공 학 석 사 학 위 논 문

머신러닝 회귀 모델 알고리즘을 적용한 기업의 판매량 예측 연구 및 분석

2021년 2월

부 경 대 학 교 대 학 원

정 보 시 스 템 학과

공 학 석 사 학 위 논 문

머신러닝 회귀 모델 알고리즘을 적용한 기업의 판매량 예측 연구 및 분석

지도교수 김 창 수

이 논문을 공학석사 학위논문으로 제출함.

2021년 2월

부 경 대 학 교 대 학 원

정 보 시 스 템 학과

정 세 훈

정세훈의 공학석사 학위논문을 인준함.

2021년 2월 19일

위 원 장 경영학박사 김 하 균 (인)

위 원 이학박사 이 경 현 (인)

위 원 공학박사 김 창 수 (인)

< 차 례 >

<그림 차례>

<표 차례>

A study on machine learning regression model for prediction corporate sales volume

Se Hun Jeong

Department of Information System
Pukyong National University

Abstract

Businesses are a profit-seeking group that puts a lot of effort into maximizing profits. The basic way to maximize profits is to minimize production costs and maximize margins. However, malicious inventory that occurs in the process of leaving margins eats up the margins that are difficult to leave. Companies are making efforts to minimize inventory by selling at a lower price or by deriving expected sales to deal with inventory. In order to predict sales volume, companies are using previously accumulated data. Therefore, in this paper, we apply machine learning algorithms to predict future sales and inventory based on previous sales performance, and compare the performance of each algorithm.

I. 서론

1. 연구 배경 및 필요성

기업은 이윤을 극대화하기 위한 방법 중 하나로 이전의 데이터를 통하여 미래를 예측해왔다. 그리고 그 예측이 맞았을 때는 막대한 이윤을 가져온다. 1970년대 프랑스는 당시에 한 가구에 자녀 한 명 이하였던 저출산국이었고 심각한 저출산으로 인해 프랑스 경제가 붕괴될 것이라 예측하였다. 이에 대한 대비책으로 이민정책을 내놓거나 막대한 자본을 투입하여 현재는 충분한 인구수를 갖춘 국가가 되었다. 노르웨이 또한 앞으로 닥칠 석유 고갈을 예측했고, 이를 대비해 석유를 대체할 다른 사업을 준비하였다[1]. 개개인도 또한 마찬가지로 자신의 노후를 위해 미래를 예측하고 투자 혹은 재테크를 하기도 한다. 앞의 사례는 개인이나 국가의 경우이지만 기업이라고 다를 바가 없다. 1970년대부터 미국이나 유럽의 국가에서 IT 산업, 자동차 산업, 화학 산업 등에서 미래예측 활동을 펼치고 있으며 기업 미래예측 전문 컨설팅 기업까지 등장하였다. 대부분의 기업들은 최소 2~5년, 최대 20~30년의 미래 예측 기간을 설정하고 있으며 평균 5~15년의 미래를 예측하는 것으로 알려져 있다[2].

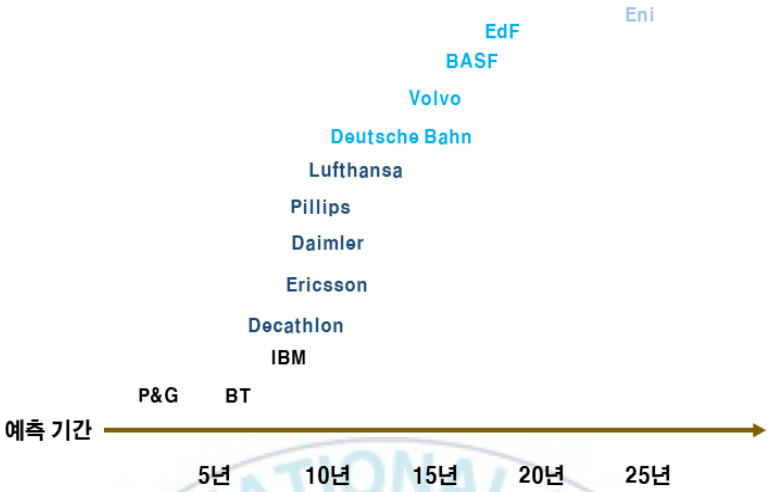

그림 1. 기업 미래예측 기간의 분포

10년 미만의 미래예측을 시행하고 있는 기업들은 P&G(Procter & Gamble), BT(British Telecommunications), IBM(International Business Machine) 등이 있으며 10~20년 사이를 차지하고 있는 기업들은 Decathlon, Ericsson, Daimler, Pillips, Lufthansa 등이 있고 25년 이상으로는 Eni가 자리하고 있다. 대부분의 저명한 기업들이 미래예측을 시행하고 있다는 것을 의미하며 그만큼 기업이 장기간 존속하기 위해서 많은 노력을 기울이고 있다는 것을 유추해 볼 수 있다.

미국의 신용평가회사 Standard & Poor's에서 제시한 미국의 대표 주가지수 S&P 500에서 책정한 미국 500대 대기업의 기업 평균 수명을 살펴보자 [3].

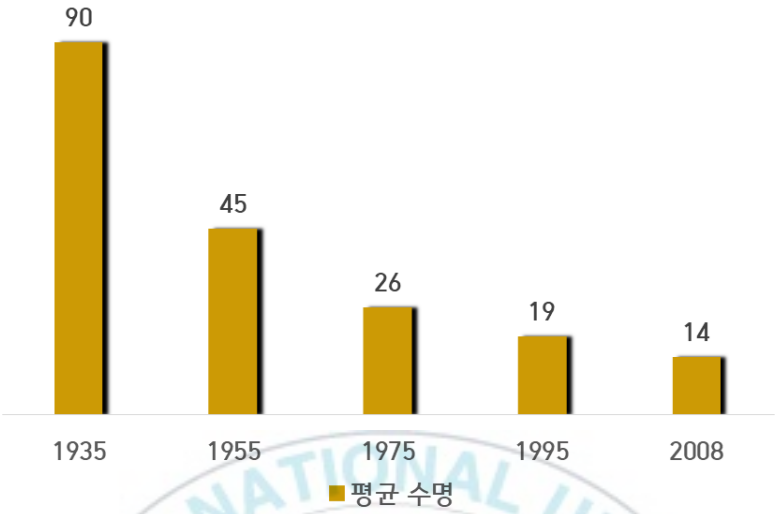

그림 2. 기업의 연도별 평균 수명

그림 2와 같이 1935년 이후로 평균 수명이 꾸준히 감소하고 있는 추세이며 2008년에는 1935년에 비해 약 84.5% 감소하였다[4]. 2020년에는 평균 수명이 10년 내외까지 더 단축될 수 있다는 분석도 나왔다. 미국을 대표하던 GM(General Motors)와 GE(General Electric)은 1900년경에 설립돼 100년 이상 미국 기업순위 최상위권에 머물러 있었지만 GM은 2019년, GE는 2016년에 경제지 포춘이 선정하는 기업순위 10위권 내에서 제외되었다고 전해진다. 이 현상에 대한 원인으로는 기술 발전 속도가 빠르고 환경이 급변하기 때문에 현재는 최상위권에 등재되어 있더라도 과거 19~20세기경에 비해 빠르게 쇠락의 길을 걷을 수 있다고 분석할 수 있다.

따라서 신속하고 정확하게 트렌드(Trend)를 읽어 시대의 흐름에 뒤쳐지지 않고 미래를 예측하는 것이 기업의 장기존속에 대한 지름길이라는 것이다.

2. 연구 목표

본 연구에서는 여러 분야, 여러 기업의 과거의 데이터를 조사하고 수집하여 적합한 머신 러닝 알고리즘을 탐색, 적용시킨다. 이 때 각각의 알고리즘의 성능을 파악하기 위해 Jupyter 플랫폼 상에서 Python 프로그래밍을 통하여 학습 샘플의 수, 테스트용 샘플의 수, 시간 단위를 적절히 조절하여 MSE(Mean Squared Error : 평균 제곱 오차), RMSE(Root Mean Squared Error : 평균 제곱근 오차), MAPE(Mean Absolute Percentage Error : 평균 절대 백분율 오차)를 파악 및 기록하는 것으로 연구를 진행한다[5][6].

II. 관련연구

1. 머신 러닝(Machine Learning)

가. 머신 러닝의 정의

기계 학습(機械 學習)이라고도 불리우는 머신 러닝은 경험을 쌓아 자동으로 개선하는 컴퓨터 알고리즘의 연구를 말한다. 머신 러닝은 인공지능(AI : Artificial Intelligence)에 속해 있는 개념으로 컴퓨터가 스스로 학습할 수 있도록 하는 알고리즘을 개발하는 분야이다. 데이터의 “알려지지 않은 속성”을 발견하는 것에 중점을 두는 데이터 마이닝과는 약간의 차이가 있는데 머신 러닝은 훈련 데이터(Training Data)를 통해 학습된 “알려진 속성”을 기반으로 예측을 하는 것에 초점을 둔다. 현재 사용되고 있는 알고리즘은 본 연구에서 사용할 서포트벡터머신(Support Vector Machine), 랜덤포레스트(Random Forest)와 그 외의 인공신경망, 유전 알고리즘, 퍼셉트론 등이 있다.

나. 머신 러닝의 역사

베이즈의 정리가 발견된 1763년부터 시작된 머신 러닝은 꽤나 오래된 역사를 가지고 있다. 컴퓨터가 발명된 1946년부터 머신 러닝의 상위 개념인 인공지능(AI)이 개발될 수 있는 주춧돌이 생겼으며 도중에 혹한기가 2~3번

찾아왔지만 한 번의 혁신으로 세간의 주목을 다시 끌기도 해 2020년 현재 오늘날의 머신 러닝을 있게 해주었다. 다음 표 1은 머신 러닝에 대한 주요 사건들을 나열한 것이다.

표 1. 머신 러닝의 역사

연도	내용
1763년	머신러닝의 토대가 되는 베이즈의 정리 발견
1946년	최초의 컴퓨터 ENIAC 발명
1950년	Alan Turing이 인공지능(AI)을 감지하기 위한 튜링 테스트(Turing Test) 설계
1952년	Arthur Samuel이 체커 게임 방법을 배우는 최초의 머신러닝 프로그램 개발
1956년	인공지능(Artificial Intelligence) 용어의 등장
1958년	최초의 인공 신경망 '퍼셉트론(Perceptron)' 설계
1967년	최근접 이웃 알고리즘(Nearest Neighbor algorithm) 탄생
1985년	Backpropagation 알고리즘의 발견으로 뇌의 모델인 Neural Network에 대한 연구가 촉진됨
1990년	머신러닝에 대한 통계적 접근 방식인 Support Vector Machine의 신개념 등장
2006년	본격적으로 상용화하기 시작. 딥 러닝(Deep Learning)이라는 용어 탄생

다. 각 머신러닝 알고리즘의 장단점

표 2. 각 머신러닝 알고리즘의 장단점

	장점	단점
k-NN	• 사용하기 간단한 모델이다. • 많은 조정 없이 좋은 성능을 발휘한다.	• 많은 데이터를 처리할 때 시간이 다소 소요된다.
나이브 베이즈 분류기	• 단순하고 빠르며 매우 효과적이다. • 매우 크거나 작은 데이터들을 처리할 때도 용이하다. • 노이즈와 결측 데이터(Missing Data)가 있어도 학습이 가능하다.	• 많은 수치로 이루어진 데이터를 학습하기 어렵다. • 특징(feature)이 서로 독립적이어야 한다.
결정 트리 (Decision Tree)	• 시각화하기 용이하며 직관적으로 구조를 파악할 수 있다. • 연속형 데이터를 처리하기에 용이하다.	• 데이터 수가 약간의 변화가 있을 경우 전혀 다른 결과가 나올 수 있다. • 데이터의 특징이 수직/수평적이지 않을 경우 분류율이 떨어진다.
랜덤 포레스트 (Random Forest)	• 결정 트리의 단점을 보완한 모델로서 과적합(overfitting)이 어느 정도 보완되었다. • 훈련이 빠르다.	• 수많은 결정 트리를 사용하기에 메모리 사용량이 매우 많다.
서포트 벡터머신 (Support Vector Machine)	• 분류, 예측문제를 동시에 쓸수 있다. • Neural Network 기법에 비해서 과적합이 덜 일어난다. • 정확도가 높고 사용하기 쉽다.	• 여러 테스트를 거쳐야 최적화된 모형을 만들 수 있어 시간이 오래 걸린다.

현재 사용되고 있는 머신 러닝 알고리즘들은 각각의 장단점이 있다. 의사결정나무(Decision Tree)는 분석 과정을 직관적으로 이해할 수 있지만 학

습 데이터를 과하게 학습하여 과적합(Overfitting)이 일어나기 쉽다. 그래서 이러한 단점을 보완하여 나온 모델이 수많은 의사결정나무를 학습시키는 랜덤 포레스트(Random Forest)가 등장했다. 서포트 벡터 머신(Support Vector Machine)은 문자 및 영상인식, 문서분류 등 여러 분야에서 우수한 성능을 보여주는 대표적인 데이터마이닝 기법 중 하나이며 분류, 회귀로 나누어 사용한다[7].

라. 알고리즘 동작 원리

본 항목에서는 적용시킬 알고리즘의 동작 원리를 정리해보았다.

(1) K-NN

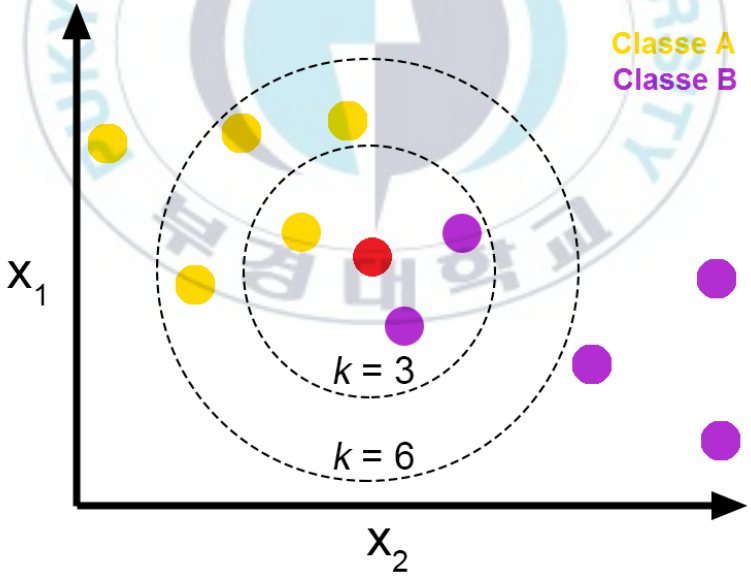

그림 3. K-NN의 작동 원리

그림 3에서 K-NN은 k라는 매개 변수를 입력시켜 작동시킨다. 그림과 같

이 클래스가 2개일수도 혹은 그 이상일수 있으며 k는 가장 가까운 이웃의 수를 의미한다. 이 때 가장 가까운 이웃을 판단하는 척도는 유클리드 거리 측정법(Euclidean distance measure)을 사용한다.

표 3. 유클리드 거리 계산

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

k가 3일 경우 위와 같은 식을 통해 가장 인접한 이웃 3개를 택하여 선호도를 도출해낼 수 있다. 그림 3의 경우 k가 3일 경우에는 Class B에 치중되어 있지만 6으로 변경하자 Class A로 치중되는 것으로 결과가 변화할 수도 있다는 것을 알 수 있다. 이 알고리즘은 선호도 이외에 실제로 고속도로 통행시간을 예측하는 등 현재까지의 데이터를 기반으로 미래를 예측할 수 있다.

(2) 랜덤 포레스트

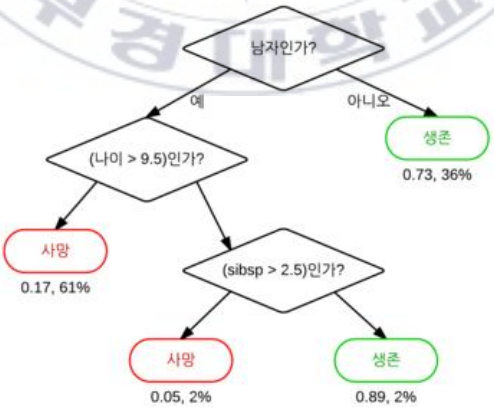

그림 4. 의사결정트리 구조

먼저 랜덤 포레스트를 언급하기 전에 기반이 되는 의사결정트리(Decision Tree)에 대해 알아갈 필요가 있다. 입력 변수를 기반으로 목표 변수의 값을 예측하는 것이 본 모델을 주 목적이며 그림 4와 같이 트리 구조로 구성되어 있다. 그러나 과적합(overfitting)이 될 가능성이 높다는 단점이 있어 이를 보완하기 위해 그림 5와 같이 여러 개의 의사결정트리를 형성하여 각 트리에 데이터를 통과시키고 각 트리의 분류한 결과를 기반으로 선호도가 가장 높은 분류 결과를 최종적으로 선택한다. 본 방식은 일부 트리는 과적합이 되지만 우선적으로 소거할 수 있다.

그림 5. 랜덤 포레스트 구조

(3) 서포트 벡터 머신

서포트 벡터 머신이란 결정 경계면(Decision Boundary), 즉 분류하기 위한 기준 선을 정의하는 모델을 말한다. 2차원, 3차원, 혹은 그 이상인 다차원에서도 적용 가능하며 n차원의 경우 n-1차원의 경계를 만들어낸다. 서포트 벡터 머신은 선형 분류, 비선형 분류와 더불어 선형 회귀, 비선형 회귀에서도 적용할 수 있는 범용성이 큰 모델이다. 그림 6의 경우 선형 분류 모델에 해당하며 마진(margin)을 최대화하는 것이 목표이다. 반대로 그림 7의 회귀 모델의 경우 일정한 마진 오류 안에서 도로 폭을 최대화하는 대신 도로 안에 최대한 많은 샘플이 들어가도록 학습한다. 도로 폭은 epsilon이라는 하이퍼파라미터로 조절한다.

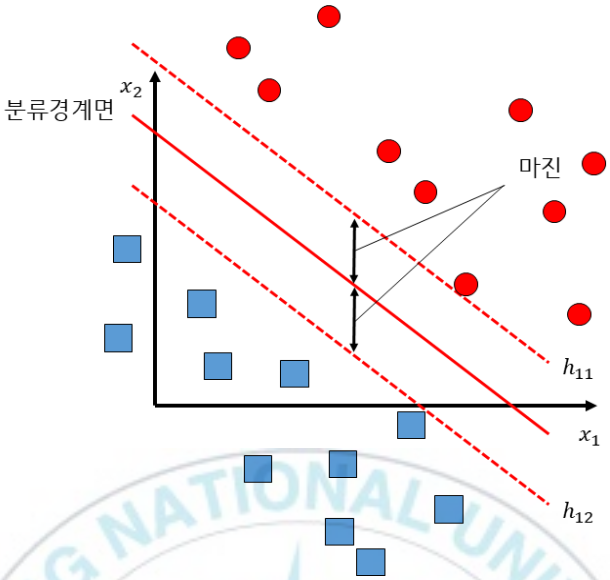

그림 6. 서포트 벡터 머신 – 분류(classification)

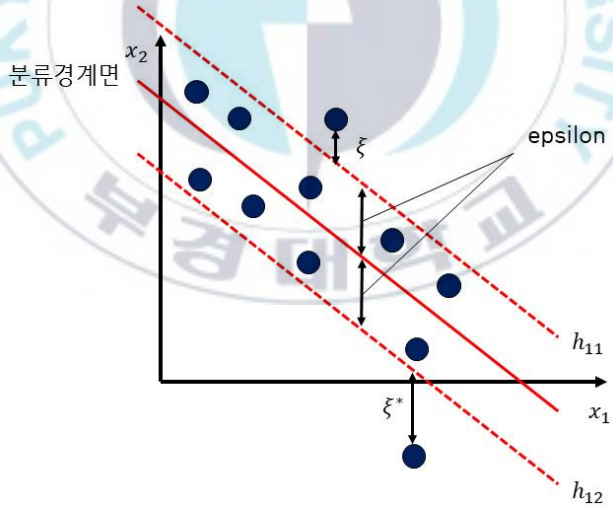

그림 7. 서포트 벡터 머신 – 회귀(Regression)

본 연구에서는 예측 모델에 적용할 회귀 모델을 사용하며 현재 널리 사용 중인 머신러닝 알고리즘 중에 가장 뛰어난 성능을 가진 것으로 알려져 있

다.

2. 머신 러닝의 현 주소

가. 네트워크 분야

네트워크 머신러닝은 아직 초기 단계이지만 지속적으로 연구가 이루어지고 있다. 유럽에서는 머신러닝을 활용한 3가지의 5G 프로젝트가 진행중이며 CogNet, SPEED-5G, SELFNET이 있다. CogNet 프로젝트는 머신 러닝 기반 5G 네트워크 관리, SPEED-5G 프로젝트는 고속 이동하는 물체(object)간 통신에 적용시킬 머신 러닝, SELFNET 프로젝트는 클라우드 컴퓨팅(Cloud Computing) 인공 지능 등의 기술을 사용하여 실시간 자동 5G 네트워크 관리를 연구했다[8]. 그 외 스페인에서는 강화학습을 적용하여 네트워크 라우팅 최적화하는 연구가 이루어졌으며 한국에서도 디도스(DDos)와 같이 이상적으로 동작하는 노드가 존재하는 네트워크를 머신 러닝 기법으로 분류하는 방법 등의 논문이 존재한다[9].

나. 보안 기술 분야

사용자가 비정상적인 패턴, 즉 이상한 행동을 취할 때 외부 경로로부터의 비인가 침입인지 탐지하는 연구로서 어느 한 논문에서는 사이버 보안상의 머신 러닝과 딥 러닝의 효용성에 대해 다루고 있다. 이 논문에서는 머신 러닝 기술을 보안 분야에 적용시킬 수는 있지만 일부 작업에 불과하며 장

단점에 대해 파악해야하며, 또한 사람의 관리가 요구되어 보안 시스템 전체를 자동화하는 것은 현재로선 어려울 것이라 말하고 있다[10].

III. 머신러닝을 적용한 수요 및 판매량 예측

1. 실험 방식

가. 실험 환경

Product_Code	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	...	W44	W45	W46	W47	W48	W49	W50	W51	MIN	MAX
P1	11	12	10	8	13	12	14	21	6	14	...	8	10	12	3	7	6	5	10	3	21
P2	7	6	3	2	7	1	6	3	3	3	...	5	1	1	4	5	1	6	0	0	10
P3	7	11	8	9	10	8	7	13	12	6	...	5	5	7	8	14	8	8	7	3	14
P4	12	8	13	5	9	6	9	13	13	11	...	3	4	6	8	14	8	7	8	2	19
P5	8	5	13	11	6	7	9	14	9	9	...	7	12	6	6	5	11	8	9	3	18
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
P815	0	0	1	0	0	2	1	0	0	1	...	1	0	0	1	0	0	2	0	0	3
P816	0	1	0	0	1	2	2	6	0	1	...	4	2	4	5	5	5	6	5	0	7
P817	1	0	0	0	1	1	2	1	1	0	...	0	2	2	0	0	0	4	3	0	4
P818	0	0	0	1	0	0	0	0	1	0	...	0	1	1	0	0	0	2	0	0	2
P819	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	3

811 rows × 54 columns

그림 8. 제품별 주간 판매량 DataFrame

실험은 'Jupyter Notebook'에서 python 언어로 진행되었다. 사용한 알고리즘은 각각 KNN(K-Nearest Neighbor), 랜덤포레스트(Random Forest), SVM(Support Vector Machine)으로 출력할 데이터의 형식이 연속된 수치

이기에 회귀(Regression) 방식을 채택하였다. 입력할 데이터는 그림 3과 같이 UCI Machine learning 사이트에서 제공하는 것으로 임의의 각 제품들의 주간 판매량을 정리해둔 것이다[11]. 그림 4는 시스템 구현 환경에 대해 정리해 둔 것이다. 라이브러리 항목에서 Sklearn은 각종 회귀 분석 알고리즘을 오픈소스로 사용할 수 있게 되어있다.

표 4. 시스템 구현 환경

구분	종류
OS	• Windows 10 64bit
Platform	• Jupyter Notebook
Language	• Python 3.7.4
Library	• Numpy, Pandas, Sklearn, Matplotlib

나. 변량 요소

총 819개의 상품의 51주간의 주간 판매량을 기록했으며 대체로 10개 내외의 판매량을 보이고 있다. 각 알고리즘을 평가할 지표는 각각 MSE(Mean Squeared Error), RMSE(Root Mean Squared Error), 예측 값이며 추가적으로 두 알고리즘 랜덤포레스트와 SVM의 결과를 합하여 단일 알고리즘의 결과와 비교 분석한다. 고정 요소와 변량 요소는 총 3가지로 샘플의 개수, 주(week) 단위, 학습 횟수이며 각각 각 기본 값은 학습 샘플 10개와 테스트 샘플 2개, 1주, 10회로 설정하였다. 또한 각각의 학습 결과 MSE와 RMSE의 평균을 구하여 산정하였다.

표 4. MSE(Mean Squared Error), RMSE(Root Mean Squared Error) 계산식

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2 \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2} \quad (2)$$

2. 실험 결과

가. 변량 요소(샘플의 개수)

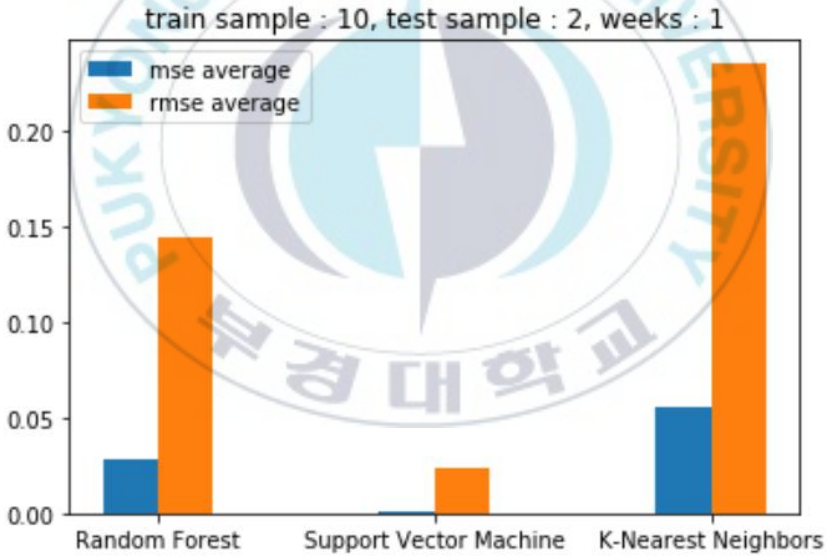

그림 9. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 10, 테스트 샘플 : 2, 주 단위 : 1)

위 그림 9의 예시의 경우 P1의 앞 10주간의 판매량 데이터를 학습 샘플로 11~12주간의 판매량을 테스트 샘플로 학습시킨 결과이다. Support Vector

Machine이 가장 작은 값의 MSE, RMSE를 나타냈고 다음 Random Forest, KNN 알고리즘 순으로 성능 지표를 나타내었다. 다음은 테스트 샘플은 2개로 고정, 학습 샘플을 20개로 늘려본 결과이다.

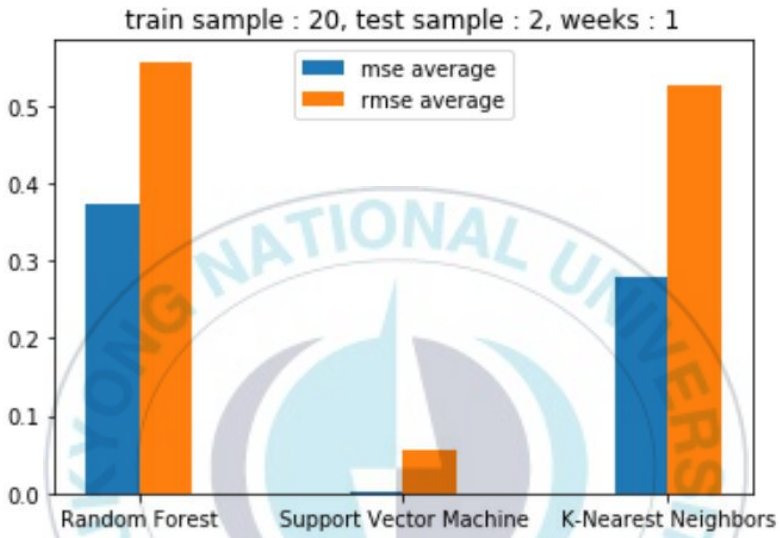

그림 10. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 20, 테스트 샘플 : 2, 주 단위 : 1)

세 알고리즘 모두 MSE, RMSE 수치가 증가되었고 이전과는 다르게 랜덤 포레스트의 성능이 KNN에 비해서 떨어지는 것을 알 수 있다. 좀 더 정확한 분석을 위해 학습 샘플을 30으로 조정했다. 다음은 테스트 샘플 개수의 변화에 따른 결과를 나타낸 것이다.

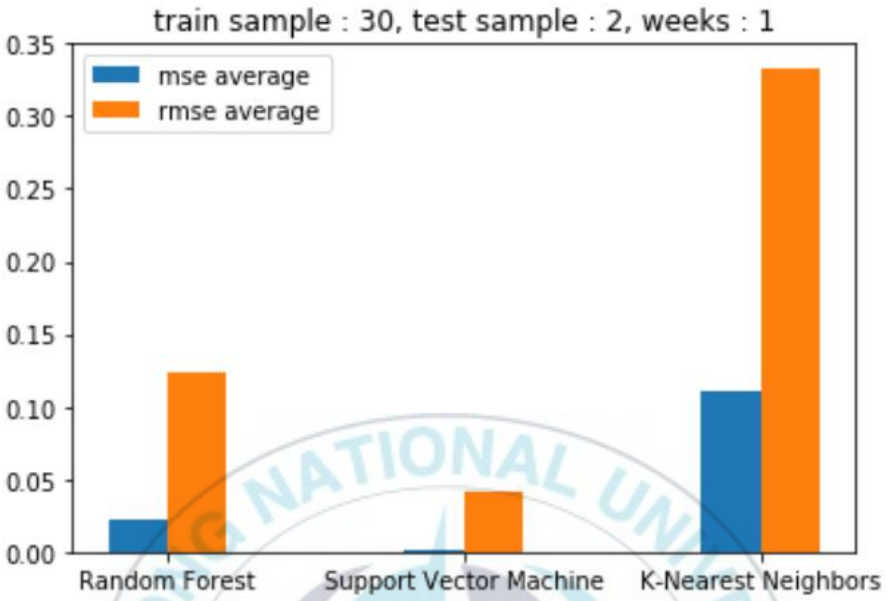

그림 11. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 30, 테스트 샘플 : 2, 주 단위 : 1)

학습 샘플이 20개일 경우에 비해 MSE, RMSE가 감소되었지만 여전히 10개일 경우에 비해 높은 수치를 나타냈다. 서포트벡터머신은 압도적인 성능을 보이고 있으며 이번엔 랜덤 포레스트가 KNN에 비해 우위를 차지했다.

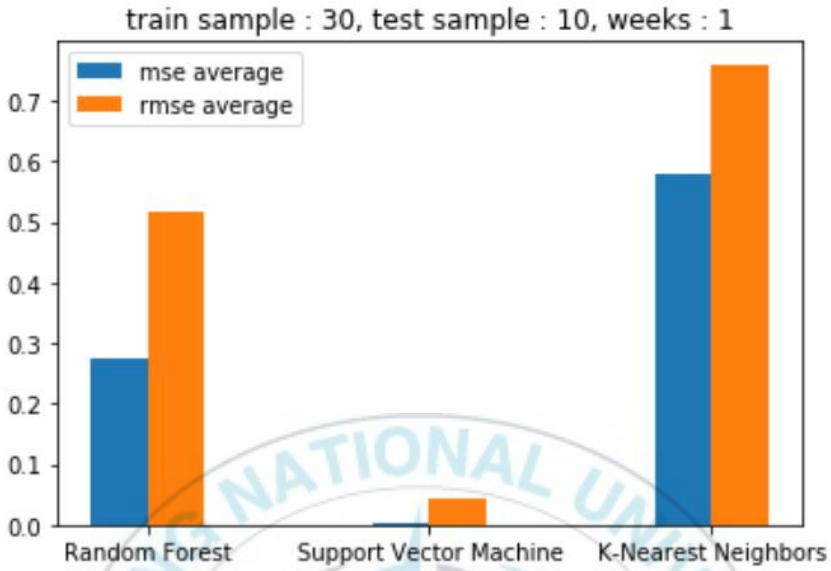

그림 12. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 30, 테스트 샘플 : 10, 주 단위 : 1)

그림 12의 실험과 비교하기 위해 테스트 샘플 개수를 2개에서 10개로 늘려 보았다. MSE 수치가 서포트벡터머신은 큰 변화가 없었지만 랜덤포레스트는 약 10배, KNN은 약 6배 가까이 증가하였다.

나. 변량 요소(주 단위)

이전 항목에서는 1주 단위의 데이터를 학습시켰지만 주 단위를 2주, 4주 단위로 늘리고자 한다. 데이터 상에서는 51주까지 구현되어 있기 때문에 학습 샘플과 테스트 샘플을 적게 설정하였다.

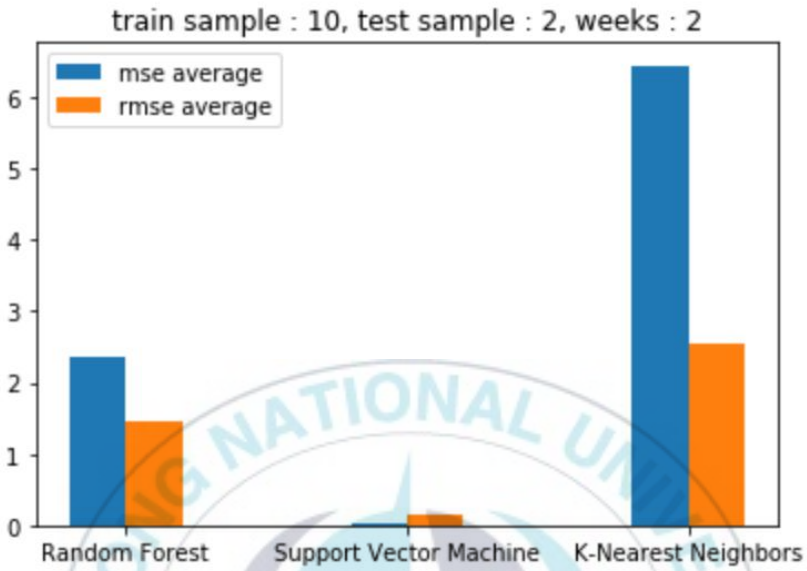

그림 13. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 10, 테스트 샘플 : 2, 주 단위 : 2)

그림 13와 동일한 학습 샘플과 테스트 샘플 개수지만 주 단위를 2주로 설정하였다. 그림 13에 비해 MSE 수치가 랜덤 포레스트와 KNN 모두 10배 이상 급증한 것으로 나타났다. 그에 비해 서포트벡터머신의 경우 큰 변화가 없었다.

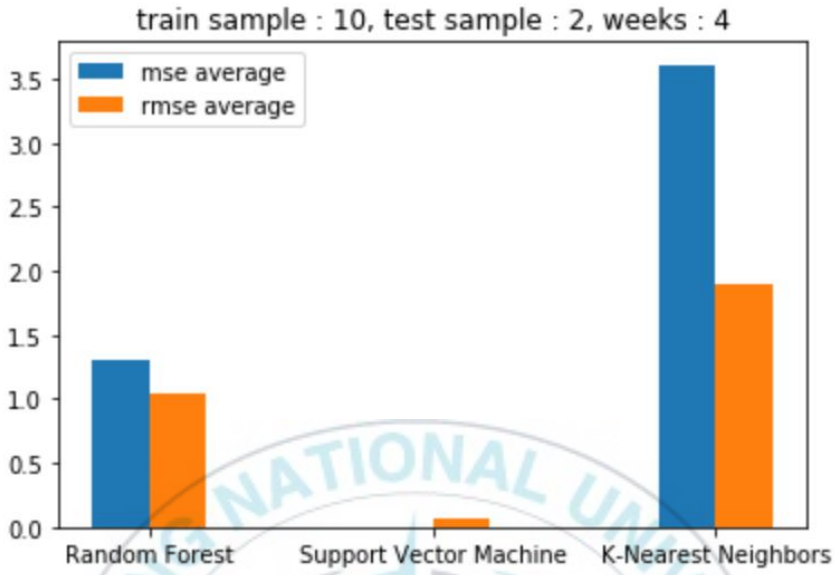

그림 14. 알고리즘 별 MSE, RMSE 평균(학습 샘플 : 10, 테스트 샘플 : 2, 주 단위 : 4)

그러나 그림 14와 같이 4주 단위로 학습 시켰을 때, MSE 수치가 다시 하락하는 경향을 보였다. 주 단위를 3주, 5주로 학습 시켜보았지만 적은 데이터로 인해 등락의 여부를 파악하기 어려웠다.

다. SVM과 랜덤포레스트의 결합

기존에 없던 방식을 사용하여 오차율을 구하고자 한다. 위에서는 각각의 알고리즘의 성능을 파악하고자 한 반면에 본 항목에서는 SVM과 랜덤포레스트의 오차율을 결합하고자 한다. SVM은 선형성과 비선형성, 랜덤포레스트는 비선형성으로 각각 다른 성격의 알고리즘이며 비선형성이 선형성에 비해 더 복잡한 관계를 표현할 수 있다고 알려져 있다. 랜덤포레스트를 단독으로 사용할 때 SVM보다 성능이 떨어지는 결과가 나왔지만 선형의 SVM과 비선형의 랜덤 포레스트 모델을 결합했을 때 어떠한 시너지를 낼지 알아보고자 한다.

표 5. SVM과 랜덤포레스트의 결합, 오류 계산식

$$total = \sum \frac{|y_{test} - y_{prerf}|}{y_{test}} + \frac{|y_{test} - y_{presvm}|}{y_{test}} \quad (1)$$

$$error_{rf} = \sum \frac{|y_{test} - y_{prerf}|}{y_{test}} \quad (2)$$

$$error_{svm} = \sum \frac{|y_{test} - y_{presvm}|}{y_{test}} \quad (3)$$

$$final = \frac{error_{rf}}{total} \times y_{prerf} + \frac{error_{svm}}{total} \times y_{presvm} \quad (4)$$

표 5의 경우 각 기법으로 학습시킨 후 도출된 예측값(y_{prerf}, y_{presvm})을 포함시켰다. y_{test}는 실제 측정된 값이며 예측값과 측정값의 차를 예측값으로 나눈 값의 합이 각각 식(2), 식(3)의 $error_{rf}$, $error_{svm}$에 해당된다. 식 (1)의 $total$은 실제로 $error_{rf}$와 $error_{svm}$을 의미하며 최종 예측값인 $final$은 $error_{rf}$와 $error_{svm}$의 비율에 각각의 예측값을 곱한 것의 합이다. 결론적으로 식 (4)는 랜덤포레스트 혹은 SVM의 오차가 작을 경우 해당 알고리즘으로 예측된 값이 크게 반영되는 공식이다.

다음 표 5는 식 (1), (2), (3), (4)를 기반으로 5가지의 시나리오를 작성하여 표로 나타낸 것이다.

표 6. SVM과 랜덤포레스트의 결합, 결과

Condition	y_test	y_prerf	y_presvm	final
[train = 10, test = 2, weeks = 1]	[11, 14]	[11.3, 13.9]	[11.03333327, 13.99333327]	[11.27333337, 13.90583338]
[train = 20, test = 2, weeks = 1]	[5, 11]	[4.7, 10.8]	[5.07777768, 11.01111109]	[4.77777768, 10.81111109]

[train = 30, test = 2, weeks = 1]	[7, 10]	[6.5, 9.9]	[7.0555557, 10.02222235]	[6.5555557, 9.92222235]
[train = 30, test = 3, weeks = 1]	[7, 10, 12]	[6.7, 9.9, 12.0]	[7.0555557, 10.02222235, 12.00000012]	[6.7555557, 9.92222235, 12.00000012]
[train = 10, test = 2, weeks = 2]	[14, 16]	[12.9, 13.5]	[13.90518395, 15.82485152]	[12.97976757, 13.65221368]

표 6의 첫 번째 시나리오의 경우 y_test와 y_prerf, y_presvm, final 간의 차의 합이 y_prerf, final, y_presvm 순으로 높게 나왔다. 두 번째, 세 번째, 네 번째, 다섯 번째 시나리오도 마찬가지의 결과를 보였다.

다음 표 7은 표 6의 결과를 기반으로 MSE 수치를 최종적으로 정리한 표이다.

표 7. SVM과 랜덤포레스트의 결합, MSE 수치

	Random Forest		SVM		RF + SVM	
Condition	MSE	RMSE	MSE	RMSE	MSE	RMSE
[train = 10, test = 2, weeks = 1]	0.050000 00000000 0176	0.223606 79774997 935	0.000577 77608889 28935	0.223606 79774997 935	0.041789 24173889 047	0.223606 79774997 935
[train = 20, test = 2, weeks = 1]	0.064999 99999999 999981	0.254950 97567963 887	0.003086 41191358 52208	0.254950 97567963 887	0.042530 88991358 516	0.254950 97567963 887
[train = 30, test = 2, weeks = 1]	0.129999 99999999 998	0.360555 12754639 89	0.001790 13432100 6258	0.360555 12754639 89	0.101790 04932100 615	0.360555 12754639 89
[train = 30, test = 3, weeks = 1]	0.033333 33333333 328	0.182574 18583505 522	0.001193 42288067 56385	0.182574 18583505 522	0.021934 12621400 8896	0.182574 18583505 522
[train = 10, test = 2, weeks = 2]	3.729999 99999999 5	1.931320 79158279 66	0.019833 53669195 661	1.931320 79158279 66	3.276487 40780142 54	1.931320 79158279 66

VI. 실험 결과 분석

1. 샘플의 개수 변화에 따른 결과

샘플의 개수는 각각 학습용 샘플, 테스트용 샘플로 나눠져 있으며 주 단위는 1주로 고정하고 학습용 샘플에 변화를 주어 MSE 수치를 구해보았다. 앞서 실험한 데이터를 살펴보면 10, 20, 30개로 점차 늘렸을 때 서포트 벡터 머신이 MSE 수치가 거의 0에 근접해있어 관련 연구 항목에서 예상했던 바와 같이 가장 압도적이고 뛰어난 성능을 보이고 있으나 나머지 회귀분석 모델인 랜덤포레스트와 K-NN은 성능 순위가 경우에 따라 변동되는 추세를 보인다. 그러나 데이터의 양이 적어 전체적으로 성능을 판단하기 어려운 것으로 보인다. 테스트용 샘플을 10개로 변화를 주었을 때 역시 마찬가지로 서포트 벡터 머신의 성능이 월등히 뛰어났고 나머지 두 모델도 이전과 비슷한 양상을 보였다. 최고 MSE 수치의 경우 각각 0.25, 0.55, 0.35, 0.75에 달했으며 큰 격차는 없는 것으로 보인다.

2. 주 단위의 변화에 따른 결과

샘플의 개수가 총 52주치이므로 학습용 샘플과 테스트용 샘플의 개수를 10, 2개로 고정하고 주 단위를 1, 2, 4주로 변화를 주어 MSE 수치를 구해보았다. 1주의 경우 최고 MSE 수치가 약 0.6에 달했고 이전 항목과 같이

서포트 벡터 머신의 성능이 압도적이었다. 그러나 주 단위를 2주로 변화를 주었을 때 최고 MSE 수치가 약 6.0으로 이전보다 약 10배 급등했으며 모델 성능 순위는 이전과 같았다. 주 단위를 4주로 변화를 주었을 때는 역시 성능 순위는 같았고 최고 MSE 수치가 약 3.5로 감소되었다. 주 단위가 1주일 경우에는 모두 최고 MSE 수치가 1.0 이하에 머물렀지만 주 단위가 2주 이상 늘어난 순간에는 급등하는 것으로 보인다.

3. SVM과 랜덤포레스트의 결합

본 항목은 표 8의 데이터를 기반으로 분석한다.

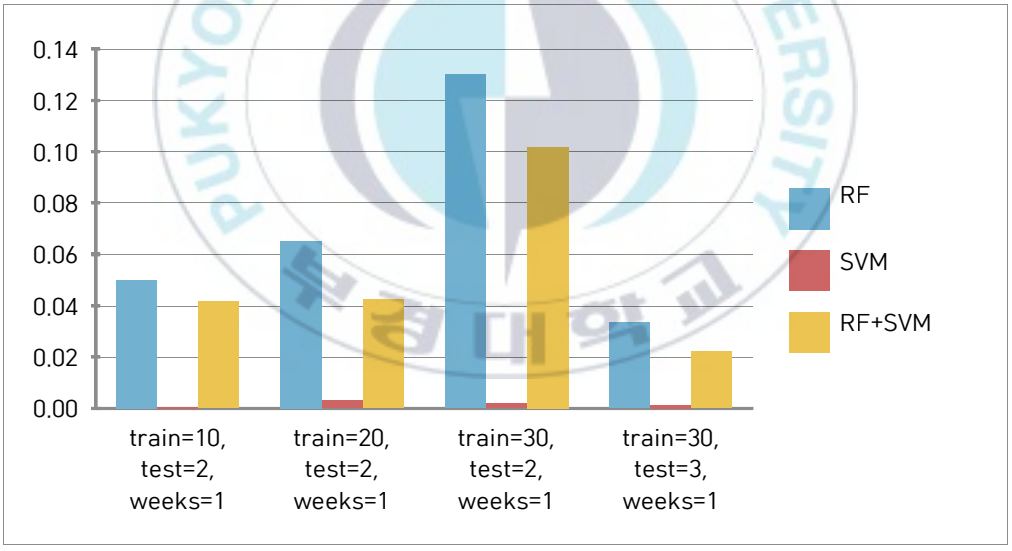

그림 15. SVM과 랜덤포레스트의 결합, 그래프

그림 15의 그래프에서 세 모델이 모든 조건에서 비슷한 양상을 보여주고 있다는 것을 알 수 있다. 학습용 데이터 샘플의 개수에 변화를 줄 때 RF와 결합 모델은 전체적으로 성능이 떨어지고 테스트용 샘플에 변화를 주었

을 때 다시 급격히 성능이 상승했다. 그림 15에는 포함되지 않았지만 주 단위를 변화시켰을 때 최고 MSE 수치가 약 74배 증가하였다.

V. 결론

본 연구에서는 데이터셋을 제공하는 사이트(Kaggle, UCI Machine Learning 등)에서 데이터를 수집하고, 수집된 정보를 회귀 알고리즘을 통해 학습시켜 예측 값을 도출하였다. 위의 실험에 따르면 SVM은 MSE 수치가 0에 근접해 뛰어난 성능을 보이지만 나머지 랜덤포레스트와 K-NN은 SVM에 비해 확실히 떨어지는 성능을 보인다. 또한 주 단위를 변경했을 때 최고 MSE 수치가 급등한 것으로 보아 데이터 쌍이 증가할수록 성능이 급격히 떨어진다. 그러나 이는 구할 수 있는 데이터 샘플의 수가 적은 한계에 봉착했기 때문에 조건(Condition)의 경우의 수가 적어 성급히 판단을 내리기엔 어렵다. 만약 데이터 샘플이 무한하다고 가정하면 적어도 수십, 수백 개의 조건을 설정해 실험하고 그 추세를 파악해야 더욱 정확한 결론을 내릴 수 있을 것이다.

본 연구를 실생활에 적용시키기 위해서는 해당 과거 기업의 제품 판매량을 크게는 연도별, 월별 작게는 주별, 일별로 수집하고 많은 데이터 샘플을 가지고 연구를 진행시켜 내린 결론에 따라 최적의 테스트용 데이터 샘플, 학습용 데이터 샘플, 시간 단위, 더 나아가 학습 횟수를 적용시키면 최적의 예측 정확도를 찾을 수 있을 것이다. 따라서 기업의 입장에서는 예측된 판매량만큼의 제품을 생산해 악성 재고량을 최소화시켜 최대의 이익을 가져올 수 있을 것이다.

<참 고 문 헌>

[1] 최항섭, “미래예측 방법론과 디지털시대의 디자인 미래”, 한국미래디자인연구원, 2010. 4.

[2] 김영환, 성경모, 홍성주, “한국기업의 미래연구 현황분석 및 역량제고 방안”, 과학기술정책연구원, 2014. 12.

[3] 윤성중, 김준식, “분산프리미엄의 수익률 예측에 대한 연구 : S&P500 및 KOSPI200 지수에 대한 증거”, 한국파생상품학회, 2014

[4] 김혜리, 허익구, “기업의 수명주기와 사내유보가 기업가치에 미치는 영향”, 한국경영교육학회, 2020. 8.

[5] 윤여진, 김민규, 이종신, “정밀 기선장을 이용한 토털스테이션의 측정오차 및 RMSE 산출”, 한국지적정보학회지, 2012. 12.

[6] 김은미, 이배호, “모멘트의 동적 변환에 의한 Kernel Relaxation의 성능과 RMSE”, 멀티미디어학회논문지, 2003. 08.

[7] 유화윤, 김성범, “LCD 디스플레이 산업에서 데이터마이닝 알고리즘을 이용한 고객 불량률 예측”, 대한산업공학회, 2016. 10.

[8] 김귀훈, 홍용근, “네트워크에서의 머신러닝 기술 동향”, 한국통신학회, 2017. 9.

[9] 권정민, 정다은, 박형곤, “머신러닝 기반 이상 동작 네트워크 노드 분류 방법에 관한 연구”, 2020. 2.

[10] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, Micro Marchetti, “On the Effectiveness of Machine and Deep Learning for Cyber Security”, IEEE, 2018

[11] Swee Chuan Tan, Jess Pei San Lau, “Time Series Clustering: A Superior Alternative for Market Basket Analysis”, Lecture Notes in Electrical Engineering, 2014