Font : 여기어때 잘난체 (licence_checked)

< 기술문서 >

팀명 : DDxDX
팀장 : 유승훈

모델은 “bert-base-multilingual-cased”를 사용하였습니다.
전체적인 모델의 틀은 html 파일에 기재된 https://velog.io/@seolini43 블로그를 참고하였습니다.

폴더 구성
1. data : Raw data(TrainSet_1차.csv, Validation_2차.csv)가 있는 폴더
2. logs/models : 학습모델이 저장된 경로
3. 모델생성스크립트(html) : Jupyter 인터프리터를 사용한 모델생성관련 html 문서가 있는 폴더
4. 외 파일
- C-statistics.txt
- DDxDX_outcome.py (Auc_Roc, confusion matrix(acc, f1 etc...) 산출)
- DDxDX_proba.py (2차 출력값 검증용 $a, b \in (0, 1)$)
- license_checklist.txt
- readme.txt
- 기술문서.pdf

※ 사용한 라이브러리는 아래와 같습니다.

라이센스 체크리스트

###인터프리터 ###		
Jupyter lab	3.4.8	BSD-3-Clause License
### 라이브러리 ###		
python	3.10.8	PSF(Python Software Foundation) License
pandas	1.5.0	BSD License
numpy	1.23.4	BSD License
sklearn	1.1.2	BSD License
scipy	1.9.2	BSD License
torch	1.12.1 + cu113	BSD license
tensorflow(keras)	2.10.0	Apache License 2.0
transformers	4.23.1	Apache License 2.0
matplotlib	3.6.1	BSD license
### 모델 ###		
bert-base-multilingual-cased		Apache License 2.0

Font : 여기어때 잘난체 (licence_checked)

[결과 보고]

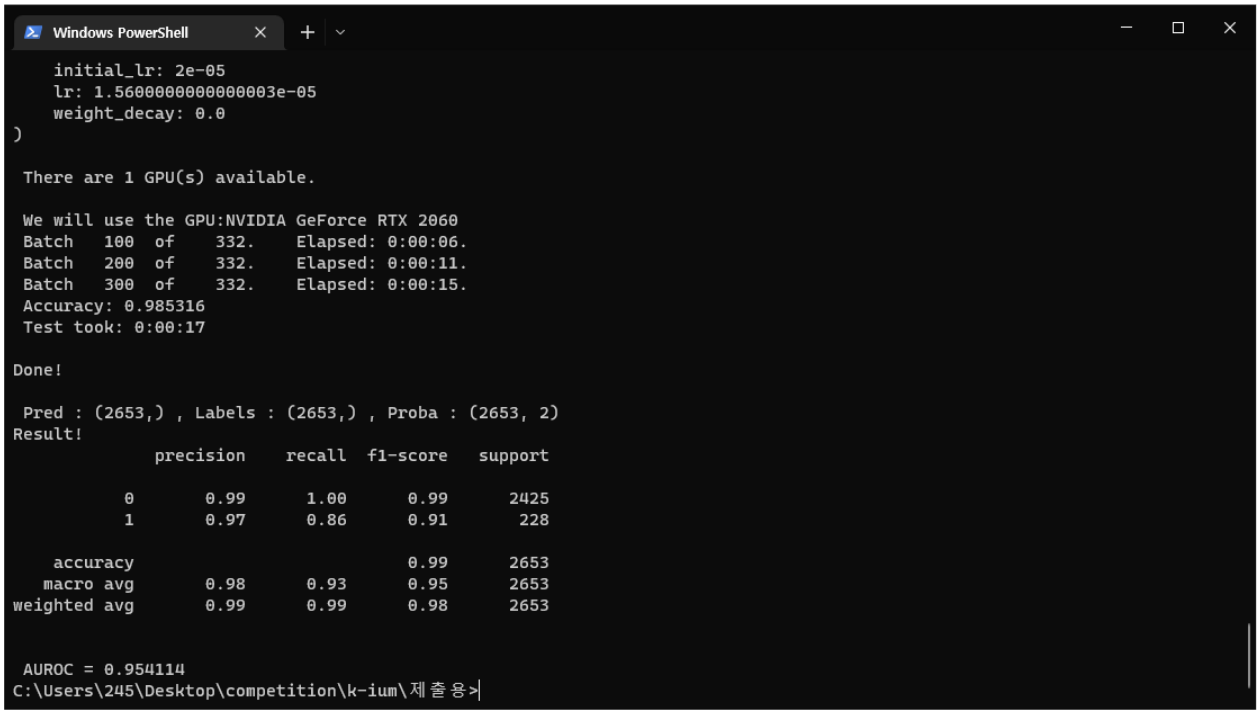

그림1. 2차 데이터 검증

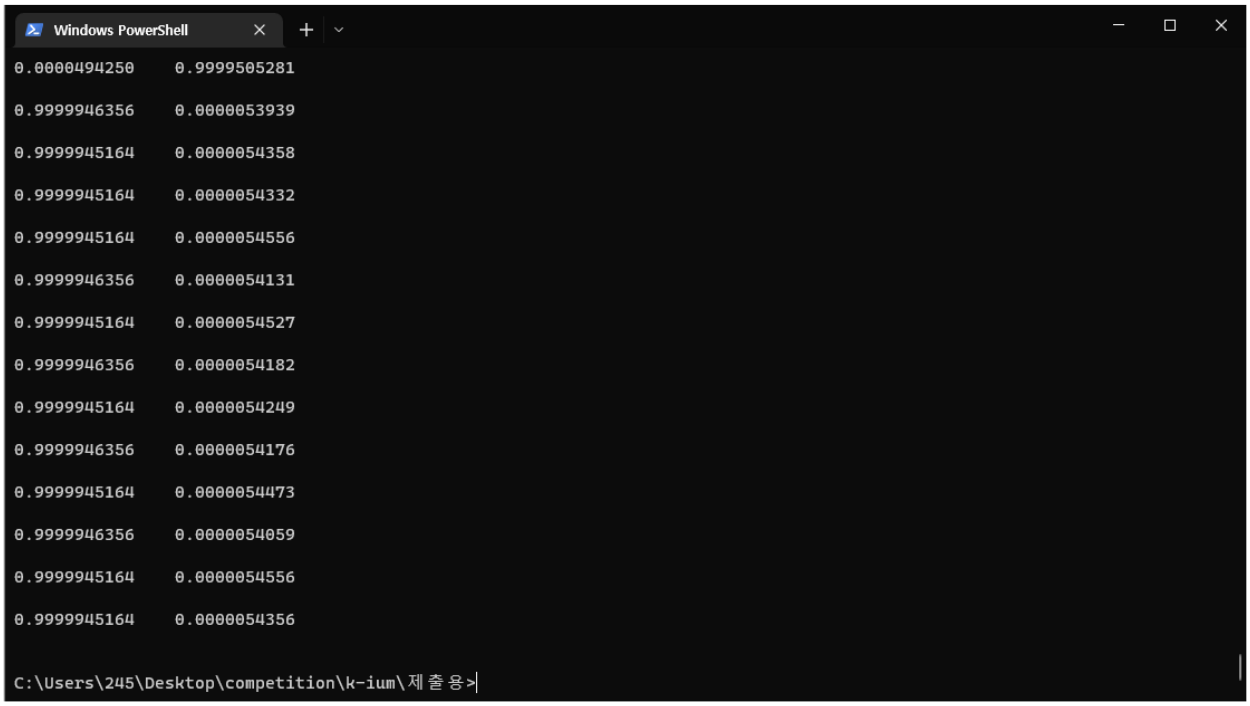

그림2. 2차 데이터 개별 확률 산출 (0, 1)

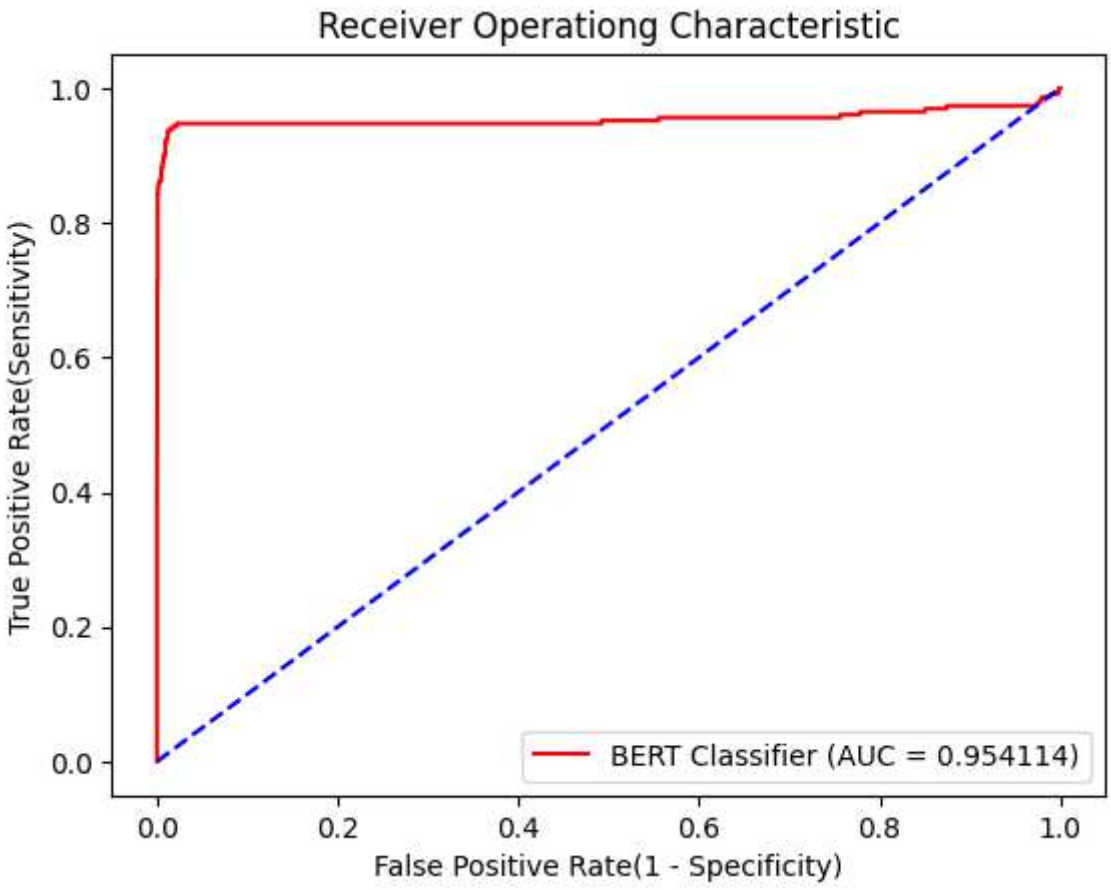

그림3. 2차 데이터 AUROC

[Conclusion]

결과 값에 대한 해석은
학습한 모델에 Validation data를 적용 시 산출된 결과로 AUC_ROC의 경우 0.954114로 기존의 학습 결과인 0.997901에 비해 다소 낮아졌으나 성적 자체는 여전히 준수한 편이라고 판단됩니다.

다만 분야가 의료라는 점에서 대회의 취지처럼 FP, FN보다 TP, TN을 더 명확히 판단 할 수 있는 모델이 필요한데 실무에 적용하기에 다소 부적합하며 추가적인 개선이 요구된다고 생각합니다.

그 이유는 무질환자에 대한 추가적인 검사는 과잉진료 선에서 끝나지만, 유질환자 판독 오류의 경우 여파가 굉장히 크리티컬하게 작용하기 때문입니다.

결론적으로 학습 및 검증 데이터의 라벨 간 비율의 불균형으로 인해 상대적으로 FP, FN에 대한 지표들이 높게 나왔고 이에 대한 Cross-validation(Straight K-fold)을 적용하였으면 결과가 조금 더 개선되지 않았을까 하는 아쉬움이 있습니다.

Font : 여기어때 잘난체 (licence_checked)

[과제 수행 절차]

1. 데이터 전처리

데이터 확인 작업을 수행함. 한글과 영문(의학용어)가 혼용된 문서로 Findings 컬럼은 영상소견을 적어 둔 글자들(string), Conclusion₩n 컬럼은 요약 소견의 글자들(string), 마지막은 이진변수 라벨로 급성 뇌경색이 있는 경우 1, 없는 경우 0. (그림1)

데이터의 특징
- 영문 비중이 높은 편이나, 간혹 한글로 된 의학용어가 들어가 있음.
- 부분 결측치가 있음.
- Findings , Conclusion 둘 다 결측인 경우는 없음.

데이터의 결측치는
Findings 1376, Conclusions 34, labels 0.

	Findings	Conclusion\n	AcuteInfarction
0	Clinical information : 두부외상 후 후유증 평가\r\n\r\nAx...	1. Encephalomalacic change in both frontal lob...	0
1	Clinical information : lung cancer\nAxial T1WI...	1. No change of focal enhancing lesion in left...	0
2	Clinical information : Multiple Sclerosis\r\n\...	No significant interval change of abnormal hyp...	0
3	Clinical information : patient with DLBCL.\r\n...	1. Decreased extent of enhancing mass in the l...	0
4	Clinical information : Transient cerebral isch...	1. Acute infarctions at right BG, right F-P-T ...	1
...	...	...	...
6185	Clinical information : s/p Removal of vestibul...	No evidence of remnant mass or remarkable post...	0
6186	CI, headache of sudden onset (known UIA).\r\nA...	1. No evidence of acute infarctions.\r\n2. Enc...	0
6187	Clinical information : patient with DLBCL.\n\n...	1. Increased size of homogeneous enhancing mas...	0
6188	Clinical information : Lung cancer patient 임.\...	No evidence of intracranial metastasis.\n	0
6189	CI, cerebellar mass (metastatic carcinoma), a ...	1. Three new hemorrhagic metastases (Rt O 1.0 ...	0

6190 rows × 3 columns

그림1. 데이터 확인

```
# 결측치 Conclusion 리스트 확인
df.loc[df.Conclusions.isna(), : ].head()
```

	Findings	Conclusions	labels
266	MRI for radiosurgery \n	NaN	0
446	MRI for radiosurgery\n	NaN	0
482	MRI for radiosurgery\n	NaN	0
537	MRI for radiosurgery\n	NaN	0

```
# 결측치 Findings 리스트 확인
df.loc[df.Findings.isna(), : ].head()
```

	Findings	Conclusions	labe
7	NaN	mild diffuse brain atrophy\r\ncomplete occlusi...	
13	NaN	interval decreased size & number of metastatic...	
28	NaN	Limited evaluation d/t motion artifact.\r\n\r\...	
32	NaN	interval increased extent of acute infarction ...	
40	NaN	still noted multifocal small nodular enhancing...	

그림2. 결측치 확인

Findings 라벨이 1에 해당하는 결측치 233개의 행은 응급상황에서의 기록 결측이라고 유추하여 "Emergency"를 줄인 "ER"로 보간.

그 외 결과값이 0인 1143의 행에 대해서는 특정 사유를 유추하기 어려워 "No Findings"를 줄인 "NF"로 보간.

Conclusion 결측치의 경우 1개의 행을 제외하고 Findings가 "MRI for radiosurgery"이므로 결측치를 "Gamma Knife"를 줄여서 GK로 보간.(그림3)

```
[8]: # Findings는 결과값이 1인행 223개에 대해서 "Emergency"를 줄인 "ER"을 대체
     idx_ER = df[(df.AcuteInfarction == 1) & df.Findings.isna()].index

[9]: # Findings는 결과값이 0인행 1143개에 대해서 "No Findings"를 줄인 "NF"을 대체
     idx_NF = df[(df.AcuteInfarction == 0) & df.Findings.isna()].index

[10]: # Conclusion 결측치의 경우 1개를 제외하고 Findings가 "MRI for radiosurgery"이므로 결측치를 "GammaKnife"를 줄여서 GK로 대체
      df.Conclusion.fillna("GK", inplace = True)

[11]: df.isna().sum()

[11]: Findings          1376
      Conclusion           0
      AcuteInfarction      0
      dtype: int64
```

그림 3. 결측치 대체

Python의 라이브러리인 re(정규식)을 통해서 특수문자를 삭제하고 각 별도의 컬럼으로 분류하여 Bert 입력형태에 맞게 전처리.

```
# 한글 분류 예제(숫자 제거)
re.compile(kor).findall(df.Findings[11])

['Clinical information ',
 ' ',
 'Axial T1WI',
 ' sagittal T1WI',
 ' axial T2WI',
```

```
# 영문 분류 예제
re.compile(eng).findall(df.Findings[11])

['획득하였으며',
 '및',
 '에',
 '대해',
 '조영증강을',
 '시행함'
```

	Findings	Conclusions	labels	Findings_eng	Findings_kor	Conclusions_eng	Conclusions_kor
4491	Clinical information : trauma 이후 기억력 저하 발생하여 평...	1. Nonspecific focal T2 hyperintense lesion at...	0	Clinical information trauma Comparison ...	이후기억력저하발생하여평가위해촬영함획득하였으며조영증강은시행하지않았음이에서관찰됨에에서를...	Clinical information trauma Comparison ...	이후기억력저하발생하여평가위해촬영함획득하였으며조영증강은시행하지않았음이에서관찰됨에에서를...
1042	Clinical information : Left paraclinoid ICA의 a...	1. s/p Stent assisted coil embolization for un...	0	Clinical information Left paraclinoid ICA ane...	의에대해을시행한환자이며에상관찰되던환자임획득하였으며에대해서조영증강을시행함	Clinical information Left paraclinoid ICA ane...	의에대해을시행한환자이며에상관찰되던환자임획득하였으며에대해서조영증강을시행함
124	Clinical information : Non-small cell lung can...	1. Focal enhancing lesion in the right IAC.\r\...	0	Clinical information Non-small cell lung canc...	획득하였으며조영증강을시행함	Clinical information Non-small cell lung canc...	획득하였으며조영증강을시행함

그림4. 정규식으로 문자별 컬럼 분류

Bert 입력 전처리
- CLS, SEP
- 서브워드 토크나이저
- 어텐션 마스크

```
# CLS, SEP 붙이기 (문장의 시작, 끝)
sentences = ["[CLS] " + str(f) + " [SEP]" for f in zip(train.Findings_eng, train.Findings_kor, train.Conclusions_eng, train.Conclusions_kor)]
```

```
sentences[0]
```

```
"[CLS] ('Clinical information  Axial T1WI sagittal T1WI axial T2WI axial FLAIR axial T2 GRE image axial DWI                    ', '획득하였으며조영증강을시행함대뇌소뇌뇌간뇌실뇌실질외공간에출혈종괴수두증위축급성또는과거의허혈성병변등의주요이상소견없음', 'Clinical information  Axial T1WI sagittal T1WI axial T2WI axial FLAIR axial T2 GRE image axial DWI                    ', '획득하였으며조영증강을시행함대뇌소뇌뇌간뇌실뇌실질외공간에출혈종괴수두증위축급성또는과거의허혈성병변등의주요이상소견없음') [SEP]"
```

그림5. cls, sep로 문장의 시작과 끝 설정

검증을 위해 1차 제공 데이터인 6190개를 셔플 후 Train 4000 / Test 2190 로 분할.

```
# 데이터 셔플
df_shuffle = df.sample(frac=1).reset_index(drop=True)
```

```
print('셔플\n', df_shuffle.Findings.head(3), '\n\n 기존\n', df.Findings.head(3))
```

```
셔플
 0    Clinical information : Brain Abscess\r\n\r\nAx...
1    CI, ischemic stroke.\r\nAxial T1WI, sagittal T...
2    Clinical information : preterm, r/o PVL\r\n\r\...
Name: Findings, dtype: object

 기존
 0    Clinical information : 두부외상 후 후유증 평가\r\n\r\nAx...
1    Clinical information : lung cancer\nAxial T1WI...
2    Clinical information : Multiple Sclerosis\r\n\...
Name: Findings, dtype: object
```

```
#train data & test data 분리 (4000, 2190)
train = df_shuffle[:4000]
test = df_shuffle[4000:]

print(train.shape)
print(test.shape)
```

```
(4000, 7)
(2190, 7)
```

그림6. 데이터 셔플 및 분할

생성된 sentences에 서브워드 토크나이저를 적용.

```
# 확인
print(sentences[0])  #토크나이징 전
print(tokenized_texts[0]) #토크나이징 후
```

```
[CLS] ('Clinical information :  Unruputured aneurysm BA aneurysm\n\nintracranial TOF MRA 획득하였으며 neck MRA에 대해서 조영증강을 시행함.\n\n\n', 'S/P Coil-embolization basilar top(2016.08.11) and left A-com aneurysm(2016.08.17).\r\n\r\nMinor recanalization at left A-com aneurysm.\r\n    -> rec) Follow-up.\n') [SEP]
['[CLS]', '(', "'", 'Clinical', 'information', ':', 'Un', '##rup', '##uture', '##d', 'ane', '##ury', '##sm', 'BA', 'ane', '##ury', '##sm', '\\', 'n', '\\', 'nin', '##tra', '##cra', '##nial', 'TO', '##F', 'MR', '##A', '획', '##득', '##하였으며', 'neck', 'MR', '##A', '##에', '대해', '##서', '조', '##영', '##증', '##강', '##을', '시', '##행', '##함', '.', '\\', 'n', '\\', 'n', '\\', 'n', "'", ',', "'", 'S', '/', 'P', 'Co', '##il', '-', 'em', '##bol', '##ization', 'basi', '##lar', 'top', '(', '2016', '.', '08', '.', '11', ')', 'and', 'left', 'A', '-', 'com', 'ane', '##ury', '##sm', '(', '2016', '.', '08', '.', '17', ')', '.', '\\', 'r', '\\', 'n', '\\', 'r', '\\', 'n', '##M', '##ino', '##r', 'reca', '##nali', '##zation', 'at', 'left', 'A', '-', 'com', 'ane', '##ury', '##sm', '.', '\\', 'r', '\\', 'n', '-', '>', 're', '##c', ')', 'Follow', '-', 'up', '.', '\\', 'n', "'", ')', '[SEP]']
```

그림7. 토크나이징 적용 전 후 비교

```
[35]: # 어텐션 마스크
      attention_masks = []

      for seq in input_ids:
          seq_mask = [float(i>0) for i in seq]
          attention_masks.append(seq_mask)
```

그림8. 어텐션 마스크 적용.

리소스가 제한되어 적용한 학습량의 차이도 있겠지만, 전반적으로 정규식을 적용했을 경우 오히려, Raw 데이터를 넣었을 때에 비해 낮은 결과가 도출됨.

```
======== Epoch 10 / 10 ========
Training...

  Average training loss: 0.074250
  Training epcoh took: 0:01:34

Running Validation...
  Accuracy: 0.942500
  Validation took: 0:00:03

Training complete!
```

그림9. 정규식 적용 후 Acc & Loss.

```
======== Epoch 22 / 100 ========
Training...
  Batch   500  of    697.    Elapsed: 0:01:44.

  Average training loss: 0.003272
  Training epcoh took: 0:02:25

Running Validation...
  Accuracy: 0.987179
  Validation took: 0:00:04
Early Stop!

Training complete!
```

그림10. Raw 데이터 결과

이유에 대한 추정으로는
정규식을 적용한 경우 각 컬럼으로 분리 후 영어와 한글이 앞뒤로 다시 합쳐지는 과정을 거치게 되어 문장의 순서와 규칙성, 줄 바꿈 등의 사용자 패턴과 같은 부분에서 데이터 형태의 변화가 일어나고 단락 변화와 같은 특수문자 [/t/n] 데이터가 소실된 부분 결손이 학습에 영향을 주었다고 판단됨.

그러므로 Raw data로 학습한 모델을 저장하고 평가를 시행하고 Roc_curve를 그래프로 산출.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5580
1	0.99	0.99	0.99	610
accuracy			1.00	6190
macro avg	1.00	0.99	0.99	6190
weighted avg	1.00	1.00	1.00	6190

```
  Batch   300  of    774.    Elapsed: 0:00:15.
  Batch   400  of    774.    Elapsed: 0:00:19.
  Batch   500  of    774.    Elapsed: 0:00:24.
  Batch   600  of    774.    Elapsed: 0:00:29.
  Batch   700  of    774.    Elapsed: 0:00:34.

Accuracy: 0.997901
Test took: 0:00:37
```

그림 11. 모델 성능 지표확인(전체 학습)

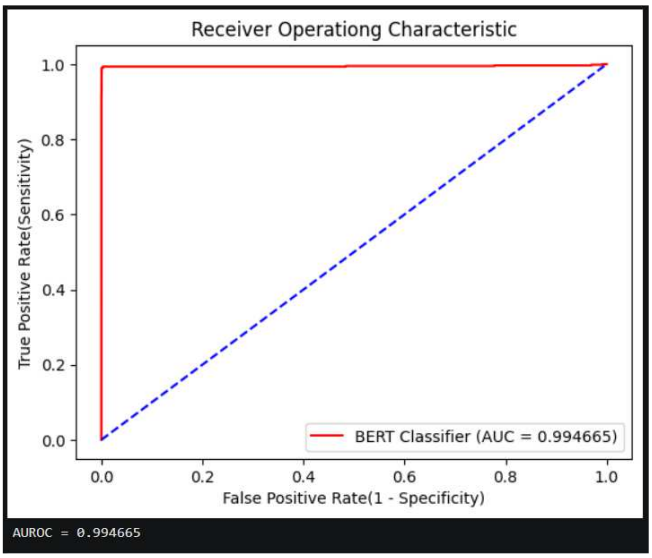

그림 12. ROC curve 시각화

그림10과 11의 검증 데이터 부분은 기존 데이터를 전체 학습하여 같은 데이터로 평가되어 신뢰성이 조금 떨어짐.

기존 6190개의 데이터를 셔플 후 분할하여 4000개를 학습하고 2160개를 평가로 사용한 경우에는 98% 정도의 정확도와 AUROC를 보임.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1977
1	0.93	0.90	0.91	213
accuracy			0.98	2190
macro avg	0.96	0.94	0.95	2190
weighted avg	0.98	0.98	0.98	2190

```
======== Epoch 11 / 100 ========
Training...

  Average training loss: 0.018090
  Training epcoh took: 0:01:34

Running Validation...
  Accuracy: 0.980000
  Validation took: 0:00:03
Early Stop! 100

Training complete!
```

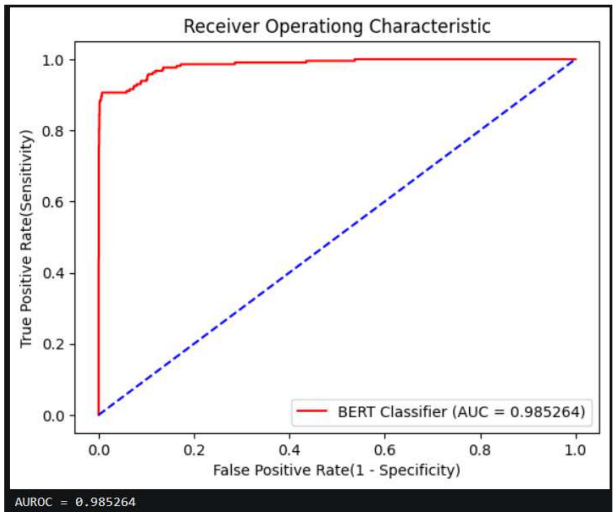

그림 13. 모델 성능 지표확인 – ROC_curve (부분 학습 4:2)

검증 후 직접 입력을 통한 테스트 창을 별도로 설정하여 Acute Stroke에 대한 증상, 검사 방법 및 결과 등 을 기입했을 때 실무에서 바로 적용할 수 있는 형태로의 문장 테스트를 수행함.

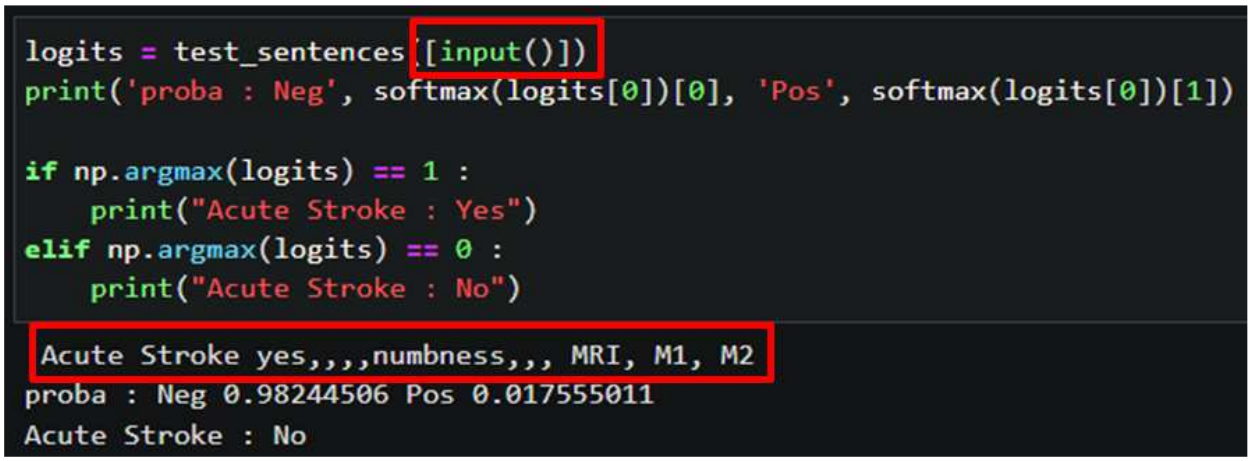

그림 14. 직접 입력을 통한 테스트

CLI 형태로 터미널에 python DDxDX.py (proto type)를 입력하고 데이터 경로를 입력하여 테스트 수행.

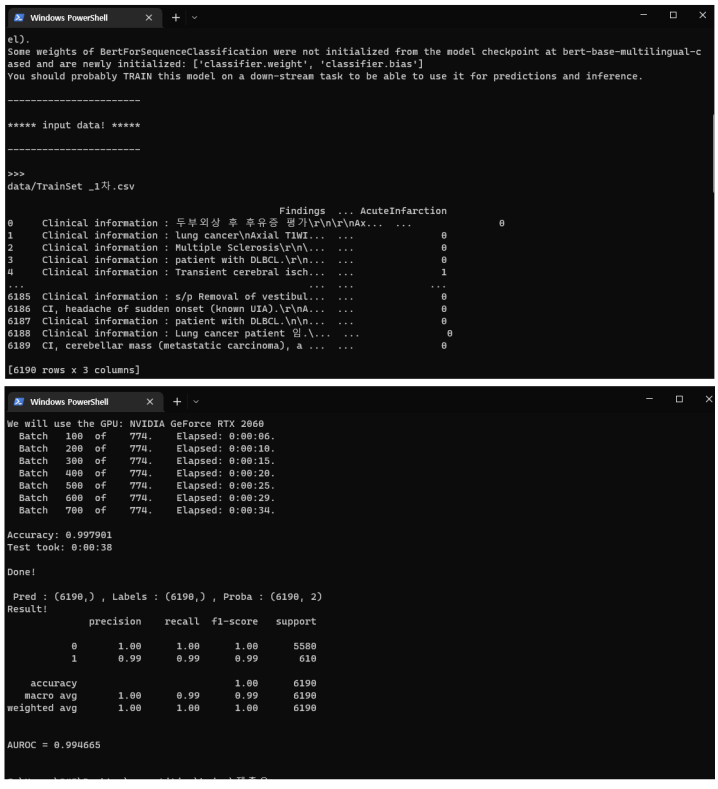

그림 15. CLI 테스트

감사합니다.