JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 6, pp. 35-47, June 2020
https://doi.org/10.9708/jksci.2020.25.06.035

# Movie Box-office Prediction using Deep Learning and Feature Selection : Focusing on Multivariate Time Series

Jun-Hyung Byun*, Ji-Ho Kim*, Young-Jin Choi*, Hong-Chul Lee*

*Student Researcher, Dept. of Industrial Management Engineering, Korea University, Seoul, Korea
*Student Researcher, Dept. of Industrial Management Engineering, Korea University, Seoul, Korea
*Student Researcher, Dept. of Industrial Management Engineering, Korea University, Seoul, Korea
*Professor, Dept. of Industrial Management Engineering, Korea University, Seoul, Korea

## [Abstract]

Box-office prediction is important to movie stakeholders. It is necessary to accurately predict box-office and select important variables. In this paper, we propose a multivariate time series classification and important variable selection method to improve accuracy of predicting the box-office. As a research method, we collected daily data from KOBIS and NAVER for South Korean movies, selected important variables using Random Forest and predicted multivariate time series using Deep Learning. Based on the Korean screen quota system, Deep Learning was used to compare the accuracy of box-office predictions on the 73rd day from movie release with the important variables and entire variables, and the results was tested whether they are statistically significant. As a Deep Learning model, Multi-Layer Perceptron, Fully Convolutional Neural Networks, and Residual Network were used. Among the Deep Learning models, the model using important variables and Residual Network had the highest prediction accuracy at 93%.

▸ **Key words**: Box-office Prediction, Feature Selection, Multivariate Time Series Classification, Random Forest, Deep Learning, Multi-Layer Perceptron, Fully Convolutional Neural Networks, Residual Network

## [요 약]

박스 오피스 예측은 영화 이해관계자들에게 중요하다. 따라서 정확한 박스 오피스 예측과 이에 영향을 미치는 주요 변수를 선별하는 것이 필요하다. 본 논문은 영화의 박스 오피스 예측 정확도 향상을 위해 다변량 시계열 데이터 분류와 주요 변수 선택 방법을 제안한다. 연구 방법으로 한국 영화 일별 데이터를 KOBIS와 NAVER에서 수집하였고, 랜덤 포레스트(Random Forest) 방법으로 주요 변수를 선별하였으며, 딥러닝(Deep Learning)으로 다변량 시계열을 예측하였다. 한국의 스크린 쿼터제(Screen Quota) 기준, 딥러닝을 이용하여 영화 개봉 73일째 흥행 예측 정확도를 주요 변수와 전체 변수로 비교하고 통계적으로 유의한지 검정하였다. 딥러닝 모델은 다층 퍼셉트론(Multi-Layer Perceptron), 완전 합성곱 신경망(Fully Convolutional Neural Networks), 잔차 네트워크(Residual Network)로 실험하였다. 결과적으로 주요 변수를 잔차 네트워크에 사용했을 때 예측 정확도가 약 93%로 가장 높았다.

▸ **주제어**: 박스 오피스 예측, 영화 흥행 예측, 주요 변수 선택, 다변량 시계열 데이터 분류, 랜덤 포레스트, 딥러닝, 다층 퍼셉트론, 완전 합성곱 신경망, 잔차 네트워크

• First Author: Jun-Hyung Byun, Corresponding Author: Hong-Chul Lee
*Jun-Hyung Byun (mylife1001@korea.ac.kr), Dept. of Industrial Management Engineering, Korea University
*Ji-Ho Kim (jihonav@korea.ac.kr), Dept. of Industrial Management Engineering, Korea University
*Young-Jin Choi (youngjin1206@korea.ac.kr), Dept. of Industrial Management Engineering, Korea University
*Hong-Chul Lee (hclee@korea.ac.kr), Dept. of Industrial Management Engineering, Korea University
• Received: 2020. 05. 04, Revised: 2020. 05. 25, Accepted: 2020. 06. 01.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

# I. Introduction

## 1. Background

### 1.1 Box-office

영화 산업에서 소비자의 반응은 박스 오피스로 파악할 수 있다. 박스 오피스(Box-office)는 영화관의 영화표 판매를 통해 얻은 매출액으로, 영화의 상업적 성공에 관한 연구에서 대표적 성과 변수로 사용된다[1]. 그러나 한국 영화의 흥행 순위는 매출액이 아닌 관객 수를 기준으로 한다[2].

### 1.2 Screen quota

한국에서는 1966년 이후 스크린쿼터(Screen Quota)를 도입하였다[3]. 스크린쿼터는 영화관에서 영화를 상영할 때 자국의 영화를 일정 기간 의무적으로 상영하도록 규제하는 것을 의미한다[3]. 한미 FTA 추진과정에서 연간 의무 상영일을 1년 중 1/5 이상, 즉 73일 이상으로 축소하여 2006년 7월 1일부터 축소 시행하고 있다[3].

### 1.3 Movie type

영화는 예술 영화와 상영 영화로 구분된다. 예술 영화는 새로운 제작 방식, 플롯, 또는 상연 방식을 색다르게 사용하는 아방가르드나 실험 영화로 정의된다[4]. 실험 영화는 일반적으로 예술가의 독특하고 색다른 관점을 표현하면서 기법이나 스토리 라인 측면에서 영화 제작자의 개인적 시각을 담고 있는 영화를 의미한다[4]. 예술 영화는 독립 영화, 실험 영화, 다양성 영화 등 예술 영화와 유사한 맥락에서 사용되는 용어들이 혼재되어 있다. 반면에 상업 영화는 대중을 즐겁게 함으로써 상업적 이익을 추구하는 영화로 정의될 수 있다[4]. 따라서 본 연구에서는 상업 영화를 대상으로 분석하였다.

### 1.4 Stakeholders

영화산업은 각 단계의 생산 주체인 제작사, 배급사, 상영관 간의 이해관계가 맞물려 있다[5]. 제작사가 배급사로부터 영화 제작비 및 투자비를 지원받고 영화를 제작하면 배급사는 영화의 판권으로 전국 영화관에 상영 권리를 재판매한다[6]. 판권이란 저작권을 가진 사람과 계약하여 저작물의 이용, 복제, 판매 등에 따른 이익을 독점할 권리를 의미한다[7]. 관객의 관심을 끌기 위해 배급사는 홍보마케팅 전략을 구축하고 적절한 시기와 최적의 장소에서 영화를 공개한다[8]. 상영 권리를 구입한 영화관은 이해관계에 따라 스크린 수를 결정하고 배분한다. 영화 제작 인력 중에 가장 중요한 인력자원은 바로 영화 감독과 배우들이다[9]. 영화 감독, 배우와 같은 인적 자원들이 영화의 흥행에 직접적으로 유의미한 영향을 미치는 것으로 나타났다[9].

### 1.5 eWOM(electronic Word Of Mouth)

온라인 리뷰는 구전효과(Word-of-Mouth)를 생성할 수 있는 중요채널로 인식되고 있다[10]. 구전효과는 마케팅 연구에서 처음 사용된 이래로 연구 분야와 연구자마다 차이는 있지만, 사람들의 입에서 입으로 전달되는 정보의 뜻으로 사용된다[11]. 영화와 같은 경험재에서는 사전 정보 습득 및 평가정보가 영화 관람 결정에 중요한 영향을 미친다[10]. 영화를 관람하기 전 관객들은 영화의 속성 및 주위 정보에 의존해서 영화를 선택하려는 경향이 있으며, 영화 관람 후의 온라인 리뷰는 관람 전 관객의 영화 선택에 영향을 미칠 수 있다[10]. 소셜 미디어의 급격한 발달로 영화 포털 사이트의 커뮤니티가 활성화되고 있으며 블로그, 뉴스 등을 통한 온라인 구전의 영향력이 커짐에 따라 소셜미디어를 활용한 영화 흥행 예측의 연구들이 많아지고 있다[12].

## 2. Research configuration

본 연구에서는 한국 영화 배경을 바탕으로 데이터를 수집하고 처리하여 실험하는 방향으로 구성되어있다. 각 장의 구성 및 내용은 다음과 같다. 2장에서는 본 연구의 주제와 관련된 사전 논문과 연구목적을 소개한다. 본 연구에 응용할 실험 방법 이론을 정리한다. 3장에서는 본 연구의 실험에 사용할 데이터를 수집하여 전처리하는 과정을 소개한다. 4장에서는 본 연구의 목적을 위한 실험 방법을 소개하고 결과를 해석한다. 마지막으로 5장에서는 실험 결과를 바탕으로 결론, 기대 효과 및 한계점과 향후 연구에 대한 고찰을 서술한다.

# II. Preliminaries

## 1. Related works

### 1.1 Predicting box-office & Important variable selection

이정미 등[13]의 연구에서는 의사 결정 나무(Decision tree), 랜덤 포레스트(Random Forest), 서포트 벡터 머신(Support Vector Machine), 신경망(Neural Network) 방법을 이용하여 개봉 1주 차 영화 관객 수를 예측하였다. 의사 결정 나무 모델로 주요 변수를 선택하였고 예측 정확도를 높인 변수들만 이용해 재예측하였다. 재예측 결과, 네 가지 모델 중 주요 변수를 이용한 서포트 벡터 머신

(Support Vector Machine)과 신경망(Neural network)이 모두 79.84%로 정확도가 가장 높았다.

정회윤 등[14]의 연구에서는 다중 회귀 분석, 의사 결정나무, 신경망을 이용하여 영화 관객 수를 예측하였다. 다중 회귀 분석으로 유의한 흥행 요소만을 주요 변수로 고려하여 재예측하였다. 결과적으로 유의한 흥행 요소들만을 고려한 예측 방법의 정확도가 모든 흥행 요소들을 고려한 예측 방법보다 평균 8.2% 향상되었으며 유의한 흥행 요소만을 고려한 신경망이 정확도 89.6%로 가장 우수한 성능을 보여주었다.

송정아 등[15]의 연구에서는 나이브 베이즈(Naive Bayes), 다층 퍼셉트론(Multi-Layer Perceptron), 서포트 벡터 머신, 랜덤 포레스트를 이용하여 개봉일, 개봉 1주 후, 개봉 2주 후 시점의 관객 수와 개봉 3주 후의 총관객 수를 예측하였다. 네 가지 모델에 새롭게 제시한 매출액 점유율, 흥행 순위, 순위 증감구분, 순위 변화폭 변수를 포함한 경우와 포함하지 않은 경우를 비교실험 하였다. 결과적으로 새롭게 제시한 변수를 포함한 모델이 정확도가 높게 나타난 경우가 많았고 통계적으로 유의했으며 개봉 3주 후의 총관객 수를 예측한 랜덤 포레스트가 정확도 88.6%로 가장 높았다.

김효동[16]의 연구에서는 공분산 분석을 이용하여 총 수입을 종속변수로 하는 애니메이션 영화의 흥행 성적에 영향을 미치는 요소를 분석하였다. 독립변수로 년도, 국가, 개봉 시기, 다음 포탈 점수에 참여한 숫자, IMDB 해외 영화 평가 사이트의 점수 및 참여자 수, 스크린 수, 연기자 지수를 이용하였고 공분산 분석으로 주 효과를 발견하고 통계적으로 검정하였다. 결과적으로 스크린 숫자와 다음 포탈 점수에 참여한 사용자의 숫자가 총수익에 영향을 준 것을 확인하였다.

허민회 등[6]의 연구에서는 관객 수에 미치는 요인들을 결합한 선형 회귀 모형을 이용하여 1주부터 6주까지 주차별 관객 수를 예측하였다. 평균 평점(최대 10점, 최소 1점), 평점 개수, 긍정 비율(8점 이상 비율), 부정 비율(3점 이하 비율)을 반영하였고 이러한 구전효과(WOM, Word Of Mouth) 요인의 유무에 따른 모형의 차이점을 비교하였다. 결과적으로 구전효과 요인을 반영한 모형이 그렇지 않은 모형보다 평균제곱오차(Mean Squared Error)가 낮았다.

### 1.2 Research purpose

사전 연구의 한계점으로 일별 데이터는 수집에 어려움이 있지만, 일별 데이터를 사용하여 흥행을 예측하면 주나 월별 데이터를 사용하여 예측할 때보다 정확히 예측가능하다. 흥행 예측 문제는 시간 개념이 포함된 여러 변수를 고려하는 다변량 시계열 데이터 분석 문제와 같다. 하지만 전통적인 시계열 분석 모델은 선형 모델을 가정하기 때문에 비현실적이고 예측 효율성이 떨어지는 문제점이 있고 새로운 분석 방법 적용의 필요성이 제기되고 있다[17]. 따라서 최근 비선형 기계학습(Machine Learning) 방법은 분류(Classification)와 회귀(Regression)분야에서 활발한 연구와 좋은 성과를 보이며 활용 가능성이 높다[17]. 일 단위 데이터를 사용할 경우, 주 단위의 데이터보다 방대해져 데이터의 양이 충분히 많을 때 성능이 좋은 딥러닝(Deep Learning) 방법을 사용할 필요가 있다.

Yi Zheng 등[18]은 다변량 시계열 분류를 위해 다채널 심층 합성곱 신경망(Multi-Channels Deep Convolutional Neural Networks) 모델을 사용하였다. 서로 다른 두 개의 데이터 세트에 다층 퍼셉트론 모델과 정확도를 비교한 결과 다채널 심층 합성곱 신경망 모델의 정확도가 최소 약 3.9%에서 최대 약 15.47% 상승했다. Fawaz 등[19]은 시계열 분류 예측을 위해 딥러닝 모델을 연구하였다. 다층 퍼셉트론, 완전 합성곱 신경망(Fully Convolutional Neural Networks), 잔차 네트워크(Residual Network)와 같이 하이브리드(hybrid)가 아닌 모델과 Encoder, MCNN, t-LeNet, MCDCNN, Time-CNN, TWIESN 같은 하이브리드 모델을 모두 12개 데이터 세트로 비교하고 실험하였다. 결과적으로 잔차 네트워크 모델과 완전 합성곱 신경망 순서로 성능이 우수한 것으로 나타났다.

전통적인 특징 기반 시계열 문제에서는 특징 추출과 학습기의 파라미터(Parameter)들을 전문가의 경험적, 실험적 요소에 따라 설정하게 된다[20]. 영상 분류에서 합성곱 신경망(Convolutional Neural Network)으로 대표되는 딥러닝의 성공은 다른 분야로도 전파되고 있다[20]. 시계열 문제에서도 기존의 전통적인 방식에서 벗어나 딥러닝을 이용하여 효율적인 특징 추출 및 기계학습을 하는 방식으로 전환되고 있다[20]. 따라서 본 연구에서는 다층 퍼셉트론, 완전 합성곱 신경망, 잔차 네트워크 모델을 사용하였다.

본 연구에서는 세 가지 연구목적을 제시한다. 첫째, 한국 영화의 일별 데이터를 수집하고 처리하는 방법을 연구하고 제안한다. 급변하는 영화 시장에서 일 단위로 예측하면 주나 월 단위로 예측할 때보다 정확히 예측할 수 있어 영화 관계자들에게 유용하다. 둘째, 영화 흥행 예측을 위해 시간 개념을 함께 고려하는 것이 통계적으로 유의함을 검정한다. 따라서 시간 개념을 함께 고려할 수 있는 비선형 모델 중 딥러닝 방법으로 예측하고 성능을 확인한다. 셋째, 효율적인 흥행 예측을 위해 예측 전에 주요한 변수만 선별하여 예측하면 통계적으로 유의미한 효과가 있는지 확인한다.

## 2. Method theory

### 2.1 Multi-Layer Perceptron

$$X=[X^1,X^2,...,X^M], \quad X^i\in\mathbb{R}^T, \quad i=1,...,M \quad (1)$$

데이터를 딥러닝에 적용할 때 수식 1과 같이 독립변수 $M$-차원 다변량 시계열 벡터 형태로 입력된다. $X$는 하나의 입력 데이터 세트를 의미하고 $M$개의 독립변수 벡터로 구성되어있다. $i$번째 독립변수 $X^i$가 범주 형태의 데이터일 경우 원-핫 인코딩 벡터(one-hot encoding vector)로 입력하고 $X^i$가 연속형 실수 형태의 데이터일 경우 변수별로 스케일링(scaling)하여 입력한다.

$$D=\{(X_1,Y_1),...,(X_i,Y_i),...,(X_N,Y_N)\}, \quad i=1,...,N \quad (2)$$

수식 2는 전체 $N$개의 훈련 데이터 세트를 $X$와 $Y$의 쌍으로 묶어 $D$로 나타낸다. $X_i$는 $i$번째 입력 데이터 세트의 독립변수 벡터 집합을 나타내고 $Y_i$는 종속변수를 범주화한 원-핫 인코딩 벡터를 나타낸다.

$$f_L(\theta_L, \quad x)=f_{L-1}(\theta_{L-1},f_{L-2}(\theta_{L-2},...,f_1(\theta_1, \quad x))) \quad (3)$$

다층신경망은 수식 3과 같이 $\theta$는 가중치이고 $x$는 독립변수이며 $f$는 분류를 위해 선형 회귀에 적용한 활성 함수이다. $L$은 신경망의 층의 깊이를 나타낸다.

$$A_{l_i}=f(\omega_{l_i}{}^*X+b), \qquad i=1,...,L \quad (4)$$

수식 4와 같이 $i$번째 층 $l_i$의 활성 뉴런 $A_{l_i}$는 가중치 $\omega_{l_i}$와 독립변수 $X$, 편향(bias) $b$로 구성된다.

$$R(z)=\max(0,z), \quad z=\omega_{l_i}{}^*X+b, \quad i=1,...,L \quad (5)$$

본 연구에서는 출력층을 제외한 모든 층에 수정된 선형 유닛(Rectified Linear Unit, ReLU)을 활성 함수로 사용하였고 수식 5와 같다.

$$\widehat{Y}_j(X)=\frac{e^{A_{L-1}{}^*\omega_j+b_j}}{\sum_{k=1}^{K}e^{A_{L-1}{}^*\omega_k+b_k}}, \quad jIN[1,K] \quad (6)$$

수식 6은 소프트맥스(softmax) 함수이다. 총 $K$개의 범주에 대해 $j$범주일 때의 확률 $\widehat{Y}_j$을 모두 계산하여 가장 큰 출력값으로 계산된 범주로 원-핫 인코딩하여 최종 분류한다. 한 데이터 세트의 각 범주에 대한 모든 확률의 합은 항상 1이다.

분류 예측을 위한 딥러닝에서 특정 가중치 값의 오차를 근사하고 손실을 정량화하기 위해 미분 가능한 손실함수(loss function)를 정의한다. 가장 많이 사용되는 손실함수는 범주형 교차 엔트로피(cross entropy)이다. 교차 엔트로피를 사용하면 출력층에서 활성화 함수의 도함수에 의한 영향을 제거할 수 있다.

$$L(X)=-\sum_{j=1}^{K}Y_j\log\widehat{Y}_j, \; J(\Omega)=\frac{1}{N}\sum_{n=1}^{N}L(X_n) \quad (7)$$

수식 7과 같이 단일 데이터 세트에서 독립변수 $X$의 다변량 시계열을 분류하기 위해 $K$개의 범주에 대해 교차 엔트로피를 모두 계산 후 합하여 손실함수 $L(X)$를 계산한다. $\widehat{Y}_j$이 $j$범주일 때의 확률이라면 $Y_j$는 실제 범주에 대한 원-핫 인코딩 값이고 전체 훈련 데이터의 손실을 살펴보기 위해 수식 2에서의 전체 훈련 데이터 세트 $D$에 대해 비용함수(cost function)도 계산한다. 비용함수는 학습된 가중치들의 집합 $\Omega$에 대해 $n$번째 데이터 세트의 독립변수 벡터 집합 $X_n$의 손실함수 $L(X_n)$의 평균 $J(\Omega)$로 계산한다.

수식 8은 경사 하강법(Gradient Descent)을 통해 최적의 가중치를 찾는다. 경사 하강법은 수식 7에서 계산한 전체 훈련 데이터 세트의 비용함수 $J$를 수식 8과 같이 가중치들의 집합 $\Omega$의 한 가중치 $\omega$에 대해 편미분 후, 학습률(learning rate) $\alpha$를 곱한 만큼 기존에 계산한 가중치 $\omega$에서 빼 새로운 가중치 $w_{new}$로 갱신하는 방법이다.

$$\omega_{new}=\omega-\alpha\frac{\partial J}{\partial\omega}\Big| \; \forall\,\omega\in\Omega \quad (8)$$

이와 같은 모든 과정은 그림 1[19]과 같이 전반적으로 시계열 분류를 위한 딥러닝 모델에 공통적으로 적용된다.

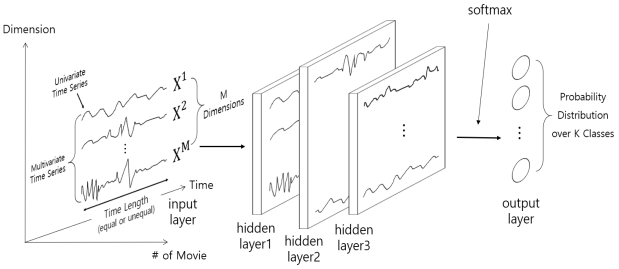



Fig. 1. A Unified Deep Learning Framework for Time Series Classification

### 2.2 Fully Convolutional Neural Networks

완전 합성곱 신경망은 합성곱 신경망을 사용한다. 합성곱 신경망은 필터(filter), 스트라이드(stride), 패딩(padding), 풀링(pooling)으로 구성된다. 이미지 데이터와 달리 시계열 데이터의 경우 필터를 2차원(폭 및 높이) 대신 1차원(시간)만 나타내고 예를 들어, 길이 3인 필터를 단변량 시계열에 합성곱하기 위해 그림 2와 같이 슬라이딩 시간 윈도우(sliding time window)를 모두 $\left[\frac{1}{3},\frac{1}{3},\frac{1}{3}\right]$로 설정하면 합성곱은 윈도우 길이로 이동 평균(Moving Average)한 결과와 같아진다.

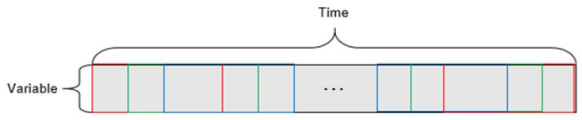



Fig. 2. Sliding Time Window

$$C_t = f\left(\omega * X_{t-\frac{l}{2}\ :\ t+\frac{l}{2}} + b\right) \Big| \forall t IN[1,T] \qquad (9)$$

슬라이딩 시간 윈도우 원리를 이용하여 수식 9와 같이 필터의 길이 $l$과 전체 시간의 길이 $T$인 독립변수 $X$, 편향 $b$로 중앙 타임 스탬프(time stamp) $t$에 대한 합성곱 $C_t$를 계산한다. $t-\frac{l}{2}$부터 $t+\frac{l}{2}$는 중앙 타임 스탬프 $t$로부터 계산된 합성곱 적용 범위다. 윈도우의 크기에 따라 노이즈(noise) 데이터가 식별되지 않을 수도 있지만[21], 노이즈 데이터가 잘 일반화할 수 있는 딥러닝 모델로 학습시키는 데 도움을 줄 수 있어 윈도우 크기를 임의로 설정하였다. 정확한 다변량 시계열의 길이 보존을 위해 모든 합성곱은 스트라이드가 1이고 제로 패딩(zero padding)을 적용한다. 또한, 파라미터 수를 줄여 과적합을 방지하기 위해 마지막 완전 연결 층(fully connected layer)의 특징 추출 과정에서 전역 평균 풀링(Global Average Pooling, GAP)을 적용한다. 그림 3[19]을 토대로, 각 합성곱 필터(convolution에 filter)의 길이와 개수를 구성한다.

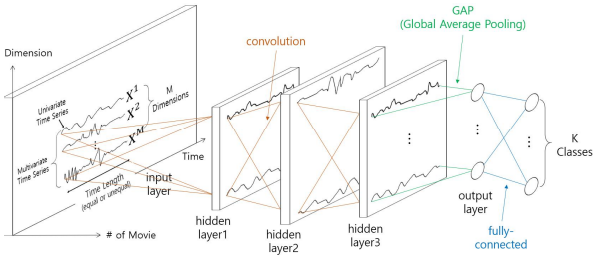



Fig. 3. Fully Convolutional Neural Network Architecture for Multivariate Time Series Classification

### 2.3 Residual Network

잔차 네트워크는 완전 합성곱 신경망과 같이 합성곱 신경망의 기본 구성 요소인 필터, 스트라이드, 패딩, 풀링을 사용한다. 그림 4[19]는 연속하는 합성곱 층 간 잔차를 직접 연결하여 학습시킴으로써 완전 합성곱 신경망에서 발생하는 경사 소실(Vanishing Gradient) 문제를 해결하는 장점이 있다[22]. 시간이 긴 시계열 데이터에 효과적이다. 잔차 네트워크도 각 합성곱에 필터 길이와 개수를 구성한다. 완전 합성곱 신경망과 마찬가지로 파라미터 수를 줄여 과적합을 방지하기 위해 마지막 완전 연결 층의 특징 추출 과정에서 전역 평균 풀링을 적용한다.

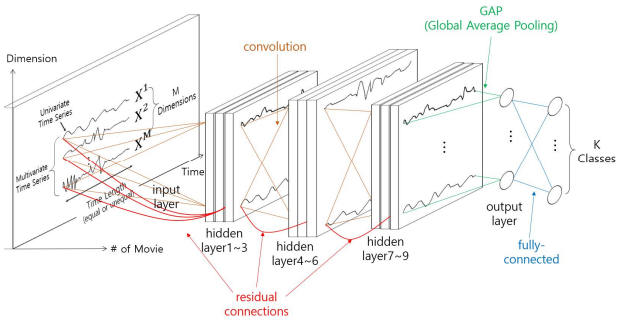


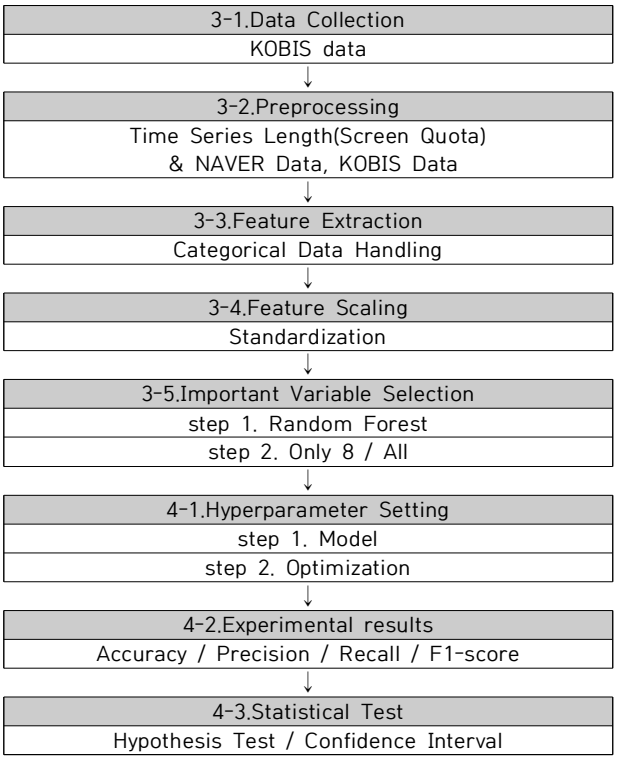

Fig. 4. Residual Network's Architecture for Multivariate Time Series Classification

## III. Data collection and preprocessing

본 연구의 3장에서는 데이터를 수집 및 처리하고 4장에서는 실험 및 결과를 분석한다. 데이터 수집부터 실험 결과 분석까지의 전체 과정은 Table 1과 같다.

3-1절에서는 수집할 KOBIS 데이터를 설명한다. 3-2절에서는 스크린쿼터제를 기준으로 서로 다른 길이의 다변량 시계열 데이터의 길이를 처리하고 해당 영화의 온라인 구전효과(eWOM) 데이터와 영화 이해 관계자들과 관련된 데이터를 수집한다. 3-3절에서는 기존의 범주형 데이터를 변환 또는 파생시켜 랜덤 포레스트(Random Forest)와 딥러닝 모형의 입력 형태에 맞게 각각 변환한다. 3-4절에서는 연속형 변수마다 다른 단위를 통일시킨다. 3-5절에서는 랜덤 포레스트 기법을 이용하여 딥러닝의 입력에 들어갈 주요변수를 선별한다. 4-1절과 4-2절에서는 딥러닝 모델의 최적화를 위한 하이퍼 파라미터(Hyperparameter) 변경과 동시에 주요 변수와 모든 독립변수의 경우를 각각 실험한다. 4-3절에서는 실험 결과를 네 개의 정량적 지표를 통해 비교 및 해석한다. 마지막 4-4절에서는 실험 결과가 통계적으로 유의한지 검정한다.

Table 1. Model flow chart




## 1. Data collection

KOBIS[23]는 한국의 영화관입장권통합전산망(KOREA Box-office Information System)으로, 전국 영화관 입장권 발권 정보를 실시간으로 집계 처리하는 서비스이다. 다양한 박스오피스 정보와 각종 영화산업 통계정보를 제공하여 본 논문의 데이터를 수집하기 위해 파이썬(Python)의 웹 수집기(Web Crawler)를 이용하였다. 데이터는 최근 5년에 해당하는 2015년 1월 1일부터 2019년 12월 31일까지, 총 1,826일 가운데 개봉해서 상영 종료한 전국 국내·외 상업 영화를 일별로 수집하였다. 흥행을 목적으로 하지 않는 예술 영화는 영화의 흥행을 예측하는 본 연구의 주제와 맞지 않기 때문에 제외하였다. 사전 연구에서 다뤘던 가능한 모든 독립변수를 우선적으로 고려하기 위해 KOBIS의 일별 박스오피스 데이터 외에 일별 체인영화관별 상영현황, 일별 좌석점유율, 일별 개봉편수, 일별 상영편수도 모두 수집하였다.

## 2. Preprocessing

Table 2는 한국 영화의 개봉일부터 1,000만 관객 달성까지 걸린 시간을 KOBIS 일별 박스오피스 데이터로 산출하여 영화별로 정리한 것이다.

Table 2. 10 million audience korean movie list

| Rank | Year | Title | Days (from release) |
|---|---|---|---|
| 1 | 2005 | Movie 1 | 66 |
| 2 | 2003 | Movie 2 | 58 |
| 3 | 2019 | Movie 3 | 53 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 25 | 2018 | Movie 25 | 14 |
| 26 | 2014 | Movie 26 | 12 |
| 27 | 2019 | Movie 27 | 11 |
| Average | | | 30.667 |
| Max | | | 66 |

2003년 영화 2를 시작으로 2019년에 개봉한 영화에 이르기까지 1,000만 관객을 달성하는데 2005년에 개봉한 영화 1이 66일로 가장 길었다. 본 연구에서는 1,000만 관객 달성 여부를 예측하기 위해 1,000만 관객 달성까지 걸린 최대 시간 66일을 기준으로 하려고 하였으나, 외국 영화와의 공정성을 위해 스크린쿼터제 기간 73일을 기준으로 설정하여 73일 이상 상영한 국내·외 영화를 대상으로 하였다. 또한, 상영 종료와 관계없이 수집한 모든 영화의 길이를 73일로 통일하였다. 1,000만 관객을 달성하는데 걸린 일 수 계산은 개봉일을 기준으로 계산하였고[24] 모든 변수에 대해 결측치 데이터를 제거하였다. 2015년 1월 1일부터 2019년 12월 31일까지, 총 1,826일 가운데 개봉일부터 73일 이상 상영 후 종료한 전국 국내·외 상업 영화 중 결측치가 없는 데이터는 총 231개다. KOBIS 데이터는 영화 상영 후 관객 수가 발생하지 않으면 해당 일의 데이터가 존재하지 않는다. 본 연구에서는 존재하지 않는 일자의 데이터를, 존재하는 전후 일의 데이터에 맞게 변수별로 관측치 값을 추가 계산 및 생성하여 일별 데이터의 연속성이 끊기지 않게 하였다. 개봉 전에 미리 상영하는 영화는 개봉일과 별개로 누적 관객 수와 관객 수가 같은 첫날을 각 영화 데이터의 실제 시작일로 하지만, 1,000만 관객이 개봉일부터 계산되기 때문에[24] 각 영화의 개봉일을 시작일로 정하였다. 그림 5는 수집한 한 영화의 개봉일부터 73일 상영 시기까지의 KOBIS에서 수집한 연속형 변수들에 대한 다변량 시계열 데이터를 예를 들기 위해 표준화(Standardization)하여 시각화한 것이다. 표준화는 이후의 3-4절 특징 스케일링(Feature Scaling)에서 설명하였다.

Table 3. Feature

| Num | Characteristic | Variable | Description | Scale |
|---|---|---|---|---|
| | 1. Time & Environment | Date | Date | Time Axis |
| 1 | | Day | Day of Date | {Sun, Mon, ..., Fri, Sat} |
| 2 | | Holiday | Holiday of Date | {No, Yes} |
| 3 | | ReleaseDate | Movie Release | {No, Yes} |
| 4 | | FixReleaseMonth | Month of ReleaseDate | {Jan, Feb, ..., Nov, Dec} |
| 5 | | FixReleaseSeason | Season of ReleaseDate | {Spring, ..., Winter} |
| | 2. Competitive factors for film stakeholders | CumAud | Cumulative audience per day | [0, ∞) |
| 6 | | AudIncDec | Audience increase/decrease | (-∞, ∞) |
| 7 | | Sales | Daily Movie Ticket Sales | [0, ∞) |
| 8 | | SalesIncDec | Sales increase/decrease | (-∞, ∞) |
| 9 | | SalesShare | Daily Movie Ticket Sales Share | [0, 1] |
| 10 | | ReleaseMovieNum | # of movies released per day | [0, ∞) |
| 11 | | ShowMovieNum | # of movies showed per day | [0, ∞) |
| 12 | | MovieTotalShow | Total # of showings for the movie in chain cinema (daily) | [0, ∞) |
| 13 | | AllTotalShow | Total # of showings for all movies in chain cinema (daily) | [0, ∞) |
| 14 | | MovieTotalShowRatio | Showing share of the movie in the chain cinema (daily) | [0, 1] |
| 15 | | MovieTotalScreen | Total # of screens for the movie in chain cinema (daily) | [0, ∞) |
| 16 | | AllTotalScreen | Total # of screens for all movies in chain cinema (daily) | [0, ∞) |
| 17 | | MovieTotalScreenRatio | Screen share of the movie in the chain cinema (daily) | [0, 1] |
| 18 | | SeatSalesRatio | Daily seat sales rate | [0, ∞) |
| 19 | | SeatShare | Daily seat share | [0, 1] |
| 20 | | Seat | Seats per day | [0, ∞) |
| | 3. Movie's own characteristics | FixMovieName | Movie Name | Total 231 |
| 21 | | FixNation | Representative Nationality | {South Korea, USA} |
| 22 | | FixRating | Viewing Rating | {All, U12, U15, U18} |
| 23 | | FixProducerScore | Replaced Producer Score | [0, ∞) |
| 24 | | FixPublisherScore | Replaced Publisher Score | [0, ∞) |
| 25 | | FixDirectorScore | Replaced Director Score | [0, ∞) |
| 26 | | FixActorScore | Replaced Actor Score | [0, ∞) |
| 27 | | FixFormat | Showing Type | {Film, 2D, 3D, 4D} |
| 28 | | FixShowTime | Showing Time(seconds) | [0, ∞) |
| 29 | 4. eWOM | NaverNtzStarpts | Netizen's Star Points | [1, 10] |
| 30 | | NaverNtzNum | # of Netizens | [0, ∞) |
| 31 | | NaverNtzPosNum | # of Positive Netizens | [0, ∞) |
| 32 | | NaverNtzNegNum | # of Negative Netizens | [0, ∞) |
| 33 | | NaverReviewNum | # of Reviews | [0, ∞) |
| 34 | | NaverBlogNum | # of Blogs | [0, ∞) |
| 35 | | NaverNewsNum | # of News | [0, ∞) |

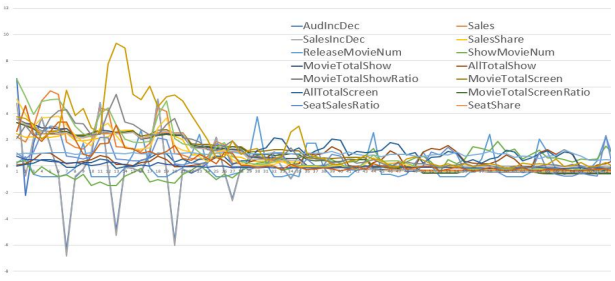



Fig. 5. Movie A's Multivariate Time Series

NAVER[25]와 NAVER 영화[26]에서 구전효과(e-WOM) 변수 7개를 추가로 수집하기 위해 231개 영화를 NAVER 영화에서 검색하여 평점 게시판과 리뷰 게시판의 데이터를 수집하였고 NAVER 검색 데이터는 검색 키워드로 영화 제목에 '영화 '를 앞에 붙여 검색한 결과를 수집하였다. 평점의 경우 네티즌 평점과 관람객 평점으로 분류되어있으며, 구분에 상관없이 합산하여 모든 일자별 데이터를 수집하였다. 마지막으로, 선별된 영화 자체와 관련된 데이터를

수집하기 위해 KOBIS에서 상영타입, 상영시간, 제작사, 배급사, 감독, 배우에 대한 필모그래피를 수집하였다.

### 3. Feature Extraction

제작사, 배급사, 감독, 배우와 같은 범주형 데이터에 대해 영화 도메인을 반영하여 연속형 실수로 변형하였다. KOBIS의 해당 영화 페이지의 제작사, 배급사, 감독, 배우를 선택하여 필모그래피를 참고하였다. 필모그래피란 일상적으로 감독, 배우, 제작자 등 영화 관계자들의 고유 영화 목록을 의미한다[7]. 따라서 각 제작사, 배급사, 감독, 배우 값을 최근까지 참여한 모든 영화 공식통계의 평균 관객 수를 계산했고 소수점 첫째 자리에서 반올림한 값으로 대체하였다. KOBIS의 필모그래피 데이터에는 공식통계 데이터와 KOBIS통계 데이터로 분류된다. 공식통계 데이터는 한국영화연감을 기준으로 산출하는 데이터이고 KOBIS통계는 단순 영화 티켓 판매를 집계한 것이다[23]. 따라서 본 연구에서는 한국영화연감을 기준으로 산출하는 공식통계 데이터를 활용하였다. 다른 범주형 데이터인 자료기준일은 요일 및 공휴일 여부를 나타내는 독립변수로 파생하였다. 개봉일의 경우 변수를 직접 사용하는 대신 개봉일 여부, 개봉월, 개봉일의 계절을 독립변수로 파생하였다. 모든 변수를 정리하면 총 35개로, Table 3과 같다.

범주형 변수는 각 변수 내 범주 간 관계가 없으므로 0과 1로 표현하는 원-핫 인코딩 방식으로 표현하였다. 총 8개의 범주형 독립변수에 원-핫 인코딩 방식을 적용하면 37차원의 범주형 독립변수와 27차원의 연속형 독립변수로, 총 64차원인 다변량 시계열 형태로 구성된다. 잔차 네트워크와 완전 합성곱 신경망은 합성곱을 계산하므로 범주형 독립변수 차원을 축소하지 않고 사용하였다. 합성곱 신경망은 영상이나 이미지와 같은 고차원의 데이터를 입력으로 받을 때 주로 사용한다. 입력층에 특징 추출망이 연결되어 특징 추출망의 출력을 다시 입력으로 받아 분류 학습하는 분별망으로 구성돼있다. 분별망에 연결된 최종 출력층으로 구성되어 기존 다층 퍼셉트론과 비교하면 특징 추출망이 추가된 형태를 갖는다[27]. 따라서 입력 데이터에 대한 합성곱 연산으로 국소적인 영역에서의 특징을 효율적으로 추출하는 기능을 하게 된다[27]. 종속변수인 누적 관객 수의 범주화를 위해 Table 4를 통해 본 연구에 사용할 231편 영화의 100만 단위 범주별 편수를 살펴보았다.

Table 4. Movie number (1 million class interval)

| Class interval (Cumulative audience) | Movie number |
|---|---|
| 10 + | 8 |
| 9 ~ 10 | 2 |
| 8 ~ 9 | 0 |
| 7 ~ 8 | 8 |
| 6 ~ 7 | 7 |
| 5 ~ 6 | 7 |
| 4 ~ 5 | 7 |
| 3 ~ 4 | 13 |
| 2 ~ 3 | 26 |
| 1 ~ 2 | 40 |
| 0 ~ 1 | 113 |
| Sum | 231 |

한국 영화의 최고 흥행 기준인 1,000만을 기준으로, 700만, 500만, 300만을 경계 기준으로 범주화하였고 Table 5와 같이 범주를 각각 A, B, C, D, E로 하였다.

Table 5. Categorization of 231 movies (day 73)

| Class interval (Cumulative audience) | Class | Movie number |
|---|---|---|
| 10 + | A | 8 |
| 7 ~ 10 | B | 10 |
| 5 ~ 7 | C | 14 |
| 3 ~ 5 | D | 20 |
| 0 ~ 3 | E | 179 |
| Sum | | 231 |

### 4. Feature Scaling

추출한 데이터 중 연속형 데이터는 표준화(Standardization)가 필요하다. 표준화는 데이터의 범위를 정규 분포로 변환하여 조정하는 방법으로, 독립변수 $x$에 대해 독립변수의 최댓값 $x_{\max}$와 최솟값 $x_{\min}$을 모르는 경우 활용한다. 본 연구에서 데이터는 73일로 시간 길이를 맞췄기 때문에 실제 상영 종료 시점을 포함하지 않는 데이터도 있다. 따라서 수식 10과 같이 표준화를 사용한다.

$$x_{standard} = \frac{x - \overline{x}}{s_x} \tag{10}$$

독립변수 $x$에 대해 $\overline{x}$는 표본 평균이고 $s_x$는 표본 표준편차를 의미한다.

### 5. Important Variable Selection

Table 3과 같이 35개의 독립변수를 사용하려면 총 누적 관객 수를 예측하는데 상대적으로 더 큰 영향을 미치는

독립변수를 선별할 필요가 있다. 이를 위해 35개의 독립변수 중 27개 연속형 독립변수들과 누적 관객 수에 대한 피어슨(Pearson) 상관분석을 적용했으며 수식 11과 같다.

$$r_{XY}=\frac{\sum_{i}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{n-1}}{\sqrt{\frac{\sum_{i}^{n}(X_i-\overline{X})^2}{n-1}}\sqrt{\frac{\sum_{i}^{n}(Y_i-\overline{Y})^2}{n-1}}},\quad i=1,\ldots,n \quad (11)$$

수식 11에서 영화의 구분 없이 모든 데이터 셋을 합쳐 총 $n$개의 관측치 중 $i$번째 관측치에 대해 $X$는 27개 연속형 독립변수를, $Y$는 누적 관객 수를 의미한다. 상관분석 결과표는 상관계수 $r_{XY}$의 절댓값이 0.8 이상인 일부를 Table 6에 정리하였다.

Table 6. Pearson correlation ($|r_{XY}| \geq 0.8$)

| Feature 1 | Feature 2 | $\lvert r_{XY}\rvert$ |
|---|---|---|
| AudIncDec | SalesIncDec | 0.99 |
| ⋮ | ⋮ | ⋮ |
| MovieTotal ScreenRatio | SeatShare | 0.95 |
| | Seat | 0.95 |
| NaverNtzNum | NaverNtzPosNum | 0.96 |

변수들간의 상관계수는 다중 공선성을 확인하는데 중요한 자료가 된다[28]. Table 6과 같이 독립변수 간의 강한 상관관계가 발생할 때 변수 선별을 위해 선형모형을 적합하면 다중 공선성이 발생한다. 따라서 다중 공선성이 발생하지 않으면서 변수를 선택하는 랜덤 포레스트를 사용하였다. 랜덤 포레스트 방법은 기존의 의사 결정 나무 방법을 합치면서 배깅(Bagging)을 적용하고 분할 변수를 임의로 선택하게 하여 다양성을 증가시킨 앙상블(Ensemble) 모델이다. 앙상블 모델을 통해 과적합(Over Fitting)을 방지하고 분류기의 성능을 향상시킬 수 있기 때문에[29], 랜덤 포레스트를 적용하였다.

본 연구에서는 데이터를 훈련용(Training)과 시험용(Test) 비율을 75:25로 나누어 선정하였다. Table 5와 같이 누적 관객 수를 범주화하여 예측한 뒤 주요한 변수를 선별하였다. 랜덤 포레스트에서의 주요 변수를 계산하는 방식은 수식 12와 같다.

$$Importance(X_i)=OOB\ Error\ \ Rate(X_i)-Error\ \ Rate \quad (12)$$

랜덤 포레스트의 오류율(Error Rate)을 계산한 뒤 앙상블한 모든 나무(tree)에서 $i$번째 변수를 임의로 다른 변수로 변경시키고 오류율을 계산하면 배깅에 사용되지 않는(Out-Of-Bag) OOB 오류가 된다. 효과가 크게 나타날수록 주요한 변수로 한다. 그림 6은 OOB 오류와 각 범주별 오류를 나타낸다.

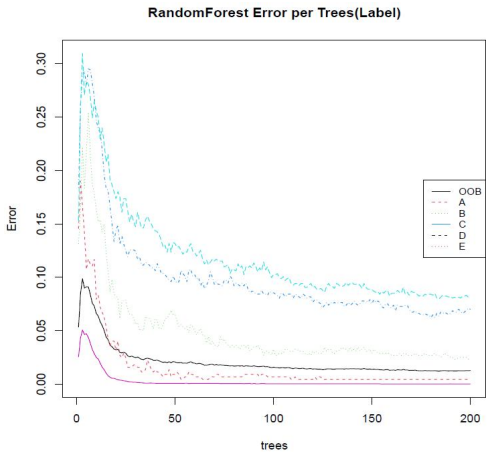



Fig. 6. Check Errors with Random Forest

주요 변수는 그림 7과 같이 배급사 점수, 상영 시간, 감독 점수, 개봉 월, 배우점수, 제작사 점수, NAVER 블로그 수, 상영 포맷 등의 순으로 크게 주요했다.

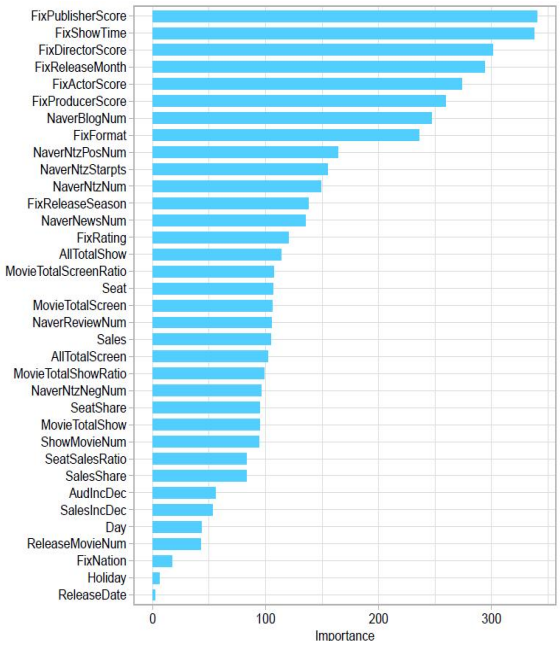



Fig. 7. Importance Variables

## IV. Experiments

앞 장의 주요 변수 선택에서 랜덤 포레스트를 통해 주요 변수 8개를 선별하였다. 본 연구에서는 딥러닝 모델의 하

이퍼 파라미터 변경과 동시에 주요 변수 8개만 사용하는 경우와 모든 독립변수 35개를 사용한 경우로 각각 실험하였다. 실험 데이터는 서로 길이가 다른 다변량 시계열에 대해 길이를 73으로 맞추어 흥행 정도를 5개로 범주화하였고 학습용 데이터와 시험용 데이터는 75:25 비율로 하였다. 실험에 사용한 데이터와 환경은 Table 7과 같다.

Table 7. Experiment Data & Environment

| Experiment Data | |
|---|---|
| Dataset | Movie20152019 |
| Old length (Time length) | 73 - 1694 |
| New length | 73 (screen quota) |
| Classes | 5 |
| Dimensions | 8(Feature) / 35(All) |
| Train | 173 (75%) |
| Test | 58 (25%) |
| Environment (Computer) | |
| Processor | Intel Core i7 6700HQ |
| VGA | Nvidia Geforce GTX 980M |
| RAM | 16GB |
| OS | Windows 10 64bit |
| Python Version | 3.6 |
| Library | tf.keras in TensorFlow 2.0 |

### 1. Hyperparameter Setting

딥러닝 모델의 하이퍼 파라미터는 Table 8과 같다.

Table 8. Model Hyperparameter

| | MLP | FCN | | | ResNet | | |
|---|---|---|---|---|---|---|---|
| Layers num | 4 | 5 | | | 11 | | |
| Conv num | 0 | 3 | | | 9 | | |
| | | 1 | 1 | 1 | 3 | 3 | 3 |
| Filter num | 0 | 128 | 256 | 128 | 64 | 128 | 128 |
| | | 128 | 256 | 256 | 128 | 128 | 128 |
| Filter length | 0 | 8 | 5 | 3 | 8 | 5 | 3 |
| Normalize | None | Batch | | | Batch | | |
| Pooling | None | None | | | None | | |
| Feature | FC | GAP | | | GAP | | |
| Activate | ReLU | ReLU | | | ReLU | | |
| Regularize | Dropout | None | | | None | | |
| Optimizer | AdaDelta | Adam | | | Adam | | |
| | Adam | | | | | | |
| Valid | Train | Train | | | Train | | |
| Loss | Entropy | Entropy | | | Entropy | | |
| Epochs | 5000 | 2000 | | | 1500 | | |
| Batch | 16 | 16 | | | 64 | | |
| Learning Rate | 0.001 | 0.001 | | | 0.001 | | |
| | 1 | 1 | | | 1 | | |
| Decay | 0.0 | 0.0 | | | 0.0 | | |

합성곱 신경망 모형을 구성하기 위해 내부 구성 층의 개수와 위치, 층 내부의 하이퍼 파라미터 값을 설정하는 정해진 방법은 없다[30]. 본 연구에서는 필터 개수, 최적화 알고리듬(Optimizer), 학습률(Learning Rate)을 변경하여 실험하였다. 필터는 완전 합성곱 신경망, 잔차 네트워크 각각 128과 256개, 64개와 128개로 실험하였고 스트라이드와 패딩의 경우 정확한 다변량 시계열의 길이 보존을 위해 모든 합성곱의 스트라이드가 1인 제로 패딩만 적용하였다[19]. 최적화 알고리듬으로 AdaDelta와 Adam을 사용하였고 학습률은 0.001, 1로 각각 실험하였다. 다층 퍼셉트론 모델은 드롭아웃(Dropout)을 적용하여 과적합을 방지하였다. 완전 합성곱 신경망과 잔차 네트워크는 마지막 완전 연결 층에 전역 평균 풀링을 적용하여 과적합을 방지하였다.

### 2. Experimental results

본 연구는 지도학습의 분류 예측 성능 지표로 널리 알려진 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1점수(F1-Score)를 사용하였다. 각 지표 계산 방법은 Table 9, 수식 13-16과 같다.

Table 9. Confusion Matrix

| | | Predicted Class | |
|---|---|---|---|
| | | Positive | Negative |
| Actual Class | Positive | True Positive(TP) | False Negative(FN) |
| | Negative | False Positive(FP) | True Negative(TN) |

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (13)$$

$$\mathrm{Precision} = \frac{TP}{TP+FP} \quad (14)$$

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

$$F1-score = \frac{2\times \mathrm{Precision}\times Recall}{\mathrm{Precision}+Recall} \quad (16)$$

수식 14, 수식 15는 정밀도와 재현율로, 평균의 평균값을 계산하는 매크로(macro) 평균값이 적용되었다. 각 데이터 별 하이퍼 파라미터를 실험한 결과는 Table 10과 같다. 딥러닝 모델에 적용하여 분석한 결과를 보면 8개의 주요 변수만 적용한 잔차 네트워크의 정확도가 0.9376, F1-score가 0.9375로 가장 높았고 모든 모델이 학습률이 0.001일 때 대체로 성능이 좋았다. 잔차 네트워크와 완전 합성곱 신경망에선 대체로 큰 필터를 사용했을 때 성능이

Table 10. Experimental results

| Model | MLP | | | | FCN | | | | ResNet | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Only 8 (MLP) Optimizer : AdaDelta | | | | Only 8 (FCN) Filter : 128 x 256 x 128 | | | | Only 8 (ResNet) Filter : 64 x 128 x 128 | | | |
| learning rate | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score |
| 0.001 | 0.8739 | 0.8795 | 0.8739 | 0.8767 | 0.9180 | 0.9182 | 0.9180 | 0.9181 | 0.9112 | 0.9315 | 0.9170 | 0.9242 |
| 1 | 0.84 | 0.8179 | 0.8268 | 0.8223 | 0.89 | 0.8715 | 0.8871 | 0.8792 | 0.9135 | 0.9114 | 0.9032 | 0.9073 |
| | Only 8 (MLP) Optimizer : Adam | | | | Only 8 (FCN) Filter : 128 x 256 x 256 | | | | Only 8 (ResNet) Filter : 128 x 128 x 128 | | | |
| learning rate | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score |
| 0.001 | **0.9074*** | 0.9131 | 0.9074 | **0.9102*** | **0.9282*** | 0.9283 | 0.9282 | **0.9282*** | **0.9376**** | 0.9375 | 0.9376 | **0.9375**** |
| 1 | 0.8970 | 0.8590 | 0.8337 | 0.8462 | 0.9001 | 0.9062 | 0.9001 | 0.9031 | 0.9335 | 0.9407 | 0.9284 | 0.9345 |
| | All (MLP) Optimizer : AdaDelta | | | | All (FCN) Filter : 128 x 256 x 128 | | | | All (ResNet) Filter : 64 x 128 x 128 | | | |
| learning rate | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score |
| 0.001 | 0.88 | 0.8601 | 0.8796 | 0.8697 | 0.9118 | 0.9196 | 0.9223 | 0.9209 | 0.9292 | 0.9263 | 0.9292 | 0.9277 |
| 1 | 0.8856 | 0.8835 | 0.8871 | 0.8853 | 0.89 | 0.8739 | 0.8795 | 0.8767 | 0.9222 | 0.9315 | 0.9170 | 0.9242 |
| | All (MLP) Optimizer : Adam | | | | All (FCN) Filter : 128 x 256 x 256 | | | | All (ResNet) Filter : 128 x 128 x 128 | | | |
| learning rate | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score |
| 0.001 | 0.9004 | 0.9067 | 0.9004 | 0.9035 | 0.9256 | 0.9260 | 0.9256 | 0.9258 | 0.9266 | 0.9265 | 0.9266 | 0.9265 |
| 1 | 0.8990 | 0.9064 | 0.8990 | 0.9027 | 0.9030 | 0.9047 | 0.9029 | 0.9038 | 0.9256 | 0.9260 | 0.9256 | 0.9258 |

*best performance in the model
**best performance in all models

더 좋았다. 합성곱을 사용하는 신경망 모델에서는 큰 필터를 사용할수록 모델의 성능이 높아졌다. 다층 퍼셉트론의 경우 AdaDelta보다 Adam을 사용했을 때 성능이 더 좋았다. 또한, 모든 독립변수를 사용했을 때 대체로 성능이 더 좋았고 완전 합성곱 신경망은 차이가 없었으며 잔차 네트워크은 8개 변수만 사용한 경우가 성능이 가장 좋았다.

### 3. Statistical Test

영화의 흥행을 예측한 사전 연구에서는 딥러닝이 아닌 기계학습 방법을 사용했었고 예측 시점별로 예측 성능이 다양했다. 본 연구에서는 73일 시계열 데이터의 약 93%의 정확도가 흥행 예측에 시간 요소를 고려하는 것이 유의미한지 실험하였다. 따라서 시간 요소를 고려한 잔차 네트워크, 완전 합성곱 신경망, 다층 퍼셉트론 모델들의 분석 정확도가 90% 이상으로 나타난 것이 통계적으로 유의미한지 검증하기 위해 단일 표본 t 검정(One Sample t-test)를 실시하였고 결과는 Table 11과 같다.

Table 11. Model effect

| One Sample t-test | | |
|---|---|---|
| ResNet | | |
| Hypothesis | p-value | 95% C.I. |
| $H_0 : \mu = 90\%$ | 0.00005665 | (0.9188, ) |
| $H_1 : \mu > 90\%$ | | |
| FCN | | |
| Hypothesis | p-value | 95% C.I. |
| $H_0 : \mu = 90\%$ | 0.07948 | (0.8983, ) |
| $H_1 : \mu > 90\%$ | | |
| MLP | | |
| Hypothesis | p-value | 95% C.I. |
| $H_0 : \mu = 90\%$ | 0.9514 | (0.8854, ) |
| $H_1 : \mu > 90\%$ | | |

잔차 네트워크는 p-value가 0.05보다 작으므로 분석 정확도가 유의미한 것으로 나타났지만 완전 합성곱 신경망과 다층 퍼셉트론은 유의미하지 않았다. 하지만 각각 95% 신뢰구간 속에 각각 약 89.8%, 약 88.5% 이상의 흥행 예측 정확도를 가지는 것으로 나타났다. 다음으로 Table 12는 랜덤 포레스트를 이용한 주요 변수 선택 효과를 이표본 t 검정(Two Sample t-test)으로 검정하였다.

Table 12. Important variables' effect

| Two Sample t-test | | |
|---|---|---|
| Hypothesis | p-value | 95% C.I. |
| $H_0 : \mu_{only8} = \mu_{all}$ | 0.5506 | (-0.0225, 0.0123 ) |
| $H_1 : \mu_{only8} \neq \mu_{all}$ | | |

검정 결과, 랜덤 포레스트로 선별한 8개 변수들만으로 흥행을 예측했을 때의 정확도가 모든 변수들로 흥행 예측했을 때와 큰 차이 없었으며 해당 8개 변수들만으로 예측한 효과를 확인하였다.

## V. Conclusions

본 연구는 영화 관련 일별 시계열 데이터를 수집하여 주요 변수를 추출한 뒤, 딥러닝을 활용하여 영화 흥행 예측 모델을 제안하였다. 영화 일별 데이터는 도메인을 참고하여 다양한 방법으로 수집 및 처리하는 방법을 제안하였다. 흥행 예측과 관련해서는 실험한 모델 중 잔차 네트워크 모델이 약 93%로 가장 높은 정확도를 보였고 통계 검정 결과, 잔차 네트워크 모델이 95% 신뢰도에 대해 약 91.8% 이상의 정확도로 예측 가능하였고 시간을 함께 고려한 흥행 예측 방법이 통계적으로 유의미한 것을 확인하였다. 또한, 8개의 주요 변수와 모든 변수를 고려한 예측 정확도를 비교했을 때 통계적으로 유의함을 검정하였다.

실제 개봉 후 흥행 예측을 위해서 분석에 필요한 데이터가 충분히 수집되어야 한다. 하지만, 현실적으로 결측치가 발생하는 경우가 많았다. 본 연구에서는 2015년부터 2019년까지 5년간의 데이터 중 231개의 영화 데이터밖에 사용하지 못했던 한계점이 있다. 향후, 사용 가능한 데이터를 충분히 수집하고, 개봉 전 데이터까지 활용한다면 딥러닝을 이용한 시계열 예측 성능이 더 상승하는지 확인할 수 있을 것이다.

## REFERENCES


[1] Ikkim, Kmchun and Hlee, "The Effect of Professiona l Critics' Reviews on Online User Reviews and Box Office: US Motion Picture Industry, 2006~2008," Korean Journal of Management, Vol. 20, No. 3, pp. 1-27, June 2012.

[2] Policy Research Team, "*2013 Korean Film Industry Settlement,*" Korean Film Council, pp. 13, 2014. *https://www.kofic.or.kr/*

[3] Yjjung and Hspark, "The impacts of screen quota in the screen industry," Journal of The Korea Society of Computer and Information, Vol. 14, No. 12, pp. 217-223, Dec. 2009.

[4] Sykim, Shim and Ysjung, "A Comparison Study of the Determinants of Performance of Motion Picture s : Art Film vs. Commercial Film," Journal of the Korea Contents Association, Vol. 10, No. 2, pp. 381-393, Feb. 2010.

[5] Chyoon and Hdkim, "The Impact of Vertical Integration on the Conducts of Multiplex Theaters in the Korean Movie Industry," Review of Culture & Economy, Vol. 15, No. 2, pp. 127-149, Aug. 2012.

[6] Mhheo, Pskang and Sjcho, "Predicting Box-office with Opinion mining reviews," Korean Institute Of Industrial Engineers, pp. 487-500, 295, The Ocean Resort, Republic of Korea, May 2013.

[7] NAVER Wikipedia, https://terms.naver.com

[8] Chroh, "A Study on the Distribution and Screening of Big Budget Movies in Korean Film Industry:Focus on the Ten Million Audiences' Movies in 2010s," Asian Cinema Studies, Vol. 12, No. 2, pp. 49-76, July. 2019.

[9] Bschon, Sbpark and Arjo, "The Effects of Movie Stars on Box-Office Performances," The Journal of Image and Cultural Contents, Vol. 18, pp. 363-389, Oct. 2019. DOI: 10.24174/jic c.2019.10.18.363

[10] Sycho, Hkkim, Bskim and Hwkim, "Predicting Movie Revenue by Online Review Mining: Using the Opening Week Online Review," Information Systems Review, Vol. 16, No. 3, pp. 113-134, Dec. 2014. DOI: 10.14329/isr.2014.16.3.113

[11] Ynhwang, Yjnam, "An Empirical Study on the Relationship between the Online WOMs and the Number of Audience of Successful Fims," Journal of The Korea Contents Association, Vol. 19, No. 5 pp. 147-162, May. 2019. DOI: 10.5392/JKC A.2019.19.05.147

[12] Shjeon and Ysson, "Prediction of box office using data mining," The Korean Journal of Applied Statistics, Vol. 29, No. 7, pp. 1257-1270, Oct. 2016. DOI: 10.5351/KJAS.2016.29.7.1257

[13] Jmlee and Gglim "*A Study on the Machine Learnin g Technique for the Prediction of the first week opening box office Using key Variable Method and Decision Tree,*" Hanyang University, pp. 1-60, Republic of Korea, Feb. 2018.

[14] Hyjeong and Hjyang, "Predicting Financial Success of a Movie Using Multiple Regression Analysis," Proceedings of the Korean Society of Computer Information Conference, pp. 275-278, Pyeongtaek University, Republic of Korea, July 2013.

[15] Jasong, Khchoi and Gwkim, "Development of New Variables Affecting Movie Success and Prediction of Weekly Box Office using Them Based on Machine Learning," Journal of Korea Intelligence Information Systems Society, Vol. 24, No. 4, pp. 67-83, Dec. 2018. DOI: 10.13088/jiis.2018.24.4.067

[16] Hdkim, "The Success of Animation in Korean Film Industry : An Exploratory Analysis," Journal of The Korean Society of

Computer and Information, Vol. 19, No. 12, pp. 57-70, Dec. 2014. DOI: 10.9708/jks ci.2014.19.12.057

[17] Swbae and Jsyu, "Predicting the Real Estate Price Index Using Machine Learning Methods and Time Series Analysis Model," Housing Studies Review, Vol. 26, No. 1, pp. 107-133, Feb. 2018. DOI: 10.24 957/hsr.2018.26.1.107

[18] F. Li, G. Li, Swhwang, B. Yao and Z. Zhang, "*Web-Age Information Management 2014*," Springer, pp. 298-310, 2014.

[19] H. Fawaz, G. Forestier, J. Weber, L. Idoumghar and P. Muller, "Deep learning for time series classification:a review," Data Mining and Knowl edge Discovery, Vol. 33, pp. 917-963, March 2019. DOI: 10.1007/s10618-019-00619-1

[20] Bhku, Gtkim, Jkmin and Hsko, "Deep Convolutional Neural Network with Bottleneck Structure using Raw Seismic Waveform for Earthquake Classification," Journal of The Korea Society of Computer and Information, Vol. 24, No. 1, pp. 33-39, Jan. 2019. DOI:10.9708/jksci.2019.24.01.033

[21] Jhcho, Lslee, "Cleaning Noises from Time Series Data with Memory Effects," Journal of The Korea Society of Computer and Information, Vol. 25, No. 4, pp. 37-45, Apr. 2020. DOI: 10.9708/jksci.2020.25.04.037

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[23] KOBIS, KOREA Box-office Information System, https://kobis.or.kr

[24] Jpyu and Ehlee, "A Model of Predictive Movie 10 Million Spectators through Big Data Analysis", The Korea Journal of BigData, Vol. 3, No. 1, pp. 63-71, Aug. 2018.

[25] NAVER, https://www.naver.com

[26] NAVER Movie, https://movie.naver.com

[27] Wscho, "Use of Machine Learning Models in the Search for New Physics," Physics and High Technology, Vol. 26, pp. 4-19, Dec. 2017. DOI: 10.3938/PhiT.26.046.

[28] Yjyi, "Testing Main Effects in Interactive Multiple Regression," Korean Academic Society Of Business Administration, Vol. 23, No. 4, pp. 183-210, Nov. 1994.

[29] Yiseo, Ehjeong and Djkim, "Deep Learning based Scrapbox Accumulated Status Measuring," Journal of The Korea Society of Computer and Information, Vol. 25, No. 3, pp. 27-32, Mar. 2020. DOI: 10.9708/jksci.2020.25.03.027

[30] Twkim, Jhkim and Hsmoon, "The Study on The Identification Model of Friend or Foeon Helicopter by using Binary Classification with CNN," Journal of The Korea Society of Computer and Information, Vol. 25, No. 3, pp. 33-42, Mar. 2020. DOI: 10.9708/jksci.2020.25.03.033

## Authors

Jun-Hyung Byun received the B.S. degree in Applied Statistics from Korea University, Korea, in 2018. He is currently a M.S. Student in the department of Industrial Management Engineering at Korea University. He is interested in Artificial Intelligence and Data Science.

Ji-ho Kim received the B.S. degree in Industrial and Information Systems Engineering from Seoul National University of Science and Technology, Korea, in 2015. He is currently a Integrated M.S/Ph.D Student in the department of Industrial Management Engineering at Korea University. He is interested in Data Mining, Artificial Intelligence and Informatics.

Young-Jin Choi received the B.S. degree in Biomedical Science from Daegu University, Korea, in 2015. He is currently a Integrated M.S/Ph.D Student in the department of Industrial Management Engineering at Korea University. He is interested in Artificial Intelligence, Health Care and Simulation.

Hong-Chul Lee received the B.S. degree in Industrial Engineering from Korea University, Korea, in 1983, M.S. degree in Industrial Engineering from Texas Arlington University, U.S. in 1988 and he received Ph.D. degree in Industrial Engineering from Texas A&M University, U.S. respectively. Dr. Lee joined the faculty of the Department of Industrial Management Engineering at Korea University, Seoul, Korea, in 1996. He is currently a Professor in the Department of Industrial Management Engineering, Korea University. He is interested in Artificial Intelligence, Manufacturing Engineering System, and Simulation.