글로벌아카데미 1차 수업복기 및 피드백

수료생 신희섬

목차

1. 수업의 전반적인 흐름도

글로벌아카데미에서의 수업은 김지량 교수님이 오전 / 김석원 강사님이 오후 수업을 도맡아 진행하였습니다.

우선 전체적인 흐름을 말씀드리자면

김지량 교수님 : AI 의 전반적인 흐름 , Transformer에 대한 간략한 요약 , CNN + Pooling 개념설명 , od에 대한 설명 , (지도 , 비지도 , 강화) 학습 및 회귀와 분류에대한 설명 , (열화상 모델 + 동물od 모델) 실습 - 1~2주차

김석원 강사님 : 문서작업(요구사항 정의서 , 화면설계서 , 테이블 명세서 , 칸트차트)작업예시 및 조별 작업지시 , DBeaver를 통한 DB관리 , 쿼리문 작성 예시 및 첨언

React 및 FastAPI에 대한 부분을 도맡아 하셨지만 수업의 대부분이 youtube 링크 공유라던지 AI한테 답변받은 내용을 메일로 전달받고 간략한 설명만 전달받아 아쉬움이 컸습니다.

미니프로젝트 기간 - 김지량 교수님께서 OCR모델 탐색에 시간을 할애해보란 조언과 함께 OCR모델 분석에 전체 3팀이 집중하였고 미니프로젝트는 크게 EasyOCR , PaddleOCR , TesseractOCR 모델 실적용 후 성능을 분석 후 Ollama Gemma와 함께 프로젝트에 붙이는데 집중하였습니다.

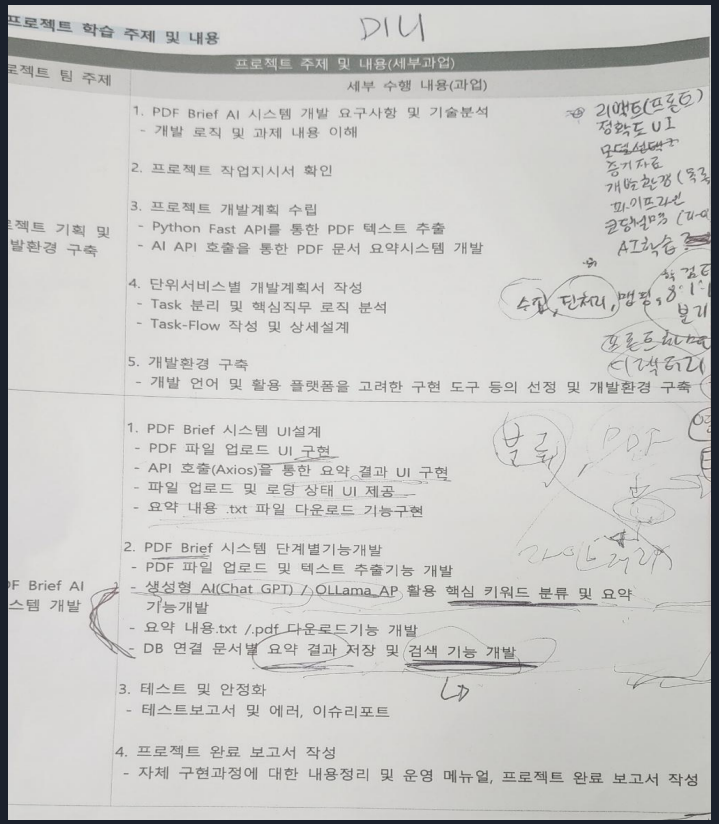
메인 프로젝트 기간 - 자율성있게 진행되었고 3팀 모두 의견을 종합하면서 프로젝트를 진행하였고 서로 원하는 분류를 해내기 위해서는 단순 프롬프트 만으로는 한계가 있다고 판단 Finetunning을 해야겠다 판단했고 이 점에서부터는 서로 다른 방식으로 진행되었습니다.

2. 미니 프로젝트

김지량 교수님이 흐름도를 잡아주셨고 그 방향대로 진행되었습니다.

OCR모델 비교분석 - EasyOCR , TesseractOCR , PaddleOCR 등 여러가지 OCR오픈소스 모델들을 서빙해서 비교해본뒤 텍스트 추출을 통해 그걸 ollama의 Gemma에 보내 키워드 추출 및 요약에 집중하였습니다.

1,2,3팀 모두 비슷하게 진행되었습니다.



프로젝트 학습 주제 및 내용 — 미니

프로젝트 팀 주제	프로젝트 주제 및 내용(세부과업) 세부 수행 내용(과업)
프로젝트 기획 및 개발환경 구축	1. PDF Brief AI 시스템 개발 요구사항 및 기술분석 - 개발 로직 및 과제 내용 이해 2. 프로젝트 작업지시서 확인 3. 프로젝트 개발계획 수립 - Python Fast API를 통한 PDF 텍스트 추출 - AI API 호출을 통한 PDF 문서 요약시스템 개발 4. 단위서비스별 개발계획서 작성 - Task 분리 및 핵심직무 로직 분석 - Task-Flow 작성 및 상세설계 5. 개발환경 구축 - 개발 언어 및 활용 플랫폼을 고려한 구현 도구 등의 선정 및 개발환경 구축
PDF Brief AI 시스템 개발	1. PDF Brief 시스템 UI설계 - PDF 파일 업로드 UI 구현 - API 호출(Axios)를 통한 요약 결과 UI 구현 - 파일 업로드 및 로딩 상태 UI 제공 - 요약 내용 .txt 파일 다운로드 기능구현 2. PDF Brief 시스템 단계별기능개발 - PDF 파일 업로드 및 텍스트 추출기능 개발 - 생성형 AI(Chat GPT) / OLLama API 활용 핵심 키워드 분류 및 요약 기능개발 - 요약 내용.txt /.pdf 다운로드기능 개발 - DB 연결 문서별 요약 결과 저장 및 검색 기능 개발 3. 테스트 및 안정화 - 테스트보고서 및 에러, 이슈리포트 4. 프로젝트 완료 보고서 작성 - 자체 구현과정에 대한 내용정리 및 운영 메뉴얼, 프로젝트 완료 보고서 작성

3. 메인 프로젝트

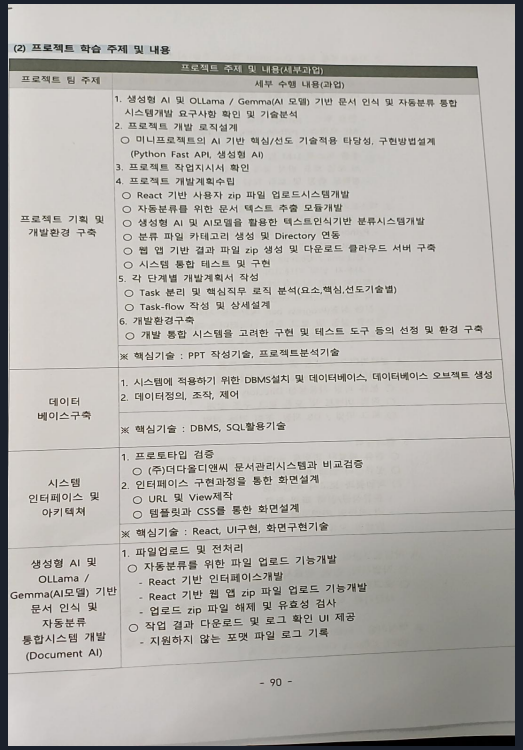
팀마다 자율성높게 프로젝트가 진행이되었고각 팀마다 제가 기억하는 집중했던 방향들을 복기해보겠습니다.

1팀 - 최신 OCR기술 PaddleVL 및 DeepseekOCR 비교 분석 및 적용 , React - vite 도입 및 React Compiler 도입 , 문서 분류를 위해 Bert의 CLS토큰 도입 , Gemma3모델 lora Finetunning

3팀 - OCR모델 구동 및 분류AI모델 구동 시의 자원관리에 따른 속도부분 개선을 위해 Selenium을 도입 , Gemma3 모델 Finetunning

2팀은 제가 기억에 남아있지않아 자료가부족합니다.

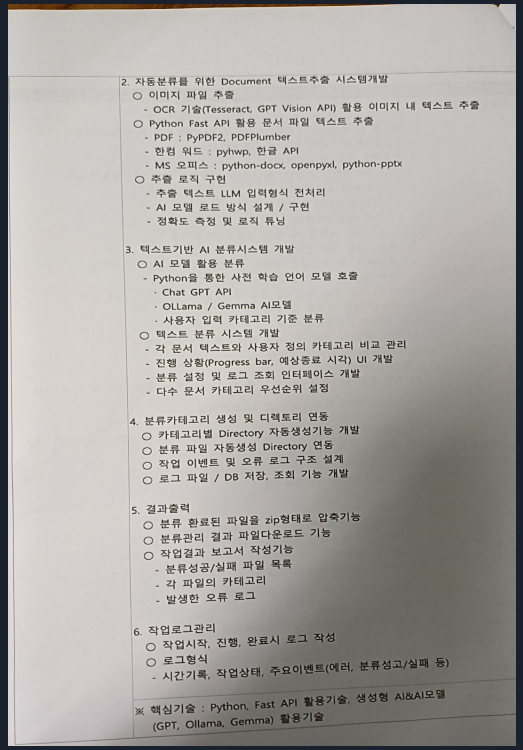
3. 메인 프로젝트



(2) 프로젝트 학습 주제 및 내용

프로젝트 주제 및 내용(세부과업)	
프로젝트 팀 주제	세부 수행 내용(과업)
프로젝트 기획 및 개발환경 구축	1. 생성형 AI 및 OLLama / Gemma(AI 모델) 기반 문서 인식 및 자동분류 통합 시스템개발 요구사항 확인 및 기술분석 2. 프로젝트 개발 로직설계 ○ 미니프로젝트의 AI 기반 핵심/선도 기술적용 타당성, 구현방법설계 (Python Fast API, 생성형 AI) 3. 프로젝트 작업지시서 확인 4. 프로젝트 개발계획수립 ○ React 기반 사용자 zip 파일 업로드시스템개발 ○ 자동분류를 위한 문서 텍스트 추출 모듈개발 ○ 생성형 AI 및 AI모델을 활용한 텍스트인식기반 분류시스템개발 ○ 분류 파일 카테고리 생성 및 Directory 연동 ○ 웹 앱 기반 결과 파일 zip 생성 및 다운로드 클라우드 서버 구축 ○ 시스템 통합 테스트 및 구현 5. 각 단계별 개발계획서 작성 ○ Task 분리 및 핵심직무 로직 분석(요소,핵심,선도기술별) ○ Task-flow 작성 및 상세설계 6. 개발환경구축 ○ 개발 통합 시스템을 고려한 구현 및 테스트 도구 등의 선정 및 환경 구축 ※ 핵심기술 : PPT 작성기술, 프로젝트분석기술
데이터 베이스구축	1. 시스템에 적용하기 위한 DBMS설치 및 데이터베이스, 데이터베이스 오브젝트 생성 2. 데이터정의, 조작, 제어 ※ 핵심기술 : DBMS, SQL활용기술
시스템 인터페이스 및 아키텍쳐	1. 프로토타입 검증 ○ (주)더다울디앤씨 문서관리시스템과 비교검증 2. 인터페이스 구현과정을 통한 화면설계 ○ URL 및 View제작 ○ 템플릿과 CSS를 통한 화면설계 ※ 핵심기술 : React, UI구현, 화면구현기술
생성형 AI 및 OLLama / Gemma(AI모델) 기반 문서 인식 및 자동분류 통합시스템 개발 (Document AI)	1. 파일업로드 및 전처리 ○ 자동분류를 위한 파일 업로드 기능개발 - React 기반 인터페이스개발 - React 기반 웹 앱 zip 파일 업로드 기능개발 - 업로드 zip 파일 해제 및 유효성 검사 ○ 작업 결과 다운로드 및 로그 확인 UI 제공 - 지원하지 않는 포맷 파일 로그 기록

- 90 -



2. 자동분류를 위한 Document 텍스트추출 시스템개발
○ 이미지 파일 추출
- OCR 기술(Tesseract, GPT Vision API) 활용 이미지 내 텍스트 추출
○ Python Fast API 활용 문서 파일 텍스트 추출
- PDF : PyPDF2, PDFPlumber
- 한컴 워드 : pyhwp, 한글 API
- MS 오피스 : python-docx, openpyxl, python-pptx
○ 추출 로직 구현
- 추출 텍스트 LLM 입력형식 전처리
- AI 모델 로드 방식 설계 / 구현
- 정확도 측정 및 로직 튜닝

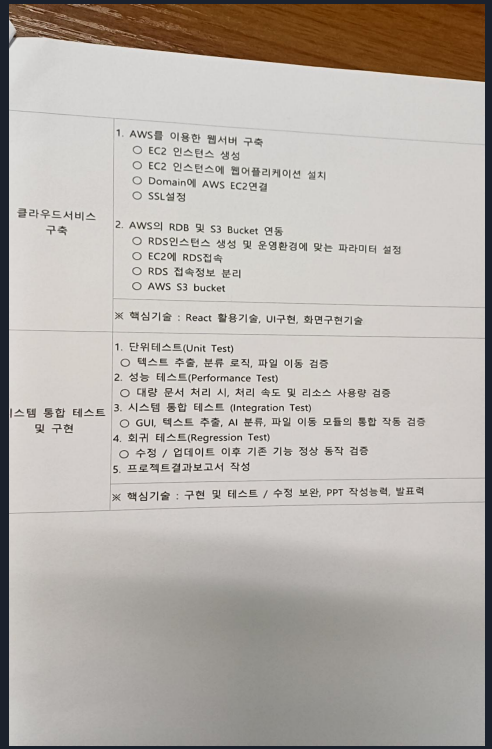
3. 텍스트기반 AI 분류시스템 개발
○ AI 모델 활용 분류
- Python을 통한 사전 학습 언어 모델 호출
· Chat GPT API
· OLLama / Gemma AI모델
· 사용자 입력 카테고리 기준 분류
○ 텍스트 분류 시스템 개발
- 각 문서 텍스트와 사용자 정의 카테고리 비교 관리
- 진행 상황(Progress bar, 예상종료 시각) UI 개발
- 분류 설정 및 로그 조회 인터페이스 개발
- 다수 문서 카테고리 우선순위 설정

4. 분류카테고리 생성 및 디렉토리 연동
○ 카테고리별 Directory 자동생성기능 개발
○ 분류 파일 자동생성 Directory 연동
○ 작업 이벤트 및 오류 로그 구조 설계
○ 로그 파일 / DB 저장, 조회 기능 개발

5. 결과출력
○ 분류 환료된 파일을 zip형태로 압축기능
○ 분류관리 결과 파일다운로드 기능
○ 작업결과 보고서 작성기능
- 분류성공/실패 파일 목록
- 각 파일의 카테고리
- 발생한 오류 로그

6. 작업로그관리
○ 작업시작, 진행, 완료시 로그 작성
○ 로그형식
- 시간기록, 작업상태, 주요이벤트(에러, 분류성고/실패 등)

※ 핵심기술 : Python, Fast API 활용기술, 생성형 AI&AI모델 (GPT, Ollama, Gemma) 활용기술



클라우드서비스 구축	1. AWS를 이용한 웹서버 구축 ○ EC2 인스턴스 생성 ○ EC2 인스턴스에 웹어플리케이션 설치 ○ Domain에 AWS EC2연결 ○ SSL설정 2. AWS의 RDB 및 S3 Bucket 연동 ○ RDS인스턴스 생성 및 운영환경에 맞는 파라미터 설정 ○ EC2에 RDS접속 ○ RDS 접속정보 분리 ○ AWS S3 bucket ※ 핵심기술 : React 활용기술, UI구현, 화면구현기술
시스템 통합 테스트 및 구현	1. 단위테스트(Unit Test) ○ 텍스트 추출, 분류 로직, 파일 이동 검증 2. 성능 테스트(Performance Test) ○ 대량 문서 처리 시, 처리 속도 및 리소스 사용량 검증 3. 시스템 통합 테스트 (Integration Test) ○ GUI, 텍스트 추출, AI 분류, 파일 이동 모듈의 통합 작동 검증 4. 회귀 테스트(Regression Test) ○ 수정 / 업데이트 이후 기존 기능 정상 동작 검증 5. 프로젝트결과보고서 작성 ※ 핵심기술 : 구현 및 테스트 / 수정 보완, PPT 작성능력, 발표력

4. 자체 피드백 및 개선사항

1. 수업 흐름

김지량 교수님이 처음에 보여주셨던AI전체적인 흐름도 부터 시작한 실습까지의 과정은 학생 만족도도 높았었고 흥미도도 많이 올라있었던 시기라고 생각이듭니다.

하지만 되게 스피드있게 진행이되었고2주차쯤부터 실습이 들어가게되었지만 그렇게 배운지식으로는각 하이퍼파라미터들이무엇을 조절하는지, 지금 어떤 지표로 정확도를 뽑아내는지, 어떤걸 기점으로 이런 모델들을 사용하였는지에대한점들을 더 배웠으면 했다라고 학생들이 많이 생각하였습니다.

그리고 프로젝트 기간동안은 수업은 없이 자율성있는 프로젝트가 진행되었지만 그렇게 진행이 되자 오히려 수업을 원하는 학생들을 많이 생겼으며 프로젝트기간에도최소 2시간정도는 수업을 진행하는 것이 좋다 생각이 듭니다.

4. 자체 피드백 및 개선사항

1.(개선사항)

제가 생각하는 AI제어를 위한 기초지식들을 나열해보았습니다.

기간안에 이룰 수 있는 학습 계획을 명확히 하자는 것이 목표입니다.

수업의 흐름도 개선

(1) AI 학습에 사용되는 공통적인 부분

(2) 정통 ML

(3) DL

4. 자체 피드백 및 개선사항

(1) AI 학습에 사용되는 공통적인 부분

1. 행렬, 벡터, 고유벡터, 고유값
2. tensor
3. Feature scaling(정규화,표준화)
4. 경사하강법(Gradient Descent)
5. 러닝레이트(Learning Rate)
6. Gradient Descent 외 AI 최적화 알고리즘 (SGD, Momentum, RMSprop, Adam)
7. K-fold cross validation
8. Confusion matrix, F1-score, ROC곡선
9. 하이퍼파라미터최적화 (Grid search, Bayesian)

4. 자체 피드백 및 개선사항

(2) 정통 ML - Iris데이터셋 활용

1. Decision tree
2. Random forest
3. KNN
4. SVM
5. 부스팅 알고리즘, 그래디언트부스팅
6. Dimension ruduction , PCA(주성분 분석) , LDA(선형 판별 분석)
7. K-평균 군집화 , 계층적 군집 분석
8. DBSCAN
9. 이상치 탐지 알고리즘
10. 자연어 처리 (Bag of words), 문서 단어 행렬 (Document-term matrix), TF-IDF, 코사인 유사도, 텍스트 전처리, 워드클라우드

4. 자체 피드백 및 개선사항

(3) DL

1. 퍼셉트론, 뉴런 네트워크
2. 활성화 함수
3. 역전파
4. 교차 엔트로피 , Sofmax(다중클래스분류)
5. Word2Vec
6. CNN
7. Autoencoder
8. 변분 오토인코더(VAE,CVAE)
9. TensorFlow vs PyTorch (Keras를 활용한 인공신경망 구축), TensorFlow 실습 (MLP)

5. 예시

KNN coding

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
df = pd.read_excel("C:₩₩Users₩₩oea₩₩datasets₩₩iris-dataset.xlsx", header = 0)

df.head()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
data = np.array(df, dtype=np.float32)

x_data = data[:, 0:-1]

y_data = data[:, [-1]]

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size = 0.2)
```

KNN coding

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

knn = KNeighborsClassifier(n_neighbors = 5)
knn.fit(x_train, y_train)

y_train_pred = knn.predict(x_train)
y_test_pred = knn.predict(x_test)

train_acc = metrics.accuracy_score(y_train_pred, y_train)
test_acc = metrics.accuracy_score(y_test_pred, y_test)

print(f'Train Accuracy: {train_acc:.3f}')
print(f'Test Accuracy: {test_acc:.3f}')
```

<hyperparameter>

The number of neighbors (default = 5)

```
Train Accuracy: 0.983
Test Accuracy: 0.933
```

5. 예시

SVM coding

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
df = pd.read_excel("C:₩₩Users₩₩oea₩₩datasets₩₩iris-dataset.xlsx", header = 0)

df.head()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
data = np.array(df, dtype=np.float32)

x_data = data[:, 0:-1]

y_data = data[:, [-1]]

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size = 0.2)
```

SVM coding

```
from sklearn.svm import SVC
from sklearn import metrics
```

<hyperparameter>

```
svm = SVC(kernel='linear', C=10, gamma = 1)
svm.fit(x_train, y_train)

y_train_pred = svm.predict(x_train)
y_test_pred = svm.predict(x_test)

train_acc = metrics.accuracy_score(y_train_pred, y_train)
test_acc = metrics.accuracy_score(y_test_pred, y_test)

print(f'Train Accuracy: {train_acc:.3f}')
print(f'Test Accuracy: {test_acc:.3f}')
```

```
Train Accuracy: 0.975
Test Accuracy: 0.967
```