

Stacking Ensemble Learning을 활용한 블록 탑재 시수 예측

권혁천¹·유원선^{2,†}

충남대학교 선박해양공학과 대학원¹

충남대학교 선박해양공학과^{2,†}

A Study on the Work-time Estimation for Block Erections Using Stacking Ensemble Learning

Hyukcheon Kwon¹·Wonsun Ruy^{2,†}

Graduate School, Dept. of Naval Architecture & Ocean Engineering, Chungnam National University¹

Dept. of Naval Architecture & Ocean Engineering, Chungnam National University^{2,†}

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

The estimation of block erection work time at a dock is one of the important factors when establishing or managing the total shipbuilding schedule. In order to predict the work time, it is a natural approach that the existing block erection data would be used to solve the problem. Generally the work time per unit is the product of coefficient value, quantity, and product value. Previously, the work time per unit is determined statistically by unit load data. However, we estimate the work time per unit through work time coefficient value from series ships using machine learning. In machine learning, the outcome depends mainly on how the training data is organized. Therefore, in this study, we use 'Feature Engineering' to determine which one should be used as features, and to check their influence on the result. In order to get the coefficient value of each block, we try to solve this problem through the Ensemble learning methods which is actively used nowadays. Among the many techniques of Ensemble learning, the final model is constructed by Stacking Ensemble techniques, consisting of the existing Ensemble models (Decision Tree, Random Forest, Gradient Boost, Square Loss Gradient Boost, XG Boost), and the accuracy is maximized by selecting three candidates among all models. Finally, the results of this study are verified by the predicted total work time for one ship among the same series.

Keywords : Ensemble learning technique(앙상블 학습 기법), Block erection work-time(블록 탑재 시수), Decision tree method(의사 결정 트리), Feature engineering(특징 추출 공학)

1. 서론

본 연구의 모든 데이터는 조선소의 보호 요청으로 구체적인 호선명과 데이터 내의 상세 내용을 밝히지는 못했음을 밝힌다.

최근 들어 인공지능에 대한 관심이 급증하고 있다. 또한, 조선 분야에서도 자율운항 선박 또는 4차 산업 기술을 산업에 적용하고자 하는 분위기가 조성되고 있으며, 현재 인간의 능력을 초월하는 복잡성을 가지면서 방대한 데이터를 포함하는 문제들이 산재해 있다. 그런 차원에서 본 연구는 방대한 데이터를 분

석하고 학습 할 수 있는 접근 방식인 기계학습을 이용하여 블록 탑재 시수 추정 문제에 적용해 보고자 한다.

조선소에서는 다른 산업분야와 다르게 블록 건조 방식을 채택하고 있다. 이는 선박의 생산 효율성을 위한 것으로 블록의 탑재시수는 조선소의 일정을 수립하고 계획하는데 매우 중요한 요소 중 하나이다. 또한 대형 조선소의 경우 단품 계획 생산이 아닌 선주로부터 주문을 받아 건조하게 되는 주문 생산 방식을 취한다. 그리고 한 척만 수주 받는 것이 아니라 선박의 단가를 낮출 수 있고 조선소로서도 동시 수주로 경쟁력과 이익률을 높일 수 있는 '시리즈 호선' 방식을 채택하고 있다. 때문에 모선의

탑재 시수를 알면 시리즈 호선 건조 시에 발생될 탑재 시수를 예측하고 계획 할 수 있으며, 이는 건조 시 생기는 불필요한 시수를 줄일 수 있는 중요한 자료로 활용할 수 있을 것이다.

조선소에서 탑재시수 계획을 위해 사용하고 있는 데이터는 원단위인데, 이는 모든 작업들을 일반화시키기 위하여 사용하는 단위로서 정밀하게 추산하기에는 어려움이 있으나, 선박의 주문 생산 방식의 특성상 원단위 외에 작업의 양을 정의할 표준이 존재하지 않는다. 대형 조선소의 경우 현재까지 수주 받은 선박의 수는 적지 않으며, 이에 따라 누적된 선박의 탑재 시수 정보도 방대하다. 본 연구에서는 이 점을 이용하여 누적된 기존 블록탑재 시수 데이터들을 기계학습에 적용하고자 했다. 현재 선박의 건조 시 수립되는 탑재 시수 계획은 그동안의 경험과 자료의 분석을 바탕으로 작업자가 직접 계획하고 있어 긴 시간이 소비되고 있으며, 비단 탑재시수 계획뿐만 아니라 전반적인 조선 생산 환경이 전문가의 경험과 지식에 의존되고 있다(Baek et al., 2018). 신조이거나 시리즈 호선인 경우에 따라서 블록탑재 시수를 계획하는데 들어가는 시간은 천차만별이지만, 신조인 경우 일주일 이상이, 시리즈 호선의 경우에도 2~3일 정도가 소요된다. 주문 생산 방식의 특성상 완전히 새로운 선박의 경우 데이터가 없어 예측이 불가능하겠지만, 비슷한 크기와 조건을 만족시키는 선박의 경우 블록탑재 시수를 예측하여 계획 단계에서의 시수 예측 단계를 간략하게 하고자 한다.

선박 건조 시 발생하는 시수나 탑재 공정, 일정 계획에 관한 연구는 꾸준히 수행되어왔으나 기계학습에 적용한 사례는 그다지 많지 않다. Oh et al.(2009)은 제품구성(Product configuration) 개념을 적용하고, 이를 기반으로 개선된 건조 공수 추정 법을 제안했다. Baek et al.(1999)은 블록 공정의 부하평준화를 위한 자원할당의 휴리스틱을 개발하고, 이를 이용하여 탑재 일정을 수립하는 방법을 제시했다. Lee & Kim (1995) 공정 계획 단계에서 도크 공기와 가용 자원의 활용도를 고려한 최적 공정 계획 수립 가능성을 증명했다. 그러나 위와 같은 연구들은 정확한 공수를 추정하거나 일정을 수립하기 위한 단계 과정에서 데이터들의 상관관계나 변수들을 고려하며 계산하므로 오래 걸린다는 단점이 있다. 반면, 조선 분야에서는 아니지만 반도체 분야에서

수행된 Kim & Baek (2012)의 연구를 보면 구축한 Ensemble 학습 모델의 우수성과 현업 적용 가능성을 확인한바 있다.

본 연구에서 확보한 선박의 블록탑재 시수의 데이터에는 WBS(Work Breakdown Structure)나 공정, 공종과 같이 작업의 대분류, 소분류에 대한 데이터들과 품목과 같이 작업의 위치 혹은 부품을 명시하는 항목이 주를 이루고 있다. 따라서 본 연구에서는 시수를 정답(label)으로 지정하고 시수를 결정하는데 들어가는 작업의 분류 및 위치 등을 특성 항목(feature)으로 삼았다. 그리고 이를 이용하여 작업자의 개인 역량 또는 성향에 따라 불규칙해지는 문제를 방지하고 불필요한 시수를 줄일 수 있는 효율적인 탑재 시수 예측 시스템을 제안하고자 한다.

2. 탑재 시수 데이터 분석

2.1 데이터의 분석

탑재 시수 예측을 위해 실제 조선소에 있는 총 3척의 호선데이터를 사용했다. 원본 데이터들을 분석한 결과 블록탑재 시수를 결정하는데 있어서 WBS, 품목, 공정, 공종, 세부, 실행Act, 품셈코드, 품셈 명, 물량내역, 단위, 대표계열, 실 계열, 적용계수 내역으로 총 13개의 항목이 존재했다. 여기서 WBS, 공정, 공종, 세부는 작업 분류 기준을 나타내고 있으며 품목의 경우에는 특정 블록과 블록의 탑재 정도를, 품셈명은 작업의 종류를 나타낸다. 그러나 원본 데이터 항목 내에 존재하는 대표 계열과 실 계열은 아무런 정보도 기입되어 있지 않아 학습에 악영향을 끼친다고 판단하여 특성 항목을 구성할 때 제외 시켰다. 또한 WBS, 공정, 종, 세부, 실행Act의 경우 공종은 공정에 의해서 결정되는 것을 확인 했으며, 같은 항목이 중복될 필요는 없으므로 공종 또한 제외 했다. 그리고 정답인 전체 블록탑재 시수의 데이터는 블록탑재 시수와 물량, 품셈 값, 계수 값이 존재하는데 블록탑재 시수는 나머지 세 항목의 곱이다. 본 연구에서는 물량을 사용자에 의해 결정되는 변수로 판단하였고 품셈 값은 품셈 코드에 의해 결정되었기 때문에 두 항목을 정답에서 제외

Table 1 Organization of block erection work-time data

WBS	Pr.	Pro.	Type	Detail	Act	Pr. code	Pr. name	Qty.	Unit	App.	Qty.	Work time	Pr. value	Coef. value
FAE4A	E28P2	K	4A	000	100	HK211	setting	block	EA	SH421	2.0	12.9	14.7	0.4
FAE4A	E28S2	K	4A	000	100	HK211	setting	block	EA	SH421	2.0	12.9	14.7	0.4
FAE4A	G14P2	K	4A	000	100	HK211	setting	block	EA	SH421	2.0	25.7	14.7	0.9
FAF4B	E28S2	K	4B	000	100	HK311	mount	mount	M	MH410	53.0	57.3	0.7	1.7
FAF43	44150	P	43	000	100	HK511	air test	TK Qty.	EA	MH410	1.0	16.9	14.5	1.2
FAF43	44250	P	43	000	100	HK511	air test	TK Qty.	EA	MH410	1.0	17.5	14.5	1.2
FAF43	44350	P	43	000	100	HK511	air test	TK Qty.	EA	MH410	1.0	17.5	14.5	1.2
FAF43	44450	P	43	000	100	HK511	air test	TK Qty.	EA	MH410	1.0	17.5	14.5	1.2
FAF43	44550	P	43	000	100	HK511	air test	TK Qty.	EA	MH410	1.0	17.5	14.5	1.2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(Pr.(product), Pro.(process), Qty.(quantity), App.(application), Coef.(coefficient))

했고, 최종적으로 특성 항목은 10개로 구성되었다. Table 1은 원본 데이터에 존재하는 항목들과 값들을 일부 나열한 것으로 대표계열과 실 계열에는 값이 없기 때문에 따로 기입하지는 않았다. 다음으로 본 연구에서 사용된 3척의 호선을 각각 A(LNG), B(CNTR), C(COT)로 명명하며, 데이터의 구성은 Table 2와 같다.

Table 2 Organization of training data

	A	B	C
No. of features and label	11	11	11
Instances	900	857	428

각 데이터의 행수는 각기 다르며 C호선의 경우 데이터의 양이 다른 두 호선의 비해서 절반 정도 밖에 되지 않는 것을 확인할 수 있었다.

2.2 Codification

본 연구의 특성 항목들은 원본데이터의 특성상 전부 문자로 이루어져 있으며, Table 1을 보면 WBS부터 적용계수 내역에 이르기까지 모든 특성 데이터의 정보들이 영어와 숫자의 조합 또는 한글로 표현된다. 컴퓨터에서는 문자를 그대로 분류할 수 없기 때문에 문자를 숫자로 변환 시켜 주어야 하는데, 이를 Codification이라고 한다. 다음으로 기계학습에서는 예측 결과의 분류를 위한 분류(Classification) 모델과 연속적인 수를 예측할 수 있는 회귀 분석(Regression) 모델이 있는데, 계수 값의 특성상 연속적인 숫자를 취하고 있으므로 본 연구에서는 결과를 예측할 때 회귀분석 모델을 사용했고 특성 항목 데이터는 정수 화를 통해 기계학습에 적용하고, 계수 값은 그대로 사용했다.

Table 3 Example of codification

Raw data	Codification data
A	0
B	1
C	2
D	3
⋮	⋮

위의 Table 3에는 기본적인 Codification의 방법을 나타내고 있으며, 표와 같이 A를 갖는 문자들은 0으로, B의 문자를 갖는 문자들은 1로 변환을 시켜준다. 이에 따라 학습에 A나 B의 문자가 아닌 0과 1과 같이 정수 화 숫자들이 대신 기계학습의 데이터로 적용된다. Codification 기법을 이용해 결과를 예측 할 때에는 raw data를 변환시킨 후 학습에 적용하고 변환된 새로운 데이터들을 입력하여 예측 결과를 도출한다. 본 연구에서는 앞서 언급했던 것처럼 분류 모델이 아닌 회귀 분석 모델이므로 결과물을 Decodification을 통해 다시 문자로 바꿔줄 필요가 없다.

2.3 Feature Engineering

본 연구에서는 데이터를 분석하여 불필요한 특성 항목들을 제거하여 학습에 적용하고자 했다. 그러나 분석한 데이터에서 사람들이 판단할 수 없는 특성 항목들의 상관관계가 존재 할 수 있고, 이를 정확히 분석하기 위해서 Feature engineering을 사용하여 특성 항목의 사용여부를 검증하고자 했다. Feature engineering이란 모델의 성능을 높이기 위해서 모델에 입력할 데이터를 만들기 위해 주어진 초기 데이터로부터 특성을 가공하고 생성하는 과정을 의미한다. 즉, 그대로 사용할 수 없는 기존의 데이터들을 학습하기에 적합한 데이터로 변환 시키는 과정으로 설명할 수 있다. 본 연구에서 Feature engineering의 방법으로 다음과 같은 두 가지 방법을 채택했다. 첫째, PCA(Principal Component Analysis)는 고차원의 문제를 저차원의 문제로 치환시키는 방법으로 특성 항목들의 각 고유 값들을 구하고, 이를 통해서 항목 간의 독립 정도를 확인할 수 있는 알고리즘이다 (Hotelling, 1933). 그러나 PCA는 단순히 특정한 항목들의 중요도를 판단해주는 것이 아니라 연관된 항목들의 고유 값을 도출하여 분류의 기준을 제시해주는 정도이다. 즉, 어떠한 항목을 사용하는지에 대한 것은 직접 결정해야한다. 또한 PCA의 경우 정답을 고려하지 않고 특성 항목들만으로 상관관계를 분석하기 때문에 결과에 악영향을 줄 가능성이 있다. 따라서 PCA와 Decision tree를 적용한 결과를 비교 하고자 한다.

Table 4 Eigenvalues for each column according to PCA

Column	A series	B series	C series
WBS	5788.65	4256.88	949.73
Pr.	648.23	873.87	584.31
Pro.	166.91	192.97	179.22
Details	6.28	11.86	4.17
Act	2.88	3.34	3.29
Pr. code	1.48	2.05	0.70
Pr. name	0.87	0.56	0.56
Quantity	0.48	0.47	0.19
Unit	0.25	0.34	0.16
App.	0.18	0.16	0.02

Table 4은 PCA를 통해 얻은 각 호선별 항목들의 고유 값을 나타내고 있다. 본 연구에서는 1보다 작은 항목들을 제외하기로 하였다. 따라서 A, B호선의 경우 품셈 명, 물량내역, 단위, 적용계수를 제외하며, C호선은 품셈코드 까지 제외 한다. 둘째로, Decision tree는 예측 가능한 규칙들의 집합을 생성하는 알고리즘으로 생성되는 규칙들에서 항목의 중요도가 나타나는데 (Rokach & Maimon, 2005), 정답을 결정하는데 있어서 큰 영향력을 가진 항목들은 tree의 가장 윗부분에 위치하게 되며 반대로 중요하지 않은 항목들은 tree의 밑 부분에 위치하거나, 심지어 사용되지 않는 경우도 존재한다. 그리고 Decision tree는 PCA와 다르게 특성 항목뿐만 아니라 정답까지 함께 고려하여

상관관계를 분석해주는데, 이는 예측 결과를 정하는데 생성된 tree의 노드들을 분석하기 때문이다.

Table 5 Whether to use each column according to decision tree

Column	A series	B series	C series
WBS	0	0	0
Pr.	0	0	0
Pro.	0	0	-
Details	0	0	0
Act	0	0	0
Pr. code	0	0	0
Pr. name	-	-	-
Quantity	0	0	0
Unit	0	0	0
App.	0	-	-

Table 5는 각 호선별 Decision tree노드에 사용되지 않은 항목들을 나타낸 것으로, A호선의 경우 품셈명이, B호선의 경우 품셈명과 적용 계수가 사용되지 않았다. 마지막으로 C호선은 공정, 품셈명, 적용 계수로 총 3개가 사용되지 않은 것을 확인 할 수 있다. 또한 본 연구에서 사용되는 데이터의 특성 항목의 수는 총 10개로 각 호선의 행 데이터 수에 비해서 적지 않은 수를 가지고 있다. 따라서 데이터들을 Feature engineering의 적용한 예측 결과와 적용하지 않은 예측 결과간의 비교를 해보고자 했다.

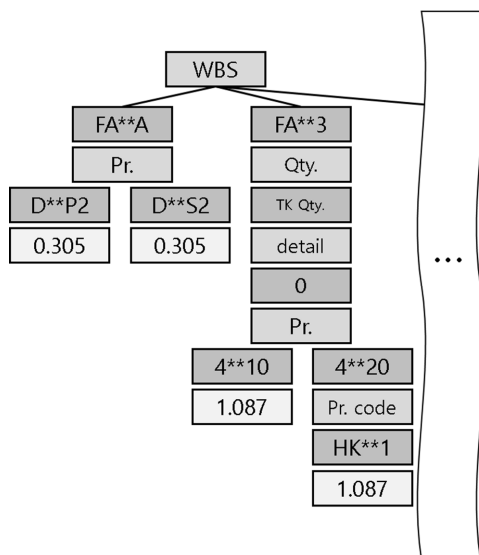


Fig. 1 Example of decision tree

Fig. 1은 C호선의 Decision tree의 일부분을 계층적으로 형상화 한 것으로 데이터의 보호를 위해 정확한 명칭은 기입하지 않았다. WBS부터 물량내역, 세부, 품목, 품셈코드까지 순서대로 중요도와 어떠한 값으로 결정되는지도 확인 할 수 있으며, 그림의 예시로 0.305의 계수 값을 예측하는데 있어서 WBS, 품

목 순으로 중요도를 표현 할 수 있다. 이를 통해서 그림에서는 WBS가 계수 값을 정하는데 있어서 가장 큰 영향을 미치고 있다고 판단 할 수 있으며 사용되지 않거나 영향력이 적은 특성 항목을 판단 할 수 있다. 또한 예측된 모든 계수 값의 상위 노드에는 WBS와 품목이 존재하고 있으며, 이것으로 블록답재 시수의 데이터는 WBS와 품목이 가장 중요한 역할을 하는 것을 확인 할 수 있었다.

3. Ensemble Learning을 통한 품셈 계수 값 예측

3.1 Ensemble Learner

본 연구에서 탑재시수를 예측하기 위한 기계학습 모델로 Ensemble 모델을 선택했다. Ensemble 모델을 사용하는 이유는 학습 모델별 장단점이 존재하고 어떤 단일 학습 모델이 적합한지 모두 테스트하기 번거롭기 때문이다. 즉, 하나의 weak learner로 결과를 정하는 것이 아니라 여러 weak learner들의 결과를 조합하여 최종 결과물을 도출한다고 할 수 있다. 또한 Ensemble 모델은 다른 단일 학습 모델에 비해서 특정 조건만 만족시키면 높은 정확성을 갖는데, 이는 Binomial distribution으로 설명할 수 있다. 단일 모델과 Ensemble 모델의 오차 확률을 비교하면 다음과 같다.

$$p_k(n, p) = \sum_{k=0}^{n/2} \binom{n}{k} p^k (1-p)^{n-k} \quad (1)$$

학습된 모델의 수를 n, 각 단일 모델들의 오차 확률을 p(모든 모델의 p는 단일 값이라 가정)라고 했을 때, Ensemble 모델의 오차 확률은 식 (1)과 같다. 여기서 Ensemble을 구성하는 모델들의 수가 많으면 많을수록 오차 확률이 줄어드는 것을 확인할 수 있다. 또한 Ensemble 모델은 모델의 과반 수 이상이 틀렸을 경우 틀렸다고 판단을 한다. 따라서 p가 0.5보다 작다면 Ensemble 모델의 오차 확률이 단일 모델의 오차 확률보다 매우 작을 수밖에 없다. Ensemble 모델은 위에서도 언급했듯이 여러 weak learner들을 조합하여 결과를 예측하고 이에 따라 단일 학습 모델에 비해서 예측 오차율이 줄어들게 된다. 이러한 Ensemble 모델에는 여러 종류가 있는데, 본 연구에서는 주로 세 가지 방법을 사용했다. 첫째로 Bagging은 데이터를 복원 추출하여 각 모델을 학습 시켜 결과를 집계하는 방법이다. 먼저 대상 데이터로부터 무작위로 복원 추출하여 데이터를 추출하고, 추출된 데이터가 일종의 표본 집단이 된다. 이 상태에서 동일한 모델을 학습 시키고 예측 변수들을 집계하여 그 결과로 모델을 생성한다. 이러한 방식을 Bootstrap aggregating이라고 부른다. 이러한 방법을 채택하는 이유는 다음과 같다. 대부분의 기계학습에 생기는 오류는 높은 편향(bias)으로 인한 과소적합(underfitting), 높은 분산(variance)로 인한 과적합(overfitting)

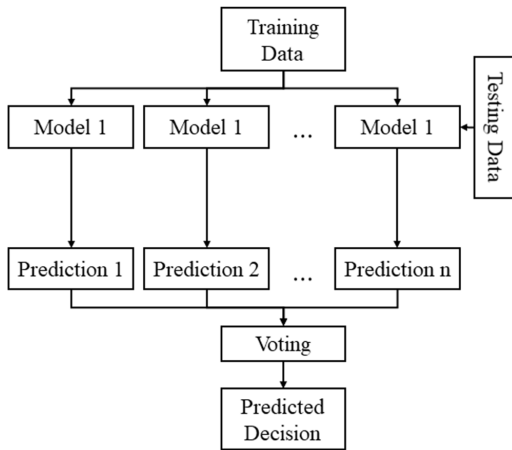


Fig. 2 Stacking ensemble algorithm overview

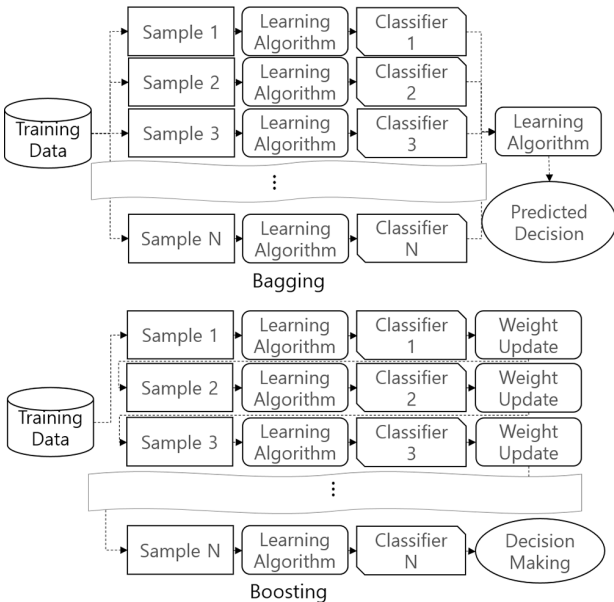


Fig. 3 Bagging & Boosting ensemble algorithm overview

인데 Ensemble 기법은 이러한 오류를 최소화 하는데 도움이 된다. 특히 Bagging의 경우 각 샘플에서 나타난 결과를 일종의 평균값으로 집계하기 때문에 과 적합을 피할 수 있다(Breiman, 1996). 둘째로, Boosting은 맞추기 어려운 문제를 예측하는데 특화되어 있으며, Bagging과 동일하게 무작위로 복원추출을 하지만, 가중치를 부여한다는 차이점이 있다(Schapire, 1990). Bagging이 병렬로 학습하는 반면, Boosting은 순차적으로 학습시킨다. 학습이 끝나면 나온 결과에 따라 가중치가 재분배되며, 오답에 대해 높은 가중치를 부여하고 정답에 낮은 가중치를 부여하기 때문에 오답에 더욱 집중 할 수 있게 되고, Bagging에 비해서 정확도가 높게 나타나지만, 그만큼 특이 값(outlier)에 취약하기도 하다. 마지막으로 Fig. 2는 Stacking ensemble 모델을 그림으로 표현한 것으로, Stacking은 서로 다른 모델들을 조합해서 최고의 성능을 내는 모델을 생성한다. Stacking모델은 다양한 알고리즘을 조합하여 구성할 수 있으며, 이러한 조합을 통해서 각 알고리즘의 장점을 취하면서 약점을 보완 할 수 있다.

즉, 일반적으로 학습하는 하나의 모델보다 성능을 향상시킨다 (Wolpert, 1992). 본 연구에서의 예시를 들면 각 모델들에 단일 Ensemble 모델인 Random forest 혹은 Gradient boost 등을 이용하여 Stacking ensemble 모델을 구성한다.

Fig. 3은 Bagging 과 Boosting ensemble의 구성을 그림으로 설명한 것으로 Bagging의 결과를 집계하여 도출하는 것과 Boosting의 가중치 업데이트를 결과를 예측하는 것을 보여준다.

3.2 단일 Ensemble 모델

본 연구에서는 각 Ensemble 모델들의 성능을 평가하기 위해서 다음과 같이 학습과 검증 과정을 거친다. 먼저 전체 학습 데이터의 70%를 학습하고 나머지 30%로 예측을 한다. 여기서 학습 데이터를 통해 학습한 후 다시 학습 데이터로 예측을 했을 때의 에러 값을 Train error라고 하며, 학습 데이터로 학습한 후 검증 데이터로 예측한 에러 값을 Test error라고 한다. 또한 데이터를 나누는 것은 무작위로 진행되기 때문에 동일한 조건에서 비교하기 위해서 같은 난수를 발생 시킬 수 있도록 seed를 고정 시켰다. 그리고 계수 값뿐만 아니라 총 블록탑재 시수의 합을 계산하여 비교하고자 했다. 따라서 동일한 seed로 물량과 품셈 값을 분류하고 예측한 계수 값을 곱해 정답 계수 값으로 구한 블록탑재 시수의 합의 오차율을 비교했다. 본 연구에서는 계수 값의 특성상 회귀 분석 모델을 통해 예측한다고 언급한 바 있다. 따라서 에러를 기존의 오차율을 통해 확인 할 수 없기 때문에 Mean Square Error를 사용했으며 식 (2)와 같이 구할 수 있다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \tag{2}$$

식 (2)의 Y_i 는 예측한 값을, \hat{Y}_i 는 정답을 의미하여 정답과 예측 값을 빼고 제곱한 값들을 더해준 뒤 평균을 구해 Mean Square Error를 구한다. 다음으로 본 연구에서 사용한 알고리즘 들은 다음과 같다. Decision tree와 Bagging의 대표적인 알고리즘으로 여러 Decision tree로 구성되며, 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과를 집계하는 Random forest(Breiman, 2001), Boosting의 Gradient boosting(Breiman, 1997)과 Square loss gradient boosting, 그리고 최근 가장 좋다고 알려져 있는 XG boosting, 총 다섯 개의 알고리즘을 사용했다.

Table 6 Single ensemble model error of ship A

A series	Ship work time		MSE	
	Relative error rate		Train	Test
Model	Train	Test	Train	Test
Decision	0.00%	3.41%	0.0000	0.0140
Square	0.71%	1.18%	0.0095	0.0122
Gradient	0.71%	1.20%	0.0095	0.0121
Random	0.50%	0.07%	0.0149	0.0207
XG	0.82%	0.33%	0.0103	0.0135

Table 7 Single ensemble model error of ship B

B series	Ship work time Relative error rate		MSE	
	Train	Test	Train	Test
Decision	0.00%	0.02%	0.0000	0.0141
Square	0.04%	1.50%	0.0076	0.0135
Gradient	0.04%	1.50%	0.0076	0.0135
Random	0.04%	1.70%	0.0161	0.0215
XG	0.40%	1.43%	0.0086	0.0145

Table 8 Single ensemble model error of ship C

C series	Ship work time Relative error rate		MSE	
	Train	Test	Train	Test
Decision	0.00%	1.71%	0.0000	0.0107
Square	0.01%	0.24%	0.0024	0.0203
Gradient	0.01%	0.15%	0.0024	0.0207
Random	0.15%	0.13%	0.0098	0.0193
XG	0.24%	0.74%	0.0034	0.0201

위 Table 6, 7, 8은 각 A, B, C 호선의 단일 Ensemble 알고리즘의 에러 값들이다. 앞서 언급했듯이 블록탐재 시수의 합 오차율(ship work time relative error rate)이란 예측한 계수 값과 물량, 품셈 값을 다시 곱한 후 더한 것의 오차율로, 예측된 블록탐재 시수의 합과 target 블록탐재 시수의 합을 비교한 값이다. 위 Table 들의 결과를 통해서 Decision tree의 경우 Train error가 0에 수렴하는데 반해 상대적으로 Test error가 큰 것을 확인 할 수 있으며, 이는 과적합의 위험이 있기 때문에 정확한 성능 평가를 하는데 어려움을 초래 할 수 있다. 따라서 A호선에서는 높은 오차율을 갖지만 나머지 두 호선에서 좋은 결과를 얻는다고 해도 가장 좋은 알고리즘이라고는 판단할 수 없다. 더군다나 본 연구의 데이터는 기계학습에 적용하기에 매우 적기 때문에 생기는 문제를 고려하지 않을 수 없다. 따라서 Stacking ensemble을 통해 더 향상된 오차율을 갖는 모델을 구축하고자 했다.

3.3 Stacking Ensemble 모델

본 연구에서 가장 중점적으로 다루고자 하는 것은 Stacking ensemble 모델로 각 모델의 장점을 취하고 단점을 보완하여 오차율을 최소화 하고자 했다. 대개 Stacking ensemble의 학습 모델들을 선정할 때 단일 모델을 많이 사용하지만 본 연구에서는 에러를 최소화하기 위해서 단일 Ensemble 모델을 다시 Stacking ensemble 모델로 구성하였다. 또한 각 학습 데이터에 따라서 좋은 결과를 갖는 알고리즘이 다르므로 사용한 단일 Ensemble 모델들과 Decision tree, 총 다섯 개의 알고리즘을 조합하여 열 개의 결과를 비교해 보았다.

Table 9 Stacking ensemble model error of ship A

Combination			Train	Test
XG	Decision	Random	0.46%	1.44%
XG	Decision	Square	0.51%	1.65%
XG	Decision	Gradient	0.51%	1.65%
XG	Random	Square	0.70%	0.70%
XG	Random	Gradient	0.70%	0.70%
XG	Square	Gradient	0.75%	0.91%
Decision	Square	Gradient	0.47%	1.94%
Decision	Square	Random	0.42%	1.73%
Decision	Random	Gradient	0.42%	1.72%
Random	Square	Gradient	0.66%	0.99%

Table 10 Stacking ensemble model MSE of ship A

Combination			Train_MSE	Test_MSE
XG	Decision	Random	0.0051	0.0112
XG	Decision	Square	0.0044	0.0094
XG	Decision	Gradient	0.0044	0.0094
XG	Random	Square	0.0105	0.0139
XG	Random	Gradient	0.0105	0.0139
XG	Square	Gradient	0.0097	0.0125
Decision	Square	Gradient	0.0042	0.0091
Decision	Square	Random	0.0048	0.0107
Decision	Random	Gradient	0.0048	0.0107
Random	Square	Gradient	0.0102	0.0134

Table 9, 10은 A호선의 Stacking ensemble 에러이며, Table 9는 전체 블록탐재 시수 합의 오차율을, Table 10은 Mean Square Error의 결과이다. 모델의 성능을 비교하기 위해서 Mean Square Test Error를 기준으로 비교하면, A호선의 경우 가장 좋은 단일 모델은 Table 6에서 확인 할 수 있듯이 Gradient boost로 0.0121의 Mean Square Error를 갖는다. 또한 Stacking ensemble은 Decision tree, Square loss gradient boosting, Gradient boosting 조합의 값이 0.0091로 가장 좋은 것으로 확인 되었다.

Table 11 Stacking ensemble model error of ship B

Combination			Train	Test
XG	Decision	Random	0.19%	0.99%
XG	Decision	Square	0.26%	0.95%
XG	Decision	Gradient	0.60%	0.95%
XG	Random	Square	0.32%	1.46%
XG	Random	Gradient	0.32%	1.49%
XG	Square	Gradient	0.40%	1.45%
Decision	Square	Gradient	0.26%	0.98%
Decision	Square	Random	0.19%	1.02%
Decision	Random	Gradient	0.19%	0.99%
Random	Square	Gradient	0.32%	1.46%

Table 12 Stacking ensemble model MSE of ship B

Combination			Train_MSE	Test_MSE
XG	Decision	Random	0.0048	0.0117
XG	Decision	Square	0.0036	0.0106
XG	Decision	Gradient	0.0036	0.0106
XG	Random	Square	0.0093	0.0144
XG	Random	Gradient	0.0093	0.0144
XG	Square	Gradient	0.0079	0.0137
Decision	Square	Gradient	0.0034	0.0105
Decision	Square	Random	0.0045	0.0115
Decision	Random	Gradient	0.0045	0.0115
Random	Square	Gradient	0.0090	0.0142

Table 11, 12는 B호선의 Stacking ensemble 에러 값이며, 단일 모델 중 가장 우수한 모델은 Table 7의 표를 통해 Square loss gradient boosting과 Gradient boosting이 동일한 0.0135 값을 가진다. 또한 가장 우수한 조합으로는 A호선과 마찬가지로 Decision tree와 Square loss gradient boosting, Gradient boosting이 조합이 0.0105의 값이 도출하는 것을 확인했다.

Table 13 Stacking ensemble model error of ship C

Combination			Train	Test
XG	Decision	Random	0.12%	0.90%
XG	Decision	Square	0.08%	0.87%
XG	Decision	Gradient	0.08%	0.85%
XG	Random	Square	0.12%	0.37%
XG	Random	Gradient	0.12%	0.37%
XG	Square	Gradient	0.08%	0.40%
Decision	Square	Gradient	0.00%	0.64%
Decision	Square	Random	0.05%	0.71%
Decision	Random	Gradient	0.05%	0.71%
Random	Square	Gradient	0.05%	0.19%

Table 14 Stacking ensemble model MSE of ship C

Combination			Train_MSE	Test_MSE
XG	Decision	Random	0.0024	0.0111
XG	Decision	Square	0.0013	0.0127
XG	Decision	Gradient	0.0013	0.0127
XG	Random	Square	0.0041	0.0174
XG	Random	Gradient	0.0041	0.0173
XG	Square	Gradient	0.0027	0.0201
Decision	Square	Gradient	0.0011	0.0130
Decision	Square	Random	0.0021	0.0113
Decision	Random	Gradient	0.0021	0.0113
Random	Square	Gradient	0.0037	0.0177

마지막으로 Table 13과 Table 14는 C호선의 에러 값을 나타내고 있다. C호선에서 가장 우수한 단일 모델은 Table 8을 통해서 Decision tree의 0.0107인 것을 알 수 있으며, Stacking Ensemble의 조합은 XG boosting, Decision tree, Random forest로 0.0111의 값이 도출 되었다. 그러나 앞의 두 호선과는 다르게 Stacking ensemble의 에러 값이 Decision tree의 에러 값 보다 큰 것을 확인 할 수 있는데, 이는 C호선의 데이터가 다른 두 호선에 비해서 학습 할 수 있는 데이터가 충분하지 않아 생긴 오차로 보인다. 또한 Table 9, 11, 13의 오차율이 가장 작은 조합은 Table 10, 12, 14의 조합과는 다른 것을 확인 할 수 있는데 이는 각각 예측된 하나의 계수 값의 오차율과 물량, 품셈 값이 상이해서 나타나는 현상으로, 블록탑재 시수의 합 오차율만으로 학습 모델의 성능을 판단할 수 없어 Mean Square Error를 사용해 비교한 것을 밝힌다.

Table 15 Stacking ensemble model error with Feature engineering of ship A

Decision	Combination		
	Decision	Square	Gradient
	Train		Test
	0.0039		0.0107
PCA	Combination		
	XG	Decision	Random
	Train		Test
	0.0050		0.0133

Table 16 Stacking ensemble model error with Feature engineering of ship B

Decision	Combination		
	XG	Decision	Gradient
	Train		Test
	0.0030		0.0133
PCA	combination		
	Decision	Square	Gradient
	Train		Test
	0.0028		0.0122

Table 17 Stacking ensemble model error with Feature engineering of ship C

Decision	Combination		
	XG	Decision	Gradient
	Train		Test
	0.0017		0.0078
PCA	Combination		
	Decision	Square	Gradient
	Train		Test
	0.0019		0.0107

Table 15 부터 Table 17까지는 각호선의 Feature engineering 을 적용한 데이터를 Stacking ensemble의 결과 중 가장 좋은 조합과 에러 값을 나타내고 있다. 결과의 비교는 Mean Square Error로 진행했다. A호선의 결과를 확인해 보면 가장 좋은 test error인 0.0091에 비해서 Feature engineering을 적용한 결과인 0.0107과 0.0133으로 차이가 일부 나타난 것을 확인 할 수 있으나, Feature engineering을 적용하지 않은 Stacking ensemble의 에러 값 중에서는 이보다 더 큰 에러를 갖는 경우도 확인 할 수 있다. 또한 C호선의 경우에는 Decision tree를 적용한 결과가 더 낮은 에러 값을 갖는 것을 확인 할 수 있는데, 이는 앞서 언급했듯이 데이터가 다른 두 호선에 비해 매우 부족하여 특성 항목들의 상관관계를 분석하기에 충분하지 않아 생기는 현상으로 보이며, PCA와 Decision tree를 비교하여 분석하면, 에러 값이 큰 차이를 보이지 않아 PCA의 정답을 고려하지 않는다는 제한점은 본 연구에서는 크게 작용하지 않은 것으로 판단된다. 마지막으로 한 시리즈 호선의 학습 데이터를 통해서 다른 시리즈 호선의 블록탑재 시수를 검증해보았다. 학습할 시리즈 호선의 데이터는 데이터의 수가 가장 많고 예측 결과가 우수했던 A호선으로 선택했고 A호선의 다른 시리즈 호선을 검증 데이터로 삼아 전체 블록탑재 시수의 합으로 오차율을 비교했으며, Stacking ensemble 모델은 A호선에서 가장 예측 결과가 좋았던 Decision tree, Square loss gradient boost, Gradient boost의 조합을 사용했다. 또한, 검증을 진행할 때 앞서 언급했던 validation방법을 사용하지 않고 A호선의 900개의 행을 가진 데이터 전부를 참여시켜 학습을 진행하였고 검증 데이터 전부가 학습된 모델을 통해 예측 값을 산출 했다. 여기서 검증 데이터의 행수는 927개이며, 총 블록탑재 시수의 합은 91705이고 예측한 값은 91944가 도출되어, 검증 데이터의 전체 블록탑재 시수 오차율은 0.26%가 나오는 것을 확인했다.

Group	품셈명	raw	predict	error rate
1	"철목(PE)"	982.624	977.2579	-0.55%
2	"건조 취부(PE)"	4565.508	4555.151	-0.23%
3	"건조 수동용접(PE)"	5502.679	5555.373	0.96%
4	"LUG 설치/해체(후PE2)"	740	737.7339	-0.31%
5	"FAIRING 작업(PE)"	545	543.331	-0.31%
6	"LUG 추가신설/제거(PE)"	327	325.9986	-0.31%
7	"건조 자동용접(PE.SUBMERGED)"	126.9918	127.4997	0.40%
8	"건조자동용접 (PE.SEG ARC)"	52.75853	52.96954	0.40%
9	"T.M.H(P.E)"	672	669.9421	-0.31%
:				
58	"탑재용 先설치"	5805.077	5860.074	0.95%
59	"탑재 설치"	2381.167	2373.005	-0.34%
60	"족장운반수거(2YD D/H외작, 건조5부)"	499.5099	496.6158	-0.58%
61	"후행 해체"	6278.148	6230.201	-0.76%
62	"후행 해체(M/UNIT)"	24.4608	24.31904	-0.58%

Fig. 4 Group error rate

Fig. 4는 검증한 시리즈 호선의 그룹별 블록탑재 시수 에러를 나타낸 것이다. error rate의 항목을 보면 그룹마다 예측 값이 증가하거나 감소하는 것을 확인 할 수 있지만 대개 1% 이내로 큰 오차가 발생하지 않았다. 결국 형상이 일부 다른 시리즈 호선의 블록탑재 시수 추정에도 사용가능함을 확인 할 수 있었

다. 명확하게 하기 위해서는 전통적인 원단위 추정과 총 블록탑재 시수와의 비교가 필요하지만, 조선소의 요청으로 비교자료를 제시할 수 없었다.

4. 결론

본 연구는 조선소에 누적되어 있는 블록 탑재 시수 데이터를 기계학습에 적용을 시도해 본 것에 대해서 설명하였다. 탑재 시수를 예측하기 위해서 최근 조명되고 있는 기계학습을 적용했으며, 그 중에서도 Stacking과 Boosting, Bagging등 여러 모델들을 이용했다. 또한 Stacking ensemble 모델에 일반적으로 단일 학습 모델을 적용하는 반면 본 연구에서는 단일 Ensemble 모델들을 다시 Stacking ensemble 모델에 적용하여 오차율을 최소화 하고자 했다. 도출된 결과를 보면 다른 단일 Ensemble 모델들에 비해서 더 적은 오차율을 갖는 것을 확인 할 수 있었다. 그러나 C호선은 다른 호선들의 경우 과적합의 문제로 인해 저평가 되었지만 Decision tree가 가장 작은 오차율을 갖는 것을 확인 했다. 이는 기계학습에 적용되는 행의 수와 열의 수가 많은 영향을 끼치는 것을 반증하며, 데이터의 수가 다른 두 호선보다 0.5배 가까이 적은 C 호선의 결과물은 신뢰도가 떨어지는 것을 확인했다. 그리고 탑재시수를 예측하는 모델을 제작하는데 그치지 않고, Feature engineering을 통해서 학습에 사용되는 특성 항목들의 상관관계 분석을 시도했으며, 각 호선의 Stacking ensemble 모델의 값 범위 내에 들어오는 것을 확인했다. 마지막으로 Ensemble 모델로 예측한 블록탑재 시수 값들을 통해 데이터가 가장 많고 예측 결과가 좋았던 A호선의 총 블록탑재 시수의 합을 비교하여 Stacking ensemble 모델이 시리즈 호선의 블록탑재 시수 추정에 사용가능 함을 확인하였고 이로써 작업자의 개인 역량 또는 성향에 의존하지 않고 데이터만으로 빠른 시간 내에 시수를 예측할 수 있다는 것을 검증했다.

후기

이 연구는 2019년도 산업통상자원부 및 산업기술평가관리원 (KEIT) 연구비 지원(10067156)과 미래창조과학부 ICT융합 Industry4.0S(조선해양) 기술개발 사업 "선박구조물 생산을 위한 Shipyard 최적 생산설비 통합관리 시스템" 과제(S0249-19-1017)의 지원을 받아 수행하였습니다.

References

Baek, T.H., Oh, D.K., Jeong, Y.G., Lee, P.L. & Kwon, Y.W., 2018. Core techniques for smartization of shipbuilding process. *Bulletin of the Society of Naval Architects of Korea*, 55(4), pp.9-15

Baek, T.H., Chung, K.H. & Park, J.C., 1999. A study on the

application of resource leveling heuristic for ship erection scheduling. *Korean Institute Of Industrial Engineers*, 12(3), pp.354-361.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 26(2), pp.123-140.

Breiman, L., 1997. *Arcing the edge*. Technical Report 486. Statistics Department, University of California, Berkeley. CA.94720.

Breiman, L., 2001. Random forests, *Machine Learning*, 45, pp.5-32

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24, pp.417-441 & pp.498-520.

Kim, M.S. & Baek J.G., 2012. Fail prediction of DRAM module outgoing quality assurance inspection using ensemble learning algorithm. *IE Interfaces*, 25(2), pp.178-186

Lee, J.W. & Kim, H.J., 1995. Erection process planning & scheduling using genetic algorithm. *Journal of the Society of Naval Architects of Korea*, 32(1), pp.9-16

Oh, D.K., Jeong, Y.H., Shin, J.G. & Choi, Y.R., 2009. Construction cost estimation on the initial design stage of naval ships based on a product configuration model. *Journal of the Society of Naval Architects of Korea*, 46(3), pp.351-361.

Rokach, L. & Maimon, O., 2005. Top-down induction of decision trees classifiers – a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 35(4), pp.476-487.

Schapire, R., 1990. The strength of weak learnability. *Machine Learning*, 5, pp.197-227.

Wolpert, D., 1992. Stacked Generalization, *Neural Network*, 5(2), pp.241-259.



권혁천



유원선