2022-2 분류/예측 1팀 2차 프로젝트:

Kaggle Scrabble Player Rating Competition

자료조사 통합 정리 문서

목차

I. Competition 기본 정보

1. 취지 및 기본사항

대회 링크: https://www.kaggle.com/competitions/scrabble-player-rating/overview

Agenda: Predict the ratings of players based on Scrabble gameplay

Evaluation: RMSE

상대 Bot 3가지: BetterBot (beginner), STEEBot (intermediate), and HastyBot (advanced)

- 취지: 지금까지의 플레이 데이터를 바탕으로, 전혀 새로운 유저의 플레이를 관전한다면 그 플레이어의 실력을 가늠할 수 있는가?

"...So you can imagine you're a Scrabble scout watching players in matches against the bots in one league ... you're familiar with these players, how good they are, and what their ratings are. Then, imagine you take a trip abroad to observe a new league with a totally different set of human players against the same bots. Can you guess what the human players' ratings are based on how they play the game against the bots?"

2. 데이터 정보

(1) **games.csv - 12 x 72,773** 크기의 게임 결과 테이블

- **game_id** - 게임 번호, **다른 테이블과의 key 변수**
- **first** - 선공 플레이어 닉네임, 봇도 섞여 있음
- **time_control_name**
- **game_end_reason** - 게임 종료 사유
- **winner**- 승리한 플레이어 **<- 승패 결과**
- **created_at** - 게임 플레이 일시
- **lexicon** - 사용한 단어 세트
- **initial_time_seconds** - 초기 세팅 시간
- **increment_seconds** - 추가 시간
- **rating_mode** - 점수 모드 Rated면 rating 점수에 영향, Casual이면 영향 x
- **max_overtime_minutes** - 최대 추가 시간 세팅
- **game_duration_seconds** - 게임 전체 플레이 시간간

(2) **sample_submission.csv - 2 x 22,363** 크기의 제출용 테이블

- **game_id** - 게임 번호, **다른 테이블과의 key 변수**
- **rating** - rating 예측값, game_id 단위로 플레이어 측의 rating값 예측

(3) **test.csv - 4 x 44,726** 크기의 테스트 데이터(게임 결과)

- **game_id** - 게임 번호, **다른 테이블과의 key 변수**
- **nickname** - 닉네임, game_id별로 두개씩 존재(봇, 플레이어)
- **score** - 게임 결과 점수, game_id별로 두개씩 존재(봇, 플레이어)
- **rating** - 게임 후 rating 점수, game_id별로 두개씩 존재(봇, 플레이어)하는데 **봇의 것은 기록되어 있고, 플레이어의 것은 NAN**

(4) **train.csv - 4 x 100,820** 크기의 학습습 데이터(게임 결과)

- **game_id** - 게임 번호, **다른 테이블과의 key 변수**
- **nickname** - 닉네임, game_id별로 두개씩 존재(봇, 플레이어)
- **score** - 게임 결과 점수, game_id별로 두개씩 존재(봇, 플레이어)

- **rating** - 게임 후 rating 점수, game_id별로 두개씩 존재(봇, 플레이어)

(5) **turns.csv - 9 x 2,005,498** 크기의 게임 진행 데데이터
- **game_id** - 게임 번호, **다른 테이블과의 key 변수**
- **turn_number -** 턴 번호, 당연히 하나의 game_id에 여러 개가 대응
- **rack** - 해당 턴에서 플레이어가 쓸 수 있는 알파벳
- **location** - 플레이어가 진행한 위치
- **move** - 플레이어가 이동하며 **새로 놓은 단어**
- **points** - 해당 턴에서 **새로 얻은 점수**
- **score** - 해당 턴까지 **누적 점수**
- **turns_type** - 턴의 진행 유형... 대개 Play

II. Competition 참가자 동향

1. 참고 가능한 Code 내용

(1) Scrabble Game Prediction
- Game_id를 기준으로 **train, turns, games 병합** 후 df로 정의
- **결측치(location, rack, move, turn_type) 최빈값으로 대치**
- Numerical_plotting, histplot, boxplot, scatter plot, reg plot, categorical_plotting, pi plot, pair plot 활용해 시각화화
- 회귀분석에서 다중공선성을 확인하기 위해 **VIF 계산** - score_y 제거
- **변수 제거** - nickname_x, nickname_y, rack, location, first, move, created_at, score_y
- 원핫인코딩, 표준화 후 XGboost+Kfold로 모델링

(2) XGBoostRegressor and Simple EDA
- **Created_at 변수를** year, month, day, hour 분할
- **범주형 변수 원핫인코딩**

(3) Basic FE and EDA of New Players: Scrabble
- **Rating 시각화** - 1500에서 huge peak -> rating 1500 default로 시작하고, 이는 new player을 의미
- rating이 1500이면 New, 아니면 Old를 나타내는 **새 변수 'isNewPlayer' 추가**(new 7.51%)
- New vs Old player의 score 분포 비교
- Nickname이 bot으로 끝나는 것들을 bot으로 분류 -> 각 bot 별 rating 분포 비교

(4) Scrabbling with Linear Regression

- Train과 test 데이터를 연결 후 game_id 내림차순 정렬
- **BetterBot, STEEBot, HastyBot을 Bot으로 지정하고, 닉네임이 Bot에 해당하지 않는 경우를 user_df, Bot인 경우 bot_df로 구분**
- user_df와 bot_df를 game_id를 기준으로 merge한 것을 main_df로 지정
- **user_freq 변수 추가** - user_name별 수
- botname **LabelEncoding**
- user_rating이 결측치가 아닌 데이터를 train, 결측치인 데이터를 test 데이터로 구분 후 Linear Regression

2. Discussion 요약

(1) **FE, Public Dataset & Notebook Example** 원문 링크:
https://www.kaggle.com/competitions/scrabble-player-rating/discussion/362735
XGBoost로 예측하는 코드 ('Rating Scrabble Players With XGB Regression'를 홍보하는 Discussion)
캐글/데이콘에서 자주 하던 파이프라인(EDA-전처리-모델 적용 및 수정-최종 모델&예측 도출)으로 XGBoost 모델 써서 rating 예측
최종 턴수, 사용시간, 1턴당 평균 점수&시간, 초반/후반 n턴간 득점 등 파생변수 group_by, 사용자 정의 함수 활용해서 game_id, nickname별로 turns.csv를 분할mutual_info_regression로 변수 추린 다음 XGBoost 모델 적용
수치형 데이터 표준화(mutual_info_regression) 적용하고 봇 플레이어 데이터를 삭제해서 성능 개선

(2) **Can we get the word list?** 원문 링크:
https://www.kaggle.com/competitions/scrabble-player-rating/discussion/363660
Woggles.io에서 사용하는 단어 데이터에 대한 질문글 - NWL20, CSW21, ECWL, NSWL20를 사용하고 있는 것으로 확인됨

- NWL20, NSWL20 데이터를 구할 수 있는 링크:
  https://github.com/scrabblewords/scrabblewords/tree/main/words/North-American
- Collins Word List 2021(CSW21) 데이터를 구할 수 있는 링크:
  https://www.collinsdictionary.com/games/scrabble/tools

(3) Ravi Ramakrishnan의 글 3가지...

평가 기준인 RMSE에 대한 레퍼런스 목록

- Competition evaluation metric RMSE -- adjutant resources for ready reference:https://www.kaggle.com/competitions/scrabble-player-rating/discussion/363660

Scrabble 게임 분석에 대한 레퍼런스 목록

- 기존 Scrabble Point Value 대회 자료나 woggles.io의 특정 bot 알고리즘 분석한 코드 등...
- Public kernels on scrabble- ready reference and starter: https://www.kaggle.com/competitions/scrabble-player-rating/discussion/362744

Scrabble 게임 규칙&플레이에 대한 레퍼런스 목록

- Scrabble- game introduction and adjutant resources for ready reference: https://www.kaggle.com/competitions/scrabble-player-rating/discussion/362747

III. Waggles.io 및 Scrabble 게임 도메인 지식

1. 게임 규칙

(1) 양측이 번갈아서 게임판에 단어를 채운다. 이미 배치되어 있는 알파벳과 인접하게 배치하여야 한다.

(2) 전체 게임 시간 내에 (20분) 사전에 있는 모든 알파벳이 소진되면 게임이 종료되고 점수가 높은 사람이 승리한다.

(3) 단, 20분내에 게임이 끝나지 않으면 overtime으로 진입할 수 있고 이때 제한된 시간(1분, 5분 등)을 모두 소진하면 점수와 무관하게 패배 처리된다.

(4) 알파벳에 표기된 숫자는 점수이며, 게임판에서 특별히 추가점수를 부여하는(2배, 3배)지점에서 단어를 완성하면 고득점이 된다.

(5) 추가 옵션

1. pass: 나의 차례를 건너뛰고 상대방에게 턴을 넘긴다.
2. exchange: 자신의 알파벳 세트에서 원하는 알파벳을 버리고 새로운 알파벳으로 교환한다.

2. 게임 플레이 결과 & 실력 유관 요소 추정

(1) 봇은 컴퓨터이므로 시간 제한이 사실상 의미가 없음. 봇의 레벨이 높을수록 더욱 더 주어진 상황에서 최선의 수를 두는 듯한 느낌임. 한번의 move로 77점을 가져가기도 함.

(2) 따라서 시간이 많이 주어질수록 플레이어의 점수는 승패와 무관하게 높을 것임. 게임 시간을 범주화 하는 것도 하나의 방법일 수 있음. 게임 시간이 긴 경우, 결과가 승리일 때 더 높은 레이팅 점수를 얻는 것으로 보임.

(3) 타임 세팅의 관점에서 overtime과 increment 중에서 무엇이 플레이어 입장에서는 시간적으로 유리할지도 고려 필요.

(4) location 활용 가능성 - 점수 획득 방식의 관점에서 가로와 세로로 다수의 단어가 동시에 완성되면 각 단어마다 점수가 획득되므로, 인접한 위치에 이미 알파벳이 위치하고 있다면 획득 점수 역시 높을 것으로 예상.

(5) move 변수를 다방면으로 활용할 수 있을 것으로 보임 - 우선, move는 실제 배치완료한 단어를 표시해주는 것인데 단어의 길이가 길수록 고득점으로 이어질 가능성이 높아 보임. 또한, 알파벳마다 점수가 다르게 부여되는데 V, Q, Z 처럼 실제 존재하는 단어의 수가 적은 알파벳으로 단어배치를 성공한 경우 더욱 고득점을 할 수 있음. 따라서 플레이어의 move를 분석할 때 알파벳의 개수 등을 파악하면 득점의 정도를 유추할 수 있을 것으로 예상.

- 실제로 hasty bot과 경기하는 플레이어를 관전했을 때 완성시키는 단어의 길이가 매우 긴 편임을 확인했음. move의 단어 길이를 수치로 바꿔서 전처리 하는 식의 방향도 생각해볼 수 있음.
- 반대로 단어를 완성시키 못할 경우 실력 부족을 의심해볼 수 있음. 즉, 패스 = 0점임을 고려해 패스가 많으면 고득점 확률이 적다는 점을 고려할 수 있음.

(6) 대부분의 게임 종료는 regular이며 타임아웃이나 기권으로는 보통 끝나지 않음. 경기 승리할 경우, rating은 승패에 따라 10~20점 내외를 각각 받거나 잃음. 또한 봇의 수준이 높을수록 승리 시 더 높은 레이팅 점수를 획득.

(7) 결국 어떤 상대를 만나든, 획득한 score가 몇점이든 간에 게임에서 승리해야만 레이팅 점수를 얻는 것이므로 게임의 승리를 1차 타겟으로 생각하고 데이터를 분석하는 게 좋을 것 같음.

(8) 자신보다 레이팅이 높은 상대와 경기하여 승리했을 경우, 더 많은 레이팅 점수를 얻고 반대의 경우는 더 많은 점수를 잃음. 스코어와 레이팅의 연관성을 심층적으로 분석해 볼 필요가 있어 보임.

3. score와 rating의 관계 추정

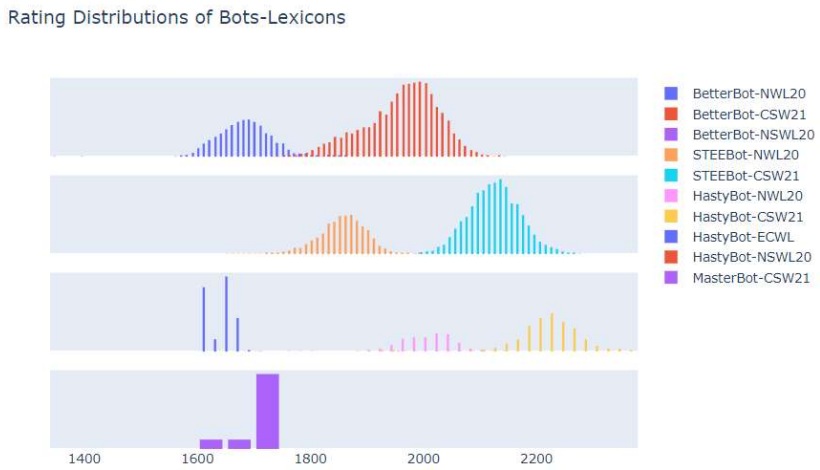
참고 링크 : https://www.kaggle.com/code/vamshichowdary/scrabble-player-rating

가정: rating을 예측하는 것이기는 하지만 score를 무시할 수는 없을 것

- **Correlation** between Game scores and rating - 0.4 정도의 상관계수가 나타남
- **train set의 rating 분포 - 1500에서 상당히 많은 유저수** 발견됨
    - 이는 처음 계정을 생성하여 플레이를 하면 디폴트 rating으로 1500에서 시작하기 때문, 이 사람들의 게임 플레이는 사실상 rating에 크게 영향을 미치지 않을 것임.
    - 8% 정도만이 유의미한 영향을 끼칠 것으로 사료됨.
    - 하지만 아쉽게도 test set에서는 개인의 이전 rating을 알아낼 수 없음
- 실제로, **뉴플레이어가 올드 플레이어보다 근소하게 score의 분포가 더 낮은** 경향
- 직접적으로 해당 유저가 new인지 old인지는 알 수 없지만, **rating_mode와 new_player의 상관관계가 꽤나 높은 편**. 이는 곧 유저가 사용하는 rating 모드(레이팅 경기 or 연습경기)에 따라 새로 유입된 유저인지 기존 유저인지 확인하는 지표가 될 수 있는 것.
- 물론 참고사항 정도이지, 이것으로 뭔가 유의미한 분석을 하기는 어려워보임.

4. Bot에 대한 분석

가정: 봇은 유저와 다르게 레이팅 점수가 정형화되어 있으므로 이를 통해 유저에게 역으로 적용하는 방안 고려 가능



- **봇의 난이도에 따라 선택하는 사전의 종류가 다름** - 게임을 반복해본 유저 입장에서 주로 사용하는 사전의 종류에 따라 해당 유저의 실력을 추측할 수 있음.
- **테스트셋은 오로지 게임 아이디, 상대한 봇과 유저의 점수, 봇의 레이팅만**을 알려주고 있음 - 점수와 봇의 레이팅을 기반으로 다른 데이터셋들을 보아야 할 것.
- 정말 rough하게 지금까지 한 것으로 예를 들면, 봇과의 경기정보를 보았을 때, CSW21 lexicon을 사용하며, rated 모드로 경기를 한 경우는 플레이어가 봇의 레이팅에 대항하는 또는 그 이상의 레이팅을 갖는다고 판단할 수 있음. 따라서 위 같은 게임정보에서의 변수를 범주화하는 식으로 레이팅에 영향을 미치는 분석요소로 사용할 수 있음.

IV. 관련 분야 참고문헌

1. Scrabble 관련 이전 Competition

https://www.kaggle.com/competitions/scrabble-point-value

Predict the point value of the next play in a game of Scrabble 대회
우리한테 주어진 데이터와 매우 유사함

(1) 여기서 우승한 데이터는 cnn/ann 모델, 따라서 우리한테 적용 불가능
이 사람이 인용한 Evaluation Function Approximation for Scrabble 논문 참고

- https://arxiv.org/pdf/1901.08728.pdf

Quackle's heuristic function used in the mid-game phase is a **sum of the move score and the value of the rack leave**

→ Quackle의 statistic evaluation을 이용하면 예측할 수 있지 않을까?

Statistic evaluation = sum of move score and leave value (same weight) ; leave value = chance of playing bingo + leaves that garner high score

게임 진행 상황에 따라 stat eval function에 weight 부여

- Bag 안에 타일이 20-80개 일 때 weight 부여

Drawback: 단순 선형 회귀용 모델이 아니기 때문에 완벽하게 적용 할 수 없다 + leave score를 우리가 구할 수 없다

→ 하지만 우리는 다음 무브를 예측하는 게 아니라 이미 주어진 무브를 분석하는 거니까 leave value를 안(혹은 다른 방법으로) 구해도 되는 거 아닐까..?

(2) https://www.kaggle.com/code/mpwolke/it-s-my-turn-scrabble

같은 대회에서 머신러닝으로 접근한 코드로 결측치는 mean으로 대입 범주형 label encoding, Decision tree (max_depth=3)로 모델링

But, 딱히 특별한 전처리 방법이 눈에 띄지는 않음

2. 점수 예측(회귀 관련)

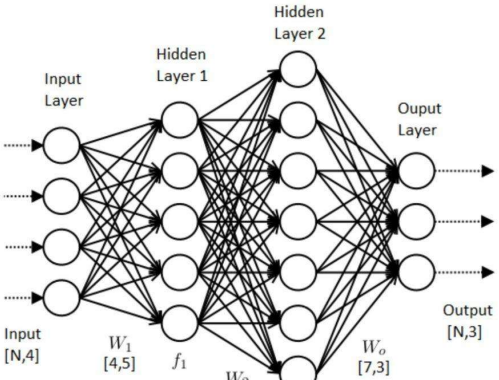
(1) Predicting football scores using machine learning techniques

- **주요 내용:** 축구 경기 점수 예측이 목표
- **피쳐 셀렉션:** 1) 배경지식을 사용하되 2) 예측하고자 하는 것에 기여를 많이 하는 피쳐를 남기는 것이 목표여야 함, 배경지식이 없을 경우에는 백지에서 시작하여서 가장 큰 영향을 주는 피쳐들을 남겨가는 방안이 바람직함. (문제는 피쳐 개수가 많으면 후자는 하기 어려워서, 논문에서는 배경지식을 활용하였다고 함) 또 피쳐를 적절하게 세분화하거나 합치는 것도 예측력 향상에 도움이 됨(파생변수)
- **모델링 관련:** 나이브 베이즈, 베이지안 네트워크, logit boost, knn, 랜덤 포레스트, ann이 도움이 된다고 또다른 선행연구에서 제시하였다. (과거의 정보를 활용하는 베이즈 계열이 잘 맞춘다고 하였음) 결과적으로는 **knn이 가장 좋은 성능을 보였고, 베이지안 네트워크**를 사용하는 계열이 나머지 것들 중에서는 나은 결과를 보였다.

(2) Machine learning & statistical models for predicting indoor air quality

- **주요 내용:** 실내 공기 퀄리티 측정이 목표였음
- **피쳐 선택:** 마찬가지로 배경지식 활용하였음
- **모델링:** ANN, 다중 회귀분석, Partial least squares, decision trees가 좋다고 나옴

  1) partial least square 장: 다중공선성 해결, 결측치 다루기에도 좋음 단: component 선택이 다소 주관적임 => cross validation으로 해결
- **ANN은 무엇인가?** - 딥러닝의 일부로 인공신경망이라 불리는 ANN은 사람의 신경망 원리와 구조를 모방하여 만든 기계학습 알고리즘



- 최적의 weight와 bias 를 찾는 역할을 하는데, **한계도 존재:**
  1) 파라미터의 최적값 결정이 어려움
  2) 오버피팅 이슈
  3) 시간이 너무 오래 걸림.

(3) Predicting user rating on Amazon Video Game Dataset

- 링크: http://jmcauley.ucsd.edu/cse190/projects/fa15/018.pdf
- 유저의 구매 정보 데이터셋을 이용하여 특정 비디오게임에 대해 유저가 평점을 어떻게 부여할지 예측
- 현재 주제와 연결짓자면, **게임 정보를 통해 해당 플레이어의 실력을 부여**한다는 점
- **시도된된 방법:** 우선 트레인셋의 레이팅 점수 평균을 산출, 주어진 피쳐 각각과의 관계(산포도)를 구함. 아마도 어떤 변수가 유의미해보이는지 추적하려는 것이 목적.
- **예측 모델:** 주로 회귀 모형을 사용함. Linear, ridge, randomforest를 사용하여 분석.
- **해당 레퍼런스에서 얻을 수 있는 아이디어:** 우선 우리의 데이터셋 역시 승패는 알고 있는 케이스임. 따라서 사실상 승과 패는 타겟에 큰 영향을 미친다기보다 그 과정에 집중하여야 함.
- 이때 플레이어의 레이팅에 대한 정보를 알 수 없기 때문에 **봇의 레이팅 점수만을 가지고 예측**해야함. 봇의 레이팅 점수는 수준에 따라서도 제각각임.
- 그렇기 때문에 굉장히 rough한 접근법으로는, 봇들의 레이팅의 평균을 구하고 그것의 상하위에 있는 봇에 대해, 만약 평균보다 상위 레이팅에 해당하는 봇에게 게임결과 승리를 거둔 플레이어는 그 실력이 뛰어나다고 판단할 수 있음.
- 결론적으로는 **봇의 실력을 구분하고 그 봇에 대항하는 플레이어의 실력을 역으로 추적**하는 느낌의 분석.