[AI.SW개론-F]

실증적 관점에서의 오픈소스 Jupyter Notebook 분석 및 Apache Zeppelin과의 비교

오픈소스 소프트웨어 탐색 보고서: 데이터 분석 Jupyter Notebook 프로젝트 분석

202578218 정지은 AISW계열

목차

Ⅰ. Jupyter Notebook 개요

Jupyter Notebook 프로젝트는 Project Jupyter의 일부로, 오픈소스 소프트웨어 중 하나이다. 계산 노트북 형식을 기반으로 하며, 데이터를 탐색 및 시각화하고 아이디어를 다른 사람들과 공유할 수 있는 서비스를 제공한다. Notebook은 기존 콘솔 기반의 인터랙티브 컴퓨팅 방식에서 더 나아가 웹 기반 애플리케이션으로 발전한 것으로, 코드 개발, 문서화, 실행, 결과 전달까지 계산 과정을 통합적으로 다룰 수 있도록 지원한다.

01. 개발 목적

Jupyter Notebook 프로젝트는 이러한 오픈소스 공유의 장을 제공함으로써 AI 시대를 선도하고자 하는 강한 목적성을 띤다. 자사 기술을 오픈소스로 공개함으로써 자사의 기술을 업계의 표준으로 만들고 기술 트렌드의 주도권을 확보하여 AI 생태계를 주도할 수 있다. 생태계를 선점하면 관련 서비스 또는 인프라에서 간접 이익을 얻을 수 있고 타사보다 유리한 입지를 얻을 수 있다. 자사 기술을 오픈 소스로 공개하면 외부 개발자들이 자발적으로 오류 지적 및 수정, 기능 추가 등의 아이디어를 제공함으로써 개발 비용을 절감할 수 있다. 수요자 관점에서의 과감하고 다양한 아이디어와 테스트 커버리지의 증가는 장기적으로 품질 향상으로 이어질 수 있다. 오픈 소스를 공개하는 것은 자사의 기술뿐만 아니라 브랜딩에도 긍정적인 영향을 미친다. 기업의 기술력과 개발 문화를 공개하며 보다 투명한 이미지를 확보할 수 있고, 소스 코드를 공개하면 보안, 개발 윤리 등에서 신뢰성을 제고할 수 있다. 오픈 소스 전략을 기술 공개로써 기업에 손해인 것처럼 보이지만, 장기적으로 전략적 이익을 취하는 셈이다.

02. 라이선스

Jupyter Notebook은 비영리 100% 오픈소스로, 모든 프로그래밍 언어에서 Interactive Data Science를 지원하기 위해 설계되어 누구나 자유롭게 사용할 수 있도록 모든 소스는 수정된 BSD 라이선스 조건 하에 배포된다. 그러나 프로젝트와 직접적으로 관련된 온라인 및 오프라인상의 모든 상호작용 및 커뮤니케이션은 Jupyter 행동 강령(Code of Conduct)의 적용을 받는다. 이 행동 강령은 다양한 배경을 가진 사용자 및 기여자들이 존중과 안전 속에서 프로젝트에 참여할 수 있도록 기대 기준을 설정한 것이므로 엄격히 유지될 것을 권고한다.

Ⅱ. 주요 기능 및 구성

Jupyter Notebook이 직접 공개한 소개에 따르면 크게 코드 자동 완성, 문법 강조, 들여쓰기 지원, 브라우저 내 코드 실행 및 결과 표시, HTML, LaTeX, 이미지(PNG, SVG) 등 리치 미디어 지원, Markdown 언어를 통한 설명 문서 작성, 수학 수식 입력 시 LaTeX 및 Math Jax 기반 실시간 렌더링의 주요 기능을 가지고 있다고 한다. 기본적으로 노트북 문서로써 작동하며 코드 입력/출력 외에도 설명 텍스트를 포함할 수 있고, 계산 기록을 완전하게 저장한다. JSON 형식으로 되어있으며 Git 등의 버전으로도 관리할 수 있고, HTML, LaTeX, PDF, 슬라이드 등 다양한 포맷으로 내보내기가 가능하다. 또한 nbviewer로 공개된 Notebook 문서는 웹페이지처럼 공유할 수 있다. 일반적으로는 로컬에서 실행되지만, 서버나 클라우드로의 데이터 자동 전송을 허용하지 않고, 원격 서버를 사용하는 경우 민감한 데이터는 IT 보안 담당자와의 논의를 필수로 하여 보안을 강하게 지키고 있다.

01. 노트북 인터페이스

노트북 인터페이스는 노트북 이름, 메뉴 바, 툴바, 코드 셀로 구성되어 있다. 노트북 이름은 상단에 표시되며, 클릭하여 이름을 변경할 수 있다. 메뉴 바는 다양한 조작 옵션을 제공하고 툴바는 자주 사용하는 기능을 아이콘으로 제공한다. 코드 셀은 기본 입력 단위로서 Shift + Enter를 통해 코드를 한 번에 실행시킬 수 있다. 셀 단위로 문제를 해결하고 결과를 확인할 수 있으며 길게 실행되는 코드는 Interrupt와 Restart로 중단하고 재시작을 할 수 있다. Ctrl+F로 텍스트 검색 기능을 제공하고 있지만 브라우저 기본 검색은 일부 콘텐츠만 검색된다. IPython 커널과 matplotlib을 연동하여 그래프를 셀 내 출력할 수 있다. 또한 높은 브라우저 호환성을 가지고 있지만 자사 공식 입장으로는 Chrome, Safari, Firefox의 최신 버전 권장한다. Opera 및 Edge도 대부분 동작하지만 오류 시에는 앞서 권장한 브라우저를 사용하기를 권장하고 Safari + HTTPS + 비 신뢰 인증서 조합은 웹 소켓 오류로 동작하지 않는다고 한다.

02. 주요 기술 스택

Jupyter Notebook는 다양한 기술 스택으로 이루어져 있으며, 대체적으로 파이썬을 핵심 언어로서, 데이터를 분석하고 시각화할 수 있는 개발 환경을 제공한다. IPython은 인터랙티브 환경을 제공하는 핵심 기술로서 파이썬 코드를 실행하고 바로 결과를 확인할 수 있도록 환경을 구축하는 데에 쓰인다. 기본적으로 웹 브라우저를 통해 접근하는 방식을 취하고 있으며 코드를 작성하고 실행한 결과를 시각화하는 데에 사용되고 있다. 웹 브라우저에서 만든 Notebook 파일은 JSON(JavaScript Object Notation)을 통해 저장되고, CSS(Cascading Style Sheets)를 통해 Notebook의 스타일을 설정한다. UI를 구성할 때는 HTML(HyperText Markup Language)가 사용되며, JavaScript를 사용하여 UI 동적 조작을 시도한다. Google Colab는 클라우드 기반의 Jupyter Notebook 환경으로, Google 드라이브와 연동하여 저장할 수 있다. Google Cloud Platform 또한 Google Colab와 같이 클라우드를 기반으로 하며 환경을 제공한다. 이때 Docker는 Jupyter Notebook 환경을 쉽게 배포하고 공유할 수 있도록 도와준다. Linux는 Jupyter Notebook을 실행하는 데 필요한 운영 체제로써 작동하고 있다.

03. 오픈 사이언스 도구로의 활용

Randles, Golshan, Pasquetto and Borgman(2017)에 따르면, “NASA가 개발한 천문학 데이터 시스템(Astrophysics Data System)을 통해 Jupyter Notebook을 언급한 논문 91편을 분석”한 결과, “Jupyter Notebook은 오픈소스 기반의 브라우저용 가상 실험 노트북으로, 연구 과정의 워크플로우, 코드, 데이터, 시각화를 통합적으로 지원한다. 이 도구는 기계와 사람이 모두 읽을 수 있는 형식을 제공함으로써 상호운용성과 학술적 소통을 용이하게 한다. 노트북은 온라인 저장소에 존재할 수 있으며, 데이터셋, 코드, 연구 방법 문서, 워크플로우, 논문 등 다양한 연구

객체와 연결될 수 있다.”고 밝혔다. 또한 “Jupyter Notebook은 오픈 사이언스를 위한 유망한 도구이며, 연구 객체를 담은 오픈 저장소들과 함께 전체 오픈 사이언스 생태계의 일부를 구성하고 있다. 개별 도구의 연구는 더 큰 과학 커뮤니케이션 문제의 해법에 기여할 수 있다.”라고 언급하며 Jupyter Notebook의 오픈 소스 도구로써의 실증적 가치를 긍정적으로 평가하였다.

더 나아가 “‘이상적인’ Jupyter Notebook은 어떤 정보를 포함해야 할까? 소프트웨어와 도구의 안정성은 어떻게 보장할 수 있을까? 연구 객체의 지속성과 연계를 위한 기술적 수단(ResourceSync 등)은 어떻게 활용해야 할까?”라는 물음을 통해 Jupyter Notebook뿐만 아니라 모든 오픈소스 기업이 앞으로의 시대에 AI시장에서 살아남기 위해, 선도하기 위해 필수적으로 혁신해나가야할 문제를 시사했다. 이 질문은 단지 기업에게만 해당하지 않는다. 공급자와 소비자 모두 고민해야 하는 문제로써 더 빠른 기술 개발과 더 안정적인 보안을 위해서 ‘이상적인 상태’를 위한 내부 혁신의 중요성이 더욱 증가할 것으로 보인다. 소비자는 기업이 선두하는 AI 산업에 그대로 이끌리지 않고 진보된 기술의 배경을 이해하고 실생활에 잘 활용하기 위해 이 질문에 대해 고민해야 한다.

Ⅲ. 해결해야 할 과제

앞서 설명한 것과 같이 접근성이 좋고 다양한 기능을 포함하고 있는 장점에도 불구하고 Pimentel, Murta, Braganholo and Freire(2019)는 “Jupyter Notebook이 나쁜 습관을 조장하고, 예측 불가능한 실행 결과를 유도하며, 재현 가능성을 저해한다는 비판이 제기되고 있다. 주요 문제로는 숨겨진 상태, 비순차적 실행, 일관되지 않은 명명, 버전 정보 누락, 의존성 기록 부족 등이 있으며, 이에 따라 노트북의 결과를 재현하기 어렵거나 불가능한 경우가 많다. 이러한 문제는 과학 계산 소프트웨어에서 소프트웨어 공학적 모범 사례가 부족할 때 발생하는 부작용들과 일치한다.”고 지적했다.

더 나아가 재현 가능성을 높이기 위한 모범 사례와 해결해야 할 과제를 제시했다. 먼저 라이브러리 버전 정보까지 명시한 경우는 5% 미만이라는 문제를 지적하며 이에 대해 “실행 환경에 필요한 모든 라이브러리 및 그 버전을 명확히 기록하고, 가능한 경우 requirements.txt 또는 environment.yml 파일로 제공해야 한다.”고 조언했다. 또한 “위에서 아래로 순차적으로 실행될 수 있도록 설계하고, 코드 셀 실행 순서를 유지해야 한다. 마크다운 셀을 적극 활용하여 코드의 목적, 입력값, 출력 해석 등을 설명함으로써, 사용자의 이해도를 높이고 활용도를 확장해야 한다. 실행 결과(출력 셀)를 포함하되, 결과가 최신인지 주기적으로 재확인해야 하며, 무작위성이 포함된 경우 seed 설정을 명시하는 것이 바람직하다. 노트북 파일의 버전을 명확히 하고, 커밋 시 변경된 셀만 반영하도록 구조를 단순화한다.”고 언급했다.

이 연구에 따르면 아직 해결되지 않은 과제로는 실행 환경 이식성 문제, 사용자 습관의 문제, 자동 분석 도구의 한계를 지적했다. 다양한 운영체제나 하드웨어 환경에서 동일한 결과를 보장하는 것은 여전히 어렵지만, 이는 종속성과 네트워크 접근 문제, 데이터 경로 설정 등과 밀접히 연관되는 문제라고 언급했다. 또한 “문서화 생략, 임의 실행 순서, 불명확한 변수 정의 등은 기술적인 해결책 이전에 사용자 교육과 문화적 변화가 필요한 부분이다. 정적 분석 도구로는 모든 실행 오류를 포착하거나 결과 일관성을 보장하기 어렵다. 동적 실행 실험과의 결합이 필수적이다.”고 설명했다. 이 문제들을 종합적 관점에서 바라보았을 때, 서비스 자체를 더 체계적으로, 안정적으로 수정할 필요가 강하게 있어 보인다. 사용자 습관에 관한 부분은 꾸준한 사용자 교육에 따른 문화적 변화가 따른다면 장기적으로 문제가 되지는 않지만, 서비스 자체의 오류, 일관성 문제는 더 많은 시행착오를 통해 필히 개선되어야 할 것이다.

Ⅳ. Apache Zeppelin과의 비교 분석

Apache Zeppelin은 Jupyter Notebook과 같이 데이터 분석을 위한 오픈소스 프로젝트 중 하나이다. Apache Software Foundation 산하의 서비스로서, 데이터 탐색, 시각화, 협업에 중심을 두어 운영되고 있다. 특히, Spark와의 콜라보를 통해 기존의 데이터 분석의 불편함을 자율성이 높은 Web기반의 Notebook 형식을 사용하여 해결을 시도했다. Jupyter Notebook과 마찬가지로 100% 오픈소스를 제공하며 데이터 시각화에 장점을 가지고 있다.

Apache Zeppelin은 기본적으로 Scala, Python (via PySpark), SQL, R, Markdown, Shell 등 다양한 언어를 제공하며 다중 언어를 한 문서 내에서 동시에 실행할 수 있다. 또한 각 언어별로 해석기를 통해 독립적으로 실행 환경을 구성할 수 있다. 클라우드와 추출된 데이터베이스로 온라인에서 작업할 수 있고 다양한 데이터베이스 형식과 시각화 기능을 통해 방대한 양의 데이터를 분석할 수 있다. 특히 Apache Zeppelin이 제공하는 SQL 문은 작업 중 의사결정에 도움을 줄 수 있다. 커뮤니티 활성화 정도가 높은 편이라 유지 보수가 비교적 빠르게 이루어진다는 장점도 있다. 그러나 UI가 Jupyter Notebook에 비해 다소 직관적이지 못하고 복잡하여 초보자가 사용자가 빠르게 적응하기 어렵다. 또한 Python을 중심으로 다양한 언어를 제공하여 데이터 사이언티스트 또는 학습자를 주 수요층으로 하는 Jupyter Notebook과 달리, Apache Zeppelin은 Spark 중심 사용자를 중심으로 설계되어 언어 중심 개발자에게는 적합하지 않을 수 있다.

Ⅴ. 결론

AI 시대를 선점하고 선도하기 위하여 치열한 오픈소스 경쟁이 벌어지는 가운데, 대규모 데이터 분석을 위한 오픈소스 소프트웨어 Jupyter Notebook은 주목할 만하다. 비영리 100% 오픈소스로 제공하며 동시에 개발자와 학습자 모두에게 기회를 제공하는 오픈소스 문화는 앞으로의 건강한 개발 경쟁 문화에 있어 긍정적인 영향을 미칠 것이다. 특히 데이터 분석 분야와 같이 방대한 양의 데이터를 다루고 시각화하는 분야의 오픈소스 생태계를 주도하고 있는 Jupyter Notebook는 데이터 분석 분야를 넘어서 다양한 오픈소스 소프트웨어 산업을 선도할 수 있을 것이다.

※참고 문헌

Bernadette M. Randles, Irene V. Pasquetto, Milena S. Golshan and Christine L. Borgman. *Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study*. Department of Information Studies University of California, Los Angeles USA. 2017. pp.1-2.

Jo˜ao Felipe Pimentel, Leonardo Murta, Vanessa Braganholo and Juliana Freire. *Jupyter Notebooks—a publishing format for reproducible computational workflows*. Universidade Federal Fluminense, New York University. 2019. pp. 1, 9-10.