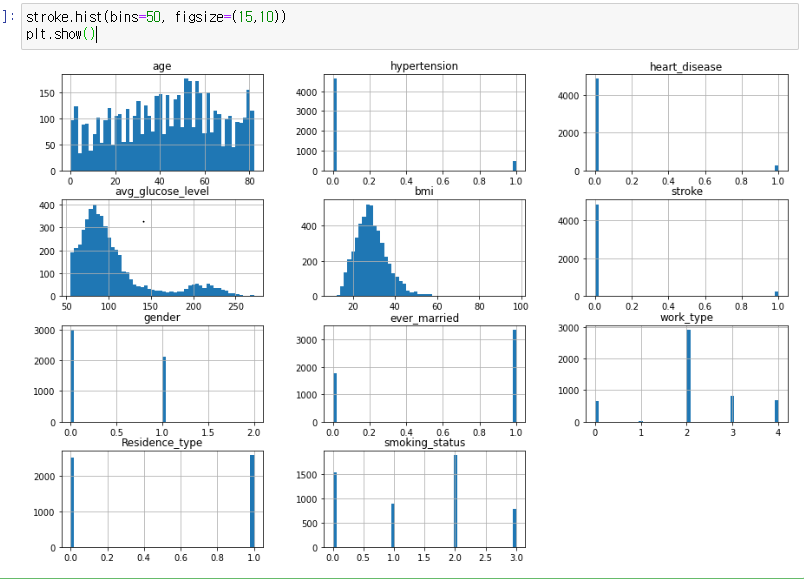
테이블이(가) 표시된 사진

자동 생성된 설명

**범주형 데이터를 변환 후 데이터셋을 보니 NaN값이 보인다**

**stroke.isnull().sum()**을 이용하여 NaN값을 확인해보니 다른특성에는 NaN값이 안보이지만 bmi특성에는 201개의 NaN값이 있었고 데이터양이 그렇게 많지 않다고 생각해서 다른값으로 교체하는것이 아니라 drop을 이용해 NaN값을 제거하였다

그 후 **stroke.hist()와 stroke[].value\_count()**를 이용하여 데이터의 전체적인 분포와 갯수를 보면 다음과 같다



테이블이(가) 표시된 사진

자동 생성된 설명

gender부분에 2값이 하나 있는 것을 볼 수 있는데 해당 값은 하나밖에 없기 때문에 혼란을 줄 것 같아 drop을 이용하여 제거하였다

**corr()**을 이용해서 상관관계를 보면 다음과 같다

테이블이(가) 표시된 사진

자동 생성된 설명

**모델 학습과정**

우선 sklearn의 train\_test\_split을 이용하여 테스트 셋과 트레인셋을 분리한 후 모델에 넣어 학습을 진행했다 학습은 SVM, DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier을 이용하였다

**성능비교**

학습한 모델들을 accuracy\_score을 이용하여 정확도를 측정해 보면 다음과 같다

텍스트이(가) 표시된 사진

자동 생성된 설명

생각보다 높은 정확도를 보이고 있다 이것만으로는 정확한 성능 측정을 할 수 없으니 아래와같이 다른 함수를 만들어 성능을 비교했다

텍스트이(가) 표시된 사진

자동 생성된 설명

3가지를 비교해보았는데 rmse값 cross\_val값 오차행렬값을 비교했다

텍스트이(가) 표시된 사진

자동 생성된 설명

상단의 사진은 rmse값으로 실제값과 엄청난 차이를 보이지 않는 것을 볼 수 있고

텍스트이(가) 표시된 사진

자동 생성된 설명

Cross\_val값은 전체적으로 큰 차이 없이 비슷하게 학습되고있다는 것을 볼 수 있다

오차행렬은 아래와 같다

테이블이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

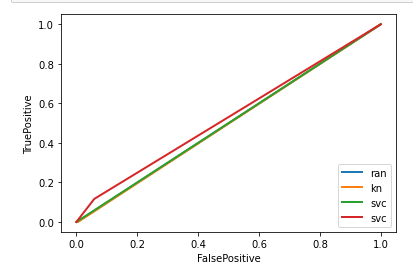
자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

해당 오차행렬의 ROC곡선을 그리면 다음과 같다



오차행렬과 roc곡선이 이렇게 낮은 값을 가지는 이유는 데이터의 불균형 때문이라고 생각한다 뇌졸중 여부의 데이터를 보면 뇌졸중이 아닌 사람이 4699명 뇌졸중인 사람이 209명 y\_test에는 뇌졸중인 사람 43명 아닌 사람이 939명으로 뇌졸중인 사람의 데이터가 많이 적다 때문에 정확도가 높게 나온 것은 학습한 모델이 테스트 셋의 대부분을 0이라 예측했기 때문에 정확도가 높게 나온 것으로 보이고 4개의 모델을 합쳐 뇌졸중을 찾아낸 것이 4개밖에 없다 그래서 오차행렬값과 roc곡선이 낮은 값을 가지고 있다고 추측하고있다

이를 해결하기 위해서 인터넷에 검색해보니 오버 샘플링이라는 데이터를 생성하여 데이터의 불균형을 맞춰주는 기법이 있어 적용한후 다시 성능비교를 해보았다

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 오버샘플링을 하여 모델을 학습한 후 성능을 비교해 봤을 때

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명

전체적인 정확도는 낮아졌지만 rmse값과 오차행렬 값은 전체적으로 올라간 것을 확인 할 수 있다 마지막으로 가장 좋은 성능을 보이고 있는 randomforest 모델을 그리드탐색 한 후 다시 학습하여 비교해보면

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 같은 결과로 파라미터의 변화 폭을 넓게 지정하지 않아서 그런지 이전과 그렇게 큰 차이는 없지만 정확도와 오차행렬이 높게나오고 rmse가 낮게나오는 것을 보아 잘 학습된것으로 보인다