<최종 발표> 발표자: 홍여원

목표 : 선수 개인의 통계뿐 아니라 관계 적용한 승패예측 모델

[방법론]

GNN 기반 Link Prediction 사용

[데이터셋]

Kaggle – UFC Complete Dataset 사용

- red 승리 비율 66% (통상적으로 도전자가 blue 배정받음)

- 개인 기록 – age, SLpM, …

- 경기 중 기록 – td\_def

- 승률과의 상관관계가 높은 15개 특성들 선택

[전처리]

1) 15개 특성 node\_feature로 활용

2) 결측치 제거 & stance를 one-hot encoding

3) 체급별 분류

4) 선수 노드, 경기 양방향 엣지로 표현

5) edge index: red, blue -> 1 0으로 라벨링

6)

[모델 구현]

레이어 추가 - Accuracy 향상

K-Fold 교차 검증 사용 – 0.01, 16, 0.2 최적의 값

[최종 성능]

SVM보다 높은 성능

나머지 3개 베이스라인보단 낮은 성능

Women’s fly 체급에서 가장 높은 성능

[결과 분석]

LogisticRegression – 높음, 안정적

- age\_diff, td\_def\_total\_diff 등 선형적 데이터 패턴에 적합

RandomForest – 라이트헤비, 헤비급에서 가장 높은 성능

- 극단적 경기 결과를 보이는 비선형적 상호작용 포착에 유리

SVM –

-

GNN – Women’s fly에서 최고 성능

- 테스트셋 경기 선수들의 엣지 수가 모두 0인 비율 : 14.29%로 해당 체급에서 최저

- 전반적으로 노드 간 엣지가 충분히 생성되지 않아 낮은 정확도로 보여

[다른 시도]

1) EvolveGCN

- 시간에 따라 동적 그래프 처리 가능

- test accuracy 0.4598로 낮은 성능

2) 노드 단순화

- 같은 특징의 선수들 모아 노드 1개 생성

- k\_means 클러스터링 적용

-

[교수님 질문]

베이스라인도 feature set를 입력해서 단순 승패예측을 하는 건지?

- Yes.

Feature set Normalization 한 건지?

- No.

- 만약 했다면 XGBoost가 더 유리했을 것.

복잡도 향상에 도움이 됐다는 게 무슨 말인지?

- hidden Channel 수를 늘렸다는 것.

Layer가 5개까지 필요했나 싶어. Overfitting의 원인이 되기도 한다.

지금도 복잡도 증가 후 accuracy를 보면 heavyweight는 감소하기도 한다.

Normalization