**Part0**

안녕하십니까 발표자 김동현입니다.

멋진 영화배우가 되기 위해서는 어떻게 해야 할까요?

많은 배우들이 돈과 명예를 얻기 위해 밤낮없이 노력하고 있습니다.

탑스타가 되기 위해 달리는 배우들에게 도움이 될 수 있는 머신러닝 모델을 소개해 보려 합니다.

발표는 다음과 같은 순서로 진행하겠습니다.

먼저 프로젝트의 구체적인 목표와 데이터셋에 대해 말씀드리겠습니다.

다음으로 EDA와 데이터 전처리 과정을 거쳐 머신러닝 모델링을 한 후 결론을 말씀드리겠습니다.

**Part1**

소속사와 배우는 상업적으로 성공하고 대중과, 평론가의 평점도 골고루 높은 작품을 고르는 것이 중요합니다.

흥행은 성공했지만 대중과 평론가의 평가가 낮다면 배우의 가치가 내려갈 것이기 때문입니다. 배우에게는 명예도 아주 중요합니다.

따라서 흥행과 작품성을 모두 충족하는 영화 '기생충' 같은 작품에 참여하는 것이 많은 배우들의 꿈입니다.

영화계에는 '믿고 보는 배우' 라는 표현이 있습니다. 특정 배우의 연기력을 신뢰한다는 말이지만 그 배우가 작품을 선택하는 능력이 좋다는 것을 의미하기도 합니다. 따라서 어떤 영화에 출연 예정인 배우 명단을 안다면 작품의 성공여부를 예측할 수 있다고 가정했습니다. 영화의 성공여부를 yes or no로 분류하는 것이 우리의 목표입니다. 모델의 성능이 유의미하다면 배우는 출연 예정인 동료 명단을 참고해 작품 합류 여부를 결정할 수 있습니다. 이를 판단하기 위해 흥행, 대중, 평론가 평가지표를 모두 고려한 Masterpiece라는 새로운 타겟을 정의하겠습니다.

데이터 셋은 캐글에서 IMDB에 기반한 영화 데이터를 참고했습니다. IMDB는 세계에서 가장 큰 영화 데이터베이스 사이트입니다. 모든 카테고리의 영화를 포함하기 때문에 다양성이 충분하고, 만표 이상의 투표를 받은 작품들로 데이터셋을 구성했으므로 통계적으로 의미있다고 판단했습니다. 또한 영화마다 출연 배우의 데이터가 있습니다. 본 데이터셋은 북미 개봉영화 기준입니다.

**Part2**

EDA와 데이터 전처리는 다음의 과정들로 이루어집니다.

데이터 셋의 칼럼은 다음과 같이 title, genre, director처럼 영화에 대한 기본 정보들로 구성되어 있습니다. Certificate는 인코더로 간단히 결측치를 대체할 수 있으므로 넘어가도 됩니다. Metascore와 gross는 타겟이기 때문에 함부로 결측치를 대체하기 어렵습니다. 따라서 데이터셋의 크기가 만개 이상으로 충분하므로 해당 관측치는 삭제하겠습니다. Gross가 0인 데이터도 이상치로 간주해 함께 제거했습니다.

모델을 원활하게 학습할 수 있도록 문자열로 된 칼럼들을 숫자열로 바꿔줍니다.

우리의 목표는 실사영화이므로 애니메이션 장르가 포함된 데이터는 제외하겠습니다.

연도별 데이터 개수를 보겠습니다. 1980년대 이전은 숫자도 적고 은퇴한 배우도 많을 것입니다. 따라서 모델 성능을 높이기 위해 1980년대 이후 데이터만 사용하겠습니다.

Gross와 votes의 분포입니다. Votes는 일반 대중들이 rating점수에 투표한 숫자인데, 대중과 관련된 지표라서 gross와 분포가 비슷한 것 같습니다.

Rating과 metascore의 분포입니다. 그래프를 보시면 Rating은 10점 만점, metascore는 100점 만점인 것을 알 수 있습니다. 두 지표다 어느정도 정규분포를 따르고 있습니다.

Gross의 분포입니다. 데이터의 분포가 굉장히 편향된 것을 알 수 있습니다. 따라서 로그 스케일링으로 분포 균형을 맞춰주겠습니다.

Rating과 metascore는 데이터의 스케일이 다르기 때문에 스케일링을 해줘야 한쪽 지표의 비중이 커지는 것을 방지할 수 있습니다. 이상치가 없기 때문에 minmaxscaling으로 0에서 1사이의 분포를 만들어주겠습니다.

우리의 목표는 흥행과 대중, 평론가의 평점이 골고루 높은 작품을 구분하는 것이기 때문에 로그 스케일링한 gross 역시 minmaxscaling으로 0과 1사이의 분포를 맞춰주겟습니다.

스케일링한 rating, metascore, gross를 더해 masterpiece rate라는 새로운 타겟을 생성했습니다. 그리고 0과 1사이의 값을 사용해 직관적인 차이를 알 수 있도록 masterpiece rate도 스케일링을 해줬습니다.

우리가 생성한 타겟이 관측치의 특성을 잘 반영하는지 샘플을 확인하겠습니다. Rating과 metascore가 높은 기생충은 스타워즈9와 gross가 10배 가까이 차이 나지만 0.92의 높은 타겟 점수를 얻었습니다. 실제로 스타워즈9는 팬들에게 혹평을 많이 받았는 데 이러한 점이 잘 반영된 것 같습니다.

이제 Masterpiece의 기준을 정해야 분류를 할 수 있습니다. 사실 좋은 작품의 기준은 사람마다 다를 수 있습니다. 어떤 영화의 Masterpiece rate가 약 0.62 이상이라면 최소한 평작은 된다는 의미이므로 이것을 기준으로 하겠습니다.

**Part3**

우리가 사용할 모델은 결정트리 기반의 Gradient Boosting을 사용하는 XGBoost입니다. 우리 데이터 셋은 명목형 특성이 많고 비선형성이 예상되기 때문에 선형회귀모델은 적합하지 않을 수 있습니다. 결정트리 모델은 불순도를 최소로 하는 지점을 찾아 데이터를 분할하기 때문에 결측치 처리도 간편하고 분류문제에 사용하기 좋습니다. Gradient Boosting은 이전 모델의 잔차를 학습하며 모델을 보완하는 방식입니다. 따라서 모델의 성능이 매우 좋습니다.

정밀도는 모델이 True라고 분류한 것중에서 실제 True인 것의 비율이고 재현율은 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율입니다. 이외에도 전체 데이터 수 중 예측 결과와 실제 값이 동일한 것의 비율인 정확도와 ROC-AUC를 지표로 모델을 평가하겠습니다. AUC는 그래프 아래의 면적값을 의미하며 0.5보다 크고 1에 가까울수록 좋습니다.

Hold Out 기법으로 데이터셋을 훈련, 검증, 시험 세트로 분리했습니다. Baseline Score는 1로 예측했을 때의 정확도로 정했습니다. 우리의 모델은 정확도가 0.5보다 좋아야 합니다.

RandomizedsearchCV로 최적의 하이퍼파라미터를 탐색했습니다. 여기서 n\_estimators는 weak learner들의 수를 결정하고 learing\_rate는 단계별로 이것을 얼마나 반영할지 결정하는데, 너무 크면 과적합이 발생할 수 있으므로 0.3까지만 탐색을 하겠습니다. max\_depth는 트리들의 깊이를 결정하는 데 역시 너무 크면 과적합이 발생할 수 있습니다. min\_child\_weight는 leaf 노드에 포함되는 관측치 수를 결정합니다. 과적합 발생 시 값을 2배씩 늘려 조절할 수 있으므로 128까지 탐색 범위에 넣었습니다.

처음 가설에 대한 모델 검증 및 해석입니다. 배우의 정보로만 모델링을 했습니다. 검증 결과 정확도가 0.53으로 baseline에 가깝습니다. 동전던지기로 맞출 확률과 크게 다르지 않기 때문에 낮은 성능입니다.

AUC도 0.55로 낮게 나왔습니다. 확실히 배우 정보로만 판단하기는 어려운 것 같습니다. 특성 중요도를 보면 주연배우인 cast1의 비중이 조금 더 높은 것으로 보아 조연보다는 주연의 비중이 더 크다는 것은 알 수 있었습니다.

Certificate, genre, director를 추가해서 모델링 했습니다. 훈련 세트의 성능은 상당히 높게 나왔는데 검증 세트는 낮은 것으로 보아 과적합이 발생한 것 같습니다.

흥미롭게도 Certificate의 비중이 높게 나왔습니다. 확실히 director와 genre가 배우들보다 영화 성공에 더 중요해 보입니다.

마지막으로 runtime을 추가해서 모델링 했습니다. 훈련 세트 성능도 좋고 검증 세트도 높게 나와 과적합이 해소된 것 같습니다.

AUC는 0.71입니다. 놀랍게도 runtime의 비중이 대단히 높습니다. 이유에 대해 분석과 고찰이 필요해 보입니다.

Runtime은 수치형 데이터라 상관계수 계산이 가능합니다. Masterpiece 타겟을 만드는 데 사용했던 gross와 rating, metascore와의 상관계수를 측정했습니다. Rating과 높은 상관관계를 나타내는데 아마도 runtime은 예산과 관련이 있고, 예산이 적은 예술이나 독립영화들은 대중의 취향과 거리가 멀 수 있습니다. 따라서 rating에 이러한 점이 반영된 것 같습니다.

**Part3**

시험 세트로 최종적인 일반화 성능을 평가하겠습니다. Cast만 사용된 모델은 baseline과 다를 바 없는 성능을 보였고, Runtime까지 포함된 모델은 우수한 정확도와 특히 높은 재현율을 보였습니다.

AUC지표도 차이가 컸습니다. Runtime 포함 모델은 0.71로 높은 성능을 보였습니다. 처음 가설과 달리 배우의 정보는 영화 성공예측에 큰 도움이 되지 않는다는 결론을 얻었습니다. 특성 공학을 통해 장르와 감독이 배우보다 더 중요하다고 나왔는데, 이것은 어느정도 통념과 일치하는 결과입니다. 흥미로운 점은 앞서 말씀드린 것과 같이 얼핏 보면 상관없을 것 같은 runtime이 큰 비중을 차지한다는 것입니다. 우리는 Runtime의 상관계수를 근거로 나름의 이유를 분석할 수 있었습니다. Certificate의 경우도 예상과 달리 비중이 높게 나왔는데 이에 대한 분석은 충분히 이루어지지 않은 점이 아쉬웠습니다. 마지막으로 모델 학습 결과에 대한 한계점은 synopsis특성을 사용하지 않은 것입니다. 아직 문자열을 다루는 테크닉이 부족하여 이것을 보완하는 것이 필요해 보입니다.

감사합니다.