Report

C&I Lab Loneliness 예측모델 프로젝트 보고서

작성자 : 양윤성

2024.06.25.

● Contents

1. Introduction

본 연구에서는 여러 가지 ML model을 사용해서 정신과 환자의 Loneliness 상태를 예측하는 모델을 개발하고, 각 모델의 성능을 비교 및 분석하고자 한다. 이를 위해 다양한 알고리즘을 적용하여 Loneliness를 라벨링한 데이터셋을 바탕으로 예측 모델을 학습시킨다. 라벨이 1일 경우 Loneliness로 판단하고 이에 대한 precision, recall, f1 score, f1 macro score을 중심으로 모델의 정확도를 평가할 것이다. 이번 연구로 Loneliness 예측에 가장 효과적인 모델을 식별하고 이를 통해 정신과 환자의 치료 및 관리에 있어 유용한 정보를 제공하는 것을 기대한다.

2. Related Work

본 연구에서는 6가지 모델을 사용하여 Loneliness를 예측하고 성능을 비교하는데, 아래에서 각 모델의 발전 과정과 기존 연구에서의 활용 사례를 간단히 살펴보고자 한다.

1) Naive Bayes Classifier (NBC)

NBC(Naive Bayes Classifier)는 Bayes 정리에 기반한 확률적 분류 알고리즘으로, 각 feature가 독립적이라고 가정하고 계산한다. 이 모델은 간단하면서도 특정 문제에 대해 효율적인 성능을 보이는데, 주어진 데이터가 특정 클래스에 속할 확률을 계산하고, 이 확률이 가장 높은 클래스로 분류한다. Bayes 정리에 따라 posterior probability를 계산하고 각 feature의 조건부 확률을 곱한다.

2) Support Vector Machine (RBF SVM)

SVM(Support Vector Machine)은 2개의 클래스 사이에 존재하는 margin을 최대화하는 평면을 찾아 두 클래스를 확실히 분류할 수 있도록 하는 알고리즘이다. 기존 NBC에 비해 두 클래스를 분리하는 margin을 최대화하여 분류하다 보니 overfitting이 방지되고 일반화 성능이 뛰어나다. 또한 SVM에 RBF 커널 함수를 사용하여 비선형적이고 높은 차원의 데이터를 보다 효과적으로 처리할 것이다.

3) ElasticNet Model

ElasticNet Model은 L1 정규화(Lasso) 항으로 중요한 변수들만 선택하고, L2 정규화(Ridge) 항으로 overfitting을 방지하고 모델의 복잡성을 줄이는 선형 회귀 모델이다. 이렇게 결합하여 변수 선택과 정규화를 동시에 수행함으로써 기존 Lasso의 단점을 보완한다. 변수 간 높은 상관관계가 있는 경우에 효과적이고 더 안정적이고 신뢰성 있는 예측을 제공한다.

4) Feed-forward Neural Network (FFN)

FFN(Feed-forward Neural Network)는 input, hidden, output layer로 구성된 가장 기본적인 형태의 신경망이다. 입력 데이터가 각 layer를 거치며 가중합과 활성화 함수를 통해 처리되고 output layer에서 최종 예측값을 도출한다. 이때 역전파 알고리즘을 통해 가주치를 조정하여 학습한다.

5) Bidirectional Long Short-Term Memory (Bi-LSTM)

기존 LSTM은 memory cell을 통해 긴 시퀀스 데이터도 활용하여 학습 가능한 방법이었는데, 여기에 양방향성을 추가한 것이 Bi-LSTM(Bidirectional Long Short-Term Memory)이다. 순방향과 역방향으로 정보를 처리하다 보니 과거와 미래 정보를 모두 활용하여 예측 성능을 향상시킬 수 있고 시퀀스 데이터에 대한 패턴도 학습 가능하다.

6) XGBoost

XGBoost는 gradient boosting 알고리즘을 기반으로 한 강력한 ML 라이브러리로 6가지 모델 중 가장 빠르고 성능이 뛰어날 것으로 예상한다. 여러 개의 decision tree를 순차적으로 학습시켜 각 단계에서 이전 모델이 만든 error를 보정하며 예측 성능을 향상시킨다. 또한 병렬 처리와 overfitting 방지를 위한 정규화 기법도 포함되어 있다.

3. Experiments

1) Datasets

테스트에는 2가지 데이터 셋을 사용한다. raw_data1.csv는 환자별로 설문조사 결과(check_val)와 받은 서비스 내역(service) column이 각 환자별로 존재한다. 환자별로 20~40번정도의 설문조사와 받은 서비스 내역을 조사했고 이 데이터셋으로 feature를 추출한다. raw_data2.csv는 병원측에서 판단하기에 해당 환자가 Loneliness인지 아닌지를 작성해놓은 데이터로 0이면 정상, 1이면 Loneliness이다. 환자별로 시계열적으로 Loneliness인지 아닌지를 체크하였는데, 환자의 처음 상태와 마지막 상태만을 고려하여 둘다 Loneliness면 최종적으로 Loneliness 환자로 간주한다.

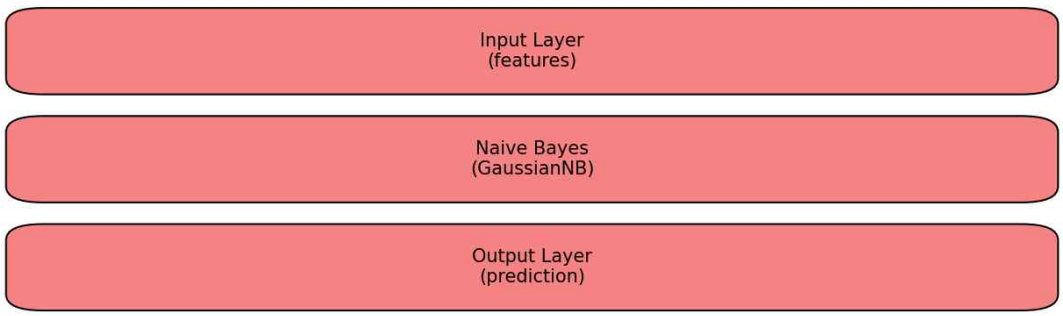
2) Data Pre-processing

불러온 두 데이터셋을 날짜 기준으로 정렬하고, raw_data1.csv로 환자의 ID를 추출한다. 이후 모델 학습에 사용할 입력 데이터(np_x)에 환자별 설문조사 결과와 서비스 내역들을 매핑하여 이후 feature로 사용한다. 출력 데이터(np_y) 배열에는 raw_data2.csv 데이터를 매핑한 y_tmp에서 위에 말한것과 동일하게 환자의 처음 상태와 마지막 상태만을 고려하기 위해 배열의 첫번째 값과 마지막 값을 곱하여 np_y에 저장한다. 이를 통해 둘다 1이여야만 Loneliness 환자로 저장한다.

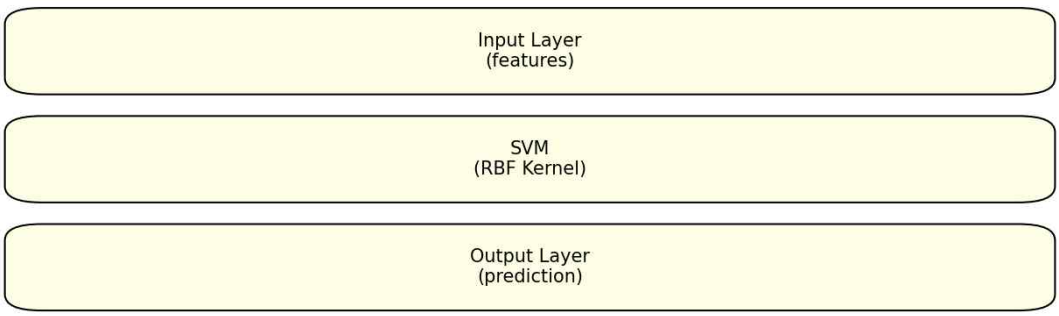
3) Model Architecture

공통적으로 모든 모델의 batch size는 1024, epoch(n_estimator)은 1000, test size는 0.2, 난수는 42로 고정하며, Loneliness의 경우 라벨이 1인 데이터가 훨씬 많아 SMOTE를 이용하며 라벨이 0인 데이터의 수를 늘려 맞춰주는 오버샘플링을 적용하여 불균형 문제를 해결했다. 그리고 np_x를 시간축으로 평균내어 (1, self.features_nums) 형태의 배열로 바꾸어 학습을 진행하였다. 모델 별 구조 및 특이 사항은 다음과 같다.

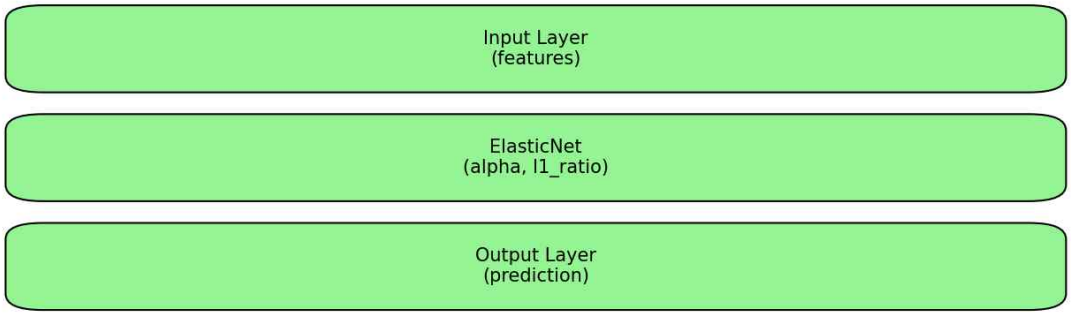
① Naive Bayes Classifier



② RBF SVM

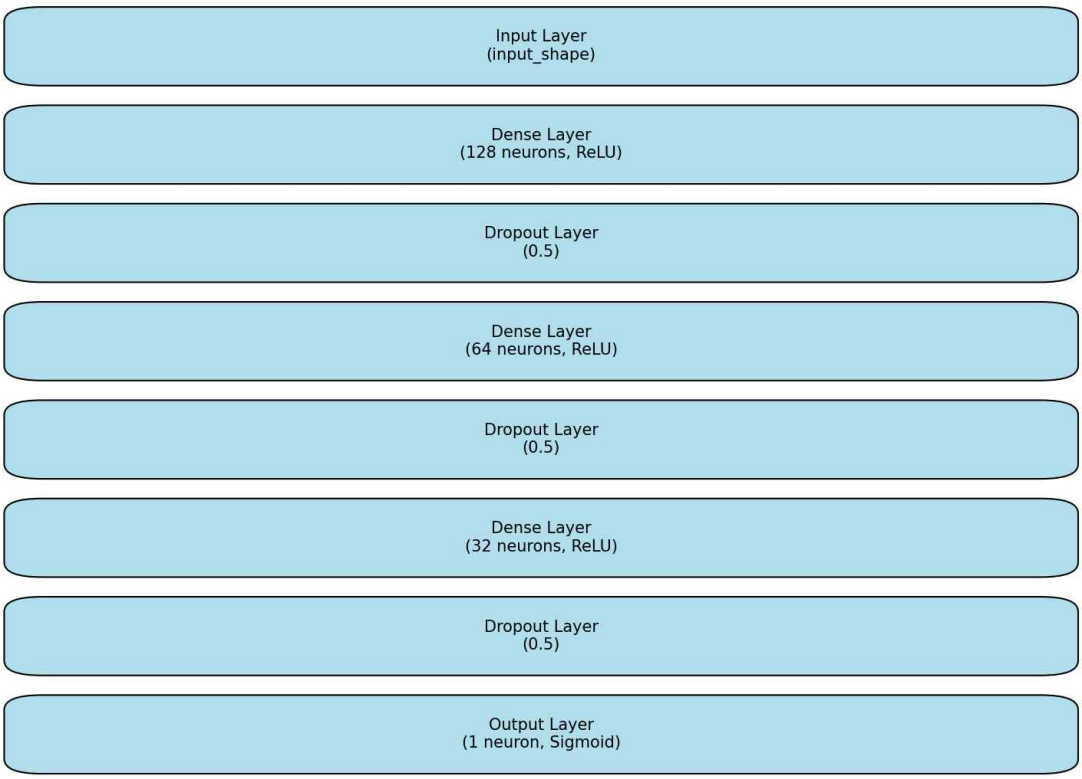


③ ElasticNet Model



크게 alpha와 l1_ratio 파라메터가 중요한데, alpha는 클수록 정규화의 강도가 강해져 overffiting을 방지할 수 있지만, 과도하게 크면 underfitting이 될 수 있다. 또한 l1_ratio는 L1과 L2의 비율을 조절하는 파라메터로 1에 가까울수록 L1이 더 많이 적용된다. 이 두 파라메터의 trade-off가 중요하기 때문에 gridsearchCV로 하이퍼파라메터 튜닝을 진행하여 최적의 파라메터 조합을 찾아 학습시켰다. alpha는 [0.1, 1, 10], l1_ratio는 [0.1, 0.5, 0.9]로 시도했고 최적의 쌍은 각각 0.1, 0.1이었다.

④ Feed-forward Neural Network

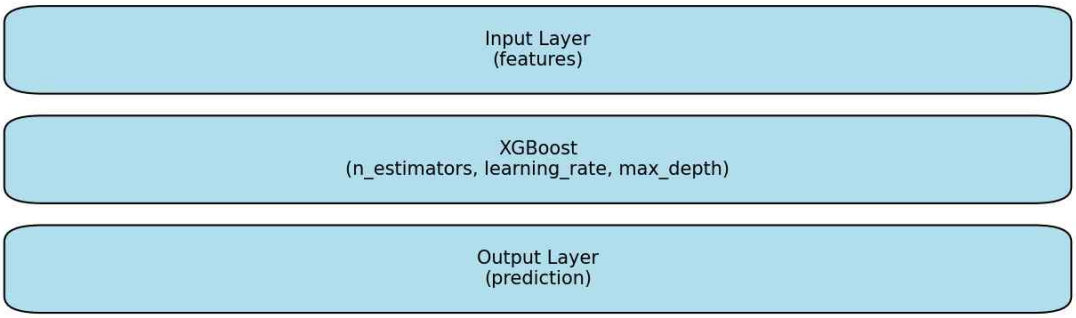


- 공통 : 드롭아웃 비율 0.5
- 첫번째 Layer : Dense Layer, 사이즈 128, ReLU
- 두번째 Layer : Dense Layer, 사이즈 64, ReLU
- 세번째 Layer : Dense Layer, 사이즈 32, ReLU
- 출력 Layer : Dense Layer, 사이즈 1, Sigmoid
- Adam optimizer 사용, 기본 학습률 0.001

⑤ Bi-LSTM

- 공통 : 드롭아웃 비율 0.5
- 첫번째 Layer : Bidirectional-LSTM Layer, 사이즈 64
- 두번째 Layer : Bidirectional-LSTM Layer, 사이즈 32
- 출력 Layer : Dense Layer, 사이즈 1, Sigmoid
- Adam optimizer 사용, 기본 학습률 0.001

⑥ XGBoost



ElasticNet Model과 같은 이유로 learning_rate와 max_depth 파라메터로 성능이 결정되고 둘의 trade-off가 중요하다. learning_rate는 너무 작으면 학습이 오래 걸릴 수 있고 너무 크면 모델이 overffiting될 수 있다. max_depth는 트리의 최대 깊이를 제한하여 모델의 복잡도를 조절하는데 깊이가 길수록 overffiting 가능성이 커지고 얕을수록 underffiting 가능성이 커진다. 그래서 동일하게 gridsearchCV로 하이퍼파라메터 튜닝을 진행하여 최적의 파라메터 조합을 찾았다. learning_rate는 [0.01, 0.1, 0.2], max_depth는 [3, 5, 7]로 시도했고 최적의 쌍은 각각 0.1, 7이었다.

4) Results

ML Method	Precision	Recall	F1	Macro F1
Naive Bayes Classifier	0.78	0.45	0.57	0.51
RBF SVM	0.78	0.78	0.78	0.62
ElasticNet Model	0.76	0.56	0.65	0.54
Feed-Forward Neural Network	0.74	0.90	0.81	0.55
Bi-LSTM	0.78	0.81	0.79	0.62
XGBoost	0.78	0.86	0.82	0.61

Table 1. 모델 비교 결과 (Feature 모두 사용)

① Naive Bayes Classifer는 높은 precision 말고는 낮은 성능이 나왔는데, 이는 NBC가 단순한 확률 모델이기에 데이터의 feature를 충분히 반영하지 못하는 결과로 해석된다.

② RBF SVM은 모든 지표에서 동일한 결과가 나오고 macro F1점수도 높아 라벨 예측 결과가 균형을 이루는 것을 알 수 있지만 다른 모델에 비해 전체적으로 특출난 지표가 존재하지 않는다.

③ ElasticNet Model은 NBC에서 precision이 낮아지고 나머지 지표가 살짝 오른 비슷한 형태로 다른 모델에 비해 성능이 떨어져 사용할 이유가 없다.

④ FFN은 가장 높은 recall를 보여 loneliness인 환자를 정상으로 판단하는 경우가 거의 없어 안전하다고 볼 수 있지만 precision과 macro F1가 낮아서 불필요한 loneliness판단이 많다고 볼 수 있다. 그럼에도 recall에 초점을 둔다면 FFN이 가장 이상적인 모델이다.

⑤ Bi-LSTM은 처음에 주어진 기본 default 모델이었는데, 기본 모델에서 smote와 np_x 평균화, 그리고 출력 layer 함수를 sigmoid로 바꾸니 성능이 비약적으로 상승했다. 가장 높은 precision과 macro F1 점수를 가진다.

⑥ XGBoost는 전반적으로 모두 뛰어난 지표를 보여주어 균형있는 loneliness 판단에 가장 효과적이라고 생각한다.

5) Feature Engineering

우선 모든 feature를 정리하면 다음과 같다.

check1_val	기분	0	편안함	check4_val	식사	0	잘함
		1	불안함			1	못함
		2	우울함				
check2_val	자살생각	0	없음	check5_val	식사횟수	1	1회
		1	있음			2	2회
						3	3회
check3_val	불면	0	없음	check6_val	외출횟수	0	안함
		1	있음			1	주 1~3일
						2	주 4~6일
						3	매일

Table 2. 설문조사(check_val) code book

service1	물품지원	0	없음	service9	안부인사	0	없음
		1	있음			1	있음
service2	식사지원	0	없음	service10	지지상담	0	없음
		1	있음			1	있음
service3	가사지원	0	없음	service11	응급개압	0	없음
		1	있음			1	있음
service4	위생지원	0	없음	service12	정서지원 센터연계	0	없음
		1	있음			1	있음
service5	병원동행	0	없음	service13	정서지원 의료기관연계	0	없음
		1	있음			1	있음
service6	차량지원	0	없음	service14	정서지원 복지서비스연계	0	없음
		1	있음			1	있음
service7	여가활동	0	없음	service15	정서지원 민간지원연계	0	없음
		1	있음			1	있음
service8	프로그램지원	0	없음	service161	정서지원 기타서비스	0	없음
		1	있음			1	있음

Table 3. 서비스지원(service) code book

다음은 정상인과 Loneliness 환자의 feature 분포이다. 데이터상 환자 수는 총 4933명이며 설문조사는 환자별로 40번씩 진행했다(무응답 포함). 그래서 환자들이 설문조사를 항목별로 평균적으로 몇번 체크했는지와 서비스를 몇번 받았는지를 나타내어 0과 1 라벨을 확실히 구분하는 feature을 추출하고자 한다.

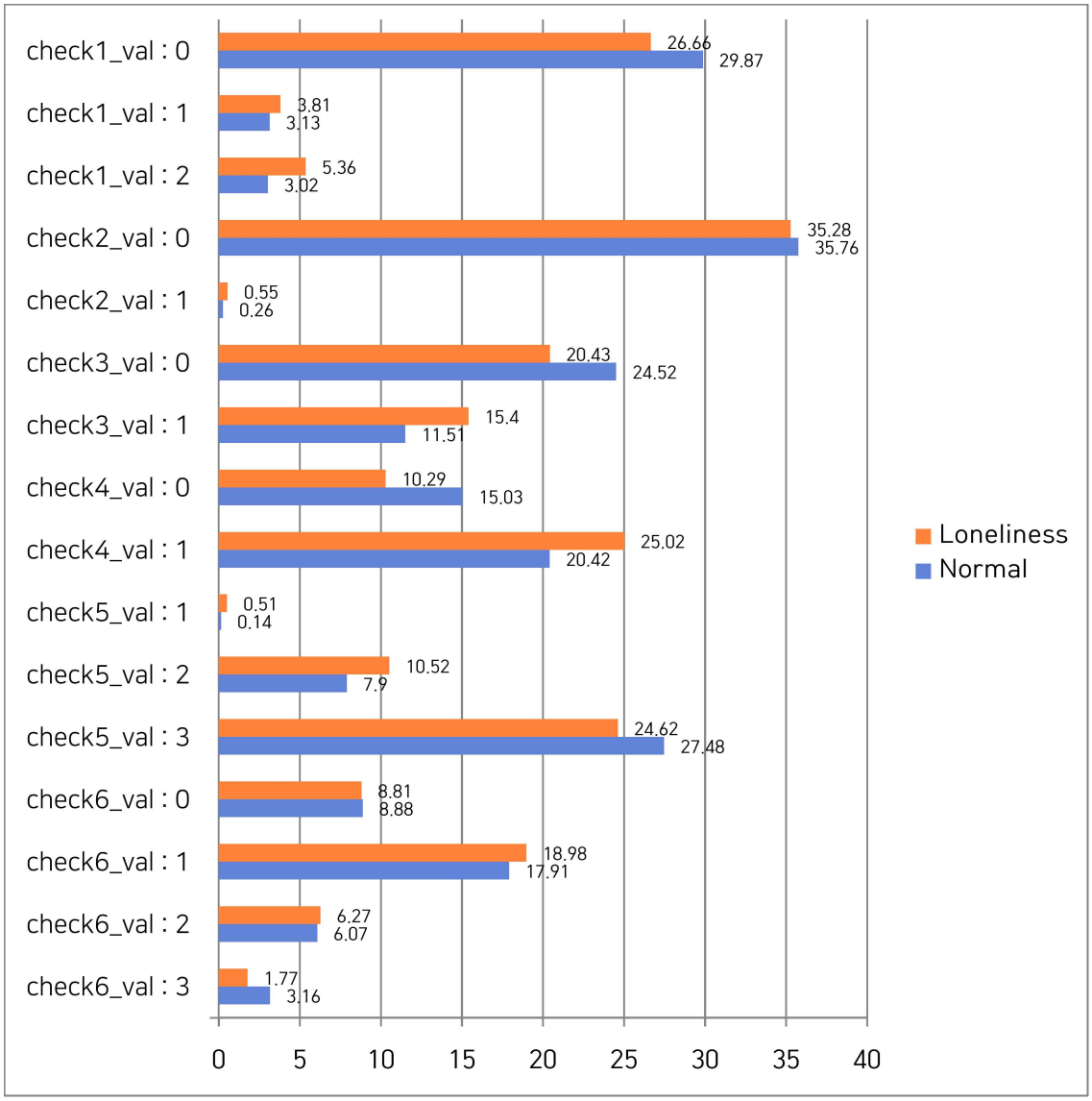


Table 4. 설문조사(check_val) 항목별 평균 체크횟수

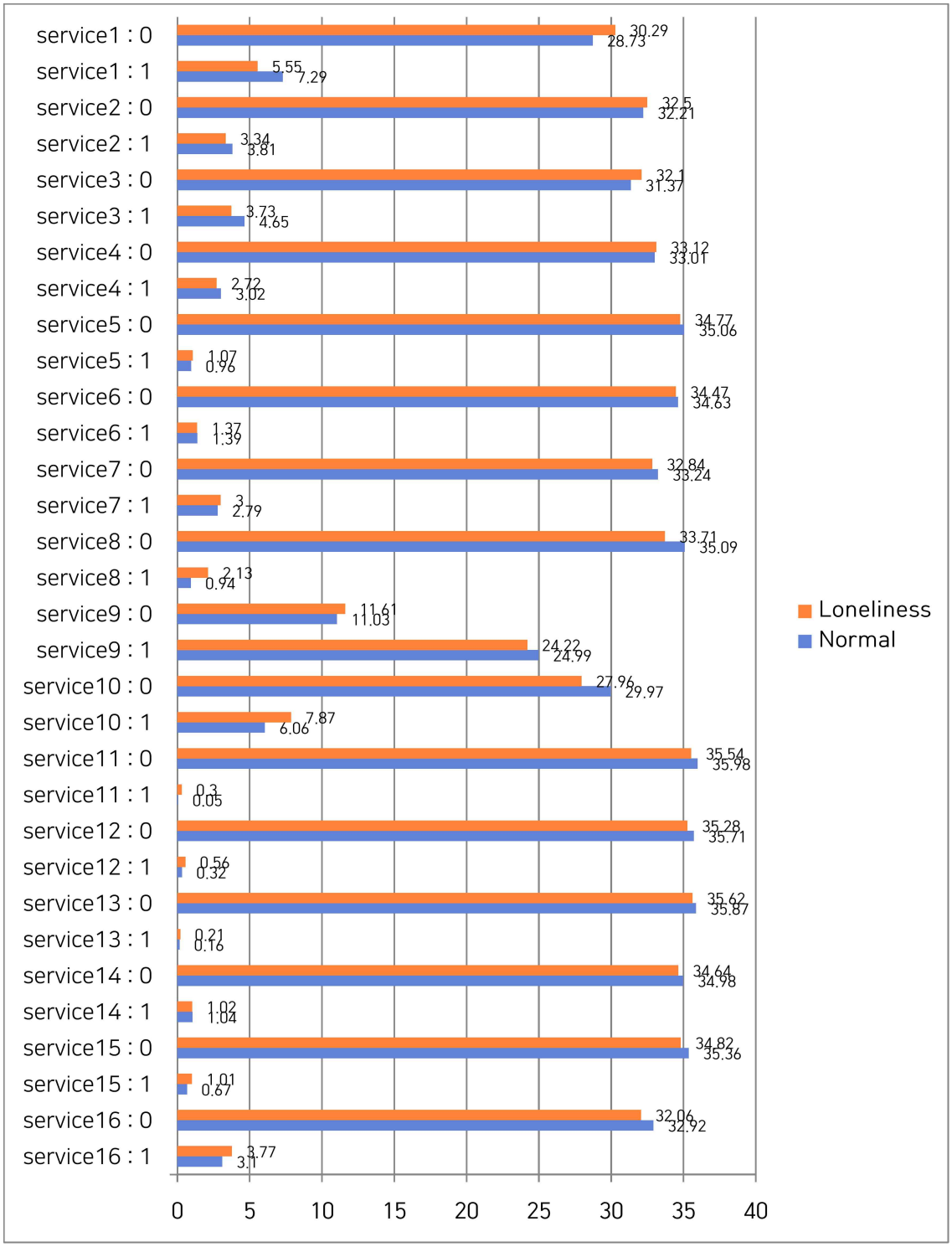


Table 5. 서비스(service) 항목별 평균 지원횟수

환자 예측은 1에 대한 recall 점수가 중요하기 때문에 recall이 가장 뛰어나면서 f1점수도 높은 FFN을 feature engineering 하여 상대적으로 낮은 precision과 macro f1 점수를 개선하고자 한다. 3-4의 모델 테스트에서는 모든 feature을 사용했는데 실제로는 feature가 많을 경우 모델이 복잡해져 연산량이 증가하고 속도가 느려질 수 있기 때문에 확실한 feature만 추출한다. 처음 All features 상태에서 feature을 하나씩 제외시켜 성능 증감이 얼마나 되었는지를 살펴보았다.

Feature	Precision	Recall	F1	Macro F1	+/-
All Features (Default)	0.74	0.90	0.81	0.55	0
remove check1_val	0.74	0.89	0.80	0.54	-0.3
check2_val	**0.75**	**0.90**	**0.82**	**0.56**	**+0.3**
check3_val	0.73	0.92	0.82	0.53	0
check4_val	0.73	0.91	0.81	0.52	-0.3
check5_val	**0.75**	**0.90**	**0.82**	**0.58**	**+0.5**
check6_val	0.74	0.88	0.80	0.55	-0.3
service1	0.74	0.88	0.81	0.56	-0.1
service2	0.74	0.89	0.81	0.55	-0.1
service3	**0.75**	**0.89**	**0.81**	**0.57**	**+0.2**
service4	0.74	0.89	0.81	0.55	-0.1
service5	0.74	0.89	0.81	0.56	0
service6	**0.75**	**0.91**	**0.82**	**0.58**	**+0.6**
service7	**0.74**	**0.90**	**0.81**	**0.56**	**+0.1**
service8	**0.74**	**0.91**	**0.82**	**0.55**	**+0.2**
service9	0.74	0.88	0.81	0.56	-0.1
service10	0.74	0.86	0.79	0.55	-0.6
service11	0.75	0.87	0.80	0.56	-0.2
service12	**0.75**	**0.90**	**0.82**	**0.57**	**+0.4**
service13	0.75	0.85	0.80	0.57	-0.3
service14	0.75	0.87	0.80	0.58	0
service15	**0.75**	**0.89**	**0.81**	**0.57**	**+0.2**
service16	0.74	0.89	0.81	0.55	-0.1

Table 6. FFN Feature Engineering

Table 6의 결과에 따라 제거시켜야 하는 feature들은 다음과 같다.

check2_val, check5_val, service3, service6, service7, service8, service12, service15

이제 이 feature를 제외하여 다음과 같은 feature set을 구성하였다.

check1_val, check3_val, check4_val, check6_val, service1, service2, service4, service5, service9, service10, service11, service13, service14, service16

이 feature set을 FFN에 적용하였더니 높은 recall은 보존되면서 실제로 precision과 macro f1이 개선됐고 f1점수도 미세하게 증가하였다.

Feature	Precision	Recall	F1	Macro F1
FFN				
All Features	0.74	0.90	0.81	0.55
Feature Set	0.76	0.90	0.82	0.59

Table 7. Feature Engineering Result

4. Conclusion

처음 연구를 시작할 때 받았던 default 모델과 연구를 통해 개선한 모델을 비교해 봤다. default 모델은 하나의 Bi-LSTM layer, 하나의 concatenate layer, softmax를 사용한 출력 layer로 구성되었으며 동일하게 epoch 1000으로 진행한 결과이다.

Feature	Precision	Recall	F1	Macro F1
Default Model	0.61	1.00	0.76	0.39
Improved Model	**0.76**	**0.90**	**0.82**	**0.59**

Table 8. Comparison of default model and improved model

이번 연구에서 모델의 성능 향상에 영향을 준 핵심 사항들은 다음과 같다.

- SMOTE 오버샘플링으로 Loneliness의 데이터 불균형을 해결
- np_x를 모두 사용하는 방법에서 평균화하는 방법으로 변경
- relu를 사용한 dense layer, sigmoid를 사용한 출력 layer로 구성된 FFN 구조로 변경
- feature engineering으로 필요한 feature만 사용

기존 default 모델은 지나치게 높은 recall에 비해 precision값이 낮은 모습을 볼 수 있는데, 이는 실제 라벨이 1인 데이터 뿐만 아니라 라벨이 0인 데이터도 1로 예측하는 경우가 많다는 의미이다. 그래서 라벨이 0일 때에 대한 지표가 좋지 않고 이는 예측 불균형 문제로 이어져 macro f1점수가 심각하게 낮았고 신뢰도 있는 예측을 제공하지 못했다.
그러나 개선 연구를 거치며 라벨이 1인 경우의 예측 정확도를 최대한 보존하면서, 불균형 문제에도 라벨이 0인 경우를 보다 정확하게 예측하는 모델을 만들 수 있었다. 특히 precision과 macro f1 점수가 각각 0.15와 0.2만큼 상승한 점이 고무적이다. 이 Improved model은 실제 환경에서도 신뢰있는 Loneliness 예측을 제공할 수 있을 거라 기대한다.