

Ai와의 대화

이미 mlb자료로 로지스틱 회귀모델을 만들었는데 이 모델 버리기는 아까운데 어떤 식으로 하면 프로젝트 잘 마무리 할 수 있을까? 통계분석응용이라는 과목 팀 프로젝트인데 주제가 "예측 모형 모델링" 이야 로지스틱회귀나 의사결정나무 사용. 추가로 팀원들끼리 mlb로 만든 모델에 kbo데이터를 예측해보는게 의견으로 나왔는데 이거 너 생각어떤지

1. 기존 MLB 모델을 활용해 프로젝트 마무리하는 방법

기존 모델(로지스틱 회귀, total_hr 제외)이 이미 잘 만들어졌으니, 이걸 기반으로 확장하면 돼. 프로젝트를 "예측 모형 모델링"으로 마무리하려면, 단순히 모델 하나 만드는 데 그치지 말고 모델 비교, 해석, 개선, 적용을 추가해서 깊이를 더 하자. 이렇게 하면 "통계분석응용" 과목에서 좋은 점수를 받기 쉽다. (예: 모델의 강점/약점 논의, 변수 중요도 분석, 실생활 적용 사례)

추천 구조: 프로젝트 보고서/발표 흐름

- 서론: MLB 데이터 소개 (2000-2015 팀 통계, 81승 초과 예측의 중요성 – 플레이오프 진출 기준). 목표: 로지스틱 회귀와 의사결정나무를 사용해 승리 예측 모델링.
- 데이터 전처리: 이미 한 부분 (NA 처리, train/test 분할, 변수 선택: ra, slg, total_bb, pitch_so, pitch_bb). total_hr 제외 이유 설명 (e.g., 홈런에 과도하게 의존하지 않기 위해, 더 일반화된 모델 만들기).
- 모델 1: 로지스틱 회귀 (기존 모델 활용):
 - 코드 그대로 사용 (glm 함수).
 - 결과 해석: 계수(coefficients) 분석 (e.g., ra(실점)가 음의 계수 → 실점 적을수록 승리 확률 ↑).
 - 성능 평가: 혼동 행렬, Accuracy, Precision, Recall, F1, AUC (코드에 이미 있음). AUC 0.96이면 "우수한 모델"로 강조.
 - 변수 중요도: summary(model)로 p-value 확인, 또는 odds ratio 계산 (exp(coef(model))).
- 모델 2: 의사결정나무 추가 (rpart 패키지 사용):
 - 로지스틱 회귀만 하면 단조로우니, 의사결정나무로 비교. (과목에서 지정한 대로)
 - 코드 예시 (아래 붙임). train/test 동일하게 사용.
 - 장점: 시각화 쉽고, 비선형 관계捕捉. 단점: 과적합 가능 (prune으로 조정).
 - 성능 비교: 로지스틱 vs. 나무 (e.g., Accuracy, AUC로 테이블화).
- 모델 비교 & 개선:
 - 테이블로 요약: e.g.,

모델	Accuracy	AUC	장점	단점	□
로지스틱 회귀	0.XXX	0.96	해석 쉬움 (odds ratio)	선형 가정	
의사결정나무	0.XXX	0.XX	시각화 강함	과적합 위험	

- 교차검증(CV) 추가: cv.glm()이나 rpart의 cp로 모델 안정성 확인.
- 개선 아이디어: 변수 추가/제거 실험 (e.g., total_hr 넣었을 때 성능 변화? → "홈런 의존성 테스트").
- 결론 & 한계: 모델의 실생활 적용 (e.g., MLB 구단 전략 수립). 한계: 데이터가 2000-2015라 최근 트렌드 반영 부족. 제안: KBO처럼 다른 리그 적용 (아래 의견 참고).
- 부록: 코드 전체, ROC 곡선 플롯, 새 팀 예측 예시 (코드에 있음).

이 구조로 하면 기존 모델 버리지 않고, 의사결정나무 추가로 주제 맞춤. 총 10-15페이지 보고서로 충분. 발표 시 ROC 곡선과 나무 플롯을 시각 자료로 써.

2. 팀원 아이디어: MLB 모델로 KBO 데이터 예측 – 내 의견

좋은 아이디어지만, 주의할 점 많아서 "확장 섹션"으로 넣는 걸 추천. 왜냐면 MLB와 KBO는 리그 규모, 규칙, 선수 수준이 달라 직접 적용 시 해석이 왜곡될 수 있음. (e.g., MLB는 162경기, KBO는 144경기; 슬러그율(slg) 기준도 다름.)

긍정적 의견:

- **재미있고 창의적:** 프로젝트에 "크로스-리그 비교"라는 차별화 요소 추가. MLB 모델로 KBO 팀(LG, 한화)의 "81승 초과 확률" 예측 → e.g., LG의 2023 정규시즌 데이터 입력해 "MLB 기준으로 보면 이 팀은 승리할까?" 분석.
- **데이터 활용 좋음:** 첨부된 KBO 데이터에 필요한 변수(slg, total_bb, pitch_bb, pitch_so, ra)가 다 있음. 시범경기/정규시즌 구분해서 예측 가능.
- **학습 효과:** 리그 차이 논의로 "모델 일반화(generalization)" 개념 설명 (e.g., MLB 모델이 KBO에 과소/과대 예측하는 이유? – KBO는 도루(SB) 비중 높음).
- **구현 쉬움:** 기존 new_team 예측 코드에 KBO 행 하나씩 넣으면 돼. e.g., LG 2023 정규시즌: slg=0.394, total_bb=583, pitch_bb=491, pitch_so=977, ra=610.

부정적 의견 & 주의점:

- **데이터 불일치:** KBO에 "win_season" 같은 실제 승패 레이블 없음. 예측만 하고 끝나서 "검증 불가" (e.g., 실제 LG 2023 승리는? – 외부 검색 필요). 만약 실제 승패 데이터 추가하면 좋음.
- **리그 차이 무시 위험:** MLB 모델은 미국 리그에 최적화됐으니, KBO 예측 정확도 낮을 수 있음. (e.g., KBO slg 평균 낮음 → 모델이 "Loss"로 과소평가). 이걸 "한계"로 논의.
- **프로젝트 초기 흐트러짐:** 주제가 MLB 예측인데 KBO까지 하면 산만. "부록"이나 "추가 분석"으로 제한.

아예 KBO만 하는건 별로임?

아예 KBO만으로 프로젝트를 진행하는 건 별로입니다.

하지만 “MLB 모델 + KBO 예측 실험” 조합은 매우 좋습니다.

아래에서 왜 KBO 단독은 비추천인지, 어떻게 하면 최고로 만들 수 있는지 명확히 정리해드릴게요.

KBO 단독 프로젝트가 별로인 이유 (5가지 핵심)

항목	문제점	영향	□
1. 승리 라벨 부재	w (승수), l (패수) 없음 → win_season 만들 수 없음	예측 모델링 불가능	
2. 데이터 규모 부족	2006~2025 × 10팀 = ~200행 (정규시즌 기준)	과적합 위험, 모델 신뢰도 ↓	
3. 변수 불완전	slg, ra, bb, so는 있지만, 팀별 경기 수, 홈/원정, 상대 강도 없음	모델 해석력 약함	
4. 프로젝트 시간 초과	데이터 정제 + 라벨 생성 + 모델링 + 검증 → 1주일 이상 소요	마감 위험	
5. 주제 이탈	기존 MLB 모델 완성 → 버리는 꼴	노력 낭비, 팀원 불만	

결론: KBO 데이터는 “예측 대상”으로만 쓰는 게 최적.

“모델 학습용”으로 쓰면 시간/품질·주제 모두 손해.

최적 전략: MLB 모델 유지 + KBO ‘예측 실험’ 부록

1. 보고서/발표 구조 (총 12~15장 추천)

markdown

복사

1. 서론

- MLB 81승 예측의 실무적 의미
- 로지스틱 회귀 + 의사결정나무 비교

2. 데이터 & 전처리 (MLB)

- 변수 선택 (total_hr 제외 이유)

3. 모델링

- 3-1. 로지스틱 회귀 (기존 코드)
- 3-2. 의사결정나무 (rpart + prune)
- 3-3. 성능 비교 테이블 (Accuracy, AUC)

4. ****부록: MLB 모델로 KBO 예측하기** ← **여기만 추가!****

- KBO 데이터 소개 (LG, 한화 2006~2025)
- MLB 모델 입력 → 81승 확률 예측
- **★★리그 차이 해석★★** (KBO는 slg 낮고, 도루 많음 → 모델이 과소평가?)

5. 결론 & 제언

- MLB 모델 우수성 입증
- KBO 적용 가능성 + 한계