📋 Day 2 완료 보고서: 프로젝트 구조 및 Git 구조 문서화

🎯 Day 2 달성 목표 및 결과

목표	상태	완료도	비고
표준 프로젝트 구조 생성	✅ 완료	100%	업계 표준 폴더 구조 적용
Git 저장소 초기화	✅ 완료	100%	로컬 및 원격 저장소 연결
기본 문서 파일 생성	✅ 완료	100%	README, requirements, .gitignore 등
GitHub 원격 연결	✅ 완료	100%	원격 저장소 연동 및 푸시
README.md 전문화	✅ 완료	100%	포트폴리오급 문서 작성

📂 최종 프로젝트 구조

🏗️ 디렉토리 구조 (Full Tree)

```
D:\dataAnalystBook\my_workspace\movie_review\
│
├── 📊 data/                          # 데이터 저장소
│   ├── raw/                          # 원본 데이터 (수정 금지)
│   │   ├── ratings_train.txt         # NSMC 훈련 데이터 (다운로드 예정)
│   │   └── ratings_test.txt          # NSMC 테스트 데이터 (다운로드 예정)
│   ├── processed/                    # 전처리된 데이터
│   │   ├── clean_train.csv           # 정제된 훈련 데이터 (생성 예정)
│   │   ├── clean_test.csv            # 정제된 테스트 데이터 (생성 예정)
│   │   └── tokenized_data.pkl        # 토큰화된 데이터 (생성 예정)
│   ├── interim/                      # 중간 처리 결과
│   │   ├── exploratory_analysis.pkl  # 탐색 분석 결과 (생성 예정)
│   │   └── feature_analysis.pkl      # 특성 분석 결과 (생성 예정)
│   └── external/                     # 외부 데이터
│       ├── korean_stopwords.txt      # 한국어 불용어 사전 (추가 예정)
│       └── sentiment_lexicon.csv     # 감정 어휘 사전 (추가 예정)
│
├── 📓 notebooks/                     # Jupyter 노트북들
│   ├── 01_data_exploration/          # 데이터 탐색 (Phase 1)
│   │   ├── data_overview.ipynb       # 데이터 기본 정보 분석 (생성 예정)
│   │   ├── text_analysis.ipynb       # 텍스트 특성 분석 (생성 예정)
│   │   └── visualization.ipynb       # 시각화 및 인사이트 (생성 예정)
│   ├── 02_data_preprocessing/        # 데이터 전처리 (Phase 2)
│   │   ├── text_cleaning.ipynb       # 텍스트 정제 (생성 예정)
│   │   ├── tokenization.ipynb        # 토큰화 및 정규화 (생성 예정)
│   │   └── data_split.ipynb          # 데이터 분할 (생성 예정)
│   ├── 03_feature_engineering/       # 특성 공학 (Phase 3)
```

```
│   │   ├── bow_tfidf.ipynb              # Bag of Words, TF-IDF (생성 예정)
│   │   ├── word_embeddings.ipynb        # Word2Vec, FastText (생성 예정)
│   │   └── feature_selection.ipynb      # 특성 선택 및 차원 축소 (생성 예정)
│   ├── 04_modeling/                     # 모델링 (Phase 4)
│   │   ├── baseline_models.ipynb        # 기본 ML 모델들 (생성 예정)
│   │   ├── deep_learning.ipynb          # LSTM, CNN 모델 (생성 예정)
│   │   ├── transformer_models.ipynb     # KoBERT, KoELECTRA (생성 예정)
│   │   └── model_comparison.ipynb       # 모델 성능 비교 (생성 예정)
│   └── 05_evaluation/                   # 모델 평가 (Phase 5)
│       ├── performance_analysis.ipynb   # 성능 지표 분석 (생성 예정)
│       ├── error_analysis.ipynb         # 오류 분석 및 개선 (생성 예정)
│       └── model_interpretation.ipynb   # 모델 해석 및 시각화 (생성 예정)
│
├── 🐍 src/                              # 재사용 가능한 소스 코드
│   ├── __init__.py                      # ✅ Python 패키지 설정
│   ├── data/                            # 데이터 처리 모듈
│   │   ├── __init__.py                  # ✅ 서브패키지 설정
│   │   ├── loader.py                    # 데이터 로딩 함수 (생성 예정)
│   │   ├── preprocessor.py              # 전처리 함수 (생성 예정)
│   │   └── validator.py                 # 데이터 검증 함수 (생성 예정)
│   ├── features/                        # 특성 추출 모듈
│   │   ├── __init__.py                  # ✅ 서브패키지 설정
│   │   ├── text_features.py             # 텍스트 특성 추출 (생성 예정)
│   │   ├── embeddings.py                # 임베딩 생성 (생성 예정)
│   │   └── feature_selector.py          # 특성 선택 (생성 예정)
│   ├── models/                          # 모델 관련 모듈
│   │   ├── __init__.py                  # ✅ 서브패키지 설정
│   │   ├── baseline.py                  # 기본 ML 모델 (생성 예정)
│   │   ├── deep_learning.py             # 딥러닝 모델 (생성 예정)
│   │   ├── transformers.py              # Transformer 모델 (생성 예정)
│   │   └── evaluator.py                 # 모델 평가 (생성 예정)
│   ├── utils/                           # 유틸리티 함수들
│   │   ├── __init__.py                  # ✅ 서브패키지 설정
│   │   ├── helpers.py                   # 공통 도우미 함수 (생성 예정)
│   │   ├── config_loader.py             # 설정 로더 (생성 예정)
│   │   └── logger.py                    # 로깅 유틸리티 (생성 예정)
│   └── visualization/                   # 시각화 모듈
│       ├── __init__.py                  # ✅ 서브패키지 설정
│       ├── plots.py                     # 기본 플롯 함수 (생성 예정)
│       ├── model_viz.py                 # 모델 시각화 (생성 예정)
│       └── report_generator.py          # 보고서 생성 (생성 예정)
│
├── 😈 models/                           # 학습된 모델 저장소
│   ├── trained/                         # 최종 학습된 모델들
│   │   ├── logistic_regression_v1.pkl   # 로지스틱 회귀 모델 (생성 예정)
│   │   ├── svm_model_v1.pkl             # SVM 모델 (생성 예정)
│   │   ├── lstm_model_v1.h5             # LSTM 모델 (생성 예정)
```

```
│   │   └── kobert_model_v1/              # KoBERT 모델 (생성 예정)
│   ├── experiments/                      # 실험용 모델들
│   │   ├── exp_001_baseline/             # 실험 001: 베이스라인 (생성 예정)
│   │   ├── exp_002_feature_eng/          # 실험 002: 특성 공학 (생성 예정)
│   │   └── exp_003_hyperparams/          # 실험 003: 하이퍼파라미터 (생성 예정)
│   └── production/                       # 운영용 모델들
│       └── final_model_v1/               # 최종 운영 모델 (생성 예정)
│
├── 📊 reports/                           # 보고서 및 시각화
│   ├── figures/                          # 그래프, 차트 이미지들
│   │   ├── data_distribution.png         # 데이터 분포 시각화 (생성 예정)
│   │   ├── model_performance.png         # 모델 성능 비교 (생성 예정)
│   │   ├── confusion_matrix.png          # 혼동 행렬 (생성 예정)
│   │   └── feature_importance.png        # 특성 중요도 (생성 예정)
│   └── documents/                        # 분석 보고서, 발표자료
│       ├── project_summary.md            # 프로젝트 요약 보고서 (생성 예정)
│       ├── technical_report.pdf          # 기술 보고서 (생성 예정)
│       └── presentation.pptx             # 발표 자료 (생성 예정)
│
├── 📈 results/                           # 분석 결과
│   ├── predictions/                      # 예측 결과들
│   │   ├── test_predictions.csv          # 테스트 예측 결과 (생성 예정)
│   │   ├── validation_predictions.csv    # 검증 예측 결과 (생성 예정)
│   │   └── error_cases.csv               # 오류 사례 분석 (생성 예정)
│   └── metrics/                          # 성능 지표들
│       ├── model_metrics.json            # 모델별 성능 지표 (생성 예정)
│       ├── experiment_results.csv        # 실험 결과 비교 (생성 예정)
│       └── performance_history.json      # 성능 변화 추적 (생성 예정)
│
├── ⚙️ config/                            # 설정 파일들
│   ├── config.yaml                       # ✅ 프로젝트 메인 설정
│   ├── model_configs/                    # 모델별 설정 (생성 예정)
│   │   ├── baseline_config.yaml          # 기본 모델 설정 (생성 예정)
│   │   ├── deep_learning_config.yaml     # 딥러닝 설정 (생성 예정)
│   │   └── transformer_config.yaml       # Transformer 설정 (생성 예정)
│   └── environment/                      # 환경별 설정 (생성 예정)
│       ├── development.yaml              # 개발 환경 설정 (생성 예정)
│       └── production.yaml               # 운영 환경 설정 (생성 예정)
│
├── 📚 docs/                              # 추가 문서들
│   ├── api_documentation.md              # API 문서 (생성 예정)
│   ├── model_documentation.md            # 모델 문서 (생성 예정)
│   └── troubleshooting.md                # 문제 해결 가이드 (생성 예정)
│
├── 🧪 tests/                             # 테스트 코드들
│   ├── __init__.py                       # 테스트 패키지 설정 (생성 예정)
│   ├── test_data_processing.py           # 데이터 처리 테스트 (생성 예정)
```

```
│   ├── test_models.py          # 모델 테스트 (생성 예정)
│   └── test_utils.py           # 유틸리티 테스트 (생성 예정)
│
├── 📋 README.md                # ✅ 프로젝트 소개 및 사용법
├── 📦 requirements.txt         # ✅ Python 패키지 의존성
├── 📦 requirements_conda.txt   # ✅ Conda 환경 정보
├── 🚫 .gitignore               # ✅ Git 무시 파일 목록
├── 📄 LICENSE                  # 라이선스 파일 (생성 예정)
├── 🔧 setup.py                 # 패키지 설치 스크립트 (생성 예정)
└── 📝 .git/                    # ✅ Git 저장소 정보 (숨김 폴더)
```

🔧 Git 저장소 구조 및 상태

📊 현재 Git 상태

```bash
# 저장소 기본 정보
Repository: nsmc-korean-sentiment-analysis
Local Path: D:\dataAnalystBook\my_workspace\movie_review
Remote URL: https://github.com/Jaehyeok-Lee-pastel/nsmc-korean-sentiment-analysis.git
Default Branch: main
```

🌿 브랜치 구조

```
main (현재 브랜치)
├── [커밋 1] feat: NSMC 프로젝트 초기 구조 및 환경 설정 완료
└── [커밋 2] docs: README.md 전면 개선
```

향후 브랜치 전략:

```
main
├── develop (개발 브랜치, 생성 예정)
│   ├── feature/data-exploration (데이터 탐색)
│   ├── feature/preprocessing (전처리)
│   ├── feature/baseline-models (기본 모델)
│   ├── feature/deep-learning (딥러닝)
│   └── feature/transformers (Transformer)
└── hotfix/* (긴급 수정용)
```

📜 커밋 히스토리

커밋 번호	메시지	날짜	변경 파일 수
[latest]	docs: README.md 전면 개선	2025-01-XX	1
[initial]	feat: NSMC 프로젝트 초기 구조 및 환경 설정 완료	2025-01-XX	30+

🔗 원격 저장소 연결 상태

bash

```
origin  https://github.com/Jaehyeok-Lee-pastel/nsmc-korean-sentiment-analysis.git (fetch)
origin  https://github.com/Jaehyeok-Lee-pastel/nsmc-korean-sentiment-analysis.git (push)

Tracking: main -> origin/main ✅
```

📝 주요 설정 파일들

🚫 .gitignore 구성

bash

```
# Python 관련
__pycache__/
*.py[cod]
*.so
*.egg-info/

# Jupyter Notebook
.ipynb_checkpoints

# 대용량 데이터 파일들
data/raw/*.txt
data/processed/*.csv
data/interim/*.pkl

# 학습된 모델들
models/trained/*.pkl
models/trained/*.h5
models/experiments/*

# 환경 파일들
.env
.venv
env/
venv/

# IDE 파일들
.vscode/
.idea/
*.swp
*.swo

# OS 파일들
.DS_Store
Thumbs.db
```

⚙ config.yaml 구성

yaml

```yaml
# 데이터 설정
data:
  raw_data_path: 'data/raw/'
  processed_data_path: 'data/processed/'
  train_file: 'ratings_train.txt'
  test_file: 'ratings_test.txt'

# 모델 설정
model:
  random_state: 42
  test_size: 0.2
  target_accuracy: 0.85

# 실험 설정
experiment:
  model_save_path: 'models/trained/'
  results_save_path: 'results/'
  figures_save_path: 'reports/figures/'
```

📦 requirements.txt 패키지 목록

```
pandas==2.x.x
numpy==1.x.x
matplotlib==3.x.x
seaborn==0.x.x
jupyter==1.x.x
requests==2.x.x

# 추가 예정 패키지들:
# konlpy (한국어 NLP)
# scikit-learn (ML)
# tensorflow (딥러닝)
# transformers (Transformer 모델)
# streamlit (웹 데모)
```

🎯 프로젝트 구조의 설계 원칙

📋 폴더별 역할 및 설계 철학

1. 데이터 관리 (data/)

- **raw/**: 원본 데이터 보존 (읽기 전용)
- **processed/**: 재사용 가능한 정제된 데이터

- **interim/**: 실험용 중간 결과물
- **external/**: 외부에서 가져온 보조 데이터

설계 원칙: 데이터 파이프라인의 각 단계를 명확히 분리

2. 코드 관리 (src/)

- **data/**: 데이터 입출력 및 변환 로직
- **features/**: 특성 추출 및 엔지니어링
- **models/**: 모델 정의 및 학습 로직
- **utils/**: 공통 유틸리티 함수
- **visualization/**: 시각화 전용 함수

설계 원칙: 단일 책임 원칙 (Single Responsibility Principle)

3. 실험 관리 (notebooks/)

- **단계별 구성**: 01~05로 분석 과정 체계화
- **목적별 분리**: 탐색, 전처리, 모델링, 평가 분리
- **재현성**: 각 노트북이 독립적으로 실행 가능

설계 원칙: 실험의 재현성과 추적성 확보

4. 결과물 관리 (models/, reports/, results/)

- **models/**: 버전별 모델 관리
- **reports/**: 시각화 및 문서 결과물
- **results/**: 수치적 결과 및 예측값

설계 원칙: 결과물의 체계적 보관 및 비교

🔄 Git 워크플로우

📝 일상적인 개발 사이클

bash

```
# 1. 작업 시작 전 최신 상태로 업데이트
git pull origin main

# 2. 새로운 기능 브랜치 생성 (필요시)
git checkout -b feature/data-exploration

# 3. 작업 진행 후 변경사항 확인
git status
git diff

# 4. 변경사항 스테이징
git add .

# 5. 의미 있는 커밋 메시지로 커밋
git commit -m "feat: NSMC 데이터 기본 탐색 완료

- data/raw/ 폴더에 NSMC 데이터 다운로드
- 01_data_exploration/data_overview.ipynb 생성
- 기본 통계 분석 및 데이터 분포 확인
- 한국어 텍스트 특성 파악 완료"

# 6. GitHub에 푸시
git push origin feature/data-exploration

# 7. 기능 완료 후 main 브랜치로 병합
git checkout main
git merge feature/data-exploration
git push origin main
```

🏷️ 커밋 메시지 컨벤션

```
bash

feat: 새로운 기능 추가
fix: 버그 수정
docs: 문서 수정
style: 코드 스타일 변경 (포맷팅, 세미콜론 등)
refactor: 코드 리팩토링
test: 테스트 코드 추가
chore: 빌드 관련, 패키지 매니저 설정 등
```

예시:

- `feat: 한국어 텍스트 전처리 파이프라인 구현`
- `fix: 토큰화 과정에서 발생하는 인코딩 오류 수정`

- `docs: 모델 성능 비교 결과 README 업데이트`

✅ Day 2 완료 체크리스트

🎯 구조 설계 완료 항목

- [x] 📂 **프로젝트 폴더 구조**: 업계 표준 기반 체계적 구성
- [x] 🐍 **Python 패키지 구조**: src/ 폴더를 모듈로 사용 가능
- [x] 📓 **노트북 구조**: 01~05 단계별 분석 과정 체계화
- [x] 🔧 **Git 저장소**: 로컬 및 원격 저장소 완벽 연동
- [x] 📝 **문서화**: README, requirements, .gitignore, config 완성
- [x] ⚙️ **설정 파일**: 프로젝트 설정의 체계적 관리

🚀 환경 구축 완료 항목

- [x] 🌍 **가상환경**: nsmc_sentiment (Python 3.9)
- [x] 📦 **패키지 관리**: requirements.txt 기반 의존성 관리
- [x] 🔗 **GitHub 연동**: 원격 저장소 연결 및 첫 푸시 완료
- [x] 📋 **README 전문화**: 포트폴리오급 프로젝트 소개

🎉 달성한 핵심 가치

1. **전문성**: 업계 표준에 맞는 프로젝트 구조
2. **확장성**: 새로운 기능 추가가 용이한 구조
3. **재현성**: 다른 환경에서도 동일하게 구현 가능
4. **협업성**: 팀 프로젝트로 확장 가능한 기반
5. **포트폴리오**: 채용 담당자에게 어필 가능한 완성도

🚀 Day 3 준비 상태

📊 준비된 개발 환경

1. ✅ **안정적인 작업 환경**: 독립적 가상환경 + 체계적 프로젝트 구조
2. ✅ **버전 관리 시스템**: Git을 통한 안전한 실험 및 백업
3. ✅ **문서화 기반**: 진행 상황을 체계적으로 기록할 수 있는 구조
4. ✅ **온라인 포트폴리오**: GitHub을 통한 실시간 프로젝트 공유

🎯 Day 3 예고: NSMC 데이터 탐색

목표: 한국어 영화 리뷰 데이터의 특성을 파악하고 분석 방향 설정

주요 작업:

1. NSMC 데이터셋 다운로드
2. 첫 번째 노트북 작성: `01_data_exploration/data_overview.ipynb`
3. 데이터 기본 통계 및 분포 분석
4. 한국어 텍스트 특성 탐색
5. 시각화를 통한 인사이트 도출

기대 효과:

- 데이터 이해를 바탕으로 한 효과적인 전처리 전략 수립
- 한국어 NLP 특성을 고려한 모델 선택 방향 설정
- 첫 번째 단계 완료를 통한 프로젝트 추진력 확보

🎉 Day 2가 성공적으로 완료되었습니다!

체계적이고 전문적인 프로젝트 기반이 완성되어, 이제 안전하고 효율적인 환경에서 본격적인 데이터 분석을 시작할 수 있습니다.