|  |  |  |  |
| --- | --- | --- | --- |
| **종합설계 프로젝트 회의록** | | | |
| 팀명 | 1팀 | | |
| 회의주제 | Kubernetes를 활용한 Hyper Parameter Tuning과 ML model 배포를 위한 MLOps플랫폼 개발 | | |
| 일시 | ○ 일 자 : 2022. 9. 8. (목)    ○ 시간 : 13:00~14:00 | | |
| 장소 | Google meet | | |
| 참석자 | 김석희, 박종민, 신시온, 이예림, 박보경 | 불참자 |  |
| 회의내용 | 1. 멘토님과의 킥오프 미팅 2. 목표 : 쿠버네티스와 도커를 다룰 수 있게 한다. – 학습하는 데에 시간이 꽤 걸릴테니, 스터디를 같이 진행을 하든 따로 하든 충분한 공부량이 필요. 3. 이번 서비스의 메인 역할 : 머신러닝을 배포하는 것.   쿠버네티스를 통해 머신러닝을 돌려 하이퍼 파라미터 튜닝을 통해 최적화 시켜주고 서빙까지 하는 작업이다.  쿠버네티스를 활용하면 여러 번 돌려 최적화 값을 찾는 과정을 한 번에 가능하다. 예를 들어100번 돌린다 치면 100개의 모델이 나오고 최적의 퍼포먼스를 보인 모델을 ML사용자가 배포를 통해 사용할 수 있게 한다.  ML사용자가 사용할 수 있는 방법   1. 소프트웨어 내에서 돌리는 방법 2. Rest api(웹)으로 올리는 방법 3. 웹 내에서 필요한 기능 4. 모델 입력 기능 5. 모델 학습 기능 6. 모델 배포 기능 7. 결과 획득(출력) 8. REST 요청 9. 전체적인 구성 필요   MLOps 구성요소가 여러가지가 있는데 일부가 하이퍼파라미터와 배포다.  텐서플로, 케라스로 머신러닝을 돌릴거고, 최종적으로는 KServer를 이용해서 배포를 하겠다 혹은 REST api를 통해 오픈하겠다 등 대략적인 틀에 대한 논의 필요.  코드를 직접 짜도 되지만 완성도 부문에서는 떨어질 수 있으니 대체적으로 오픈소스 서비스에 이미 만들어진 프레임워크가 있으니 그걸 이용하는 것이 안정성이나 퍼포먼스 측면에서 좋을 수 있다.   1. 쿠버네티스 환경   서버를 다 살 수 없으니까 서버를 여러 개 묶어서 사용하는 것 -> virtual machine 즉, 가상으로 서버를 만들어 낼 수 있다. 여기에 리눅스를 서리하고 이거를 쿠버네티스로 묶어버리면 환경이 완성된다.  위와같은 방법으로 하기 힘들기 때문에 AWS같은 클라우드를 쓰면 될거다. 인스턴스를 만들 수 있고, 서버를 생각하고 클러스터로 쓸 수 있다. AWS를 써보지 않았다면 가입해 무료계정을 받을 수 있다.  컴퓨터 사양이 16GB 이상이면 겨우 돌아갈 것이다. | | |
| 개인별  역할 분담 | 1. 김석희, 박보경  * 프론트 구현공부와 쿠버네티스 공부 * 웹 페이지에서 시스템 제어를 위한 전반적인 웹 프론트엔드  1. 신시온  * 백엔드 구현 공뷰와 쿠버네티스 공부 * 쿠버네티스 환경과 ML model 배포를 위한 연결부 부분 공부  1. 박종민, 이예림  * 쿠버네티스를 활용한 Hyper parameter Tuning과 모델 배포를 위한 엔진 개발에 대한 공부 | | |
| 진행할  업무 | 1. 쿠버네티스에 대한 기본적인 공부 2. 전체적인 흐름 3. 라이브러리 익히기 4. 클러스트링, CI/CD, 젠킨스, virtual box, virtual machine 학습 5. 1학기 동안 수행할 수행계획서 작성 필요 6. 어떤 툴을 쓸지 고민 필요   추천 – 깃허브, 마크다운 형식   1. 각자 맡은 분야에 대한 개인적인 공부 | | |
| 회의 사진 |  | | |
| 다음 회의 때 논의해야할 사항 | 1. 쿠버네티스 공부 후 관련 스터디 진행 2. 전체적인 구성 및 기능의 세부적인 사항 조정 필요   MLOps 구성요소가 여러가지가 있는데 일부가 하이퍼파라미터와 배포다. 우리가 직접 짜도 되지만, 이미 만들어진 프레임워크가 존재한다.  텐서플로, 케라스로 머신러닝을 돌릴거고, 최종적으로는 KServer를 이용해서 배포를 하겠다 혹은 REST api를 통해 오픈하겠다 등 대략적인 틀에 대한 논의 필요. 코드 또한 직접 짜도 되지만 완성도 부문에서는 떨어질 수 있으니 대체적으로 오픈소스 서비스에 이미 만들어진 프레임워크가 있으니 그걸 이용하는 것이 안정성이나 퍼포먼스 측면에서 좋을 수 있다. | | |