표 형식 데이터에 대한 딥러닝을 활용한 암 발병 예측

팀 명 : esc

지도교수 : 송길태

부산대학교 전기컴퓨터공학부 정보컴퓨터전공

201824604 최지광

201924493 서진욱

201924496 송민재

목차

1. 수정사항

1.1 기존 연구 목표 및 제약 사항

기존 연구에서 '표 형식 데이터에 대한 딥러닝을 활용한 반려견 질병 예측 및 예방'을 주제로 프로젝트 및 연구를 진행하려 하였으나 사용할 데이터에 대한 한계가 있었다. 반려견의 feature 값과 반려견의 의학 정보들에 대한 연관성을 찾기 어려웠으며 이 때문에 인해 반려견 질병을 예측의 정확성이 매우 낮았다(LightGBM 기준 recall score 0.3448).

의학정보의 한 예시로는 C-반응성 단백질 (CRP)이라는 개의 매우 민감하고 특이한 전신 염증 표지 자가 있었다. 하지만 조사 결과 해당 단백질은 염증 발생의 결과로 수치가 높아지는 것일 뿐 높은 단백질 수치가 염증 원인이 되지는 않았다. 이 때문에 반려견의 feature와 CRP 사이의 연관성을 찾기 어려웠다.

그 외에도 CRP 수치 증가에는 폐렴, 췌장염 등 다양한 질병이 영향을 미친다는 점, CRP 수치가 정상 범위에서 벗어나는 데이터 수가 적다는 점 등 다양한 이유 때문에 주제 변경을 선택하였다.

1.2 연구 수정사항

가지고 있는 데이터로 반려견 질병을 예측하기엔 한계가 있다고 판단해 '표 형식 데이터에 대한 딥러닝을 활용한 사람의 암 예측'을프로젝트의 주제로 변경하였다. 사람의 Age, Gender, BMI, Smoking, CancerHistory, AlcoholIntake, PhysicalActivity와 같은 feature들을 통해 암의 발병을 예측하는 딥러닝 모델을 개발하고 성능을 향상시켜 위의 7가지 입력값을 받았을 때 암을 예측할 수 있도록 할 것이다. 각 입력 데이터의 특성은 아래와 같다.

Age : 20 ~ 80까지의 범위의 나이로 정수값으로 표현
Gender : 남자는 0 , 여자는 1로 표현
BMI : 체질량 지수로 15 ~ 40까지 연속된 값으로 표현
Smoking : 비흡연자이면 0, 흡연자이면 1로 표현

PhysicalActivity : 일주일에 몇 시간동안 신체적인 활동을 하는지 0~10까지 연속된 값으로 표현

AlcoholIntake : 일주일에 소비하는 알코올 unit의 수를 0~5까지 연속된 값으로 표현

CancerHistory : 암에 대한 개인 병력이 없으면 0, 있으면 1로 표현

Diagnosis : 암이 아니면 0, 암이 맞으면 1로 표현

```
data.head()
```

	Age	Gender	BMI	Smoking	GeneticRisk	PhysicalActivity	AlcoholIntake	CancerHistory	Diagnosis
0	58	1	16.085313	0	1	8.146251	4.148219	1	1
1	71	0	30.828784	0	1	9.361630	3.519683	0	0
2	48	1	38.785084	0	2	5.135179	4.728368	0	1
3	34	0	30.040296	0	0	9.502792	2.044636	0	0
4	62	1	35.479721	0	0	5.356890	3.309849	0	1

암의 발병 예측을 위해 사용한 데이터 예시

LightGBM / Tabnet / MLP / Ensemble(MLP, Random Forest, XGBoost) / SAINT / Random Forest 이 6가지 모델을 사용해 암을 예측하고자 하였으며 모델들에 대한 성능 평가에는 Recall score에 중점을 두었다.

2. 설계 상세화 및 변경 내역

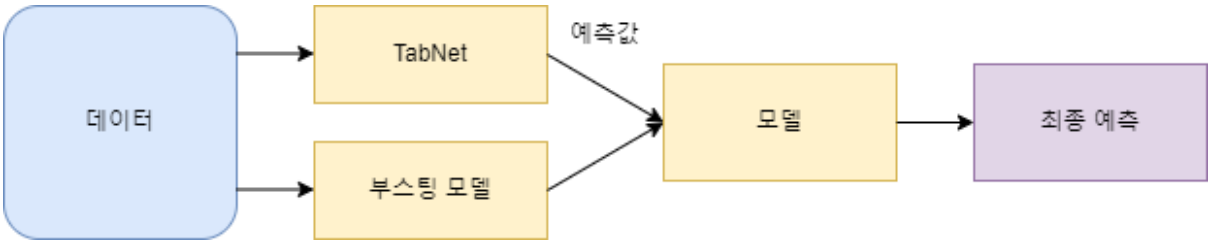
2.1 모델 설계

본 과제에서는 TabNet과 부스팅 모델을 결합한 앙상블 모델을 설계해 예측 성능을 향상하고자 한다. TabNet의 특성화 부스팅 모델의 장점을 결합함으로 안정적인 예측 결과를 도출할 수 있을 것이다.

- 데이터 전처리

TabNet은 정규화 없이도 작동할 수 있지만, 부스팅 모델의 성능을 최적화하기 위해 전처리를 진행함으로써 안정적인 학습을 할 수 있도록 하였다.

- 모델 구조



딥러닝 모델 구상화

TabNet과 부스팅 모델 간의 앙상블 기법을 통해 예측 성능을 개선할 것이다. 앙상블 방식으론 스태킹을 고려해 설계를 진행한다. 부스팅 모델은 표 형식 데이터에 좋은 성능을 보이던 트리 기반 알고리즘을 선택했다. 모델 선택 전 MLP, Random Forest, XGBoost를 앙상블 기법을 사용해 예측 성능을 향상한 결과가 있기에 이와 같은 방법을 사용하기로 했다.

- 손실 함수 및 최적화

암 유무를 예측하는 이진 분류 문제이기 때문에 Binary Cross-Entropy를 사용해 모델의 오차를 계산할 것이다.

- 성능 평가 지표

Accuracy, Precision, Recall, F1-score등을 기준으로 하고, Recall에 중점을 둔다. 또한 ROC커브를 사용해 모델의 종합적인 예측 성능을 평가할 것이다.

3. 보고 시점까지의 과제 수행 내용 및 중간결과

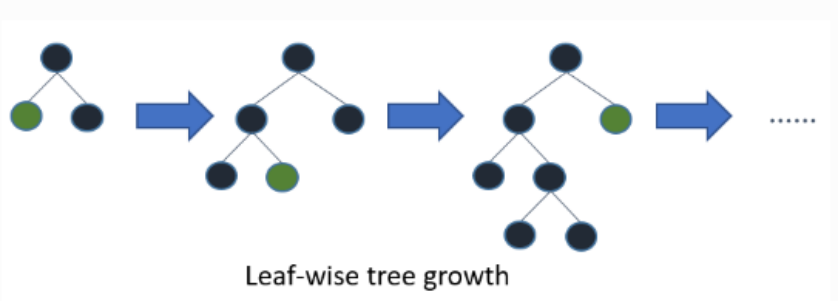
3.1 기존 모델과의 성능 결과 확인 및 비교

표 형식 데이터를 다루기 적합한 기존 머신러닝 모델과 딥러닝 모델을 비교해 성능을 비교했다.

- LightGBM
    - f1-score : 0.8860
    - Recall : 0.8834

LightGBM은 트리 기반의 학습 알고리즘으로 대용량 데이터에 대해 메모리는 적게 사용하지만 빠르고 정확한 학습이 가능하므로 사용된다. 대부분의 트리 기반 알고리즘은 트리가 수평적으로 확장되지만 LightGBM은 수직으로 확장된다는 차별점이 있다.

일반적인 트리 확장



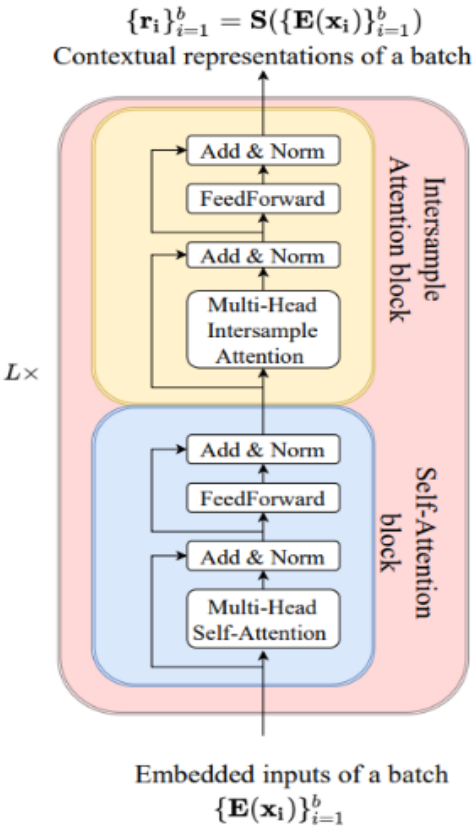
LightGBM의 트리 확장

LightGBM의 대표적인 parameter로는 트리의 형태를 결정하는 num_leaves, max_depth가 있으며 이중/다중 분류를 결정하는 objective 등이 있다. 결과 측정에서 사용한 코드는 아래와 같다.

```
gbtree = lgb.LGBMClassifier(learning_rate=0.05, n_estimators=116, num_leaves= 9,
                            max_depth=3, subsample=0.74, colsample_bytree=0.69,
                            objective = 'binary', is_unbalance = 'true', random_state=21)
```

LightGBM Code

- SAINT (Self-Attention and Intersample Attention Transformer)
    - f1-score : 0.8876
    - Recall : 0.8902



SAINT 모델 구조

SAINT 모델은 Transformer의 Encoder 부분을 활용하고 Self-Attention 및 Intersample Attention Block이 L번 반복되는 구조이다. MSA(Multi-Head Self-Attention)와 FF(FeedForward)/GELU , 두 계층으로 구성되며 Add & Norm 계층을 통해 skip connection과 층정규화가 적용된다. Intersample Attention Block은 Self-Attention Block과 거의 같으며 MSA가 MISA(Multi-Head Intersample Attention)으로 대체된다. Self-Attention은 한 데이터 내의 features 간

관련성을 파악하는 것이고 Intersample Attention은 입력된 배치 사이즈 내 데이터 간 연관성을 파악하는 방법이다.
모델을 컴파일 할때 활성화함수는 GELU 함수를 사용하였고 손실함수로는 크로스 엔트로피 오차(Cross Entropy)를 사용하였으며 옵티마이저로는 Adam을 선택하였다.

```
input_dim = x_train.shape[1]
hidden_dim = 64
output_dim = len(np.unique(y_train))

saint_model = SAINT(input_dim, hidden_dim, output_dim)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(saint_model.parameters(), lr=0.001)
```

SAINT Code

전체 훈련 데이터셋에서 일정한 묶음으로 나누어 처리할 수 있는 배치와 훈련횟수인 epoch 선택도 중요한데 배치 사이즈는 32로 하였으며 epoch수는 120으로 모델을 학습시켰고 test_size=0.1로 테스트 데이터셋을 전체 데이터셋의 10%로 설정했을 때 성능이 좋게 나왔다.

- RandomForest
    - f1-score : 0.9047
    - Recall : 0.8995

여러 개의 결정트리(Decision Tree)를 조합하여 더 강력한 분류 모델을 구축하는 방법으로 과적합(Overfitting) 을 줄이고 예측 성능을 향상하는데 효과적이다. Scikit-learn (sklearn)에서 RandomForest 를 구현한 다양한 클래스와 함수를 제공하며 필요한 라이브러리를 import한다. test_size=0.2로 테스트 데이터 세트를 전체 데이터 세트의 20%로 설정해주었을 때 가장 성능이 좋았다.
불순도 측정방식인 criterion은 엔트로피로 설정하였으며 ccp_alpha는 0.001로 해 각 가지치기 과정이 효과가 있도록 하였다.

```
random_forest_clf = RandomForestClassifier(ccp_alpha = 0.001, criterion = 'entropy', random_state = 123)

random_forest_clf.fit(x_train, y_train)
```

RandomForest Code

Scikit-learn의 'RandomForestClassifier' 클래스를 사용해 모델을 초기화하고 fit 메서드를 사용해 모델을 학습시켰다. 그리고 predict 메서드와 테스트 데이터를 사용해 예측을 수행하고 그 후 성능을 평가하였다.

- MLP
  - f1-score : 0.8559
  - Recall : 0.8655

```
model = Sequential([
    Dense(64, input_shape=(X_train_final.shape[1],), activation='relu', kernel_regularizer=l2(0.001)),
    Dropout(0.5),
    Dense(32, activation='relu', kernel_regularizer=l2(0.001)),
    Dropout(0.5),
    Dense(1, activation='sigmoid')  # 이진 분류를 위한 출력층
])
```

MLP Code

입력층 및 첫 번째 은닉층은 64개의 뉴런으로 구성되고, ReLU를 사용하고 L2 정규화를 적용했다. 또한, 과적합을 방지하기 위해 드롭아웃 비율을 0.5로 설정하여 일부 뉴런을 무작위로 학습에서 제외했다.

두 번째 은닉층은 32개의 뉴런을 가지고 있으며, 첫 번째 은닉층과 같게 ReLU 활성화 함수와 L2 정규화, 드롭아웃을 적용했다.

출력층은 이진 분류 문제를 처리하기 위해 1개의 뉴런과 Sigmoid 활성화 함수를 사용했다.

Adam 옵티마이저를 사용, 학습률을 0.001로 설정했다. 손실함수는 binary_crossentropy를 사용했다. 학습데이터로 20 에포크동안 모델을 학습시키고, 한번에 32개의 샘플을 처리했다.

- Ensemble(MLP, Random Forest, XGBoost)
    - f1-score : 0.8918
    - Recall : 0.8655

```
# 메타 모델을 위한 데이터 생성
stacked_predictions = np.column_stack((rf_pred, xgb_pred, nn_pred))

# 메타 모델 정의
meta_model = LogisticRegression()
meta_model.fit(stacked_predictions, y_test)
```

Ensemble Code

Stacking 기법을 사용해 각 기본 모델의 예측 결과를 결합하고, 메타 모델로 학습하여 최종 예측을 수행했다. MLP 모델은 기존에 만들었던 모델을 사용했다. 각 모델의 예측값을 메타 모델에 입력할 데이터로 사용한다. 메타 모델로는 Logistic Regression을 사용했다. 메타 모델에 만들어진 데이터와 test 셋을 사용해 최종예측을 수행했다.

- Tabnet
    - f1-score : 0.9003
    - Recall : 0.8976

Tabnet은 4가지 특성을 가지고 있는 딥러닝 모델이다.

1. Gradient descent-based optimization으로 학습을 진행하여 전처리 없이 입력에서 결과를 이끌어 낸다.
2. Sequential attention을 사용하여 순차적인 학습을 진행하기 때문에 더 높은 성능의 학습이 가능하다.
3. Tabular Data에 대한 학습에서 좋은 성능을 보인다.
4. Label이 없는 데이터로 학습을 진행하는 Self-supervised learning을 사용하여 높은 성능을 보인다.

```
tb_cls = TabNetClassifier(optimizer_fn=torch.optim.Adam,
                          optimizer_params=dict(lr=1e-3),
                          scheduler_params={"step_size":10, "gamma":0.9},
                          scheduler_fn=torch.optim.lr_scheduler.StepLR,
                          mask_type='entmax' # "sparsemax"
                          )
```

Tabnet Code

3.2 피드백에 대한 결과

- 모델 성능 평가 지표

모델 성능 평가 지표로 재현율(Recall)을 추가 활용하기로 했다. 높은 재현율은 실제 암 환자를 모델이 얼마나 잘 식별하는지를 나타낼 것이다. 그러나 재현율만을 높이면 암이 아닌 사람을 암으로 예측하는 사례가 증가할 수 있기 때문에 기존 성능 평가 지표이던 f1-score를 사용해 정밀도와 재현율의 균형을 확인할 것이다.

- 입력 데이터 선별

입력 데이터 중 일반인이 입력할 수 없는 경우 학습에서 제외하였고 어려운 경우 표, 그림 등을 통해 입력할 수 있도록 변경할 것이다. 입력 데이터 중 암에 대한 유전적 위험을 의미하는 GeneticRisk라는 feature가 있다. 데이터의 설명에 의하면 “위험도가 낮으면 0, 보통이면 1, 높으면 2”로 데이터가 결정된다. 하지만 위험도의 낮고 높음을 결정하는 기준이 설명되어 있지 않다. GeneticRisk를 넣은 딥러닝이 제외한 딥러닝보다 f1 score 기준 0.06 높은 점수가 나오지만, 일반인 기준 입력이 어렵다 판단되어 제외하기로 결정했다(10회 학습 기준 포함한 f1 score은 평균 0.8605, 제외한 f1 score는 평균 0.7936을 기록하였다).
또한 일주일 동안 마시는 알코올 유닛을 의미하는 AlcoholIntake라는 feature가 있다. 알코올 유닛이라는 단위는 일반인들에게 생소한 단위로 입력 단계에서 어려움이 존재할 수도 있다. 우리는 이를 해결하기 위해 다양한 주류들의 알코올 유닛을 조사하였고 이후 시각화 과정에서 그림, 표, 또는

기타 형식을 통해 입력을 쉽게 도울 것이다. 아래는 이후 시각화 과정에서 그림을 사용한다면 입력을 도울 예시이다.

AlcoholIntake 예시

4. 갱신된 과제 추진 계획

4.1 딥러닝 모델 성능 향상

딥러닝 모델 성능 향상을 위해 다음과 같은 방식을 시도해 볼 것이다.

- epoch, batch_size 등과 같은 Parameter 수정 및 테스트 시도
- 다양한 부스팅 모델과의 앙상블 방식 시도
- category 형식의 feature만 모아서 tabnet의 parameter에 적용 시도
- GeneticRisk에 대한 정보를 획득 시도 (GeneticRisk와 같이 학습할 시 더 높은 정확도를 보임)
- 그 외 다양한 딥러닝 모델 향상 방식 찾아보기

4.2 딥러닝 모델 시각화

입력한 값들을 기반으로 암을 예측하여 출력하고 암 예방법 설명, 근처 병원 위치 확인 등과 같은 부가적인 서비스를 제공하는 웹을 제작할 계획이다. 웹의 요구사항은 아래와 같다.

- 나이는 20 - 80세 사이를 직접 입력하는 input type = “number”을 사용한다
- 성별, 흡연 유무, 암 걸린 기록은 그룹 중 하나만 선택할 수 있는 input type = “radio”를 사용한다
- BMI, 활동량, 알코올 섭취량은 주어진 범위 내의 값을 입력하는 input type = “text”를 사용한다
- 어떤 값을 입력해야 하며 어떤 단위, 조건을 사용하는지 반드시 명시한다
- 주어진 입력 조건에 일치하지 않는 경우 경고문을 출력하고 다시 입력을 할 수 있도록 진행한다
- BMI는 사이트 안에서 직접 계산할 수 있는 시스템을 구현한다
- 알코올 섭취량을 사이트 안에서 직접 계산할 수 있는 시스템을 구현한다
- 암 예방 방법 설명, 현재 위치 근처 병원 표시 등과 같은 부가적인 서비스는 필요할 경우 구현한다

시각화를 완료하였을 때 대략적인 사이트의 모습은 아래와 같다.

입력 화면 예시

출력 화면 예시

4.3 이후 일정

8월		9월				10월			
3	4	1	2	3	4	1	2	3	4
딥러닝 모델 선정 및 성능 향상									
		시각화							
						최종 보고서 작성 및 발표 준비			

5. 구성원별 진척도

5.1 구성원별 진척도 현황

이름	역할
최지광	- MLP 모델 학습 및 성능 평가 - Ensemble(MLP, Random Forest, XGBoost) 모델 학습 및 성능 평가
서진욱	- SAINT 모델 학습 및 성능 평가 - Random Forest 모델 학습 및 성능 평가
송민재	- TabNet 모델 학습 및 성능 평가 - LightGBM 모델 학습 및 성능 평가
공통	- 보고서 작성 - 주제 수정 및 데이터 탐색

6. 참고 문헌

1. Alcohol units - what's safe?
   https://safercornwall.co.uk/alcohol-units-whats-safe/

2. Cancer Prediction Dataset
   https://www.kaggle.com/datasets/rabieelkharoua/cancer-prediction-dataset/data

3. LightGBM documentation (2023)
   https://lightgbm.readthedocs.io/en/stable/Features.html

4. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training (2021) [2106.01342]

SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training (arxiv.org)

5. TabNet: Attentive Interpretable Tabular Learning (2019)

https://arxiv.org/pdf/1908.07442