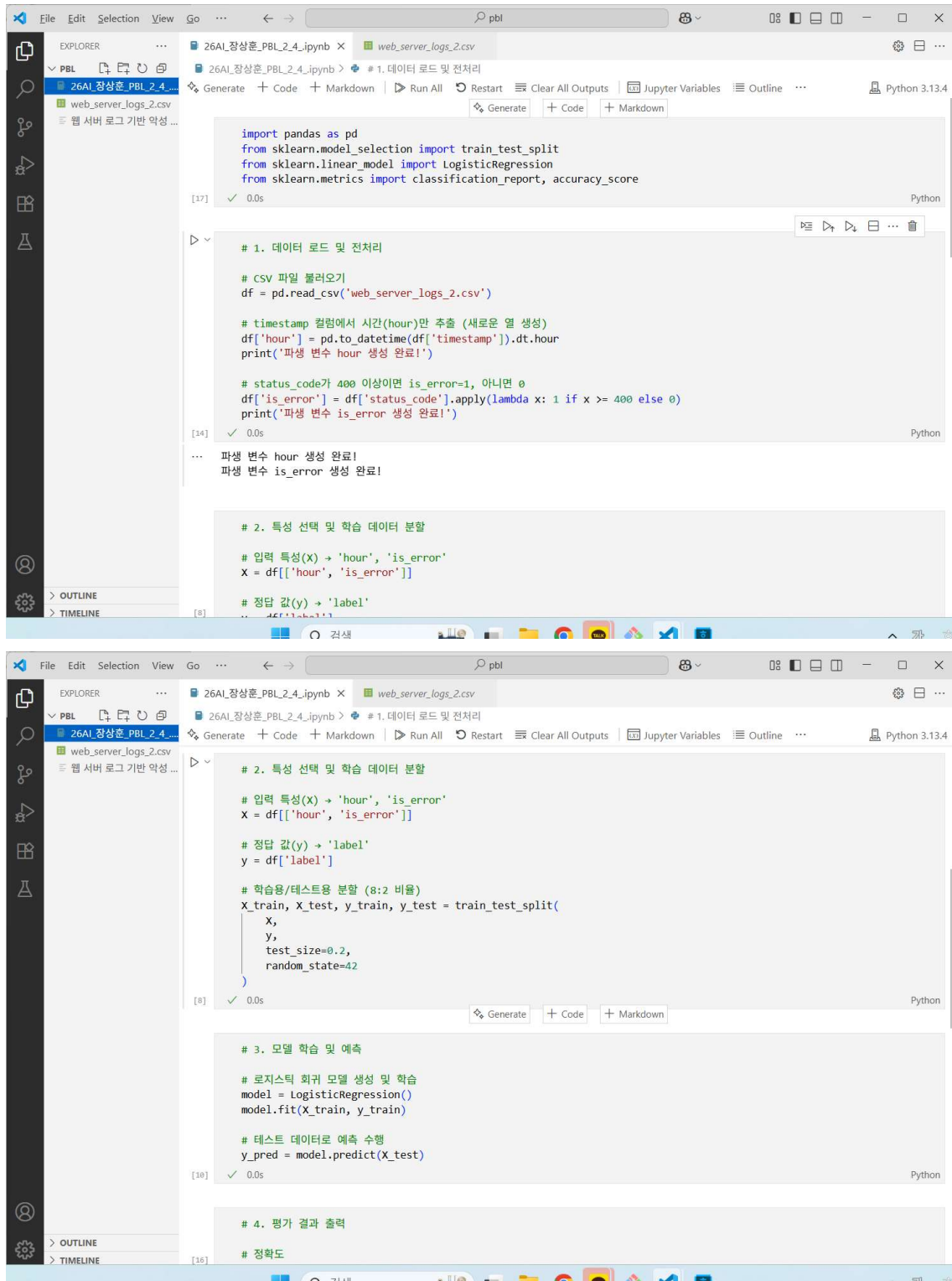


웹 서버 로그 기반 악성 요청 탐지를 위한 분류 모델 구축

26기_장상훈

프로그램 실행 결과



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# 1. 데이터 로드 및 전처리

# csv 파일 불러오기
df = pd.read_csv('web_server_logs_2.csv')

# timestamp 컬럼에서 시간(hour)만 추출 (새로운 열 생성)
df['hour'] = pd.to_datetime(df['timestamp']).dt.hour
print('파생 변수 hour 생성 완료!')

# status_code가 400 이상이면 is_error=1, 아니면 0
df['is_error'] = df['status_code'].apply(lambda x: 1 if x >= 400 else 0)
print('파생 변수 is_error 생성 완료!')

# 2. 특성 선택 및 학습 데이터 분할

# 입력 특성(X) -> 'hour', 'is_error'
X = df[['hour', 'is_error']]

# 정답 값(y) -> 'label'
y = df['is_error']

# 학습용/테스트용 분할 (8:2 비율)
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42
)

# 3. 모델 학습 및 예측

# 로지스틱 회귀 모델 생성 및 학습
model = LogisticRegression()
model.fit(X_train, y_train)

# 테스트 데이터로 예측 수행
y_pred = model.predict(X_test)

# 4. 평가 결과 출력

# 정확도
```

26AI_장상훈_PBL_2_4.ipynb

web_server_logs_2.csv

1. 데이터 로드 및 전처리

Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline Python 3.13.4

web_server_logs_2.csv

웹 서버 로그 기반 악성 ...

```
# 4. 평가 결과 출력

# 정확도
accuracy = accuracy_score(y_test, y_pred)
print(f"정확도: {accuracy:.4f}")

# 정밀도, 재현율, F1-score 등 상세 평가
print("상세 평가 보고서:")
print(classification_report(y_test, y_pred, digits=4))
```

[16] ✓ 0.0s Python

정확도: 0.8933

상세 평가 보고서:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.9303 | 0.9122 | 0.9212 | 205 |
| 1 | 0.8182 | 0.8526 | 0.8351 | 95 |
| accuracy | | | 0.8933 | 300 |
| macro avg | 0.8743 | 0.8824 | 0.8781 | 300 |
| weighted avg | 0.8948 | 0.8933 | 0.8939 | 300 |

> OUTLINE

> TIMELINE