



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

조선소 의장품 조달관리를 위한  
데이터마이닝 방법론에 관한 연구

A Study of Data-Mining Methodology in Offshore Plant's  
Outfittings Procurement Management



2016년 4월

한국해양대학교 대학원

조선해양시스템공학과  
함동균

본 논문을 함동균의 공학석사 학위논문으로 인준함.

위원장 남 중 호 (인)

위원 이 승 재 (인)

위원 우 중 훈 (인)



2016년 4월

한국해양대학교 대학원

# 목 차

List of Tables .....	iv
List of Figures .....	vi
Abstract .....	viii
<b>1. 서 론</b>	
1.1 연구 배경 .....	1
1.1.1 생산적 관점에서의 조선해양 .....	1
1.1.2 빅데이터 개요 .....	2
1.2 관련 연구 동향 .....	3
1.2.1 제조업 빅데이터 이용 사례 .....	3
1.2.2 조선업 빅데이터 이용 사례 .....	5
1.3 연구 목적 .....	6
<b>2. 조선해양산업의 배관의장 조달관리</b>	
2.1 조선해양산업의 배관의장 조달관리 .....	7
2.2 조선해양산업의 배관제작 및 설치공정의 이해 .....	8
2.3 수집 데이터 정의 .....	10
<b>3. 빅데이터 분석 방법론</b>	
3.1 KDD 분석 방법론 .....	13
3.2 CRIPS-DM 분석 방법론 .....	13
3.3 SEMMA 분석 방법론 .....	16
3.4 분석 방법론 제안 .....	18

#### 4. 예측모델의 이론적 고찰

4.1 일반적 통계기법의 회귀모형 .....	22
4.1.1 다중 선형회귀모형 .....	22
4.1.2 Ridge 회귀모형 .....	23
4.1.3 PLS 회귀모형 .....	23
4.1.4 Poisson 회귀모형 .....	25
4.2 빅데이터의 기계학습 알고리즘 .....	27
4.2.1 의사결정나무 회귀모형 .....	27
4.2.2 인공신경망 회귀모형 .....	29

#### 5. 배관재 공정 리드타임 예측 모델

5.1 IBM Modeler 17을 활용한 데이터전처리 .....	33
5.2 IBM SPSS 21을 활용한 분석 .....	37
5.2.1 다중 선형회귀모델 .....	37
5.2.2 PLS 회귀모델 .....	38
5.3 R을 활용한 분석 .....	40
5.3.1 다중 선형회귀모델 .....	40
5.3.2 Ridge 회귀모델 .....	43
5.3.3 PLS 회귀모델 .....	47
5.3.4 Poisson 회귀모델 .....	50
5.3.5 인공신경망 회귀모델 .....	52
5.3.6 의사결정나무 회귀모델 .....	54
5.3.7 R을 활용한 회귀분석 결과 비교 .....	57
5.4 Microsoft Azure를 활용한 분석 .....	58
5.4.1 인공신경망 회귀모델 .....	58
5.4.2 의사결정나무 회귀모델 .....	61

#### 6. 변수 민감도 분석

6.1 민감도 분석 모델 .....	65
6.2 민감도 분석 결과 .....	66

7. 결론 .....	70
참고문헌 .....	72
부록 A .....	74
부록 B .....	86
부록 C .....	109



## List of Tables

Table 1 의장품 조달관리 시스템의 주요 구성요소 .....	7
Table 2 예측모델 구축을 위한 변수정의 .....	11
Table 3 KDD, CRISP, SEMMA 프로세스 비교(Umair & Haseeb, 2014) .....	20
Table 4 R을 사용한 다중 선형회귀분석 코드전문 .....	41
Table 5 다중 선형회귀모델 평가데이터-계수 조정 함수 .....	43
Table 6 ridge 회귀모형 초기 lambda와 선택 lambda .....	43
Table 7 R을 사용한 다중 선형회귀(Ridge)분석 코드전문 .....	44
Table 8 결측이 포함된 이분형 필드 값 제거함수 .....	47
Table 9 PLS 회귀모형 최종 ncomp와 R2 .....	48
Table 10 R을 사용한 PLS 회귀분석 코드전문 .....	48
Table 11 SelectBestPLS 함수 .....	49
Table 12 R을 사용한 Poisson 회귀분석 코드전문 .....	50
Table 13 R을 사용한 인공신경망 회귀분석 코드전문 .....	52
Table 14 R을 사용한 의사결정나무 회귀분석 코드전문 .....	55
Table 15 SelectBestGbm 함수 .....	56
Table 16 Azure 인공신경망 회귀분석 조건 .....	60
Table 17 Azure 의사결정나무 회귀분석 조건 .....	62
Table 20 SPSS 다중선형회귀 계수(Making) .....	86
Table 21 SPSS 다중선형회귀 계수(Painting) .....	88
Table 22 SPSS 다중선형회귀 계수(Out Stock) .....	90
Table 23 SPSS 다중선형회귀 계수(In Stock) .....	92
Table 24 SPSS 다중선형회귀 계수(Stand Install) .....	94
Table 25 SPSS 다중선형회귀 계수(Install) .....	96

Table 26 SPSS PLS회귀 계수(Making) .....	98
Table 27 SPSS PLS회귀 계수(Painting) .....	99
Table 28 SPSS PLS회귀 계수(Out Stock) .....	101
Table 29 SPSS PLS회귀 계수(In Stock) .....	103
Table 30 SPSS PLS회귀 계수(Stand Install) .....	105
Table 31 SPSS PLS회귀 계수(Install) .....	106
Table 32 2차 데이터전처리 R 스크립트 전문 .....	109



## List of Figures

Fig. 1 해양플랜트의 후행공정 과다에 따른 작업지연 .....	2
Fig. 2 조선소 배관 발주 프로세스와 현행 업무 체계 .....	8
Fig. 3 배관공정의 일반적인 흐름도 .....	9
Fig. 4 배관재 공급망 데이터 구성 현황 .....	10
Fig. 5 KDD 분석 프로세스(Fayyad, 1996) .....	14
Fig. 6 CRISP-DM 분석 프로세스 .....	16
Fig. 7 SEMMA 분석 프로세스 .....	18
Fig. 8 배관재 공급망 데이터 분석 프로세스 .....	21
Fig. 9 생물학적 뉴런의 신경전달 구조 .....	30
Fig. 10 인공 뉴런의 방향성 네트워크 다이어그램 .....	31
Fig. 11 Painting과 Out Stock의 공정기간 생성 노드 .....	34
Fig. 12 로그스케일을 수행한 Weight(무게) .....	35
Fig. 13 로그스케일을 수행한 Length(길이) .....	35
Fig. 14 범주형 변수의 이분화 노드 .....	36
Fig. 15 1차 데이터전처리 모델(이분화노드 미포함) .....	37
Fig. 16 1차 데이터전처리 모델(이분화노드 포함) .....	37
Fig. 17 SPSS를 사용한 다중 선형회귀분석 모델 .....	38
Fig. 18 SPSS를 사용한 PLS 회귀분석 모델 .....	39
Fig. 19 SPSS를 사용한 회귀분석 결과(MAE) .....	35
Fig. 20 SPSS를 사용한 회귀분석 결과(MAPE) .....	40
Fig. 21 R을 사용한 회귀분석 결과(MAE) .....	58
Fig. 22 R을 사용한 회귀분석 결과(MAPE) .....	58
Fig. 23 Azure ML을 활용한 분석모델(전체) .....	59

Fig. 24 Azure ML을 활용한 분석모델(Making) .....	60
Fig. 25 Azure ML을 사용한 인공신경망 회귀분석 결과(MAE) .....	61
Fig. 26 Azure ML을 사용한 인공신경망 회귀분석 결과(MAPE) .....	61
Fig. 27 Azure ML을 사용한 의사결정나무 회귀분석 결과(MAE) .....	62
Fig. 28 Azure ML을 사용한 의사결정나무 회귀분석 결과(MAPE) .....	61
Fig. 29 회귀분석 전체 결과(MAE) .....	64
Fig. 30 회귀분석 전체 결과(MAPE) .....	64
Fig. 31 민감도 분석 모델(Making) .....	66
Fig. 32 Weight 히스토그램 .....	74
Fig. 33 Length(길이) 히스토그램 .....	75
Fig. 34 DIA(직경) 히스토그램 .....	75
Fig. 35 Num of Parts(부품 수) 히스토그램 .....	76
Fig. 36 Num of Joints(연결 수) 히스토그램 .....	76
Fig. 37 Stage(작업수행 공정시점) 빈도분석 .....	77
Fig. 38 Making Vendor(제작업체) 빈도분석 .....	78
Fig. 39 Paint Code(도장 방법) 빈도분석 .....	78
Fig. 40 Material(배관 재질) 빈도분석 .....	79
Fig. 41 Warehouse(물류업체) 빈도분석 .....	79
Fig. 42 Painting Vendor(도장업체) 빈도분석 .....	80
Fig. 43 Sch(배관 두께) 빈도분석 .....	80
Fig. 44 Main Problem(주요문제) 빈도분석 .....	81
Fig. 45 Making(제작 관련 공정) 빈도분석 .....	82
Fig. 46 Painting(도장 관련 공정) 빈도분석 .....	83
Fig. 47 Out Stock(사외재고 관련 공정) 빈도분석 .....	83
Fig. 48 In Stock(사내재고 관련 공정) 빈도분석 .....	84
Fig. 49 Stand Install(설치대기 관련 공정) 빈도분석 .....	84
Fig. 50 Install(설치관련 공정) 빈도분석 .....	85

# A Study of Data-Mining Methodology in Offshore Plant's Outfittings Procurement Management

Ham, Dong kyun

Division of Naval Architecture and Ocean Systems Engineering  
Graduate School of Korea Maritime University

## Abstract

선박 및 해양플랜트 건조에는 방대한 물량의 의장품이 사용되며 공급되는 경로 또한 공급망 프로세스와 협력업체에 따라 다양한 경로가 존재한다. 최근 해양플랜트 건조의 경우 다양한 의장품의 사양과 경로에 따른 조달지연이 충분히 고려되지 못하고 있어 조기에 공급되어 불필요한 적치가 증가하거나 공급이 지연되어 설치가 연기되는 등 손실이 발생하고 있다. 이를 해결하기 위해서는 우선적으로 의장품의 사양과 공급경로를 고려한 공급 리드타임에 대한 예측이 필요하다.

이러한 예측모델 구축을 위해 최근 이슈가 되고 있는 빅데이터 기술을 살펴보고 이를 조선해양 생산 분야에 적용하여 효용성을 살펴보고자 한다. 빅데이터의 기술은 데이터의 생성, 수집, 저장, 분석, 표현의 전 과정에 걸쳐 사용되며 이 중 분석, 표현의 기술은 여러 데이터마이닝 기법 및 시각화 기법을 사용하여 데이터를 통해 유의미한 정보를 찾아내는 것이다. 본 논문에서는 선박 및 해양플랜트의 배관재 공급망 데이터를 여러 데이터마이닝 기법을 사용하여 예측 모델을 개발하였다. 예측 모델은 다중선형, Ridge, PLS, Poisson, 의사결정나무, 인공신경망을 SPSS와 R 그리고 Microsoft Azure의 기계학습을 사용하여 구축하였고 각각의 예측 성능을 MAE(Mean Absolute Error)와 MAPE(Mean Absolute Percentage Error)를 통해 비교하였다.

또한 변수 민감도분석을 통해 예측모델에 대한 각각의 변수들의 영향력 분석하였고 현업의 공정개선 시 이를 참고할 수 있도록 하였다. 본 연구의 예측모델과 민감도분석 결과를 통해 기존 배관재 생산계획 단계에서 보다 합리적인 의사결정을 할 수 있을 것으로 예상된다.

**KEY WORDS:** Shipbuilding 선박건조; Production planning 생산계획; Big data 빅데이터; Machine learning 기계학습; .



# 제 1 장 서 론

## 1.1 연구 배경

2010년대 심해 자원 개발에 대한 관심이 증대 되면서 해양플랜트 수요가 증가하였다. 당시 상선 수주 부진을 겪고 있던 국내 조선사들은 글로벌 해양플랜트의 수요를 거의 독점하다시피 하여 건조물량 확보에는 성공하였다. 하지만 부작용은 수주된 해양플랜트가 본격적으로 생산되기 시작하면서 나타났는데, 현재까지 국내 주요 조선 3사의 해양플랜트 관련 누적작자가 10조원에 육박할 것으로 추산하고 있다(KMRI, 2015(12)). 신조 수주로 현재의 경영 악화를 완화하여 하였으나 사상초유의 유가하락으로 해양플랜트의 수요가 줄어 그것마저도 효과를 거두지 못하고 있는 실정이다. 이렇듯 현재 국내 조선소들이 겪고 있는 위기의 원인을 분석해보면 크게 3가지로 구분할 수 있다. 첫 번째는 국내 조선소들 간의 가격경쟁으로 인한 저가수주이다. 일반적으로 FPSO-FLNG의 경우 적당 50억불(약6조원)에 수주되는데 국내 모 중공업의 경우 3조 2000억 원 가량에 다소 낮은 가격으로 계약되었다. 이는 국내 3사의 사업영역이 중복되어 과당경쟁 체제로 인한 부작용으로 볼 수 있다. 두 번째는 경험부족으로 인한 공기지연 문제이다. 해양플랜트의 경우 조선업에서는 신규 사업으로 설계 및 건조계획 시 공기예측 및 원가산정이 거의 불가능한 상황이다. 또한 기본설계에 해당하는 FEED를 해외 해양엔지니어링 회사에서 수행하기 때문에 기자재 발주처 선정이 불가능하고 설계변경이 자주 발생하여 공기지연이 야기되었다. 세 번째로는 유가하락이다. 최근 미국의 셰일오일 생산이 증가되면서 국제유가가 30달러 선까지 떨어지게 되고 이로 인해 심해시추의 채산성이 감소하여 해양플랜트에 대한 수요가 줄어들고 있다. 이로 인해 선주 측에서의 인도지연이 발생하게 되었고 이는 영업손실을 야기하는 결과를 낳았다.

### 1.1.1 생산적 관점에서의 해양플랜트

앞서 밝힌 손실원인들 중 공기지연은 생산과 밀접한 관련성을 가지고 있고

이 부분에서도 여러 원인들을 찾을 수 있다. 해양플랜트 관련 국제표준 부재, 후행의장공정에서의 과도한 업무증가를 들 수 있다. 이중 후행의장작업에 대한 업무증가는 해양플랜트의 고유의 특성을 반영하는 부분으로, 해양플랜트는 육상의 플랜트가 선박의 Topside 위에 탑재되는 형태로 기존의 상선에서의 Topside의 의장작업은 후행공정으로 분류하여 선행공정에서 완료되지 못한 작업들을 수행하고 있다. 일반적으로 후행공정은 호선이 F/O(Floating:진수)되어 안벽에 계류한 상태에서 수행하는데, 일반 상선은 선행공정에 비해 후행공정의 비중이 크지 않기 때문에 설치계획 및 설치순서를 체계적으로 관리하지 않고 있다. 하지만 해양플랜트의 경우 이 후행공정이 선행공정보다 더 커지게 되었고 작업량을 많아지는 상황에서 기존의 생산 체계가 이를 수용하지 못하게 되어 지연이 발생하는 것이다. 특히 의장품 중 각종 자원이공 모듈에 쓰이는 배관재의 경우 적게는 3만개에서 많게는 10만개의 물량을 다루고 있기 때문에 비효율적인 설치 작업으로 인한 공수 낭비, 분실로 인한 추가 발주 등으로 지연에 많은 영향을 미치고 있는 실정이다.

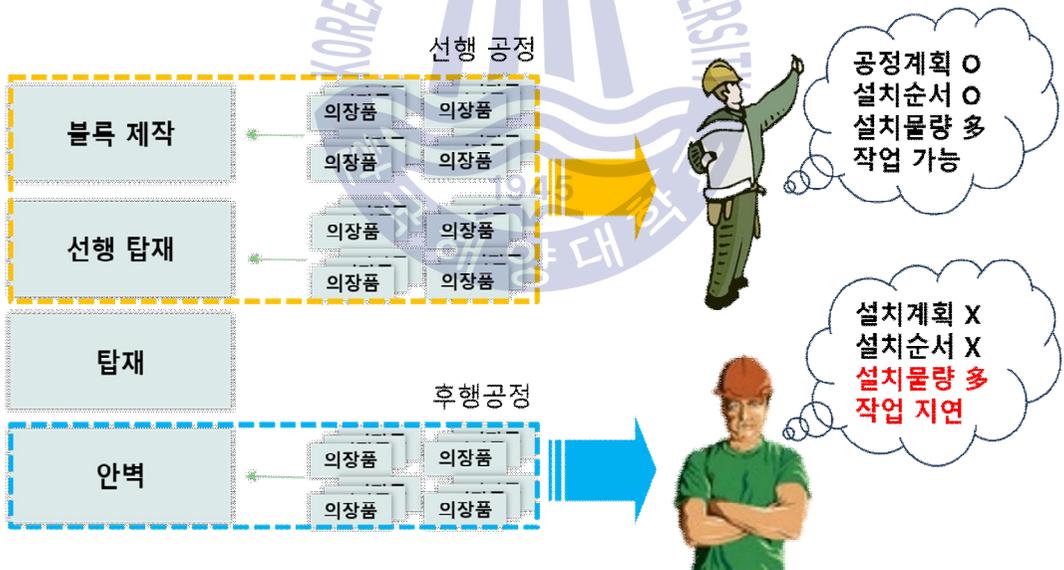


Fig. 1 해양플랜트의 후행공정 과다에 따른 작업지연

### 1.1.2 빅데이터 개요

빅데이터는 최근 각광 받고 있는 신기술로 다양한 종류의 대규모 데이터를

실시간으로 빠르게 분석하여 의미 있는 결과를 얻는데 사용하는 기술 집합을 의미한다. 이와 같은 핵심 개념을 3V와 4V로 설명할 수 있는데, 3V는 Volume(크기), Velocity(속도), Variety(종류)의 약자이며 여기에 Veracity(진실성)이라는 개념이 포함되면 4V가 된다. 빅데이터라는 기술은 기존에 없던 전혀 새로운 기술이 아니라 기존 대용량 데이터 관리와 분석이 발전한 하드웨어와 네트워크 인프라를 배경으로 더 다양한 종류의 정보와 더 빠른 속도를 포함하는 것으로 기존의 데이터마이닝(Data Mining) 개념에 향상된 데이터 처리기술이 접목된 형태라 할 수 있다. 데이터마이닝이란, 데이터 내에 존재하는 관계, 패턴, 규칙 등을 탐색하고 찾아내어 유용한 지식을 추출하는 일련의 과정들로서 분류, 추정, 예측, 연관성규칙, 군집화, 프로파일링 등을 수행한다. 특히 추정과 예측은 수집된 데이터의 변수들과 결과간의 인과관계를 정립하여 새로운 데이터가 들어왔을 때 결과를 추정 및 예측하는 기술로서, 최근 많은 분야에서 활용되고 있다. 본 연구에서는 이러한 빅데이터의 방대한 데이터를 빠른 시간으로 분석하여 패턴을 찾아내는 기술을 활용하여 배관재의 리드타임을 예측하는 예측모델을 개발하고자 한다.

## 1.2 관련 연구 동향

### 1.2.1 제조업 빅데이터 이용 사례

해외에서는 일찍이 스마트팩토리(Smart Factory)라는 이름으로 빅데이터의 개념을 활용하고자 하였다. 하지만 제조공정에서 만들어지는 대용량의 데이터들을 수집, 저장, 분석, 가시화할 수 있는 능력의 한계로 인해 주춤하다가 분산처리기술 및 분석기법의 발전으로 다시금 제조분야에 이슈가 되고 있다.

제조업의 빅데이터 이용 사례를 살펴보면 먼저 Merck사의 약품 생산라인을 들 수 있다. Merck사의 경우 전체 공정을 상호 연관성 있는 공정단위로 군집화한 뒤 상호연관관계를 분석하여 9개의 방해인자를 발견하고 이를 개선하여 500만 달러의 비용 절감 효과를 거두었다(Mckinsey&Company,2014). 또한 Intel사는 2012년부터 품질관리 공정에서 생성되는 제조로그데이터를

분석하여 품질불량 예측모델을 개발하여 품질검사 대상을 현격하게 줄임으로써 3백만 달러의 제조비용을 절감하는 효과를 얻었다. 글로벌 반도체 기업인 마이크론 테크놀로지는 로봇의 움직임 로그데이터를 분석하여 효율적인 동작 모델을 개발하고 이를 로봇에 적용하였다. 이후 이상 징후가 발견되면 동작을 수정하는 방식으로 가용성을 높여 생산성을 10% 향상시키고 380억 원의 비용절감 효과를 거두었다(KIET industrial economic review(KIET), 2014).

국내의 경우 국외에 비해 제조업의 역사가 길지 않아 현재 빅데이터 활용사례가 상대적으로 적지만 최근 정부의 K-ICT 빅데이터센터 오픈 및 여러 기업들의 투자로 점점 활용사례가 늘어가는 추세이다. 국내사례의 경우 한국남동발전 발전설비 운영효율 극대화 사례를 들 수 있다. 해당 사례의 경우 시스템 및 연료 설비로부터 생성되는 데이터들을 수집·분석하여 장비고장 예측 모델을 개발하였다. 이 모델은 장비 고장을 예측하여 적절한 정비시기를 결정하여 연료비 절감 및 효율적인 발전기운용을 가능하게 하였고 비계획발전손실율<sup>1)</sup> 0.02%를 절감할 수 있었다. 또한 웅진케미컬 정수기 필터 사업부는 제조과정에서 생성되는 생산데이터와 공급망-전사시스템에서 생성되는 기업 내 운영데이터를 통합하고 이를 분석하여 고객의 니즈를 신속히 파악하면서 최적의 생산계획을 수립하여 재고를 줄이고 납기율을 상승시켰다(Jang, 2012).

제조업에서 빅데이터 활용 형태를 살펴보면 3가지로, 공정 분석 및 개선, 제조데이터 활용, 운영통합데이터 활용과 같다. 먼저 공정 분석 및 개선의 경우 Merck사 사례에서도 볼 수 있듯이 기존의 공정프로세스를 단위 공정 별로 분석하여 비효율적인 공정을 개선하는 형태로, 이러한 형태의 적용은 주로 생산라인이 고정되어 있는 경우 효과적일 수 있다.

두 번째 제조업 빅데이터 활용은 생산과정에서 발생하는 제조데이터를 활용하는 방식이다. 제조업은 수많은 장비와 센서들을 사용하기 때문에 장비로그데이터와 센서데이터가 대량으로 생성되어 저장된다. 이러한 데이터들을 분석하여 재공품의 품질을 예측하거나 제조장비의 고장을 예측하여

---

1) 비계획발전손실율 : 주어진 일정 시간 동안 갑작스런 고장 등 계획되지 않은 발전기 정지율을 의미

공정운영 효율을 높이는 방식으로, 현재 제조업에서 가장 많이 활용되고 있다. 공정 분석 및 개선 활동과 제조데이터 활용은 적용방법이 다소 유사해 보이나 제조데이터 활용은 주로 장비고장 예측이나 품질 예측에 집중되어 있는 반면 공정 분석 및 개선 활동은 기존의 데이터분석에서는 프로세스마이닝에 해당하면 공정 간에 관계를 개선하는데 초점이 맞춰져 있다.

세 번째는 자재 조달에서 생산, 판매에 이르는 전사적인 운영데이터를 제조업 관점에서 활용하는 방식이다. 제조공정 개선 문제는 제조공정 자체 뿐 아니라 제조와 관련된 자재구매, 수요, 재고 등과 같은 조달 공급망데이터가 큰 영향을 미치는데 이러한 운영데이터를 활용하여 생산자재수요 예측 및 판매수요 예측 모델을 개발하는데 활용할 수 있다. 그러나 실제 현업에서는 제조정보와 경영정보가 별도로 관리되는 경우가 많기 때문에 이들을 통합하는 과정은 또 다른 해결과제라 할 수 있다. 최근 빅데이터의 영향으로 여러 제조분야에서 제조공정관리에 경영정보를 반영하려는 노력이 늘고 있고 실제로 웅진케미컬과 같이 효과를 거두고 있는 기업이 생기고 있다. 이러한 움직임은 제조와 경영이 밀접한 관계를 가지고 있는 조선업에서도 적용할 수 있음을 시사한다.

## 1.2.2 조선업 빅데이터 이용 사례

조선업은 일반제조업과 같이 조립, 도장, 검수의 과정을 가진다는 점에서 빅데이터의 활용 모습이 유사할 수 있으나, 몇 가지 조선업 고유의 특성을 고려하는 것이 필요하다. 제조업은 대개 대량생산체제의 생산방식을 가지고 있으나 조선업은 대표적인 수주산업으로 호선 하나가 하나의 프로젝트에 해당한다. 또한 일반제조업은 품질관리를 제외하면 거의 자동화 생산시스템을 가지고 있는데 반해 조선업은 일부 절단 및 용접공정을 제외하면 거의 수작업으로 진행된다. 마지막으로 제조 프로세스가 라인생산의 일반제조업에 비해 매우 유동적인데, 전공정과정 및 공법은 큰 틀에서는 정해지나 세부 실행단계에서는 유동적으로 수행된다. 이런 조선업의 특징은 인적요소의 개입이 많이 발생되고 공정관련 데이터에 노이즈가 많이 발생된다는 점이다.

제조업 분야에서의 빅데이터 활용은 계속 늘어나는 추세이나 조선업의 경우

국내 대우조선해양과 현대중공업 이외는 학술적 성과로 찾을 수 있는 것이 전무한 상황이다. 대우조선해양의 경우, MRO(Maintenance, Repair & Operation) 라고 하는 빅데이터 기반 선박신수요 예측 플랫폼 및 MRO 서비스 사업을 추진하고 있다. 해당 사업의 경우 선박 수요 예측에 특화된 빅데이터 분석 플랫폼을 개발하고, 데이터 분석을 통한 선박 신수요 예측 결과를 신제품 기획 및 개발에 활용하고자 하는 것이다. 수요 예측 항목으로는 소모성 선박 용품 공급과 기부품, 기자재/장비 공급, 기자재/장비 및 수리 서비스 공급, 신제품개발 등을 포함한다(Hankook MediaNetwork, 2015). 현대중공업은 제조 프로세스 분석을 위한 빅데이터 클라우드 서비스 사업을 추진 중으로 해당 사업은 빅데이터 클라우드를 기반으로 공정 데이터 분석 시스템을 개발하고 이를 통해 다중 프로젝트 공정들의 가시성을 확보하고 계획 일정의 정확도를 증대하며 공정 부하 및 지연을 해소하고자 하는 목적을 가지고 있다(Hankook Media Network, 2015).

### 1.3 연구 목적

선박 및 해양플랜트 건조에는 방대한 양의 의장품들이 취급되며 공급되는 경로 또한 공급망 프로세스와 협력사에 따라 여러 경로가 존재한다. 최근 해양플랜트의 건조의 경우 여러 의장품들의 사양 및 경로에 따른 지연이 충분히 고려되지 못하고 있어 조기에 적치가 되거나 또는 적치가 지연되는 현상으로 인해 공수가 낭비되고 설치지연이 발생하는 문제가 발생하고 있다. 따라서 본 논문은 기존의 일괄적인 의장공정계획에서 합리적인 의사결정을 가능하도록 다양한 데이터 마이닝 기법을 사용하여 예측모델을 개발하고 이를 통해 조선업종의 통계적 방법론 적용을 위한 가이드라인을 제시하고자 한다. 특히 본 논문에서는 여러 의장품 종류 중에서 많은 비중을 차지하고, 또한 지연을 야기하고 있는 배관재 공정을 중심으로 연구를 수행하였다.

## 제 2 장 조선해양산업 배관의장 조달관리

### 2.1 조선해양산업의 의장 조달관리 프로세스

조선해양산업의 의장 조달관리 프로세서는 조선소마다 약간의 차이는 있으나 큰 틀에서는 유사하게 진행되며 해양플랜트의 경우 전체공정 중 의장작업이 차지하는 비율이 상선보다 더 크다. 조선소의 의장품 조달관리 시스템 구성요소를 보면 Table 1과 같다(Okumoto, 2012).

Table 1 의장품 조달관리 시스템의 주요 구성요소

구성요소	설명
스케줄링 처리	선박번호, 잡코드(Job code), 작업개시일, 자재 배치일, 정리일, 발송일(자재 필요처), 실적 변동일수를 등록
설계에서 요구처리	물품 제작정보(품명, 수량, 제품사양, 도장사양) 물건의 납입에 필요한 정보(잡코드, 납입처, 조정일수 등) 등록
구입에서 주문처리	조달요구표, 제작도, 사양서를 근거로 의장품 발주처 및 가격 결정 후 주문처리
납기변경 처리	납기지연에 의한 변경 정보 등록
납품, 검수 및 분류와 입고실적 조회처리	사양의 적합성, 수량, 하자, 도장상태 점검 후 제품의 납입과 검수처리 수행

‘설계에서 요구처리’에서 조정일수라는 것은 장착 전 물품입하 시기를 의미하는 것으로, 초기치는 -n일로 설정하는데 동력기계 및 엔진, 대형 의장품의 경우 n일 전에 입하시켜도 놓아둘 장소가 없고 가접하는 것 또한 용이하지 않기 때문에 조정일수를 0일로 설정하는 경우도 존재한다. 또한 조선소의 입지환경, 유통환경 및 거래처와 계약 조건 등에 따라 달라질 수도 있다. Fig. 2과 같이 블록디비전 후 상세설계 도면이 출도되면 제작업체로 제작지시가 발행되는데 현재 조선소에서는 일회 발주되는 양이 많기 때문에 한번에 요청되는 의장품들에 대해서 조정일수를 일괄적으로 부여되고 있다.

이런 과정으로 인해 의장품들이 조기에 납품되거나 아니면 지연되는 현상이 발생하여 설치공정이 비효율적으로 수행되어 공수 낭비 및 공정지연을 발생하고 있다. 그러므로 주문처리가 이루어지는 과정에서 공급망 사이의 예측된 리드타임을 활용하여 계획을 수행한다면 공수 낭비 및 공정지연을 줄일 수 있을 것이다(Ham, et al., 2015).

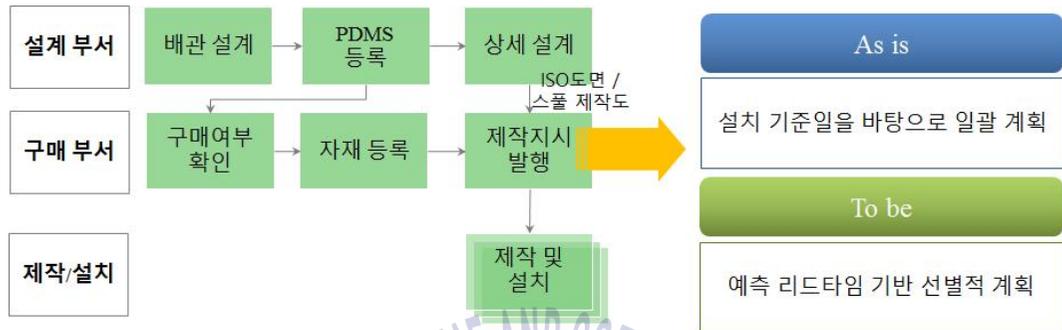


Fig. 2 조선소 배관 발주 프로세스와 현행 업무 체계

## 2.2 조선해양산업의 배관제작 및 설치 공정의 이해

조선소의 의장은 크게 배관의장과 철의장으로 나눌 수 있는데, 이중 해양플랜트의 경우 배관의장이 차지하는 비율이 상대적으로 큰 편으로, 적게는 3만개에서 많게는 10만개의 배관을 취급한다. 따라서 선행 의장공사의 많은 비중을 차지하는 배관의장 공사가 효율적으로 진행될 경우 전체 공기단축에 상당한 영향을 줄 수 있다. 배관 공정흐름은 전체 공정 중 한부분에 국한 되지 않고 전 공정에 걸쳐 이루어지기 때문에 다소 불규칙한 구조를 가지고 있으나 배관공정의 일반적인 흐름을 정의하면 Fig. 3와 같다.

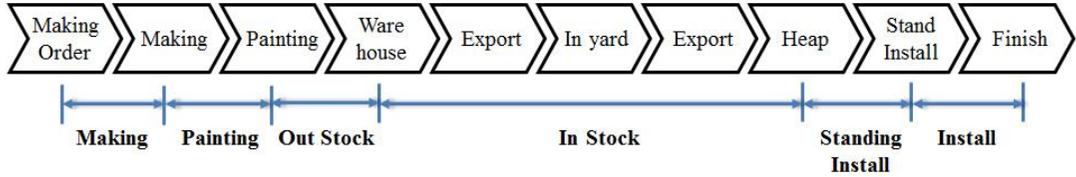


Fig. 3 배관공정의 일반적인 흐름도

Making은 배관재를 제작하는 공정기간으로 보통 제작협력업체에 이루어진다. Making은 제작지시가 발행된 시점부터 제작이 완료되기까지의 기간으로 이 안에는 제작과 비파괴검사 등이 포함된다. 제작순서는 보통 발주된 순서대로 진행하지만 긴급물량이나 품질검사 결과에 따라 완료날짜가 변경될 수 있다. Making에서 특이점(Outlier)으로 작용할 수 있는 요소로는 검사업무를 들 수 있다. 품질검사는 검사관에 의해서 이루어지기 때문에 여러 가지 이벤트(검사관 부재, 장비고장 등)의 여부에 따라 리드타임의 변동이 발생한다. Painting은 제작이 완료된 시점부터 도장이 완료되는 시점까지를 의미한다. Painting에서는 배관재에 따라 수행여부가 달라지는데 후행도장 대상 품목이나 긴급입고일 경우에 Painting과정을 거치지 않고 제작 후 바로 야드 내로 입고되는 경우가 존재한다. Out Stock은 도장이 완료되고 난 뒤 야드에 입고되기 전에 잠간 사외 적치장에 적치되어 있는 기간으로 도장이 없는 배관재는 해당 적치장을 거치지 않고 바로 야드로 입고된다. In Stock은 야드 입고된 후 배관재가 작업장 주위로 적치되기까지의 기간으로 일반적으로는 해당 작업자에게 인계된 날짜를 기준일로 삼지만 현재 조선소의 배관 운용상 인계되기 전에 적치를 먼저 수행하는 경우가 많기 때문에 적치일을 기준으로 삼았다. In Stock의 경우 해당 적치장에 호선 하나의 배관만 적치되는 것이 아니라 여러 호선의 배관재들을 동시에 관리하기 때문에 찾는 데 소요되는 시간이 많을 뿐 아니라 분실 또한 빈번히 일어나고 있어 이런 요소들이 특이점으로 작용할 수 있는 가능성을 가지고 있다. Standing Install은 배관재가 실제 설치될 장소에 적치되기까지의 기간으로 여기에는 가접되는 경우와 최종 설치되는 경우로 나뉜다. Install은 배관재가 설치장소에 적치된 뒤 설치가 최종 완료되는 시점까지를 의미한다.

### 2.3 수집 데이터 정의

의장품 중 배관재 공급망 중심으로 현재 조선소에서 사용 중인 공급망 데이터를 수집한 뒤에 2.2절을 기준으로 데이터를 정의하였다. 데이터는 Fig. 4와 같이 크게 리드타임과 관련된 시계열데이터와 배관재의 사양데이터, 그리고 제작업체 및 기타 제조환경과 관련된 기타데이터로, 총 3부분으로 구성된다.

No.	시계열정보						사양정보			기타정보	
	주문일	....	제작일	도장일	....	설치일	중량	직경	....	제작업체	....
1	2014-10-11	....	2014-11-05	2014-11-20	....	2015-02-18	12	23	....	A사	....
2	2014-10-20	....	2014-11-04	2014-11-20	....	2015-02-01	8	12	....	A사	....
3	2014-11-11	....	2014-11-16	null	....	null	7	8	....	C사	....

Fig. 4 배관재 공급망 데이터 구성 현황

데이터는 121개의 필드와 12323개의 레코드의 크기를 가지고 있다. 121개의 필드 중 본 연구에서는 생산 관련된 12개의 필드를 선정하였다. 13개 필드 중 수치형변수는 5종, 범주형변수는 8종으로 이루어져 있다. 13개의 설명변수(입력변수)와 함께 6개의 공정 별 리드타임을 목표변수(종속변수)를 Table 2과 같이 정의하였다.

**Table 2** 예측모델 구축을 위한 변수정의

Type	Description	
Independent Variable	Making term(day) = Lead time to make pipe at vendors	
	Painting term(day) = Lead time to paint pipe at vendors	
	Out Stock term(day) = Lead time to be heaped at out-warehouse	
	In Stock term(day) = Lead time to be heaped or moved in shipyard	
	Standing install term(day) = Lead time to get somewhere to be installed	
	Install term(day) = Lead time to be installed	
Dependent Variable	Continuous data	Diameter (inch)
		Length (mm)
		Weight (kg)
		Number of parts (piece)
		Number of Joint (piece)
	Discrete data	Vendors for making
		Vendors for painting
		Warehouse
		Material property
		Thickness
		Paint code
		Pre-weld Stage
		Main problem

변수정의에서 수치형변수 중 Diameter는 배관의 외방 직경을 의미하고 단위는 인치(inch)이고 Length는 배관의 길이를 의미하고 단위는 밀리미터(mm), Weight는 배관의 무게를 의미하고 단위는 킬로그램(kg), Num of part-Joint는 각각 부재수와 조인트(연결부) 개수를 의미하고 단위는 개(piece)로 정의한다. 범주형변수 중 Vendors for making은 배관제작 협력업체를 의미하고 Vendors for painting은 배관도장 협력업체, Warehouse는 야드 입고 전 적치되는

외주창고, Thickness는 배관의 두께, Paint Code는 도장 방법 및 유체특성에 따른 코드, Pre-weld Stage는 가접 및 최종용접이 수행되는 공정단계, Main problem은 도장 이전까지의 문제점을 의미한다.



## 제 3 장 빅데이터 분석 방법론

### 3.1 빅데이터 분석 방법론 소개

최근 여러 산업, 비즈니스에서는 이전에는 사용하지 않던 방대한 량의 데이터를 분석하여 트렌드 및 여러 패턴들을 알아내는 능력이 매우 중요한 과업이 되었다. 데이터의 크기가 커질수록 이러한 데이터를 활용하기 위한 데이터마이닝에 대한 관심이 증가하게 되었고 1980년대 등장한 이 단어는 오늘날까지도 사용되고 있는 기술이 되었다. 데이터마이닝의 등장과 함께 여러 가지 분석방법론도 함께 나타나기 시작했고, 대표적으로 KDD(Knowledge Discovery Databases), CRISP-DM(Cross-Industry Standard Process for Data Mining) 그리고 SEMMA(Sample, Explore, Modify, Model and Assess)가 있다. KDD는 반복적이고 순환적인 모델로 총 9개의 단계를 가지고 있고 데이터에 대한 정보를 찾는 것을 목적으로 하고 높은 수준의 데이터마이닝 기법을 사용하는 것을 중요하게 생각한다. CRISP-DM은 1996년에 Daimler Chrysler (then Daimler-Benz), SPSS (then ISL)와 NCR(Corp.)에 의해 고안된 것으로 수년 동안 갱신되고 있고 6개의 단계를 포함한다. SEMMA는 SAS사에서 개발한 것으로 5개의 서로 상이한 단계를 가지고 있다(Umair & Haseeb, 2014). 본 연구에서는 이들 분석 방법론에 대해 살펴본 뒤 조선업의 배관재 리드타임 예측모델을 구축하는데 적합한 방법론을 사용할 것이다.

### 3.2 KDD 분석 방법론

KDD는 Knowledge Discovery Databases의 약자로, 기술과 데이터베이스를 중심으로 한 통찰력을 발견하기 위한 절차와 단계를 정리한 것이다. 해당 방법론은 KDD(Knowledge Discovery and Data Mining) 컨퍼런스에서 Gregory I. Piatetsky-Shapiro에 의해 1989년 처음 언급된 프로세스로 데이터가 생성되는 분야(Domain)의 이해를 바탕으로 한 데이터마이닝 자체가 핵심이 되는 방법론이다. KDD는 Fig. 5와 같이 9개의 진행 프로세스를 가지고 있는데, 필요에 따라 전단계의 과업을 반복수행하거나 세부조건을 조절하는 반복적이고

순환적으로 진행되는 것이 특징이다.

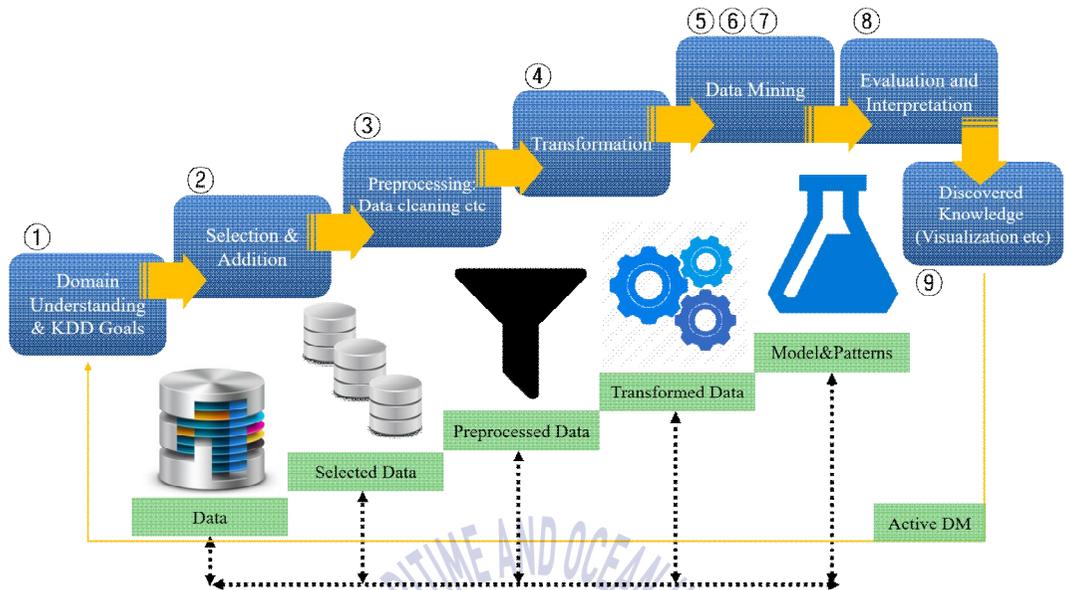


Fig. 5 KDD 분석 프로세스(Fayyad, 1996)

KDD 프로세스 중 첫 번째 Developing an understanding of the application domain은 최초 준비 단계로 이때 이루어진 해당 분야 파악 정도에 따라 데이터변환, 분석알고리즘, 가시화 등에 대한 결정이 이루어진다. KDD 프로젝트의 분석가는 해당 분야의 분석목표와 비즈니스 환경 등에 대한 이해가 필수적이다.

두 번째는 Selecting and creating a data로 정의된 분석목표에 따라 가용 데이터를 추출 및 수집하는 단계로 Aggregation된 데이터 집합에 따라 분석결과가 달라지기 때문에 중요 특성을 놓치지 않도록 하는 것이 필요하다. 또한 순환 분석 단계에서 모델의 결과에 따라 데이터 관측 범위를 확장하여 추가데이터를 통합하는 방법을 사용할 수도 있으나 이는 비용을 발생시키므로 비즈니스 관점에서 적절한 Trade-off가 필요하다.

세 번째는 Preprocessing and cleansing으로 데이터의 신뢰성을 강화하기 위하여 노이즈 값이나 극값을 제거하거나 결측값을 제거하는 data cleansing을 수행한다. 만약 어떤 예측모델을 만들기 위해 사용한 필드 중 결측값이 많은

필드를 포함할 경우 해당 필드를 목표변수로 한 예측모델을 선행적으로 개발하여 설명변수를 강화한 뒤 최종 목표변수를 지도로 한 예측 모델을 개발할 수도 있다.

네 번째는 Data transformation으로, 이 단계에서는 해당 데이터마이닝에 최적화된 데이터로 변환하는 단계이고 데이터변환 과업으로는 차원감소, 데이터이산화 등이 있다. KDD 프로세스 과정에서 한번 데이터변환을 하였다 하더라도 후행 프로세스의 필요에 의해 Data transformation은 반복적으로 수행될 수 있다.

다섯 번째부터 일곱 번째까지는 KDD 프로세스에서 데이터마이닝에 해당하는 부분으로 비즈니스 및 해당 분야의 분석목적에 따라 분석과업을 결정하고 분석과업을 풀기위한 알고리즘을 결정한 뒤에 여러 조선을 달리하면서 원하는 결과가 산출될 때까지 분석을 반복하게 된다. 분석과업은 분류, 추정, 예측, 연관분석, 프로파일링, 군집화 등이 있고 분석알고리즘은 분석과업마다 하나 또는 복수개로 구성되어있다. 또한 분석 알고리즘마다 분석조건(파라미터)을 다르게 할 경우 분석결과가 달라지므로 이들을 변화시키면서 최적의 모델이 생성될 까지 반복 분석을 수행한다. 가령 의사결정나무 모델을 만들 때 나무의 가지 수를 달리하여 여러 결정나무를 생성하여 상호 비교분석을 수행할 수 있다.

여덟 번째는 평가단계로 분석결과로 나온 모델이 첫 번째 단계에서 정의한 비즈니스의 분석목적에 대한 유의미한 결과인지를 평가한다. 만약 효과적인 측면(정확도 등)에서 보완이 필요할 경우 Preprocessing 단계에서의 보완이 이루어져야 한다. 본 과정의 결과는 추후 현업의 사용을 위해 적절한 문서화가 필요하고 실제 비즈니스의 적용 피드백을 수행할 수 있도록 체계화가 되어 있어야 한다.

마지막 단계는 분석결과와 적용단계이다. 해당단계는 데이터가 발생된 비즈니스 및 업무환경에서 앞의 데이터마이닝으로 발견한 지식이나 통찰력 등을 활용하여 효과가 어느 정도인지 측정하는 단계이다. 과거에는 평가 측정이 최종 결과 문서 또는 결산 시점에서 알 수 있었으나 최근에는

실시간으로 효과를 판단하는 경우도 있다.

### 3.3 CRISP-DM 분석 방법론

CRISP-DM은 Cross-Industry Standard Process Data Mining의 약자로 1996년 유럽연합의 ESPRIT(European Strategic Program on Research in Information Technology) 펀딩 운동에서 언급되었고 이 프로젝트에는 SPSS, Teradata, Daimler AG, NCR Corporation 그리고 보험회사인 OHRA가 참여하였다. 현재는 공식적인 활동은 중단한 상태이고 IBM의 SPSS Modeler 가이드로 업계에서 사용되고 있다. 단계는 총 6개로 구성되며 진행 프로세스는 Fig. 6과 같이 수행된다(Pete, et al., 2000).

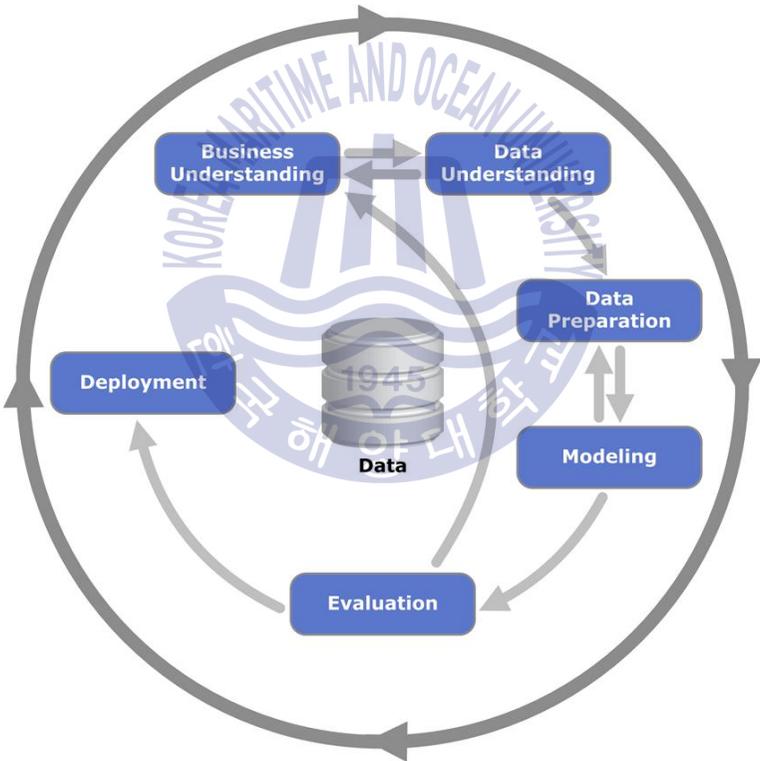


Fig. 6 CRISP-DM 분석 프로세스

CRISP-DM 프로세스 중 비즈니스 이해(Business Understanding)는 고객의 요구 사항과 비즈니스 목표를 이해하고 이를 데이터마이닝의 목표로 전환하는 단계로 결과에 영향을 주는 항목을 도출하고 데이터마이닝 목표를 달성하기

위한 예비 계획을 수립한다. 이 단계의 주요 과업으로 4가지로 비즈니스 관점에서 고객 요구를 이해하는 Determine Business Objects, 프로젝트에 영향을 미치는 요인에 대한 사실 조사를 하는 Assess Situation, 기술적인 측면에서 프로젝트 목표를 결정하는 Determine Data Mining Goals, 프로젝트의 목표를 달성하기 위한 세부 계획을 준비하는 Produce Project Plan가 있다.

데이터 이해(Data Understanding)는 초기 데이터를 수집한 뒤 데이터의 품질을 정의하고 몇 가지 가설을 구성하기 위한 데이터 셋을 설정하는 단계이다. 이 단계의 주요 과업으로 4가지로 초기 데이터를 수집하는 Collect Initial Data, 수집된 데이터의 특성을 확인하는 Describe Data, 질의, 시각화, 보고서 등 데이터마이닝 목표에 부합하는 데이터를 추출하는 Explore Data, 데이터의 품질을 검사하는 Verify Data Quality가 있다.

데이터 준비(Data Preparation)는 모델링 도구의 입력이 될 데이터를 원시 데이터부터 추출하는 단계로 5가지 과업을 통해서 수행된다. 분석에 사용할 데이터를 선별하는 Select Data, 분석 기법에서 요구한 수준으로 데이터의 품질을 향상하는 Clean Data, 분석을 위한 구조화된 데이터를 준비하는 Construct Data, 분석을 위해 여러 테이블에서 데이터를 통합하는 Integrate Data, 사용하려는 모델에 적합하도록 데이터의 구문을 수정하는 Format Data가 있다.

모형(Modeling)은 모델링 기법을 선택하고 최적의 값을 가진 파라미터 변수를 설정하는 단계로 4가지 과업으로 구성된다. 먼저 모델링의 기법을 선택하는 Select Modeling Technique, 모델의 유효성검사 및 품질테스트 절차를 생성하는 Generate Test Design, 모델링 기법을 실행하는 Build Model, 결과의 해석, 평가, 비교를 여러 평가 기준에 따라 모델의 순위를 지정하는 Assess Model이 있다.

적용(Deployment)은 프로세스를 통해서 획득한 지식을 가공하여 고객에게 제시하는 단계로, 비즈니스에 프로젝트 결과를 반영하기 위한 전략을 수립하는 Plan Deployment, 산출물의 유지 보수 전략을 준비하는 Plan Monitoring Maintenance, 프로젝트의 최종 보고서를 작성하는 Produce Final Report, 최종

보고서를 검토하는 Review Project를 수행한다.

### 3.4 SEMMA 분석 방법론

데이터마이닝의 분야가 점점 발전함에 따라 비즈니스와 경영적 접근이 아닌 데이터마이닝 자체의 단순하고 표준화된 마이닝 프로세스 확립에 대한 필요성이 대두 되었다. 비즈니스와 경영 전략을 중심으로 데이터마이닝을 적용하는 KDD와 CRISP-DM와 달리 SAS사에서 고안한 SEMMA(Sample, Explore, Modify, Model and Assess)은 데이터마이닝 틀에 맞추어진 데이터분석 방법이라 볼 수 있다. SEMMA는 통계적 관점의 방법론으로써 Fig. 7과 같이 총 5단계로 구성되며, 비즈니스 상에서 필요한 요구사항 도출/정리/처리하는 단계를 포함하지 않는다.

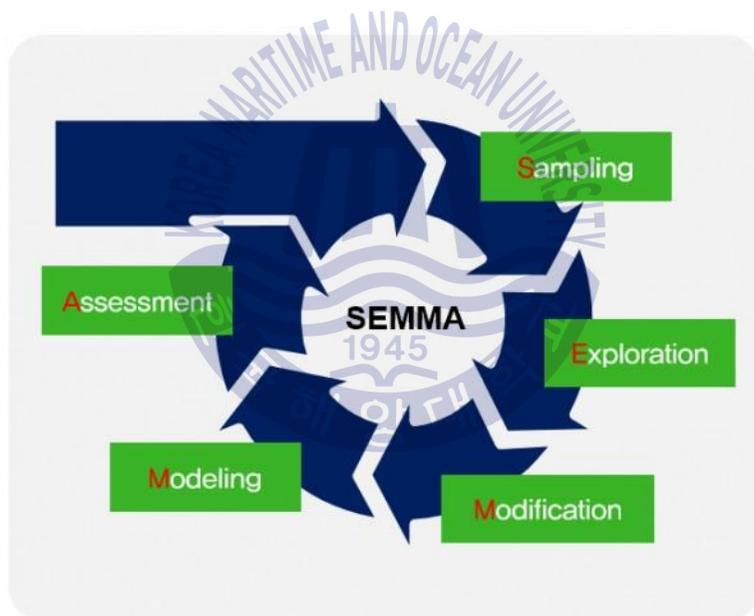


Fig. 7 SEMMA 분석 프로세스

Sampling은 분석할 데이터를 통계적 추출 혹은 일정한 조건에 따라 추출하는 과정으로 산출될 모델을 평가하기 위한 준비를 하는 과정이다. Explore는 분석할 데이터를 탐색하는 과정으로 기초 통계 분석을 통해 데이터의 특성을 파악하고 변수의 유의성을 확인하다. 데이터의 탐색은 평균이나 히스토그램

침도 등의 기초통계와 요인별 분할표, 군집화, 상관분석을 통해 수행하며 상관분석의 경우 각 변수간의 상관성과 공선성을 확인하기 위함이다. 이 과정에서 데이터의 오류를 검출해 낼 수 있으며, 데이터의 현황을 보고 비즈니스의 더욱 이해 할 수 있다. 또한 각 필드의 Outlier(극값)를 확인하고 이를 처리할 기준을 마련하게 된다. Modify는 비즈니스 적으로 의미가 있거나 분석을 보다 더 용이하게 할 수 있는 파생변수를 도출한다. 이를 위해 수량화, 표준화, 그룹화, 기타변환 작업을 사용한다. 해당 단계의 목적을 데이터가 지닌 정보의 표현을 극대화하며 최적의 모델을 구축할 수 있도록 다양한 형태로 변수를 생성, 선택, 변형하기 위함이다. Modeling은 실제로 분석 알고리즘을 사용하여 분석모델을 구축하는 과정이다. 인공신경망, 의사결정나무, 로지스틱 회귀, 전통적 통계기법을 사용할 수 있으며 이를 통해 데이터의 숨겨진 패턴을 발견한다. 예전에는 하나의 비즈니스 문제를 다루기 위해 하나의 알고리즘을 차용하는 방식으로 수행하였으나 최근 분석처리속도 및 분석알고리즘의 발전으로 복합-다중 알고리즘을 사용하여 새로운 모델을 개발하는 사례가 늘고 있다. Assessment는 모델의 평가 및 검증이 이루어지는 단계로 여러 모델 중 어떤 모델이 적합할 것인가 검증하고 도출된 모델이 얼마나 유의미한지 또는 영향력이 있는지를 파악하는 과정이다. 이를 통해 검증된 모델에 대해 가설을 확장하는 등의 과정을 거치고 추가 분석 수행 여부를 결정하게 된다.

### 3.5 분석 방법론 제안

3가지의 데이터 분석 방법론을 비교해보면 다소 유사한 특성을 가진다. 각각의 단계들을 유사한 것 끼리 묶은 후 비교하면 Table 3와 같다.

**Table 3** KDD, CRISP, SEMMA 프로세스 비교(Umair & Haseeb, 2014)

Data Mining Process Models	KDD	CRISP-DM	SEMMA
No. of Steps	9	6	5
	Domain Understanding & KDD Goals	Business Understanding	-
	Selecting and creating a data	Data Understanding	Sample
	Preprocessing: Data cleansing etc		Explore
	Data Transformation	Data Preparation	Modify
	Choosing the suitable Data Mining Task	Modeling	Model
	Choosing the suitable Data Model Mining Algorithm		
	Employing Data Mining Algorithm		
	Evaluation and Interpretation	Evaluation	Assessment
	Using Discovered Knowledge	Deployment	-

SEMMA의 경우 KDD와 CRISP-DM과 달리 비즈니스 전략 분석 및 적용에 대한 단계가 존재하지 않고 데이터마이닝 중심으로 프로세스가 진행된다. 그런 점에서 SEMMA 방법론은 비교적 소규모 프로젝트나 단위 비즈니스 분석에 적합하고 KDD와 CRISP-DM은 보다 대규모 프로젝트나 재사용이 필요할 경우 적용할 수 있는 프로세스이다.

본 논문에서는 KDD 프로세스를 보다 단순한 형태로 변형하여 분석을 수행하였다(Fig. 8). Developing and Understanding of the Application의 경우 앞의 1장과 2장을 통해서 배관재 공급망 프로세스에 대해서 이미 살펴보았다. 또한 본 연구는 조선업의 빅데이터 적용의 선행적 연구로써 데이터분석을 수행하였기 때문에 Discovered Knowledge 단계에 해당하는 현업에 대한 적용

사례 및 효과에 대한 부분은 제외하였다. Selection & Addition의 데이터 수집단계의 경우는 현재 S중공업에서 사용 중인 배관재 공급망 Raw 데이터를 그대로 사용하였기 때문에 데이터 로딩 및 추가 수집 과정은 제외하였다.

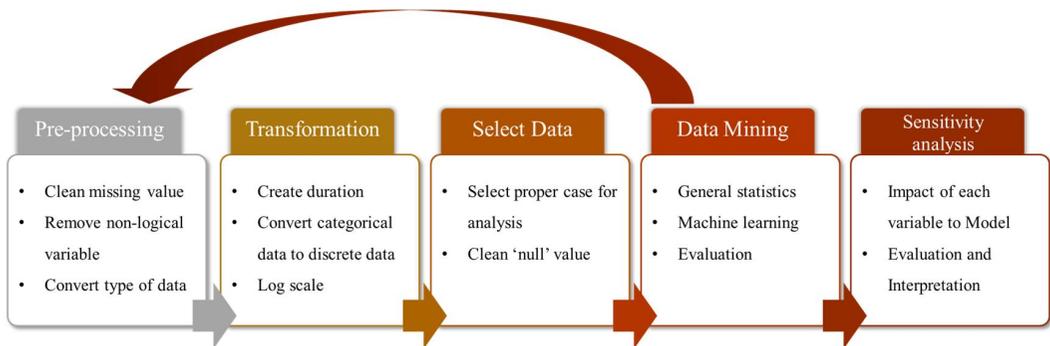


Fig. 8 배관재 공급망 데이터 분석 프로세스

Pre-processing은 데이터의 전처리 부분으로 데이터의 결측값 및 비논리값을 제외하고 데이터타입을 변환해주는 작업을 수행한다. 결측값과 비논리값은 시스템 오류 및 인적과실로 발생하는 경우가 많기 때문에 업무 프로세스 및 비즈니스의 환경적 요소를 파악하는 것이 필요하다. 데이터타입의 경우 원본데이터를 CSV형태로 분석 플랫폼에 로드하기 때문에 경우 따라 숫자형태의 문자데이터, 조선소 고유의 날짜 표기 방식과 같은 예외적인 상황이 발생할 수 있다. 그래서 데이터정의 단계에서 필드 속성 및 의미를 파악하여 해당 단계에서 정성적으로 고려해주었다.

Transformation은 분석 모델에 적합한 형태로 데이터를 변화해주는 과정으로 리드타임에 해당하는 6개의 공정기간 필드를 생성하고 범주형자료를 이산형자료로 변환해주는 과정을 포함한다. 또한 선행 데이터마이닝 결과 후 모델의 강화를 위해 데이터정규화 과정을 수행한다.

Select Data은 KDD 단계 중 Selecting and creating a data에 해당하는데 Transformation 단계의 뒤에 자리한 이유는 배관재마다 공정이 고정되어 있지 않기 때문에 먼저 목표변수를 먼저 산출한 뒤에 결측값을 확인한 후 공정별로 데이터를 선택하기 위해서이다.

## 제 4 장 예측모델의 이론적 고찰

### 4.1 일반적 통계기법의 회귀모형

#### 4.1.1 다중 선형회귀모형

다중회귀분석은 단순선형회귀의 확장으로 설명변수가 두 개 이상인 회귀모형에 대한 분석으로 회귀모형 식 (1)과 같다.

$$y = X\beta + \epsilon, \epsilon \sim N(0, \sigma^2 I) \quad (1)$$

여기서, 오차항은 평균이 0이고 분산이 일정한 정규분포를 따르고 서로 독립이라고 가정한다. 회귀계수벡터( $\beta$ )와 오차항 벡터( $\epsilon$ )는  $\beta = (\beta_0, \beta_1, \dots, \beta_p)'$ ,  $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_p)$ 이다.  $\beta$ 는 최소자승법으로 추정하며 잔차의 제곱합(Q)이 최소가 되도록 회귀계수를 구한다. Q는 식 (2)과 같이 정리할 수 있으며,  $\beta$ 의 최소제곱추정량( $\hat{\beta}$ )은 식 (3)과 같다.

$$Q = \sum [y_i - (\beta_0 + \beta_1 X_{1i} + \dots + \beta_p X_{pi})]^2 \quad (2)$$

$$\hat{\beta} = (X^t X)^{-1} X^t y \quad (3)$$

식 3에  $\hat{\beta}$ 을 계산하기 위해서는  $X^t X$ 의 역행렬이 존재하여야 한다. 역행렬이 존재하기 위해서는 행렬의 행렬식이 0이 아니어야 하고 2)full-rank여야 한다.  $rank(A_{n \times n}) = n$ 이면 full-rank라고 하며, 이를 만족하지 못하는 경우 역행렬이 존재하지 않아  $\hat{\beta}$ 의 계산이 불가능하다.

설명변수의 개수가 관측치보다 많거나, 설명변수들 간에 선형관계가 있는 경우  $X^t X$ 가 full-rank를 만족하기 힘들어  $X^t X$  역행렬의 계산이 어렵게 된다. 특히, 설명변수들 간에 선형관계가 있는 경우 역행렬이 계산되어  $\hat{\beta}$ 을 계산하더라도  $\hat{\beta}$ 의 분산이 매우 커지게 되어 모수추정에 어려움이 있다. 이러한 현상을 다중공선성(Multicollinearity)이라고 한다. 다중공선성을 진단하는 방법으로는 분산팽창계수(VIF<sub>j</sub>)를 이용하며, 일반적으로 VIF<sub>j</sub>가 5~10이상이면 다중공선성이

2) full-rank : 행렬이 역행렬을 가질 수 있는 상태

존재한다고 볼 수 있다. 다중공선성의 문제를 해결하기 위해서는 사전에 변수 선택 과정에서 상관계수가 높은 변수의 유무를 확인하여 차원 축소를 하는 것이 바람직하다. 그러나 이 경우도 반응변수를 설명하는데 중요한 변수를 제외시킬 가능성이 있다. 설명변수를 제거하지 않는 방법으로 능형회귀, PLS 회귀분석 등이 있다(Kim, 2012).

#### 4.1.2 Ridge 회귀모형

일반적으로 회귀분석은 최소제곱 추정량  $\beta = (X^t X)^{-1} X^t y$ 에서  $X^t X$ 가 역행렬이 존재한다는 가정 하에서 회귀계수를 추정한다. 하지만 역행렬이 존재하지 않는 경우 추정에 모순이 발생하게 된다. 이 경우 대각원소에 임의의 수를 더해서 식 (4)와 같이 추정치를 구할 수 있다. 이후 원래의 회귀계수 추정량을 구하는 최소제곱 방법에서 식 (5)와 같이 제약조건( $\sum_{j=1}^p \beta_j^2 \leq t^2$ )을 부여하여 회귀계수를 추정하게 된다. 이를 랑그랑주승수법을 통해 식 (6)과 같은 방정식으로 정리할 수 있다. 최소제곱법의 추정량을 살펴보면, 회귀계수가 2개일 경우 타원의 형태를 띠고 있고 제약조건은 원의 형태를 띤다.

$$\hat{\beta}^{ridge} = (X^t X + \lambda I)^{-1} X^t y \quad (4)$$

$$\hat{\beta}^{ridge} = \operatorname{argmin} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \quad (5)$$

$$\hat{\beta}^{ridge} = \operatorname{argmin} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \quad (6)$$

이 Ridge estimator는 Horel and Kennard(1970)이 소개하였고,  $p > n$  일 때 최소제곱추정량을 계산하기 위해 고안 되었다. 제약조건에 따라 Ridge와 Lasso 회귀로 나뉘는데 Lasso 회귀의 경우 차원축소가 크게 일어나므로 변수들을 최대한 줄이지 않기 위해서는 Ridge 회귀를 수행한다.

#### 4.1.3 PLS 회귀모형

PLS(Partial Least Square) 회귀분석 방법은 1966년 Herman Wold가 고안한 분석방법으로 사회과학, 특히 경제분야에 적용하기 위해 개발되었다. PLS는

주성분을 추출하는데 있어 설명변수뿐만 아니라 반응변수간의 공분산을 추가로 고려하기 때문에 반응변수를 보다 효율적으로 예측할 수 있다는 장점을 지닌다. PLS는 현재 계량분석화학(Chemometrics)분야에서 광범위하게 활용되고 있으며, Naes와 Martens(1985, 1989), Helland(1988), de Jong(1993) 등 많은 연구자에 의해 다양한 알고리즘이 개발되었다(Kim, 2012).

회귀분석에서 일반적으로 사용되는 최소자승법(Least Square)은 설명변수의 수보다 표본의 크기가 클 때 사용된다. 그러나 생명정보학, 분석화학, 생리학, 의학 등 많은 분야에서 관측 개체의 수가 설명변수의 수보다 적은 경우가 많고 변수 간 높은 상관관계를 나타내는 경우가 많다. PLS회귀는 이러한 설명변수의 수가 개체 수에 비해 많은 경우나 변수가 선형성이 높을 경우 유용한 방법이다. PLS회귀는 PCR(Principal Component Regression:주성분분석)과 자료의 변수 특성을 잘 대표한다는 점에서 공통점을 갖는다. 그러나 PCR의 경우 주성분 추출 시 설명변수만을 고려하는 반면, PLS회귀는 설명변수와 반응변수를 동시에 고려한다는 점에서 차이점이 있다. PLS의 주요 목적은 반응변수를 잘 예측할 수 있는 잠재변수를 추출하고 잠재변수를 잘 설명하는 설명변수들의 가중치를 설정하는 것이다. 그러나 지나치게 많은 잠재변수를 선정하면 분석시간이 많이 소요되고 데이터의 과적합(Overfitting)을 발생시킬 가능성을 증가시킨다.(Lee, 2007).

PLS 모델은 X(설명변수)와 Y(공정기간)의 상관관계를 이용하여 모델링을 한다.

$$X = \sum_a^A t_a p_a^t + E_A \quad (7)$$

$$Y = \sum_a^A u_a q_a^t + F_A \quad (8)$$

$t_a$ 는 Y를 고려한 축소공간에서의 이력을 나타내는 a번째 주성분 벡터,  $u_a$ 는 X를 고려한 Y 축소공간에서의 이력을 나타내는 a 번째 주성분 벡터이다.  $p_a$ 는  $X^t X$ 의 고유벡터인 loading 벡터,  $E_A$ 는 잔차행렬(Residual matrix)이고, Y도 비슷한 의미를 갖는다. 그리고 X와 Y의 주성분 벡터 사이의 내적관계(Inner

relation)로 회귀계수( $b = u^T t / t^T t$ )를 구해 모델링한다.

결정계수( $R^2$ )는 전체 제곱합(SST)에 대한 회귀 제곱합(SSR)의 비율로 모델 적합도의 척도로 사용된다. 이 값은 0과 1사이의 값을 가지며 1에 가까울수록 적합도가 높으며 0에 가까울수록 적합도가 약하다고 할 수 있다.

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{Y}_i - \bar{Y})^2}{\sum(Y_i - \bar{Y})^2} \quad (9)$$

$$RMSEP = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (10)$$

여기서,  $\hat{Y}_i$ 는 추정치,  $\bar{Y}$ 는 관측치의 평균치,  $Y_i$ 는 관측치,  $N$ 은 관측 표본 수를 나타내며, 모델을 만든 후 예측성능을 평가하는 척도로서 RMSEP가 사용된다(Yun, 2001).

배관 제작 및 설치 공정은 의장품들 중에서도 변동성이 크고 주관적인 판단에 의한 인적 오류가 포함될 가능성을 많이 가지고 있는 공종이다. 따라서 차원감소 효과 뿐 아니라 다중공선성하 뛰어난 예측 성능을 가지고 노이즈가 포함된 데이터를 분석하는 데 효과적인 PLS 회귀방법을 다중회귀분석과 함께 사용하여 비교할 것이다.

#### 4.1.4 Poisson 회귀모형

Consul and Jain(1973)이 소개한 일반화된 포아송모형(General Poisson Model)은 기존의 표준포아송모형을 확장한 모형이다. 회귀분석이나 분산분석은 종속변수가 정규분포 형태를 따르는 연속형 변수이다. 그러나 실제 종속변수가 정규분포를 따르지 않는 데이터를 분석해야할 경우 기존의 최소자승법에 의한 선형회귀모형이 부적합할 수 있다. 이러한 경우, 일반화 선형모형을 적용할 수 있다. 일반화 선형모형이란, 종속변수가 정규분포가 이외의 분포를 따를 때, 가령 Poisson, Binomial, Gaussian 등과 같은 분포를 따를 경우 사용할 수 있는 회귀모형이다.

확률변수  $Y$ 가 일반화된 포아송분포를 따른다고 하면  $Y$ 는 1부터  $n$ 까지의

값을 가진다고 가정할 수 있다. 포아송분포는  $\mu > 0$  인 조건에서 확률밀도 함수  $f(y)$ 는 식 (11)과 같다.

$$f(y) = P(Y=k) = e^{-\mu} \frac{\mu^k}{k!}, \text{ for } k=0,1,\dots \quad (11)$$

여서서 포아송변수(Poisson random variable)는  $\mu$ 와 같다. 포아송 확률모델은 일반적으로 3가지 조건을 가정한다. 첫 번째, 비중첩된 시간으로 이루어진 사건들은 각각에 대하여 독립적이다. 두 번째, 짧은 기간 동안 사건이 발생할 확률은 작다. 세 번째, 짧은 간격동안 사건이 발생할 확률은 간격의 전체 길이에 비례한다. 첫 번째 가정의 경우 포아송분포가 지수 확률분포를 따르기 때문에 일반화된 포아송분포모델 방법은 예측값이 포아송분포와 관련되어 사용된다. 포아송모델의 반응변수  $y$ 는 regressor 변수  $x_1, \dots, x_p$ 에 종속되어 있다고 가정한다. 평균이 양수이기 때문에  $y$ 에 의한 예측식은 식 (12)와 같다.

$$\mu = E(y|x_1, \dots, x_n) = e^{\beta_0 + \beta_1 x_1 + \beta_p x_p} \quad (12)$$

포아송분포는 고유하며 평균과 분산이 각각 같다. 만약 포아송의 평균이 식 (12)의 regressor와 관련되어 있다면  $Y$ 의 분산은  $var(Y) = e^{\beta_0 + \beta_1 x_1 + \beta_p x_p}$ 와 같다.

우도함수(Maximum likelihood estimation)는 일반적으로 포아송 회귀모델의 파라미터를 추정할 때 사용한다.  $y_i$ 을 예측변수  $x_i$ 에 종속되고 표본추출이  $i = 1, \dots, n$ 을 따르는 포아송변수라고 두면, 우도함수는 식 (13)과 같다.

$$L(\beta_0, \beta_1) = \frac{\prod_{i=1}^n e^{-\mu} \mu^{y_i}}{y_i!} \quad (13)$$

우도함수에 로그함수를 취해주면 식 (14)과 같다.

$$l(\beta_0, \beta_1) = \sum_{i=1}^n [-\mu_i + y_i \log(\mu_i) - \log(y_i!)] \quad (14)$$

결과적으로 식 (11)이 최대가 되는  $\beta_0$ 과  $\beta_1$ 을 찾는 것이 최종 목표가 된다. R에서는 포아송 회귀분석을 수행하기 위해 glm가 있으며 ‘family’ 파라미터에 ‘poisson’을 입력하여 분석이 가능하다.

## 4.2 빅데이터의 기계학습 알고리즘

### 4.2.1 의사결정나무 회귀모형

빅데이터는 크게 분석적인 측면에서의 데이터마이닝과 시스템구축과 관련된 분산처리시스템으로 구분해서 살펴볼 수 있다. 데이터마이닝에는 수많은 종류의 분석기법들이 존재하는데 그중 의사결정나무이나 인공신경망, 유전자알고리즘과 같이 기존의 하나의 선형식이 아닌 수많은 케이스를 생성한 뒤에 다수의 규칙들을 발견하는 기법을 기계학습이라 한다. 본 연구에서는 기계학습 기법 중 의사결정나무와 인공신경망을 사용하였다.

의사결정나무란, 의사결정 규칙을 나무 구조로 도표화하여 분류와 예측을 수행하는 분석 방법으로 분류 또는 예측의 과정이 나무 구조에 의한 추론 규칙에 의해서 표현되는 방법론이다. 이는 다른 기계학습 방법론에 비해 의사결정과정이 쉽게 이해되고 각 단계별 의사결정에 가장 큰 영향을 미친 요소가 무엇인지 알아내는 것이 용이다.

의사결정나무는 부모노드와 자식노드로 구성되는데, 부모마디에서 자식노드로 분기되는 과정에서 순수도를 증가시키는 방향으로 분기가 이루어진다. 순수도를 평가하는 척도는 데이터가 범주형인지 수치형인지에 따라 다른 척도를 사용한다. 범주형데이터의 경우 지니(Gini), 엔트로피(=정보이익, Information gain), 정보이익비율(Information gain ratio), 카이제곱 검정(Chi-square test)이 있으며 수치형데이터의 경우 분산의 감소(Reduction in variance), F검정 방법을 사용할 수 있다.

의사결정나무의 노드의 분할이 더 이상 이루어지지 않으면 그때의 상태를 최대나무(full tree)라고 한다. 최대나무의 경우 해당 데이터에 대해 과도하게 적합(Fitting)되어 있는 상태를 의미하는데, 이 경우 과적합 현상을 방지하고 보다 일반적인 분리 규칙을 만들기 위해 가지치기 방법을 수행한다. 가지치기 방법으로는 CART 가지치기와 C5 가지치기 등이 있으며 일반적으로 검증을 위한 데이터가 충분하지 못할 때 C5를 사용한다.

의사결정나무의 경우 보통 분류문제를 적용되는 방법이지만 본 연구에서는 추정문제, 즉 리드타임이라는 연속형변수를 추정하는 문제를 해결하는데 사용하였다. 이렇게 결정나무의 목표변수가 실수를 가지는 경우를 회귀나무(regression tree)라고 한다.

본 연구에서 사용한 알고리즘의 경우 MART(Multiple Additive Regression Trees)로 불리는 회귀나무 알고리즘을 사용하였다. 오픈소스 R에서는 Generalized Boosting Model(gbm)으로 불리는 이 회귀나무 알고리즘은 기존의 의사결정나무 알고리즘에 Gradient boosting 기법을 사용한 것이다. Gradient boosting이란, 분류와 추정문제를 위한 기계학습 기법으로 일반적으로 의사결정나무와 같은 예측력이 약한 모델들의 결합으로 예측모델을 생성하는 기법이다. 다른 부스팅 기법과 마찬가지로 단계별 방법으로 모델을 빌드하는데 연속적성을 갖는 손실함수를 만족하는 방향으로 모델을 생성한다.

Gradient boosting은 주로 기저분류기(base learner)중 크기가 고정된 의사결정나무(특히 CART)에 사용한다. m번째 단계의 일반적인 Gradient boosting은 유사잔차(pseudo-residuals)  $h_m(x)$ 을 따른다.  $J$ 을 잎의 개수(leave)라 하면, 나무는 입력공간을  $J$ 만큼의 분리된 공간  $R_{1m}, \dots, R_{jm}$ 으로 분리시키고 각각의 영역에서 상수값을 예측한다. 대수기하학(indicator notation)을 사용하면,  $h_m(x)$ 은 식 (15)와 같이 표현할 수 있다.

$$h_m(x) = \sum_{j=1}^J b_{jm} I(x \in R_{jm}) \quad (15)$$

식 (12)에서  $b_{jm}$ 은  $R_{jm}$  영역에서 예측된 값이다. 그 후  $b_{jm}$ 는  $\gamma_m$ 와 곱해지는데,  $\gamma_m$ 는 손실함수가 최소가 되도록 하는 값이 선택되고 식 (16)와 (17)와 같이 나타난다.

$$F_m = F_{m-1}(x) + \gamma_m h_m(x) \quad (16)$$

$$\gamma_m = \operatorname{argmin} \sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \quad (17)$$

식 (13)에서  $F_m$ 은 1과 M단계 사이에 있는 불완전한 모델이다. Friedman은

전체 나무에 대한  $\gamma_m$  대신에 최적의 분리된  $\gamma_{jm}$  값을 찾을 수 있도록 알고리즘을 변형할 것을 제안하였고, 이를 TreeBoost라고 불렀다.  $b_{jm}$ 는 나무가 피팅되는 과정에서 삭제될 수 있고 식은 와 같이 변형된다.

$$F_m = F_{m-1}(x) + \sum_{j=1}^J \gamma_{jm} h_m(x) I(x \in R_{jm}) \quad (18)$$

$$\gamma_{jm} = \operatorname{argmin}_{x_i \in R_{jm}} \sum L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \quad (19)$$

나무의 크기  $J$ 는 모델의 변수들 간의 상호작용 최대 허용수준을 정하는 파라미터로 크기가 2일 경우 변수 사이의 상호작용이 없는 One-level decision tree에 해당하게 되고 3일 경우 상호작용을 2개까지 허용한다는 것을 의미한다. 일반적으로 4에서 8사이를 사용한다(Hastie, et al., 2009).

#### 4.2.2 인공신경망 회귀모형

인공신경망은 1980년대 이후 역전파(back propagation)알고리즘의 발견과 컴퓨터 성능의 발전으로 현재까지 활발한 연구가 이루어지고 있는 분야이다. 특히 인공지능, 로봇모션, GPU(Graphics Processing Unit) 산업에서 적극적으로 활용되고 있으며, 특히 메모리에 적재된 데이터를 빠르게 처리하여 화면에 표시하는 과정에서 인공신경망 알고리즘을 활용하여 이전보다 월등히 향상된 성능을 구현하고 있다.

인공신경망은 어떻게 생물적으로 뇌가 감각 입력의 자극에 반응하는지에 대한 이해로부터 얻어진 모델을 사용해서 입력 신호와 출력 신호 간의 관계를 모델화하는 알고리즘이다. 마치 뇌가 막대한 병렬 프로세서를 생성하기 위해 뉴런이라는 세포로 연결된 망을 사용하듯이 인공신경망은 학습 문제를 풀기 위해 인공 뉴런이나 노드의 망을 사용한다.

인공신경망은 생물체의 뉴런의 신경 전달 방식에서 착안된 알고리즘이기 때문에 먼저 뉴런이 어떻게 기능하는지 이해하는 것이 도움이 된다. 그림 9와 같이 입력 신호는 상대적인 중요도나 빈도에 따라 가중치가 부여된 강도를 생화학적 과정을 통해 세포의 수상돌기로 받아들인다. 세포 몸체는 입력된

신호를 축적해 경계 값에 도달하면 세포는 출력 신호를 전기 화학적 신호로 축색돌기(axon)에 전달한다. 축색돌기의 끝단에서 화학적 신호로 처리된 전기 신호는 다시 스냅스(synaps)로 알려진 작은 간극을 통해 이웃 뉴런에 전달한다.

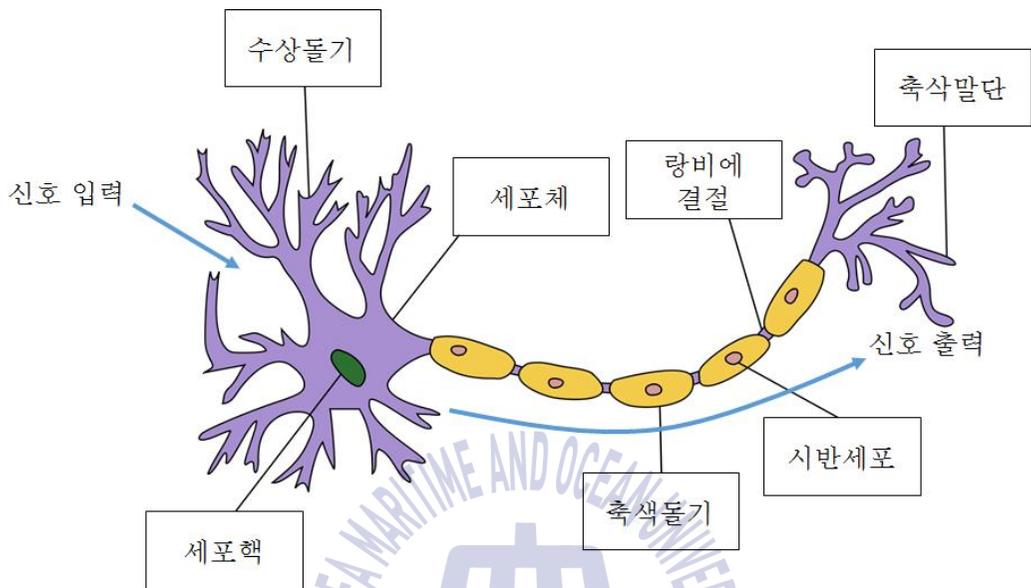


Fig. 9 생물학적 뉴런의 신경전달 구조

인공 뉴런의 모델은 생물학적 모델과 유사한 용어로 이해될 수 있다. 그림 10과 같이 인공 뉴런은 수상돌기에 입력되는 신호( $X_i$ )와 결과 신호( $Y$ ) 간의 관계를 정의한다. 생물체의 뉴런과 같이 각 돌기 신호는 중요도에 따라 가중치( $w_n$ )가 부여된다.

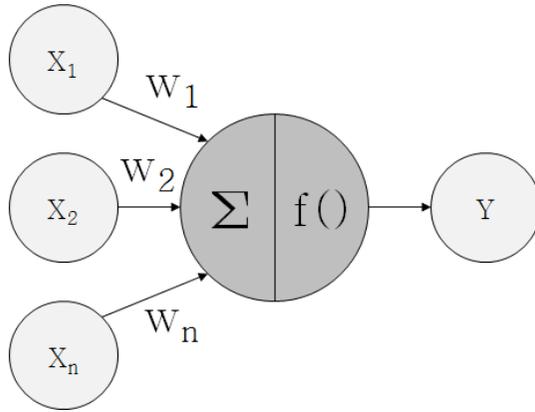


Fig. 10 인공 뉴런의 방향성 네트워크 다이어그램

$n$ 개의 입력 수상돌기를 가진 일반적인 인공 뉴런은 식 (20)과 같이 표현할 수 있다. 가중치  $w$ 는 입력신호의 총합에 영향을 주며 총 합이 활성화함수  $f$ 에 입력되면 결과 신호로써  $y(x)$ 가 축색돌기로서 출력된다(Brett, 2015).

$$y(x) = f\left(\sum_{i=1}^n w_i x_i\right) \quad (20)$$

활성화함수는 모든 입력을 하나의 값으로 통합시키는 결합함수(combination function)와 결합함수의 값을 전환하는 전환함수(transfer function)로 구성된다. 가장 흔한 결합함수는 가중 합으로, 각 입력이 가중치만큼 곱해진 후 모두 합쳐진 값이다. 전환함수의 경우 시그모이드(로지스틱), 선형, 하이포보릭 탄젠트가 있다. 선형의 경우 선형 회귀와 동일하게 작동되기 때문에 일반적으로 시그모이드와 하이포보릭 탄젠트를 사용하는데 시그모이드의 경우 출력값의 범위가 0과 1사이이고 하이포보릭 탄젠트의 경우 -1과 1사이이다.

인공신경망에서 가중치를 조정함에 있어 피드포워드(역전파 알고리즘)를 사용하는데, 피드포워드를 사용하기 위한 신경망은 3가지 층, 입력층(Input layer), 은닉층(hidden layer), 출력층(output layer)으로 구성되어 있다.

입력층은 원본데이터 필드에서 -1과 1사이의 값으로 맵핑된 값을 읽어온 뒤 출력변수로 출력하는 기능을 한다. 은닉층은 입력층에서 전달되는 출력값과 가중치를 입력값으로 취하고 마지막 층인 출력층으로 출력값과 가중치를 전달하는 층으로 입력데이터와 최종예측값 모두와 직접적으로 연결되어 있지

않기 때문에 은닉층으로 불린다. 출력층은 신경망의 최종예측값과 은닉층의 모든 단위들과 연결되어 있다. 일반적으로 하나의 출력값을 가지나 추정문제의 경우 복수개의 출력값 가지는 인공신경망을 사용하게 된다. 각각의 층(Layer)들은 앞서 설명한 결합함수를 가진 노드로 이루어져 있고 은닉층의 경우 일반적으로 2개에서 8개의 노드수를 가지며 은닉층 수가 2개 이상일 경우를 MLP(Multi Layer Perceptron)라 하며 많은 노드 수와 복수개의 은닉층이 더 정확한 결과를 낼 수도 있지만 과적합(Overfitting)의 위험을 가지고 있으므로 은닉층을 과도하게 확장해선 안 된다.



## 제 5 장 배관재 공정 리드타임 예측 모델

### 5.1 IBM Modeler 17을 활용한 데이터전처리

배관재 공정 예측모델을 생성하기 전 데이터전처리와 변환 작업을 IBM Modeler 17을 사용하여 수행하였다. 전처리는 크게 1차와 2차로 나누어 수행하였고 더미변수가 있는 것과 없는 것을 나누어 두 개의 데이터 셋을 분석에서 사용하였다. 2차 전처리는 1차 전처리 후 생성된 데이터 셋을 R과 Microsoft Azure ML에서 분석 시 필요할 때 각각의 플랫폼에서 추가적으로 수행한 전처리이다.

전처리 과정에서 우선 6개의 공정별로 목표변수를 생성하였다. 6개의 공정은 각각 Making, Painting, Out Stock, In Stock, Stand Install, Install과 같고 이들의 공정에 해당하는 시작 이벤트와 종료 이벤트간의 기간을 계산하여 6개의 새로운 필드를 생성하였다. 공정별 구분 이벤트는, Making은 제작지시가 발행된 시점부터 반출된 날짜까지의 기간을 계산하였고 Painting은 제작 후 반출된 시점부터 도장 후 물류업체로 반출되는 날짜까지의 기간을, Out Stock은 물류업체에 입고된 시점부터 야드 내로 입고되는 날짜까지의 기간을, In Stock은 야드 내로 입고된 시점부터 설치장소 근처에 적치되는 날짜까지의 기간을, Stand Install은 설치장소 근처에 적치되는 시점부터 데크 위로 올려진 날짜까지의 기간을, Install은 데크 위에 적치된 시점부터 실제 설치가 완료된 날짜까지를 계산하여 6개의 필드를 생성하였다. 단 Painting과 In Stock의 경우 배관재마다 도장을 포함하지 않은 경우가 있었기 때문에 그림 11과 같이 조건문을 사용하여 만약 도장이 없는 경우 배관 제작이 완료된 후부터 야드 입고 전까지를 Out Stock으로 계산되도록 하고 Painting은 도장 후 반출 이벤트가 존재하는 경우에만 기간을 계산하도록 코딩하였다.



Painting 기간 생성 노드



Out Stock 기간 생성 노드

Fig. 11 Painting과 Out Stock의 공정기간 생성 노드

데이터전처리 및 변환단계에서 중요한 과업 중 하나로 추후 데이터마이닝 시 필요한 설명변수와 목표변수의 분포 및 빈도분석을 수행해야 한다. 본 연구에서 사용한 데이터의 분포와 빈도분석 결과는 부록 A에 첨부한다. 데이터 분포 상태에서 무게와 길이의 경우 최소값과 최대값의 차이가 크며 지나치게 왼쪽으로 왜곡되어 있어 초기 선형회귀분석 시 계수 간 스케일 차이가 많이 나는 현상이 발생하여 그림 12와 13과 같이 로그스케일을 취해주었다.

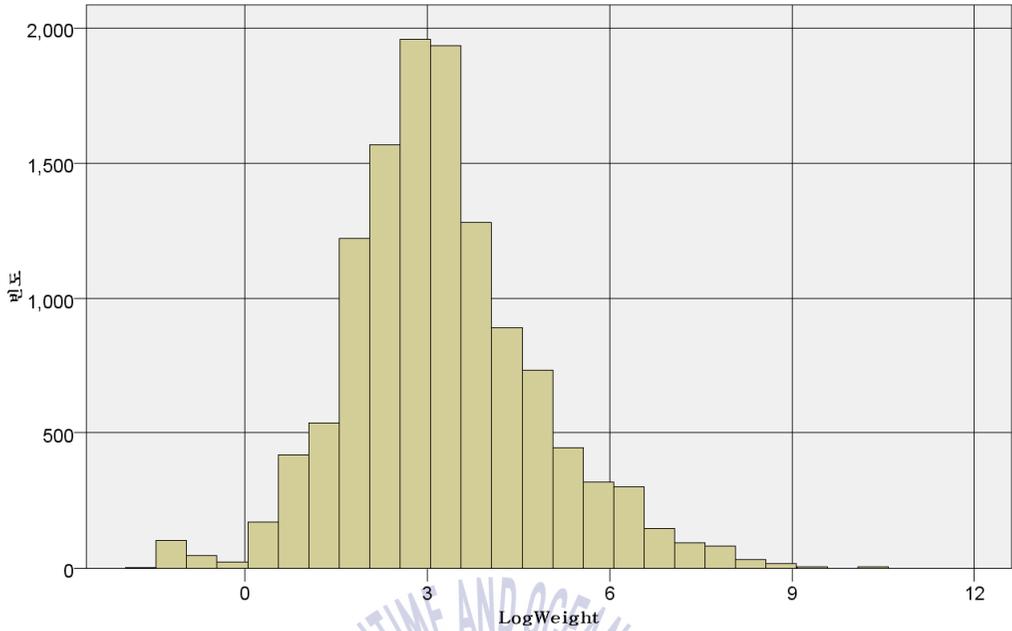


Fig. 12 로그스케일을 수행한 Weight(무게)

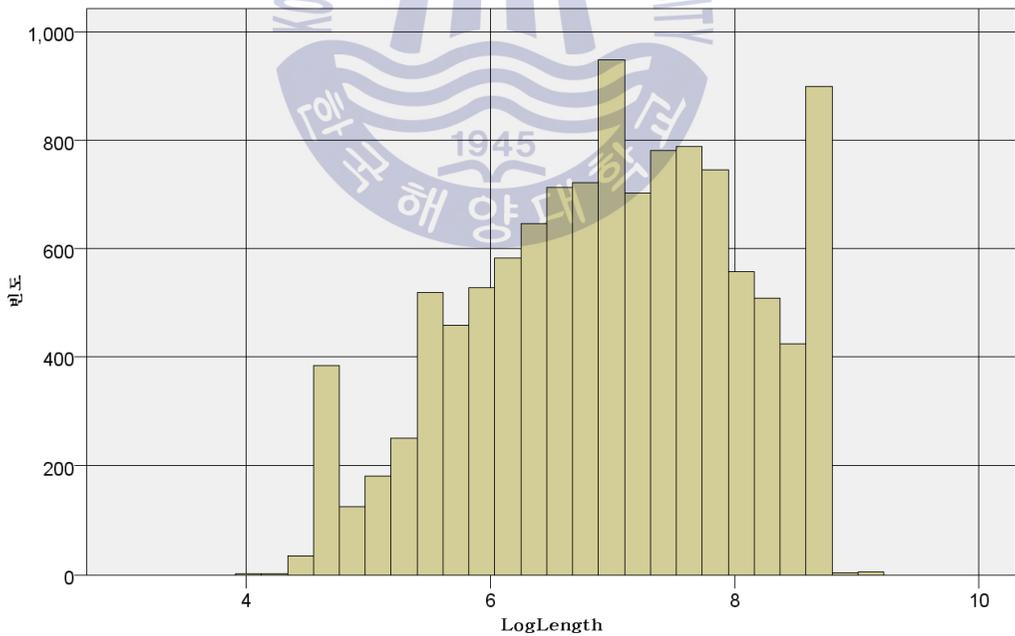


Fig. 13 로그스케일을 수행한 Length(길이)

1차 데이터전처리 과정에서 IBM Modeler에서 SPSS의 분석엔진을 탑재하여 선형회귀분석 및 PLS회귀분석을 수행할 수 있다. 선형회귀분석의 경우 생성된 모델에 평가용 데이터 노드를 연결해주면 바로 예측값을 산출해주기 때문에 결과값을 그대로 사용하였다. PLS회귀분석의 경우 계수를 뽑은 뒤에 이를 평가용 데이터에서 선형곱을 통해 예측값을 산출해야하기 때문에 범주형 변수의 이분변환을 수행하였다. 이분화 과정에서 하나의 범주형 변수의 이분화 범주 수는 전체 범주의 수의 1을 뺀 것과 같은데, 뺀 범주는 어느 것을 선택해도 상관없으나 되도록 NONE에 해당하는 범주를 제외시켰다.

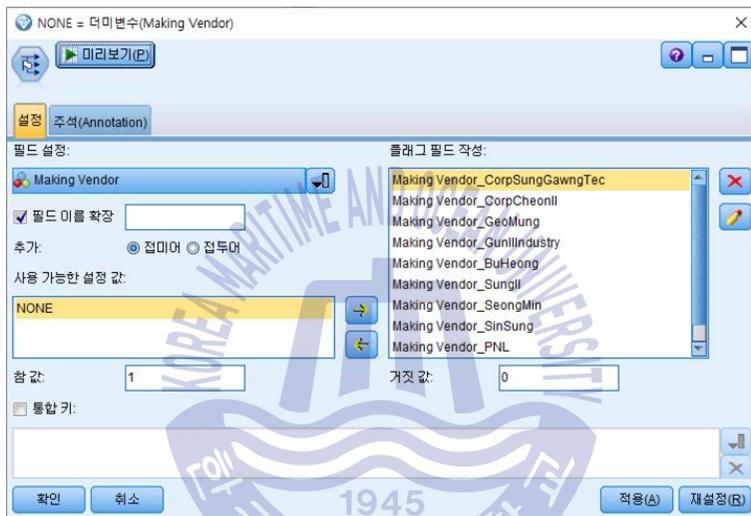


Fig. 14 범주형 변수의 이분화 노드

1차 데이터 전처리 모델은 그림 15와 16과 같다.

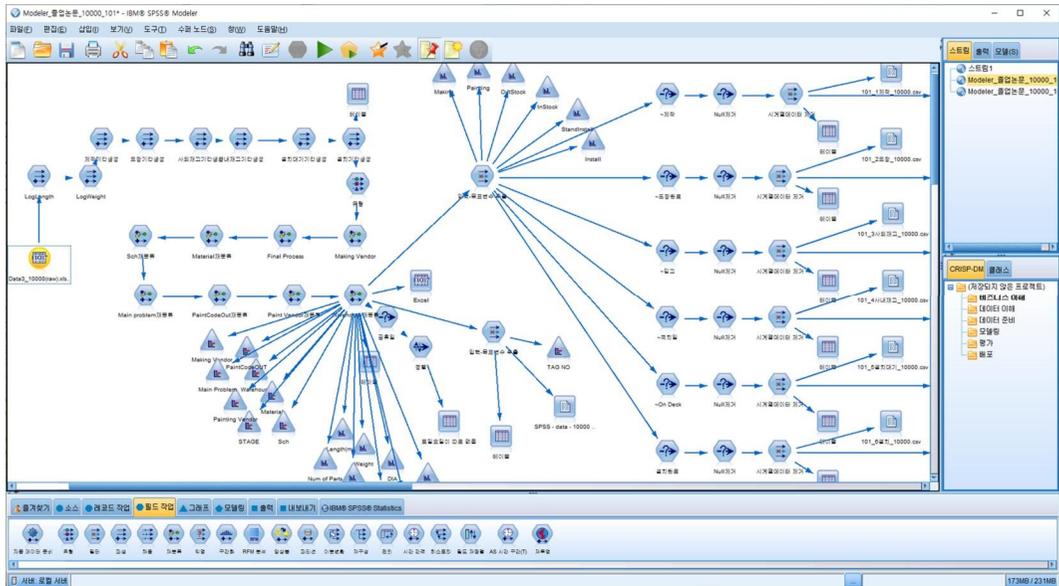


Fig. 15 1차 데이터전처리 모델(이분화노드 미포함)

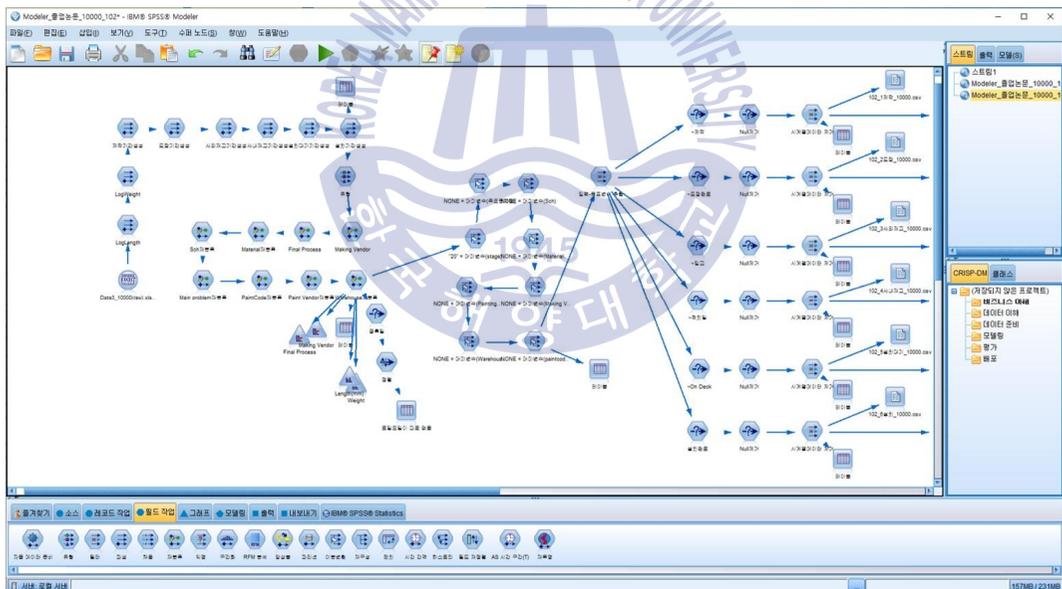


Fig. 16 1차 데이터전처리 모델(이분화노드 포함)

## 5.2 IBM SPSS을 활용한 분석

### 5.2.1 다중 선형회귀모형

SPSS을 활용한 다중 선형회귀분석은 IBM Modeler로 만든 데이터전처리 모델에서 SPSS 분석노드를 추가하여 수행하였다. 변수 선택방식은 동시입력 방식을 사용하였고 훈련과 평가용 데이터는 7대 3으로 분리하여 분석을 수행하였다. 선형회귀의 경우 예측모델의 예측값은 생성된 모델에 평가용 데이터 노드를 연결하면 자동으로 예측을 값 필드를 생성해주기 때문에 결과값을 엑셀형태로 추출한 뒤에 추가적인 평가지표 산출하였다.

다중 선형회귀분석 모델은 그림 17과 같다. 해당 모델에서 최초 분석을 통해 변수들에 대한 유의확률을 확인한 뒤 이를 제거한 뒤 다시 분석하기 위함이다. 단계별 변수선택을 할 경우 다수를 차지하고 있는 범주형 변수(이분형)들이 제거되는 문제가 발생하기 때문에 동시입력 방식의 회귀분석을 2차례 수행하였다.

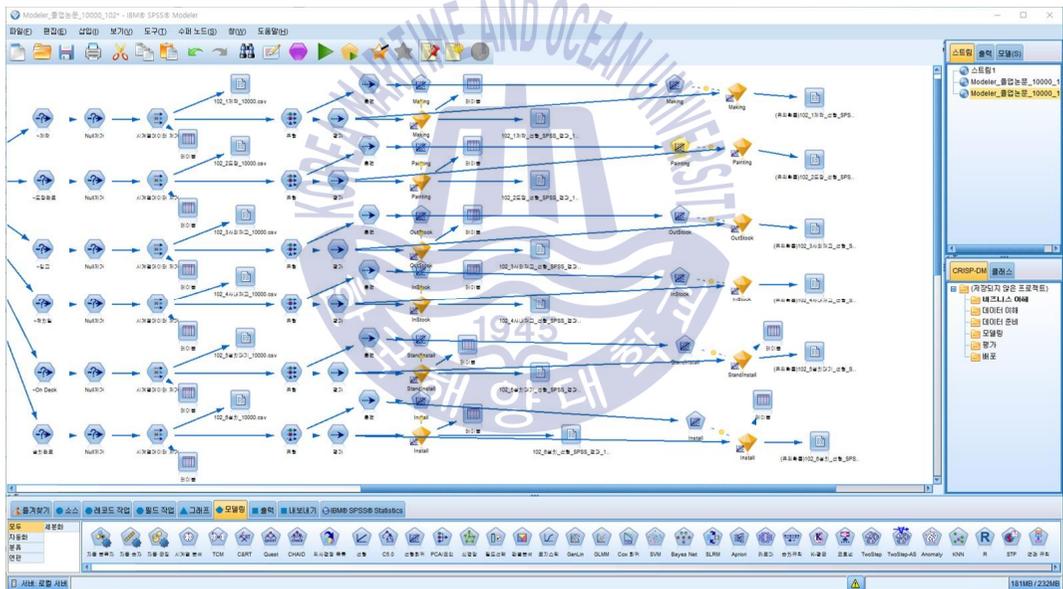


Fig. 17 SPSS를 사용한 다중 선형회귀분석 모델

### 5.2.2 PLS 회귀모형

SPSS를 사용한 PLS의 경우 기본적으로 탑재 되어 있지 않기 때문에 Python으로 구현한 분석 번들을 SPSS에 탑재하여 사용하였다. SPSS에 탑재되어있는 PLS 번들을 Modeler에서 스크립트 형태로 요청하고 결과를 받기

때문에 선형회귀분석과 같이 평가용 데이터를 예측모델노드에 연결하여 예측 결과값을 산출할 수 없다. 따라서 결과 스크립트의 계수와 평가용 데이터를 엑셀형태로 추출한 뒤에 선형공급을 취하여 예측결과값을 산출하였다. 분석모델은 그림 18과 같고 SPSS를 사용한 다중 회귀분석과 PLS분석 계수는 부록 B에 첨부한다.

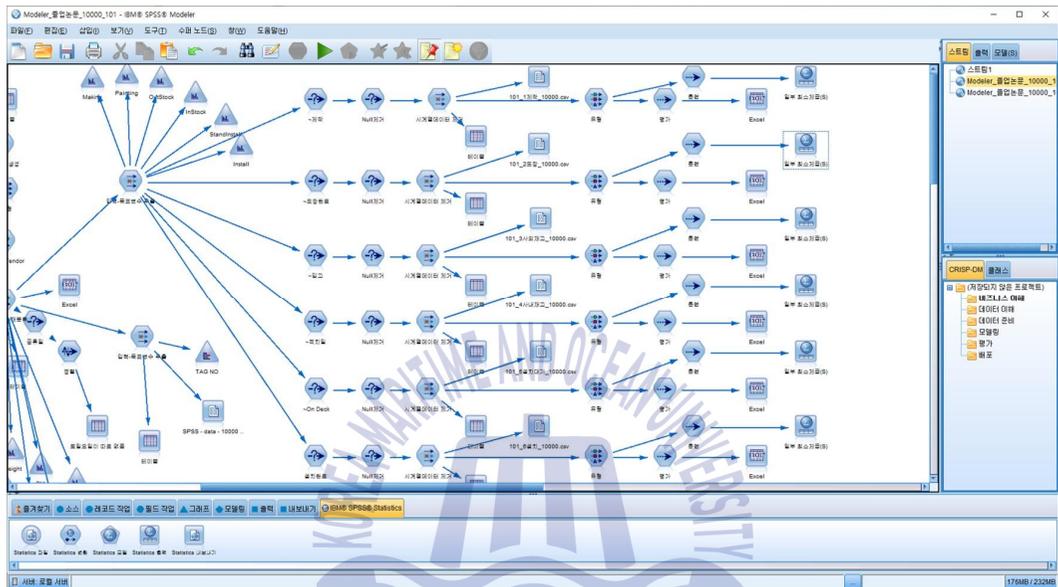


Fig. 18 SPSS를 사용한 PLS 회귀분석 모델

예측결과값은 MAE(Mean Absolute Error)와 MAPE(Mean Absolute Percentage Error) 지표를 사용하여 결과를 나타내었다. MAE는 실적값과 예측값의 절대오차값의 평균이고 MAPE는 실적값과 예측값의 절대오차비율값의 평균이다. 현장에서는 MAE가 활용 가능성이 높지만 보다 쉬운 정확도 비교를 위해 MAPE를 함께 사용하였다.

다중 선형회귀모델과 PLS 회귀모델의 결과에서 알고리즘 변화에 따른 큰 차이를 보이지 않는다. 이를 통해 배관재 공급망 데이터에서 변수 간 다중공선성의 효과가 미비한 것으로 판단할 수 있다.

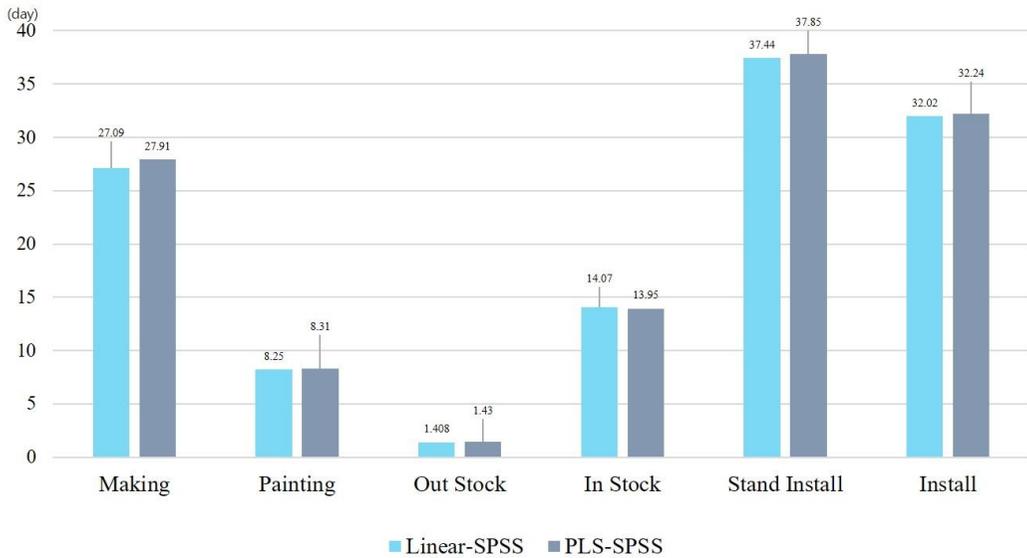


Fig. 19 SPSS를 사용한 회귀분석 결과(MAE)

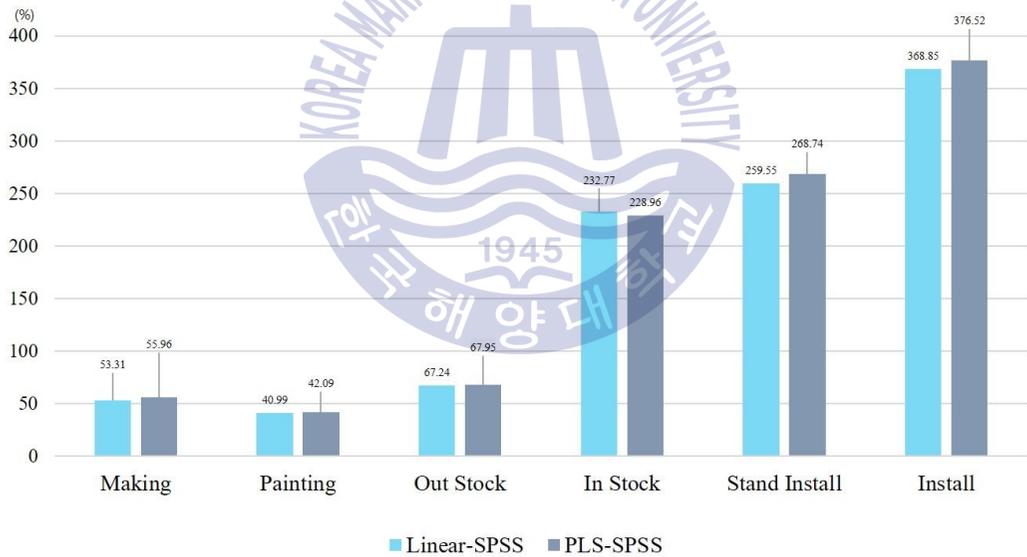


Fig. 20 SPSS를 사용한 회귀분석 결과(MAPE)

## 5.3 R을 활용한 분석

### 5.3.1 다중 선형회귀모델

R을 활용한 분석은 총 6가지의 회귀모형을 사용하여 만들었고 각각의 모형은 R에서 제공하는 분석패키지를 사용하였다. 앞서 언급했던 것처럼 IBM Modeler로 1차 데이터전처리를 수행한 6개 공정별 데이터를 R에 로딩한 뒤 R 분석패키지에 맞는 형태의 데이터 셋을 만드는 2차 데이터전처리를 수행하였다. R의 경우 3.2.4 버전을 사용하였고 보다 효과적인 코딩 및 소스코드 관리를 위해 R의 여러 IDE(Integrated development Environment)들 중 R Studio를 사용하였다.

2차 데이터전처리의 경우 데이터형태나 데이터 셋을 구성하고 학습과 평가 데이터를 분리시키는 단계까지는 모든 회귀모형에서 동일하며, 다만 분석패키지에 따라 데이터 셋 클래스 형태나 결측값 처리, 파라미터 설정, 설명변수 설정 등의 분석패키지별로 상이하게 수행해 주었다. 공통 전처리 소스코드 전문은 부록 C에 첨부한다.

R을 활용한 분석에서 다중 선형회귀분석은 기본적으로 탑재되어 있는 회귀분석 패키지인 lm패키지를 사용하였다. lm의 다중 회귀분석은 모델의 계수와 평가용 데이터를 따로 추출해서 서로 선형곱을 취하는 방식으로 예측값을 추출하였다. 다만 학습용과 평가용 데이터를 분리하는 과정에서 상대적으로 데이터수가 적은 범주의 변수가 이탈되는 현상이 발생하여 이를 평가용 데이터 범주와 회귀계수의 종류가 맞도록 추가 function을 만들어 사용하였다. 회귀분석 코드전문과 추가 function 코드전문은 Table 4, 5와 같다.

**Table 4** R을 사용한 다중 선형회귀분석 코드전문

---

```
##제작
MakingLinearModel <- lm(MakingTrain$Making~., data=MakingTrain)
write.csv(cbind("variable"=as.data.frame(names(coef(MakingLinearModel))),
"value"=as.data.frame(coef(MakingLinearModel))[, 1]),"1MakingLinearModel.csv")
summary(MakingLinearModel)
MakingTestMatrix <- as.data.frame(model.matrix(~.,data=MakingTest)) #평가데이터
행렬형태로 출력
MakingTestMatrix1 <- ConvertTestdataMeetedCoef(Model
=MakingLinearModel,testMatrix =MakingTestMatrix )
write.csv(MakingTestMatrix1, "1MakingTest.csv")

##도장
PaintLinearModel <- lm(PaintTrain$Painting~., data=PaintTrain)
```

---

---

```

write.csv(cbind("variable"=as.data.frame(names(coef(PaintLinearModel))),
"value"=as.data.frame(coef(PaintLinearModel))[,1]),"2PaintLinearModel.csv")

PaintTestMatrix <- as.data.frame(model.matrix(~.,data=PaintTest))
PaintTestMatrix1<- ConvertTestdataMeetedCoef(PaintLinearModel,PaintTestMatrix)
write.csv(PaintTestMatrix1, "2PaintTest.csv")

##사외재고
OutStockLinearModel <- lm(OutStockTrain$OutStock~., data=OutStockTrain)
write.csv(cbind("variable"=as.data.frame(names(coef(OutStockLinearModel))),
"value"=as.data.frame(coef(OutStockLinearModel))[,1]),"3OutStockLinearModel.csv")

OutStockTestMatrix <- as.data.frame(model.matrix(~.,data=OutStockTest))
OutStockTestMatrix1<-
ConvertTestdataMeetedCoef(OutStockLinearModel,OutStockTestMatrix)
write.csv(OutStockTestMatrix1, "3OutStockTest.csv")

###사내재고
InStockLinearModel <- lm(InStockTrain$InStock~., data=InStockTrain)
write.csv(cbind("variable"=as.data.frame(names(coef(InStockLinearModel))),
"value"=as.data.frame(coef(InStockLinearModel))[,1]),"4InStockLinearModel.csv")

InStockTestMatrix <- as.data.frame(model.matrix(~.,data=InStockTest))
InStockTestMatrix1<-
ConvertTestdataMeetedCoef(InStockLinearModel,InStockTestMatrix)
write.csv(InStockTestMatrix1, "4InStockTest.csv")

##설치대기
StandInstallLinearModel <- lm(StandInstallTrain$StandInstall~.,
data=StandInstallTrain)
write.csv(cbind("variable"=as.data.frame(names(coef(StandInstallLinearModel))),
"value"=as.data.frame(coef(StandInstallLinearModel))[,1]),"5StandInstallLinearModel.csv")

StandInstallTestMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTest))
StandInstallTestMatrix1<-
ConvertTestdataMeetedCoef(StandInstallLinearModel,StandInstallTestMatrix)
write.csv(StandInstallTestMatrix1, "5StandInstallTest.csv")

##설치
InstallLinearModel <- lm(InstallTrain$Install~., data=InstallTrain)
write.csv(cbind("variable"=as.data.frame(names(coef(InstallLinearModel))),
"value"=as.data.frame(coef(InstallLinearModel))[,1]),"6InstallLinearModel.csv")

InstallTestMatrix <- as.data.frame(model.matrix(~.,data=InstallTest))
InstallTestMatrix1<- ConvertTestdataMeetedCoef(InstallLinearModel,InstallTestMatrix)
write.csv(InstallTestMatrix1, "6InstallTest.csv")

```

---

**Table 5** 다중 선형회귀모델 평가데이터-계수 조정 함수

```
ConvertTestdataMeetedCoef <- function(Model, testMatrix){
  MakingTestMatrix2 <- data.frame()
  for( i in names(Model$coefficients) ){

    for(ii in names(testMatrix) ){

      if(ii==i){ MakingTestMatrix2 <- c(MakingTestMatrix2, testMatrix[ii]) }

    }
  }
  MakingTestMatrix2 <-
  c(MakingTestMatrix2,testMatrix[names(testMatrix)[length(testMatrix)] ])
  MakingTestMatrix2 <- as.data.frame(MakingTestMatrix2)
  return(MakingTestMatrix2)
}
```

### 5.3.2 Ridge 회귀모델

Ridge 회귀모형은 MASS패키지의 `lm.ridge` 함수를 사용하여 분석을 수행하였다. `lm.ridge` 함수의 경우 초기 `lambda`를 `seq(0.001, 20, 0.001)`로 설정하고 분석을 수행한 뒤 `select` 함수를 사용하여 GCV(Generalised Cross Validation)값이 최소가 되는 `lambda`값을 찾아낸 뒤 최종 Ridge모형을 생성하였다. `lambda` 초기값과 최종 입력된 `lambda` 값은 Table 6과 같다.

**Table 6** ridge 회귀모형 초기 `lambda`와 선택 `lambda`

공정	초기 <code>lambda</code>	최종 <code>lambda</code>
Making	<code>seq(0.001, 20, .001)</code>	6.938
Painting	<code>seq(0.001, 20, .001)</code>	1.409
Out Stock	<code>seq(0.001, 20, .001)</code>	6.422
In Stock	<code>seq(0.001, 20, .001)</code>	25.264
Stand Install	<code>seq(0.001, 20, .001)</code>	5.727
Install	<code>seq(0.001, 20, .001)</code>	307.439

lm.ridge 함수의 경우 data frame 형식이 아닌 matrix 형태로 변환해야 해주어 하며 이 과정에서 범주형자료의 경우 이산화변환이 수행된다. 회귀분석 코드 전문과 결측이 존재하는 이분형 필드 값 제거함수는 Table 7, 8과 같다.

**Table 7** R을 사용한 다중 선형회귀(Ridge)분석 코드전문

---

```

##제작
MakingTrainMatrix <- as.data.frame(model.matrix(~.,data=MakingTrain))
MakingTestMatrix <- as.data.frame(model.matrix(~.,data=MakingTest))

MakingTrainMatrix1 <- MakingTrainMatrix[,-128]
MakingTrainMatrix1 <- MakingTrainMatrix1[,-1]
MakingTrainMatrix1<-RemoveMissingForLidgeMatrix(MakingTrainMatrix1)

MakingRigdeModel <- lm.ridge(MakingTrainMatrix$Making~., data
=MakingTrainMatrix1 , lambda = seq(0.001, 20, .001))

select(MakingRigdeModel)

MakingRigdeModel <- lm.ridge(MakingTrainMatrix$Making~., data
=MakingTrainMatrix1 , lambda = 6.938 )
write.csv(cbind("variable"=as.data.frame(names(coef(MakingRigdeModel))),
"value"=as.data.frame(coef(MakingRigdeModel))[,1]),"1MakingRigdeModel.csv")

MakingTestMatrix1 <- MakingTestMatrix[,-128]
MakingTestMatrix1 <- RemoveMissingForLidgeMatrix(MakingTestMatrix1)
MakingTestMatrix1 <- cbind(MakingTestMatrix1, MakingTestMatrix[,128])
write.csv(MakingTestMatrix1, "1MakingTest.csv")

##도장
PaintingTrainMatrix <- as.data.frame(model.matrix(~.,data=PaintTrain))
PaintingTrainMatrix <- PaintingTrainMatrix[PaintingTrainMatrix$Painting >0,]
PaintingTestMatrix <- as.data.frame(model.matrix(~.,data=PaintTest))

PaintingTrainMatrix1 <- PaintingTrainMatrix[,-115]
PaintingTrainMatrix1 <- PaintingTrainMatrix1[,-1]
PaintingTrainMatrix1<-RemoveMissingForLidgeMatrix(PaintingTrainMatrix1)

PaintingRigdeModel <- lm.ridge(PaintingTrainMatrix$Painting ~., data
=PaintingTrainMatrix1, lambda = seq(0.001, 20, .001))

select(PaintingRigdeModel)

PaintingRigdeModel <- lm.ridge(PaintingTrainMatrix$Painting~., data
=PaintingTrainMatrix1 , lambda = 1.409)
write.csv(cbind("variable"=as.data.frame(names(coef(PaintingRigdeModel))),

```

---

---

```
"value"=as.data.frame(coef(PaintingRigdeModel))[,1]),"2PaintRigdeModel.csv")
```

```
PaintingTestMatrix1 <- PaintingTestMatrix[,-115]
PaintingTestMatrix1<-RemoveMissingForLidgeMatrix(PaintingTestMatrix1)
PaintingTestMatrix1 <- cbind(PaintingTestMatrix1, PaintingTestMatrix[,115])
write.csv(PaintingTestMatrix1, "2PaintTest.csv")
```

```
##사외재고
```

```
OutStockTrainMatrix <- as.data.frame(model.matrix(~.,data=OutStockTrain))
OutStockTestMatrix <- as.data.frame(model.matrix(~.,data=OutStockTest))
```

```
OutStockTrainMatrix1 <- OutStockTrainMatrix[,-122]
OutStockTrainMatrix1 <- OutStockTrainMatrix1[,-1]
OutStockTrainMatrix1<-RemoveMissingForLidgeMatrix(OutStockTrainMatrix1)
```

```
OutStockRigdeModel <- lm.ridge(OutStockTrainMatrix$OutStock~., data
=OutStockTrainMatrix1 , lambda = seq(0.001, 20, .001))
```

```
select(OutStockRigdeModel)
```

```
OutStockRigdeModel <- lm.ridge(OutStockTrainMatrix$OutStock~., data
=OutStockTrainMatrix1 , lambda = 6.422)
write.csv(cbind("variable"=as.data.frame(names(coef(OutStockRigdeModel))),
"value"=as.data.frame(coef(OutStockRigdeModel))[,1]),"3OutStockRigdeModel.csv")
```

```
OutStockTestMatrix1 <- OutStockTestMatrix[,-122]
OutStockTestMatrix1<-RemoveMissingForLidgeMatrix(OutStockTestMatrix1)
OutStockTestMatrix1 <- cbind(OutStockTestMatrix1, OutStockTestMatrix[,122])
write.csv(OutStockTestMatrix1, "3OutStockTest.csv")
```

```
##사내재고
```

```
InStockTrainMatrix <- as.data.frame(model.matrix(~.,data=InStockTrain))
InStockTestMatrix <- as.data.frame(model.matrix(~.,data=InStockTest))
```

```
names(InStockTrainMatrix)
InStockTrainMatrix1 <- InStockTrainMatrix[,-121]
InStockTrainMatrix1 <- InStockTrainMatrix1[,-1]
InStockTrainMatrix1<-RemoveMissingForLidgeMatrix(InStockTrainMatrix1)
```

```
InStockRigdeModel <- lm.ridge(InStockTrainMatrix$InStock~., data
=InStockTrainMatrix1 , lambda = seq(0.001, 100, .001))
```

```
select(InStockRigdeModel)
```

```
InStockRigdeModel <- lm.ridge(InStockTrainMatrix$InStock~., data
=InStockTrainMatrix1 , lambda = 25.264)
```

---

---

```

write.csv(cbind("variable"=as.data.frame(names(coef(InStockRigdeModel))),
"value"=as.data.frame(coef(InStockRigdeModel))[,1]),"4InStockRigdeModel.csv")

InStockTestMatrix1 <- InStockTestMatrix[,-121]
InStockTestMatrix1<-RemoveMissingForLidgeMatrix(InStockTestMatrix1)
InStockTestMatrix1 <- cbind(InStockTestMatrix1, InStockTestMatrix[,121])
write.csv(InStockTestMatrix1, "4InStockTest.csv")
nrow(InStockTestMatrix1)

##설치대기
StandInstallTrainMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTrain))
StandInstallTestMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTest))

names(StandInstallTrainMatrix)
StandInstallTrainMatrix1 <- StandInstallTrainMatrix[,-115]
StandInstallTrainMatrix1 <- StandInstallTrainMatrix1[,-1]
StandInstallTrainMatrix1<-RemoveMissingForLidgeMatrix(StandInstallTrainMatrix1)

StandInstallRigdeModel <- lm.ridge(StandInstallTrainMatrix$StandInstall~., data
=StandInstallTrainMatrix1 , lambda = seq(0.001, 20, .001))

select(StandInstallRigdeModel)

StandInstallRigdeModel <- lm.ridge(StandInstallTrainMatrix$StandInstall~., data
=StandInstallTrainMatrix1 , lambda = 5.727)
write.csv(cbind("variable"=as.data.frame(names(coef(StandInstallRigdeModel))),
"value"=as.data.frame(coef(StandInstallRigdeModel))[,1]),"5StandInstallRigdeModel.csv")

StandInstallTestMatrix1 <- StandInstallTestMatrix[,-115] # 종속변수 제거
StandInstallTestMatrix1<-RemoveMissingForLidgeMatrix(StandInstallTestMatrix1)
StandInstallTestMatrix1 <- cbind(StandInstallTestMatrix1, StandInstallTestMatrix[,115])
write.csv(StandInstallTestMatrix1, "5StandInstallTest.csv")

##설치
InstallTrainMatrix <- as.data.frame(model.matrix(~.,data=InstallTrain))
InstallTestMatrix <- as.data.frame(model.matrix(~.,data=InstallTest))

InstallTrainMatrix1 <- InstallTrainMatrix[,-108]
InstallTrainMatrix1 <- InstallTrainMatrix1[,-1]
InstallTrainMatrix1<-RemoveMissingForLidgeMatrix(InstallTrainMatrix1)

InstallRigdeModel <- lm.ridge(InstallTrainMatrix$Install~., data =InstallTrainMatrix1 ,
lambda = seq(0.001, 500, .001))

select(InstallRigdeModel)

InstallRigdeModel <- lm.ridge(InstallTrainMatrix$Install~., data =InstallTrainMatrix1 ,

```

---

---

```

lambda = 307.439)
write.csv(cbind("\variable"=as.data.frame(names(coef(InstallRigdeModel))),
"value"=as.data.frame(coef(InstallRigdeModel))[,1]),"6InstallRigdeModel.csv")

InstallTestMatrix1 <- InstallTestMatrix[,-108]
InstallTestMatrix1<-RemoveMissingForLidgeMatrix(InstallTestMatrix1)
InstallTestMatrix1 <- cbind(InstallTestMatrix1, InstallTestMatrix[,108])
write.csv(InstallTestMatrix1, "6InstallTest.csv")

```

---

**Table 8** 결측이 포함된 이분형 필드 값 제거함수

---

```

RemoveMissingForLidgeMatrix <- function(matrix){

for(i in length(names(matrix)):1 ){
  nm <-sum(as.data.frame(matrix[,i]))
  cat(i, names(matrix[i]), nm)
  cat("\n")
  if(nm==0) matrix <- matrix[, -i]
}
return(matrix)
}

```

---

### 5.3.3 PLS 회귀모델

PLS 회귀모델의 경우 ‘pls’ 패키지를 사용하였고 잠재변수(Latent Variables), ncomp의 초기치는 30으로 설정한 뒤 예측오차가 가장 낮은 값을 사용하였다. 검증은 LOO(Leave-one-out)과 CV(Cross-Validation) 중 CV를 사용하였고 segment계수는 default값인 10을 사용하였다. R의 ‘pls’ 패키지는 다양한 함수를 제공하는데 잠재변수 별 설명계수를 볼 수 있는 R2 함수와 모델이 자동으로 새로운 데이터의 예측값을 산출해주는 predict 함수를 통해 보다 효율적으로 결과값을 추출할 수 있다. 최종 입력한 ncomp값과 설명계수 값은 Table 9와 같다.

**Table 9** PLS 회귀모형 최종 ncomp와 R<sup>2</sup>

공정	최종 ncomp	R <sup>2</sup>
Making	26	0.3543
Painting	30	0.2068
Out Stock	19	0.2031
In Stock	15	0.2049
Stand Install	28	0.3955
Install	30	0.2069

PLS 회귀모형의 경우 예측오차가 가장 낮은 잠재변수를 찾기 위해 SelectBestPLS 함수를 별도로 만들어 사용하였다. 회귀분석 코드전문과 SelectBestPLS 함수는 Table 10, 11과 같다.

**Table 10** R을 사용한 PLS 회귀분석 코드전문

```
install.packages("pls")
library(pls)
require(pls)

##제작
names(MakingTrain)
MakingPLSModel <- plsr(Making~., ncomp = 30, data=MakingTrain, validation = "CV")
MakingPLSResult <- predict( MakingPLSModel, newdata = MakingTest)
MakingResult <- SelectBestPLS(MakingPLSResult, MakingTest, 14)
write.csv(MakingResult, "1MakingResult.csv")

##도장
names(PaintTrain)
PaintPLSModel <- plsr(Painting~., ncomp = 30, data=PaintTrain)
PaintPLSResult <- predict( PaintPLSModel, newdata = PaintTest)
PaintResult <- SelectBestPLS(PaintPLSResult, PaintTest, 14)
write.csv(PaintResult, "2PaintResult.csv")

##사외재고
names(OutStockTrain)
OutStockPLSModel <- plsr(OutStock~., ncomp = 30, data=OutStockTrain)
OutStockPLSResult <- predict( OutStockPLSModel, newdata = OutStockTest)
```

---

```
OutStockResult <- SelectBestPLS(OutStockPLSResult, OutStockTest, 14)
write.csv(OutStockResult, "3OutStockResult.csv")
```

```
##사내재고
names(InStockTrain)
InStockPLSModel <- plsr(InStock~., ncomp = 30, data=InStockTrain)
InStockPLSResult <- predict( InStockPLSModel, newdata = InStockTest)
InStockResult <- SelectBestPLS(InStockPLSResult, InStockTest, 14)
write.csv(InStockResult, "4InStockResult.csv")
```

```
##설치대기
names(StandInstallTrain)
StandInstallPLSModel <- plsr(StandInstall~., ncomp = 30, data=StandInstallTrain)
StandInstallPLSResult <- predict( StandInstallPLSModel, newdata = StandInstallTest)
StandInstallResult <- SelectBestPLS(StandInstallPLSResult, StandInstallTest, 14)
write.csv(StandInstallResult, "5StandInstallResult.csv")
```

```
##설치
names(InstallTrain)
InstallPLSModel <- plsr(Install~., ncomp = 30, data=InstallTrain)
InstallPLSResult <- predict( InstallPLSModel, newdata = InstallTest)
InstallResult <- SelectBestPLS(InstallPLSResult, InstallTest, 14)
write.csv(InstallResult, "6InstallResult.csv")
```

---

**Table 11** SelectBestPLS 함수

---

```
SelectBestPLS <- function(PLSResult, testdata, valN)
{
  MaeList <- data.frame()
  PLSResultDB <- as.data.frame(PLSResult)
  count_ncomp <- length(names(as.data.frame(PLSResult)))

  for(i in 1:count_ncomp)
  {
    Level_Test <- cbind(testdata[,valN], PLSResultDB[[i])

    names(Level_Test)[1] <- "Real"
    names(Level_Test)[2] <- "level"
    Level_Test$MAE <- abs(Level_Test$Real - Level_Test$level)
    Level_Test <- na.omit(Level_Test)
    MAE <- mean(Level_Test$MAE)

    MaeList[i,1] <- MAE
```

---

---

```

}

ProperNcomp <- which(MaeList$V1==min(MaeList$V1))
print(count_ncomp)
print(MaeList)

return(resultDB <- cbind(testdata, PLSResultDB[[ProperNcomp]])
}

```

---

### 5.3.4 Poisson 회귀모델

Poisson 회귀모형의 경우 목표변수가 poisson 분포를 따를 때 주로 사용하는 GLM(General Linear Model)로 예측값이 로그(log) 형태로 나오는 것이 특징이다. 분석패키지는 R의 ‘glm’을 사용하였고 포아송분포를 설정하기 위해 함수의 family 값에 poisson을 설정하였다. 또한 선형회귀분석과 마찬가지로 모델과 평가용 데이터 간의 계수를 맞추기 위해 별도의 함수를 정의하여 사용하였고 CSV파일로 추출한 뒤 선형곱을 해주어 예측값을 산출하였다. 회귀분석 코드전문은 Table 12와 같다.

Table 12 R을 사용한 Poisson 회귀분석 코드전문

---

```

##제작
MakingTrainMatrix <- as.data.frame(model.matrix(~.,data=MakingTrain))
MakingTrainMatrix <- MakingTrainMatrix[,-1]
MakingTestMatrix <- as.data.frame(model.matrix(~.,data=MakingTest))

MakingPoissonModel <- glm(Making~., MakingTrainMatrix, family = poisson)
MakingPoissonModel1 <- na.omit(coef(MakingPoissonModel)) 삭제
MakingTestMatrix1 <-
ConvertTestdataMeetedCoefList(MakingPoissonModel1,MakingTestMatrix)
write.csv(MakingPoissonModel1, "1MakingPoissonModel.csv")
write.csv(MakingTestMatrix1, "1MakingTest.csv")

##도장
PaintTrainMatrix <- as.data.frame(model.matrix(~.,data=PaintTrain))
PaintTrainMatrix <- PaintTrainMatrix[,-1]
PaintTrainMatrix <- PaintTrainMatrix[PaintTrainMatrix$Painting>=0,]

PaintTestMatrix <- as.data.frame(model.matrix(~.,data=PaintTest))

```

---

---

```

PaintPoissonModel <- glm(Painting~., PaintTrainMatrix, family = poisson)
PaintPoissonModel1 <- na.omit(coef(PaintPoissonModel))
PaintTestMatrix1 <-
ConvertTestdataMeetedCoefList(PaintPoissonModel1,PaintTestMatrix)
write.csv(PaintPoissonModel1, "2PaintPoissonModel.csv")
write.csv(PaintTestMatrix1, "2PaintTest.csv")

##사외재고
OutStockTrainMatrix <- as.data.frame(model.matrix(~.,data=OutStockTrain))
OutStockTrainMatrix <- OutStockTrainMatrix[,-1]
OutStockTrainMatrix <- OutStockTrainMatrix[OutStockTrainMatrix$OutStock>=0,]
OutStockTestMatrix <- as.data.frame(model.matrix(~.,data=OutStockTest))

OutStockPoissonModel <- glm(OutStock~., OutStockTrainMatrix, family = poisson)
OutStockPoissonModel1 <- na.omit(coef(OutStockPoissonModel))
OutStockTestMatrix1 <-
ConvertTestdataMeetedCoefList(OutStockPoissonModel1,OutStockTestMatrix)
write.csv(OutStockPoissonModel1, "3OutStockPoissonModel.csv")
write.csv(OutStockTestMatrix1, "3OutStockTest.csv")

##사내재고
InStockTrainMatrix <- as.data.frame(model.matrix(~.,data=InStockTrain))
InStockTrainMatrix <- InStockTrainMatrix[,-1]
InStockTrainMatrix <- InStockTrainMatrix[InStockTrainMatrix$InStock>=0,]
InStockTestMatrix <- as.data.frame(model.matrix(~.,data=InStockTest))

InStockPoissonModel <- glm(InStock~., InStockTrainMatrix, family = poisson)
InStockPoissonModel1 <- na.omit(coef(InStockPoissonModel))
InStockTestMatrix1 <-
ConvertTestdataMeetedCoefList(InStockPoissonModel1,InStockTestMatrix)
write.csv(InStockPoissonModel1, "4InStockPoissonModel.csv")
write.csv(InStockTestMatrix1, "4InStockTest.csv")

##설치대기
StandInstallTrainMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTrain))
StandInstallTrainMatrix <- StandInstallTrainMatrix[,-1]
StandInstallTrainMatrix <-
StandInstallTrainMatrix[StandInstallTrainMatrix$StandInstall>=0,]
StandInstallTestMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTest))

StandInstallPoissonModel <- glm(StandInstall~., StandInstallTrainMatrix, family =
poisson)
StandInstallPoissonModel1 <- na.omit(coef(StandInstallPoissonModel))
StandInstallTestMatrix1 <-
ConvertTestdataMeetedCoefList(StandInstallPoissonModel1,StandInstallTestMatrix)
write.csv(StandInstallPoissonModel1, "5StandInstallPoissonModel.csv")

```

---

---

```
write.csv(StandInstallTestMatrix1, "5StandInstallTest.csv")
```

```
##설치
```

```
InstallTrainMatrix <- as.data.frame(model.matrix(~.,data=InstallTrain))
```

```
InstallTrainMatrix <- InstallTrainMatrix[,-1]
```

```
InstallTrainMatrix <- InstallTrainMatrix[InstallTrainMatrix$Install>=0,]
```

```
InstallTestMatrix <- as.data.frame(model.matrix(~.,data=InstallTest))
```

```
InstallPoissonModel <- glm(Install~., InstallTrainMatrix, family = poisson)
```

```
InstallPoissonModel1 <- na.omit(coef(InstallPoissonModel))
```

```
InstallTestMatrix1 <-
```

```
ConvertTestdataMeetedCoefList(InstallPoissonModel1,InstallTestMatrix)
```

```
write.csv(InstallPoissonModel1, "6InstallPoissonModel.csv")
```

```
write.csv(InstallTestMatrix1, "6InstallTest.csv")
```

---

### 5.3.5 인공신경망 회귀모델

인공신경망 회귀모형은 ‘nnet’ 라는 분석패키지를 사용하였다. nnet의 경우 Brian Ripley와 William Venables가 작성한 단일 은닉층 구조의 피트포워딩 방식을 적용한 인공신경망 분석 패키지이다. nnet 함수에는 은닉층의 노드 개수를 의미한 size와 반복횟수를 의미하는 maxit, 출력값의 형태를 선택하는 linout, 가중치들의 과적합 방지를 위한 decay가 있다. size와 maxit의 경우 크게 설정한다고 해서 무조건 좋은 모델이 만들어 지는 것은 아니다. 본 연구에서는 컴퓨터의 사양 및 처리속도를 고려해서 각각 5개와 1000회로 설정하여 분석하였다. nnet의 경우 predict 함수를 지원하기 때문에 예측값 추출을 위한 별도의 엑셀작업을 수행하지 않았다. 회귀분석 코드전문은 Table 13과 같다.

Table 13 R을 사용한 인공신경망 회귀분석 코드전문

---

```
install.packages("nnet")
```

```
library("nnet")
```

```
##제작
```

```
MakingTrainMatrix <- as.data.frame(model.matrix(~.,data=MakingTrain))
```

```
MakingTrainMatrix <- MakingTrainMatrix[,-1]
```

```
MakingTestMatrix <- as.data.frame(model.matrix(~.,data=MakingTest))
```

```
MakingNnet <- nnet(Making~., data=MakingTrainMatrix, size=5, maxit=1000, linout=T,
```

---

---

```

decay=0.01)
MakingNnetResult <- predict(MakingNnet,MakingTestMatrix)

MakingNnetResult <- cbind(as.data.frame(MakingTestMatrix),MakingNnetResult)
write.csv(MakingNnetResult,"1MakingANNModel.csv")

##도장
PaintTrainMatrix <- as.data.frame(model.matrix(~.,data=PaintTrain))
PaintTrainMatrix <- PaintTrainMatrix[,-1]
PaintTestMatrix <- as.data.frame(model.matrix(~.,data=PaintTest))

PaintNnet <- nnet(Painting~., data=PaintTrainMatrix, size=5, maxit=1000, linout=T,
decay=0.01)
PaintNnetResult <- predict(PaintNnet,PaintTestMatrix)

PaintNnetResult <- cbind(as.data.frame(PaintTestMatrix),PaintNnetResult)
write.csv(PaintNnetResult,"2PaintANNModel.csv")

##사외재고
OutStockTrainMatrix <- as.data.frame(model.matrix(~.,data=OutStockTrain))
OutStockTrainMatrix <- OutStockTrainMatrix[,-1]
OutStockTestMatrix <- as.data.frame(model.matrix(~.,data=OutStockTest))

OutStockNnet <- nnet(OutStock~., data=OutStockTrainMatrix, size=5, maxit=1000,
linout=T, decay=0.01)
OutStockNnetResult <- predict(OutStockNnet,OutStockTestMatrix)

OutStockNnetResult <- cbind(as.data.frame(OutStockTestMatrix),OutStockNnetResult)
write.csv(OutStockNnetResult,"3OutStockANNModel.csv")

##사내재고
InStockTrainMatrix <- as.data.frame(model.matrix(~.,data=InStockTrain))
InStockTrainMatrix <- InStockTrainMatrix[,-1]
InStockTestMatrix <- as.data.frame(model.matrix(~.,data=InStockTest))

InStockNnet <- nnet(InStock~., data=InStockTrainMatrix, size=5, maxit=1000, linout=T,
decay=0.01)
InStockNnetResult <- predict(InStockNnet,InStockTestMatrix)

InStockNnetResult <- cbind(as.data.frame(InStockTestMatrix),InStockNnetResult)
write.csv(InStockNnetResult,"4InStockANNModel.csv")

##설치대기
StandInstallTrainMatrix <- as.data.frame(model.matrix(~.,data=StandInstallTrain))
StandInstallTrainMatrix <- StandInstallTrainMatrix[,-1]

```

---

---

```

StandInstallTestMatrix <- as.data.frame(model.matrix(~..data=StandInstallTest))

StandInstallNnet <- nnet(StandInstall~, data=StandInstallTrainMatrix, size=5,
maxit=1000, linout=T, decay=0.01)
StandInstallNnetResult <- predict(StandInstallNnet,StandInstallTestMatrix)

StandInstallNnetResult <-
cbind(as.data.frame(StandInstallTestMatrix),StandInstallNnetResult)
write.csv(StandInstallNnetResult,"5StandInstallANNModel.csv")

##설치
InstallTrainMatrix <- as.data.frame(model.matrix(~.,data=InstallTrain))
InstallTrainMatrix <- InstallTrainMatrix[,-1]
InstallTestMatrix <- as.data.frame(model.matrix(~.,data=InstallTest))

InstallNnet <- nnet(Install~, data=InstallTrainMatrix, size=5, maxit=1000, linout=T,
decay=0.01)
InstallNnetResult <- predict(InstallNnet,InstallTestMatrix)

InstallNnetResult <- cbind(as.data.frame(InstallTestMatrix),InstallNnetResult)
write.csv(InstallNnetResult,"6InstallANNModel.csv")

```

---

### 5.3.6 의사결정나무 회귀모델

본문의 의사결정나무 회귀모델의 경우 Greg Ridgeway가 Friedman's gradient boosting 기법을 적용하여 제작한 분석패키지이다. gbm 함수에는 출력값의 형태를 정의하는 distribution과 tree의 총 개수를 의미하는 n.tree, tree의 깊이를 의미하는 interaction.depth, 배깅(bagging) 시 집합 선택 크기를 의미하는 bag.fraction, 학습 집합 크기를 나타내는 train.fraction, 학습률을 의미하는 shrinkage 등으로 구성되어 있다. 이중 n.tree와 interaction.depth 값을 크게 설정할 경우 해당 집합에만 과적합 되는 현상이 발생할 수 있다. 본 연구에서는 tree 수는 1000개, tree 깊이는 3으로 설정하여 분석을 수행하였고 리드타임 예측은 추정문제에 해당하므로 distribution은 “gaussian” 으로 설정하였다. shrinkage의 경우 작게 할수록 최적값에 가깝게 수렴할 수 있으나 시간이 많이 걸리고 자칫 local minimum(국소 최적값)에 수렴하는 문제가 발생할 수 있다. 본 연구에서는 0.1로 설정하여 분석을 수행하였다.

PLS 분석과 마찬가지로 오차평균값이 가장 낮은 tree 수를 구하기 위해 SelectBsetGbm 함수를 정의하였다. 회귀분석 코드전문 및 SelectBestGbm 함수는 Table 14, 15와 같다.

**Table 14** R을 사용한 의사결정나무 회귀분석 코드전문

---

```
##제작
MakingGbmModel <- gbm(Making ~., data=MakingTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
MakingGbmResult <- SelectBestGbm(MakingTest,MakingGbmModel,1000,14)
write.csv(MakingGbmResult,"1MakingGbmModel.csv")

##도장
as.factor(names(PaintTrain))
PaintGbmModel <- gbm(Painting ~., data=PaintTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)

PaintGbmResult <- SelectBestGbm(PaintTest,PaintGbmModel,1000,14)
write.csv(PaintGbmResult,"2PaintGbmModel.csv")

##사외재고
as.factor(names(OutStockTrain))
OutStockTrain1 <- na.omit(OutStockTrain) # 869 > 862
OutStockGbmModel <- gbm(OutStock ~., data=OutStockTrain1 , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)

OutStockTest1 <- na.omit(OutStockTest)
OutStockGbmResult <- SelectBestGbm(OutStockTest1,OutStockGbmModel,1000,14)
write.csv(OutStockGbmResult, "3OutStockGbmModel.csv")

##사내재고
as.factor(names(InStockTrain))
InStockGbmModel <- gbm(InStock ~., data=InStockTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
InStockGbmResult <- SelectBestGbm(InStockTest,InStockGbmModel,1000,14)
write.csv(InStockGbmResult, "4InStockGbmModel.csv")

##설치대기
as.factor(names(StandInstallTrain))
```

---

---

```
StandInstallGbmModel <- gbm(StandInstall ~ ., data=StandInstallTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
StandInstallGbmResult <-
SelectBestGbm(StandInstallTest,StandInstallGbmModel,1000,14)
write.csv(StandInstallGbmResult, "5StandInstallGbmModel.csv")
```

```
##설치
```

```
as.factor(names(InstallTrain))
InstallGbmModel <- gbm(Install ~ ., data=InstallTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
```

```
InstallGbmResult <- SelectBestGbm(InstallTest,InstallGbmModel,1000,14)
write.csv(InstallGbmResult, "6InstallGbmModel.csv")
```

---

**Table 15** SelectBestGbm 함수

---

```
##제작
```

```
MakingGbmModel <- gbm(Making ~., data=MakingTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
MakingGbmResult <- SelectBestGbm(MakingTest,MakingGbmModel,1000,14)
write.csv(MakingGbmResult,"1MakingGbmModel.csv")
```

```
##도장
```

```
as.factor(names(PaintTrain))
PaintGbmModel <- gbm(Painting ~., data=PaintTrain , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
```

```
PaintGbmResult <- SelectBestGbm(PaintTest,PaintGbmModel,1000,14)
write.csv(PaintGbmResult,"2PaintGbmModel.csv")
```

```
##사외재고
```

```
as.factor(names(OutStockTrain))
OutStockTrain1 <- na.omit(OutStockTrain) # 869 > 862
OutStockGbmModel <- gbm(OutStock ~., data=OutStockTrain1 , n.trees=1000,
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,
shrinkage=0.1, keep.data=TRUE)
```

```
OutStockTest1 <- na.omit(OutStockTest)
OutStockGbmResult <- SelectBestGbm(OutStockTest1,OutStockGbmModel,1000,14)
```

---

---

```
write.csv(OutStockGbmResult, "3OutStockGbmModel.csv")
```

```
##사내재고
```

```
as.factor(names(InStockTrain))
```

```
InStockGbmModel <- gbm(InStock ~., data=InStockTrain , n.trees=1000,  
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,  
shrinkage=0.1, keep.data=TRUE)
```

```
InStockGbmResult <- SelectBestGbm(InStockTest,InStockGbmModel,1000,14)
```

```
write.csv(InStockGbmResult, "4InStockGbmModel.csv")
```

```
##설치대기
```

```
as.factor(names(StandInstallTrain))
```

```
StandInstallGbmModel <- gbm(StandInstall ~ .. data=StandInstallTrain , n.trees=1000,  
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,  
shrinkage=0.1, keep.data=TRUE)
```

```
StandInstallGbmResult <-
```

```
SelectBestGbm(StandInstallTest,StandInstallGbmModel,1000,14)
```

```
write.csv(StandInstallGbmResult, "5StandInstallGbmModel.csv")
```

```
##설치
```

```
as.factor(names(InstallTrain))
```

```
InstallGbmModel <- gbm(Install ~ .. data=InstallTrain , n.trees=1000,  
distribution="gaussian", interaction.depth=3, bag.fraction=0.5, train.fraction=1.0,  
shrinkage=0.1, keep.data=TRUE)
```

```
InstallGbmResult <- SelectBestGbm(InstallTest,InstallGbmModel,1000,14)
```

```
write.csv(InstallGbmResult, "6InstallGbmModel.csv")
```

---

### 5.3.7 R을 활용한 회귀분석 결과 비교

R의 6가지 분석패키지를 사용하여 회귀분석을 수행한 후 각각의 결과를 MAE와 MAPE로 표시하면 그림 21, 22와 같다. 결과에서 Tree(Boosted Decision Tree Regression: 의사결정나무 회귀)가 가장 낮은 오차를 가지고 있었고 ANN(Artificial Neural Network: 인공신경망 회귀)가 가장 높은 오차를 가지고 있었다. 선·후행 공정간 비교에서는 상대적으로 선행 단계인 Making, Painting, Out Stock 공정이 In Stock, Stand Install, Install 보다 더 정확했다.

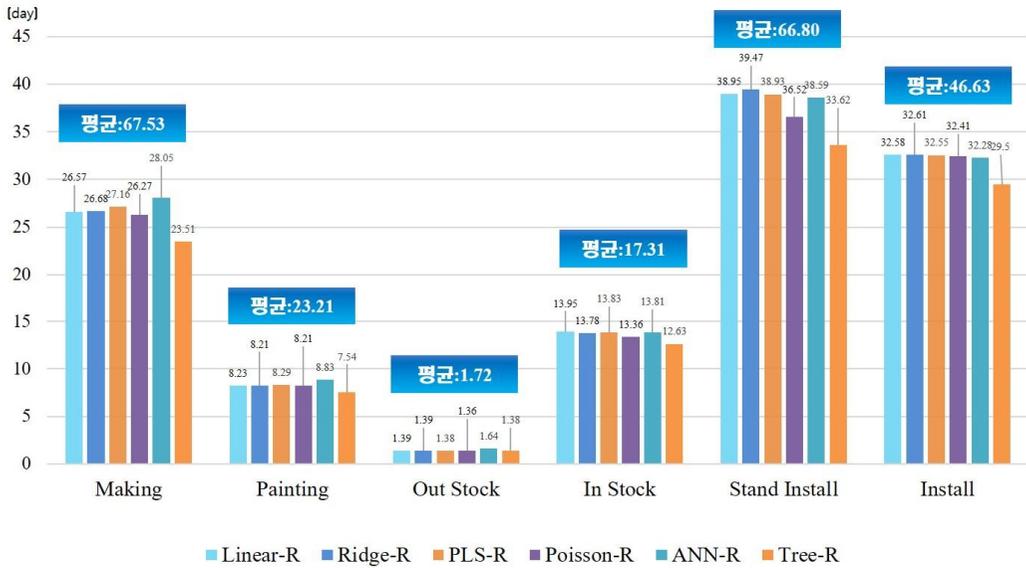


Fig. 21 R을 사용한 회귀분석 결과(MAE)

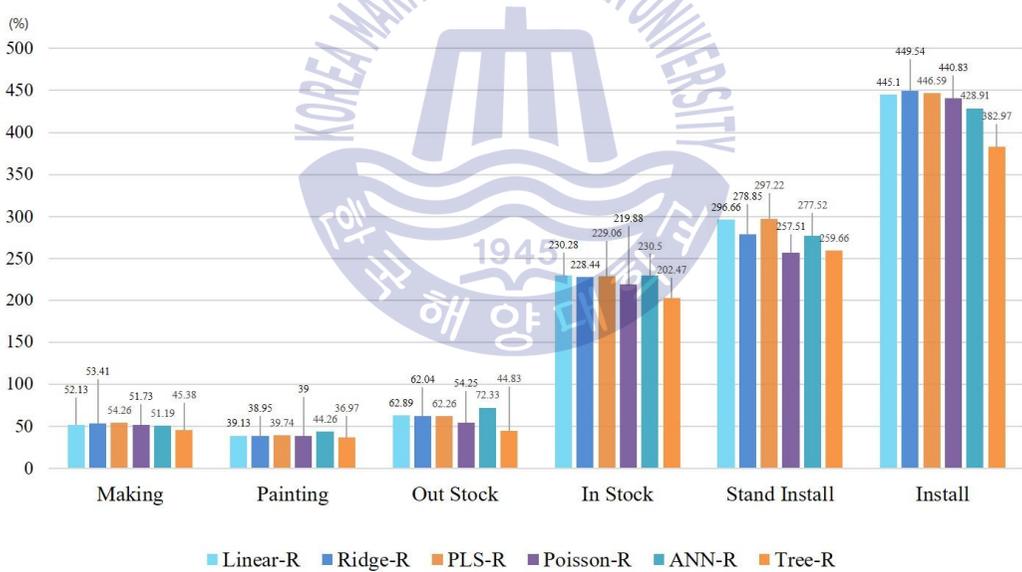


Fig. 22 R을 사용한 회귀분석 결과(MAPE)

## 5.4 Microsoft Azure를 활용한 분석

### 5.4.1 인공지능망 회귀모델

Azure ML은 Microsoft에서 제공하고 있는 오픈 분석 플랫폼으로 데이터전처리 및 변환을 위한 다양한 기능 및 분석모듈을 제공하고 있다. Microsoft Azure의 경우 분석모듈을 GUI(Graphical User Interface) 형태로 작성할 수 있도록 구성되어 있기 때문에 비교적 쉽게 모델을 만들 수 있으며, Server-Side Program 형태의 분산처리시스템을 구축하고 있어 Client의 컴퓨터 성능과 관계없이 매우 빠른 분석 속도를 나타낸다. 또한 개발한 분석모듈을 웹상에서 API 형태로 배포할 수 있다. 본 연구에서는 Azure ML의 여러 알고리즘들 중 인공신경망 회귀와 의사결정나무 회귀를 사용하여 분석하였다. 모델구성 방법은 Azure ML에 정의하고 있는 가이드라인을 따랐고 앞서 수행했던 1차 전처리데이터를 Microsoft의 DATASETS에 탑재한 뒤 작업공간에서 간단한 결측값 제거만 수행한 뒤 분석을 수행하였다. 분석모델은 그림 23, 24와 같다.



Fig. 23 Azure ML을 활용한 분석모델(전체)

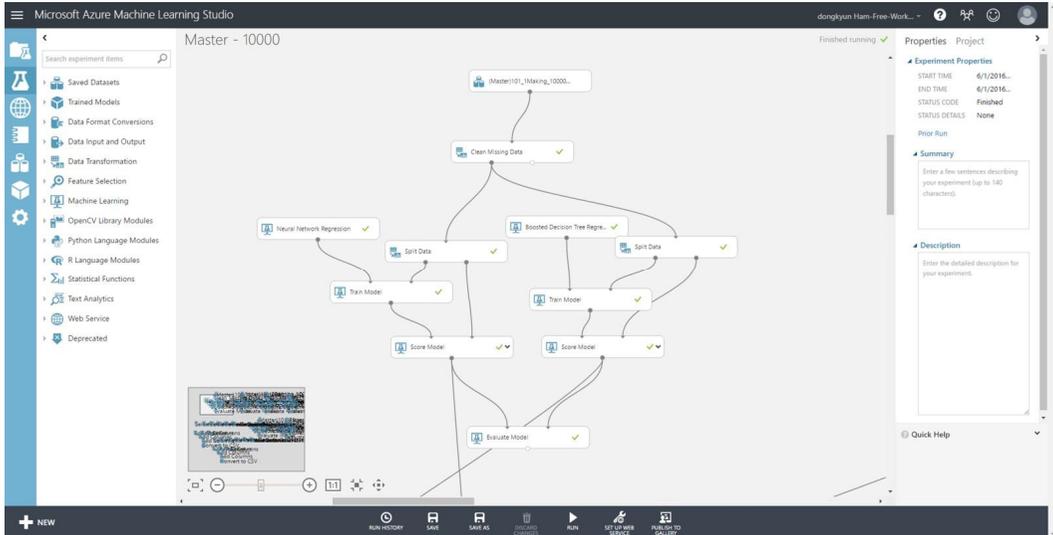


Fig. 24 Azure ML을 활용한 분석모델(Making)

인공신경망 회귀분석에서 조건을 Table 16과 같이 3가지로 구분해서 결과를 산출하였다. 분석결과를 MAE와 MAPE로 나타낸 결과는 그림 25, 26과 같다. 분석결과에서 은닉층의 노드 수와 증가함에 따라, learning rate가 좁아짐에 따라 상대적으로 오차가 줄어드는 경향을 보이고 있고 iteration의 경우 MAE는 점차 낮아지나 MAPE는 경우 따라 다른 것을 확인할 수 있다.

Table 16 Azure 인공신경망 회귀분석 조건

	iteration	node	learning rate
R	1000	5	0.01
Azure_001	1000	5	0.01
Azure_002	100	100	0.005
Azure_003	2000	100	0.005

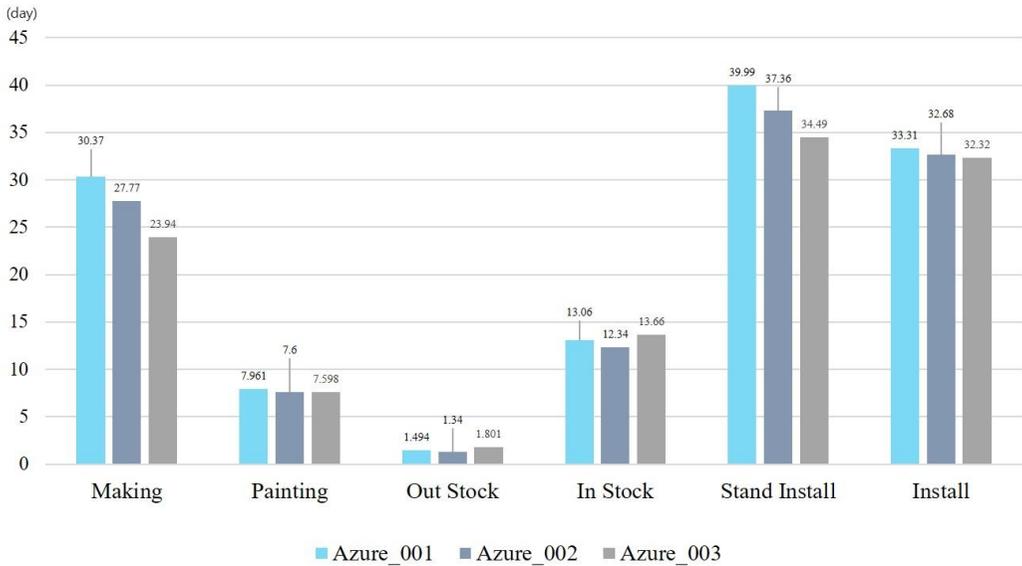


Fig. 25 Azure ML을 사용한 인공지능망 회귀분석 결과(MAE)

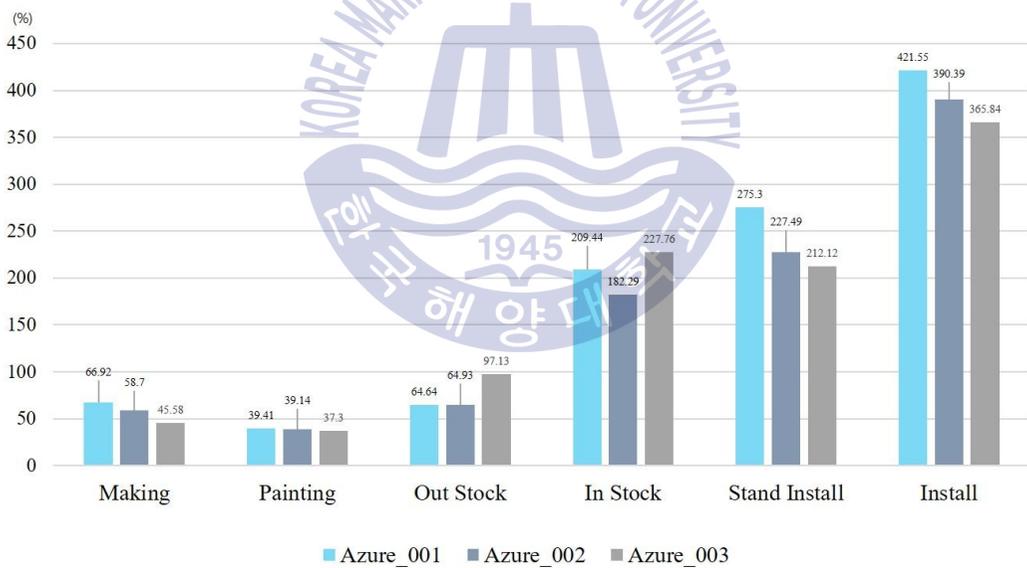


Fig. 26 Azure ML을 사용한 인공지능망 회귀분석 결과(MAPE)

#### 5.4.2 의사결정나무 회귀모델

의사결정나무 회귀분석에서 사용한 분석조건을 Table 17과 같다. 분석은 R과 동일한 조건을 설정한 것, 나무의 수는 줄인 것, 전체적으로 개수를 크게

증가시킨 경우로 나누어 분석을 수행하였고 MAE와 MAPE를 지표로 사용하여 결과를 그림 27, 28과 같이 나타내었다. 분석결과에서 트리 수의 증가에 따라 오차가 줄어드는 것을 확인할 수 있었으나 증가에 따른 오차감소 정도가 다소 낮기 때문에 처리속도를 고려한 합리적인 트리 수 설정이 필요하다. Azure\_003 케이스에서 트리의 수 및 가지 수, 가지 당 샘플데이터의 수를 모두 증가시키고 학습률은 보다 좁게 설정에 따른 오차감소 효과가 거의 없는 것을 확인할 수 있었다.

Table 17 Azure 의사결정나무 회귀분석 조건

	tree	shrinkage	leaf per tree	sample per leaf
R	1000	0.1	-	-
Azure_001	1000	0.1	20	10
Azure_002	100	0.1	20	10
Azure_003	2000	0.05	30	20

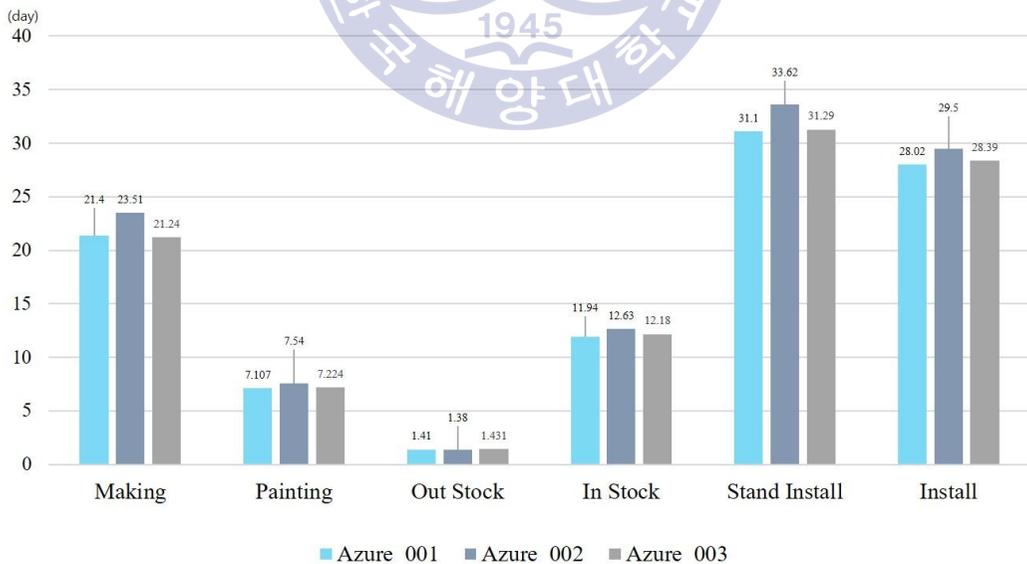


Fig. 27 Azure ML을 사용한 의사결정나무 회귀분석 결과(MAE)

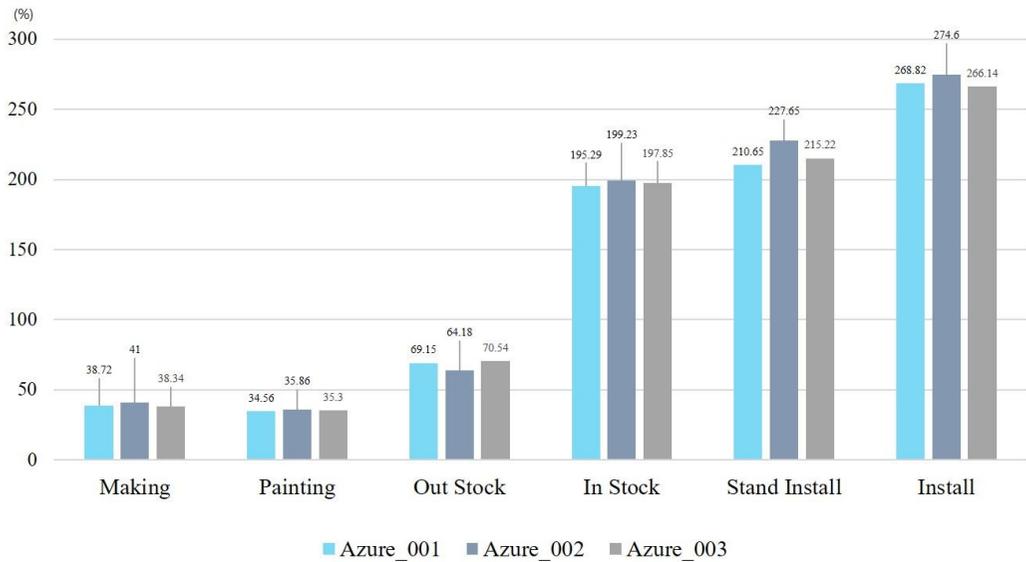


Fig. 28 Azure ML을 사용한 의사결정나무 회귀분석 결과(MAPE)

본 연구에서 분석했던 결과들을 하나의 그래프로 도식화하면 그림 29, 30와 같다. 분석 틀 간 비교에서 전반적으로 큰 차이가 없으나 선행에서 후행으로 갈수록 오차가 급격히 증가하는 현상을 보였다. 알고리즘 간 비교에서는 R과 Azure에서 수행했던 의사결정나무 회귀가 가장 낮은 오차를 나타내었다. 결과 비교를 통해 조선 배관재 공급망 데이터 분석 시 기계학습 알고리즘이 일반적 통계기법의 회귀모형보다 예측성능이 우수한 것을 알 수 있고 이는 조선업의 생산 데이터가 비선형적인 특성을 가지고 있음을 말한다. 또한 인공지능망의 경우 추정문제에서 일반적으로 좋은 결과를 나타내는 인공지능망이 예상외로 낮은 성능을 가짐으로써 배관재 데이터에 인공지능망이 local minimum하게 만드는 극값(외생변수)이 다소 포함되어 나타난 결과로 볼 수 있다.

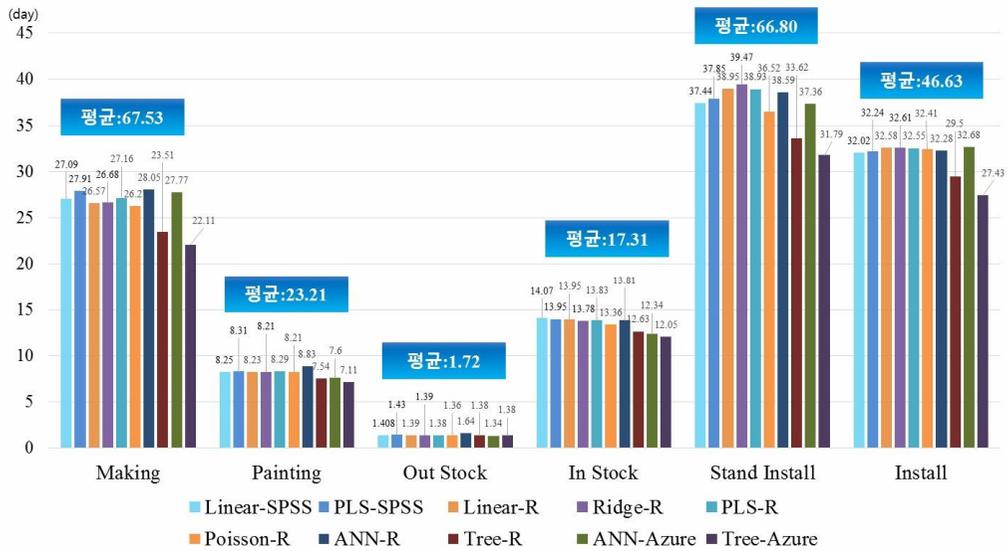


Fig. 29 회귀분석 전체 결과(MAE)

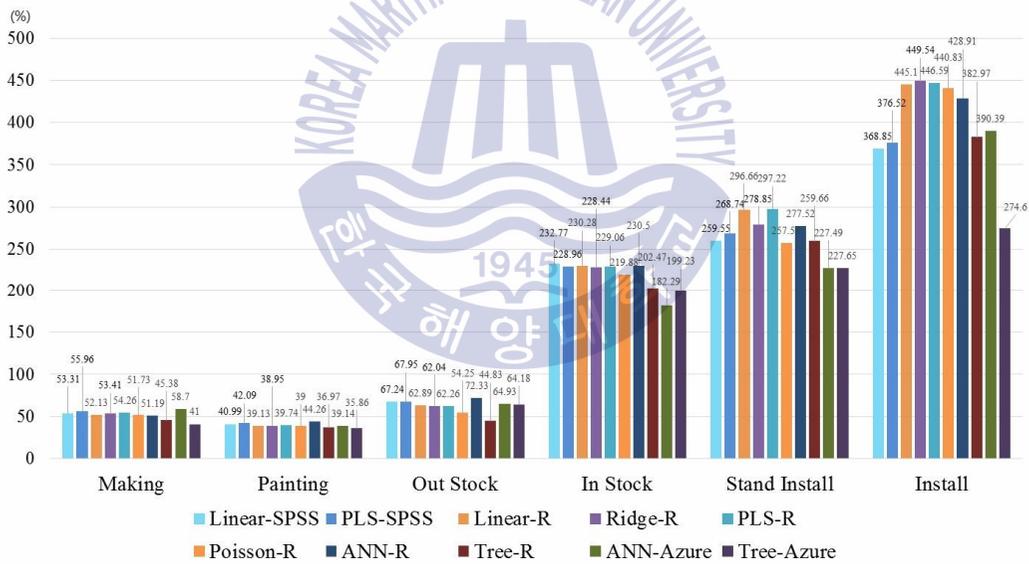


Fig. 30 회귀분석 전체 결과(MAPE)

## 제 6 장 변수 민감도 분석

### 6.1 민감도 분석 모델

앞의 분석결과에서 인공지능망 회귀나 의사결정나무 회귀의 경우 내부 알고리즘이 어떻게 이루어져 있는지 알기 어렵다. 인공지능망의 내부는 블랙박스과 같아서 입력이 어떻게 입력이 처리되어 출력되는지 또는 가중치가 어떻게 조합되었는지를 알기가 어렵다. 또한 추정 문제를 위한 의사결정나무 회귀의 경우 트리의 수가 많기 때문에 전체 나무를 한꺼번에 도식화하는 것이 어려우며 전체를 펼쳤다 하더라도 사람의 인식할 수 있는 범위를 넘어가는 경우가 많다. 이로 인해 기계학습의 불투명한 모형들의 작동원리에 대해 이해하기 위해 민감도분석을 수행하였다. 민감도 분석은 테스트 집합을 사용하여 각 입력에 대하여 출력이 얼마나 민감한지를 보여줄 수 있다(Michael & Gordon, 2012).

본 연구의 민감도 분석은 기존의 예측 모델에 사용된 변수들을 하나씩 제거한 뒤에 예측 모델을 재 산출하는 방식으로 진행하였다. 예측 모델은 가장 낮은 오차를 나타냈던 Azure의 의사결정나무 회귀를 사용하였고 6개 공정별로 각각의 모델을 만들어 분석을 수행하였다. 평가지표는 MAE(Mean Absolute Error:절대오차평균)을 사용하였는데 이는 해당 변수를 제거한 예측모델에서 오차값의 변화를 알아보는 것이 현업에서 영향정도를 파악하기에 용이하기 때문이다.

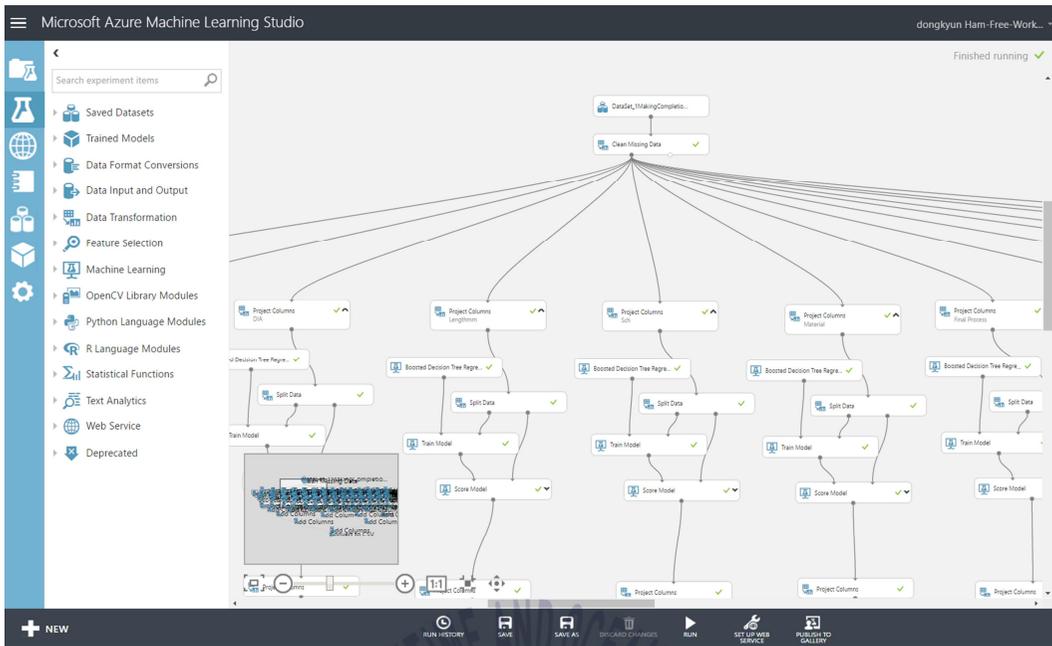


Fig. 31 민감도 분석 모델(Making)

## 6.2 민감도 분석 결과

민감도 분석 결과는 Table 18, 19와 같다. 결과에서 모든 변수가 입력되었을 때의 오차를 Origin으로 두고 우측 변수가 제거되었을 때의 오차를 각각의 공정마다 나타낸 것이다. 보다 용이한 비교를 위해 그래프로 도식화한 것이 Table 19와 같다. Table 19에서 초록색으로 표시된 그래프가 해당 변수를 제거했을 때 증가한 오차에 해당하고 노란색으로 표시된 그래프는 오차가 줄어든 정도를 의미한다. 공정별로 살펴보면 Making의 경우 대부분의 변수가 예측모델에 대하여 Positive한 영향력을 가지고 있었다. 우선순위별로 나타내면 Main Problem, Making Vendor, STAGE, Num of Parts 와 같은데, Making Vendor의 경우 제작업체를 의미하는 변수이고 Main Problem은 제작단계전까지의 문제점을 의미하는 변수로 분석결과가 일반적인 통념과도 부합한다고 볼 수 있다. Painting의 경우 STAGE, Warehouse, Making Vendor, Main Problem 순으로 Positive한 영향력을 나타냈다. 도장작업에 해당하는 Painting에서 예상외로 Painting Vendor가 큰 영향력을 가지지 않았고 선행과 후행공정에 해당하는 Making Vendor와 물류업체를 의미하는 Warehouse가

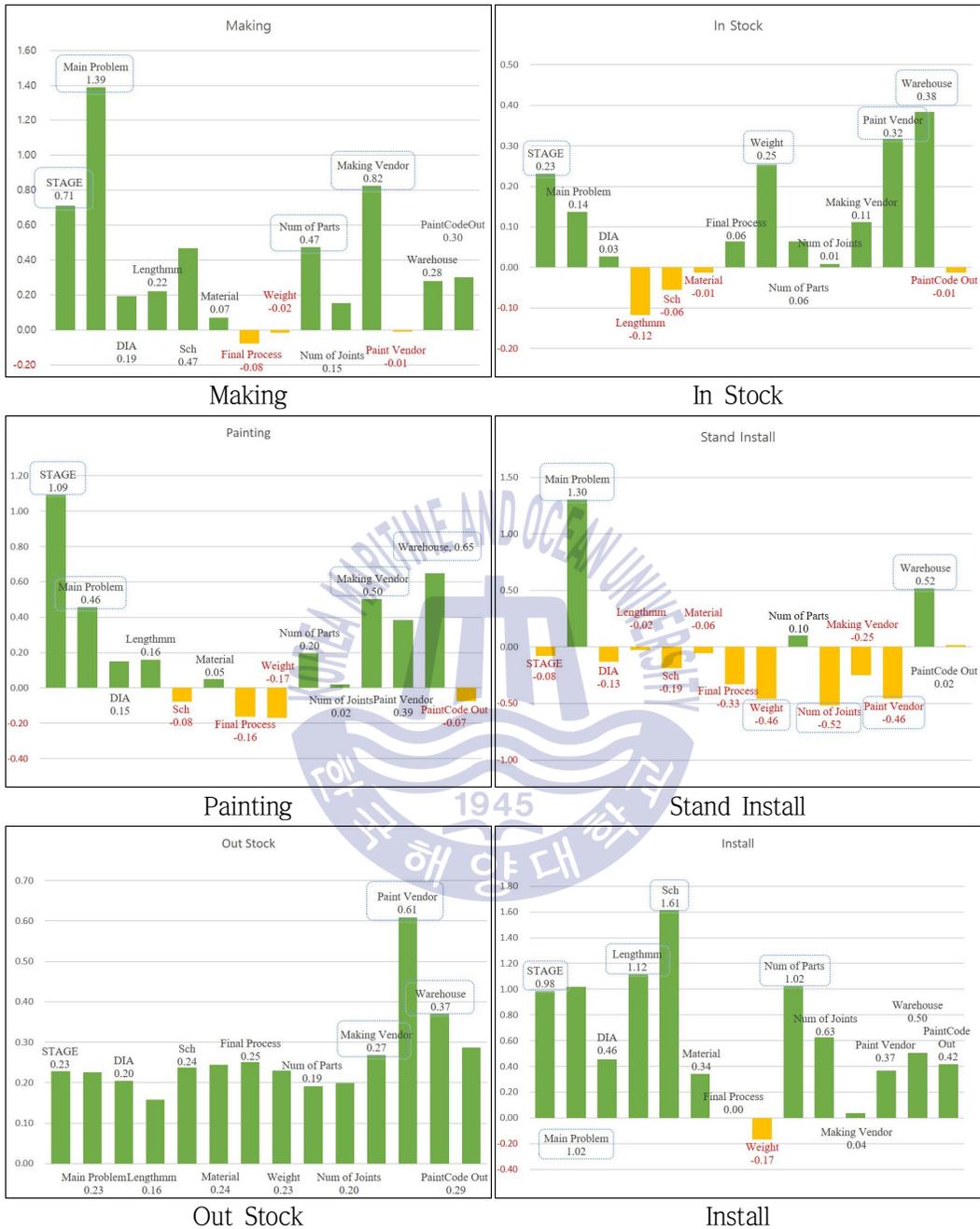
우선순위의 영향력을 가지고 있었다. Out Stock의 경우, 모든 변수가 Positive한 영향력을 가지고 있었고 Paint Vendor, Warehouse, Making Vendor 순으로 나타났다. 야드로 입고되기 전 물류업체에서 잠시 적치되는 기간을 의미하는 Out Stock의 경우 Paint Vendor에 가장 큰 영향을 받는 것으로 나타났다. 야드 내로 입고되어 작업장 근처에 적치되기까지의 기간을 의미하는 In Stock의 경우 Warehouse, Paint Vendor, Weight 순으로 나타났는데, 선행단계에 해당하는 변수인 Paint Vendor와 Warehouse에 영향을 받고 있음을 알고 있고 특히 무게를 의미하는 Weight의 영향이 큰 것으로 야드 내에서의 물류 흐름이 무게에 영향을 받고 있음을 알 수 있다. Stand Install의 경우 다른 공정들과 달리 대부분의 변수가 Negative한 영향력을 가지고 있었다. Install의 경우 Sch, Length, Num of part와 같은 배관재의 사양과 관련된 변수가 큰 영향력을 가지고 있었다. 실제 배관재 의장품을 설치하는 과정에서 길이가 길거나 부품 수가 많고 두께가 두꺼운 배관재의 경우 작업 시간이 지연될 수 있다.



Table 18 민감도 분석 결과(MAE)

	Making	Painting	Out Stock	In Stock	Stand Install	Install
Pre-weld stage	16.3	15.4	2.3	8.9	29.1	30.1
Main problem	17	14.8	2.3	8.8	30.5	30.1
Diameter	15.8	14.5	2.3	8.7	29.1	29.5
Length	15.8	14.5	2.3	8.6	29.2	30.2
Thickness	16.1	14.2	2.3	8.7	29	30.7
Material	15.7	14.4	2.4	8.7	29.2	29.4
Final Process	15.5	14.1	2.4	8.8	28.9	29.1
Weight	15.6	14.1	2.3	9	28.8	28.9
Parts	16.1	14.5	2.3	8.8	29.3	30.1
Joints	15.7	14.3	2.3	8.7	28.7	29.7
Making vendor	16.4	14.8	2.4	8.8	29	29.1
Painting vendor	15.6	14.7	2.7	9	28.8	29.5
Warehouse	15.9	15	2.5	9.1	29.7	29.6
Paint-Type	15.6	14.3	2.1	8.7	29.2	29.1
Origin (기준)	15.6	14.3	2.1	8.7	29.2	29.1

Table 19 민감도 분석 결과 그래프



## 제 7 장 결 론

최근 여러 산업에서 빅데이터를 활용하여 생산성을 향상시키거나 불합리한 공정을 개선함으로써 손실을 줄이고 비용을 절감하고자 하는 시도들이 점차 늘어나고 있는 추세이다. 그러한 관점에서 조선업의 축적되어 있는 방대한 데이터에 빅데이터 기법을 적용하는 것이 필요한데, 현재 조선업의 생산데이터의 분석 사례가 거의 없으며 적용 및 활용 방식에 대한 검증이 부족한 상태이다.

조선업의 빅데이터 적용 및 분석방법론으로 제안할 수 있는 것으로는 크게 3가지, KDD, CRIPS-DM, SEMMA가 있으며 분석전략계획 및 데이터마이닝의 균형적인 접목이 가능한 KDD방법론을 사용하였다.

빅데이터 분석을 위한 플랫폼은 상용 소프트웨어와 오픈 소프트웨어로 나뉘며 분석 목적과 환경에 맞는 툴을 사용할 수 있다. SPSS, Azure ML과 같은 상용 툴의 경우 상대적으로 쉽고 잦은 예외상황을 일일이 컨트롤할 필요가 없으나 분석의 재사용성이 떨어질 수 있고 특정 어플리케이션과 연동하여 사용하기 어려운 단점이 있다. 반면 R과 같은 오픈소스의 경우 분석을 위한 여러 전처리 작업을 코드화 해야 하고 여러 예외처리 기능을 구현해야하는 단점이 있지만 여러 알고리즘을 사용할 수 있고 넓은 확장성을 가지며, 최근 대두되는 분산처리 기능을 구현하는데 용이하다.

조선업 배관재 공급망 분석은 일반적인 통계기법의 회귀모형보다 빅데이터의 기계학습 알고리즘이 적합하며 인적요소가 포함된 전형적인 비선형성을 가지는 데이터에 해당한다. 선행에서 후행으로 갈수록 이러한 현상을 심해지며 실제 분석에서도 큰 오차율을 나타냈다. 추후 보다 나은 성능을 가지는 예측모델을 구현하기 위해서는 추가 변수의 획득과 선-후행간의 시계열데이터를 활용한 구조방정식 정의가 필요하다.

예측모델에 대한 변수들의 영향력을 분석한 민감도분석 결과, Stand Install외에 대부분의 변수들이 Positive한 영향력을 가지고 있고 공정에 따라 영향력의 우선순위를 분석함으로써 현업의 공정개선 활동 시 이를 활용할 수

있다.



## 참고문헌

KOREAMARITIME RESEARCH INSTITUTE(KMRD), 2015. The top 7 maritime news according to MONTHLY MARITIME KOREA(해양한국이 뽑은 2015년 해사산업 별 7대 뉴스). *MONTHLY MARITIME KOREA*, 2015(12), pp.28-29.

Mckinsey&Company, 2014. *How big data can improve manufacturing*[Online] (Updated 3 July 2014) Available at : [http://www.mckinsey.com/insights/operations/how\\_big\\_data\\_can\\_improve\\_manufacturing](http://www.mckinsey.com/insights/operations/how_big_data_can_improve_manufacturing)

Jang, Y.J., 2012. Big-data technique utilization to field of manufacturing(제조 분야에서의 빅데이터 기술 활용). *Journal of The Korean Institute of Communication Sciences*, 29(11), pp.30-35.

Hankook Media Network, 2015. *DSME, which remote to reinforce shipbuilding's competitiveness by Big-data*(대우조선해양, '빅데이터' 로 조선 경쟁력 강화 추진[Online] (Updated 14 October 2015) Available at : [http://daily.hankooki.com/lpage/economy/201508/dh2015081712031413\\_8080.htm](http://daily.hankooki.com/lpage/economy/201508/dh2015081712031413_8080.htm)

Hankook Media Network, 2015. *Hyundai Heavy Industries, which pioneers total service market by Smart-ship*(현대중공업, 스마트십으로 토털서비스 시장 개척[Online] (Updated 14 October 2015) Available at : [http://daily.hankooki.com/lpage/economy/201508/dh2015081712031413\\_8080.htm](http://daily.hankooki.com/lpage/economy/201508/dh2015081712031413_8080.htm)

Okumoto Yasuhisa, 2012. Shipbuilding Technology & Production System. Translated by Kim, Y.S. GS-intervision: Seoul

Ham, D.k. Back, M.G. Park, J.G. & Woo, J.H., 2016. A Study of Piping Leadtime Forecast in Offshore Plant's Outfittings Procurement Management. *Journal of the Society of Naval Architects of Korea*, 53(1), pp.29-36.

Umair, S. & Haseeb, Q., 2014. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1), pp.217-222.

Oded Maimon & Lior Rokach, 2010. *DATA MINING AND KNOWLEDGE DISCOVERY HANDBOOK*. 2nd Ed. Springer: International.

Pete, Chapman. et al., 2000. CRISP-DM 1.0. 1<sup>st</sup> Ed. SPSS: USA.

Lee, J.W., 2007. *Variable selection and its application in PLS regression(PLS 회귀분석에서의 변수선택과 응용)*. Master's Thesis. Korea National University.

Kim, D.J., 2012. Models of local finance decisions using PLS regression analysis : cities and counties in Gyeongnam. Master's Thesis. Gyeongsang National University.

Yun, K.U. Lee, Y.H. & Han, J.H., 2001. The Prediction of effluent NOx composition from Fired Heater using PLS(PLS를 이용한 Fired Heater의 NOx 배출 농도 예측). The Korean Institute of Chemical Engineers, 7(2), pp.3191-3195.

Hastie, T. Tibshirani, R. Friedman, J. H., 2009. The Elements of Statistical Learning (2nd ed.). Springer: New York.

Brett, Lantz., 2015. Machine Learning with R. Translated by Jeon, C.U. acorn: Seoul

Michael J.A. & Gordon S., 2012. Data Mining Techniques. Translated by Kim, J.W. & Kim, S.T. hankyungsa: Seoul



## 부록 A

### A.1 수치형 설명변수 히스토그램

히스토그램이란, 수치형 변수의 분포를 알 수 있도록 하는 그래프로 가로축이 크기이고 세로축이 빈도수이다. 회귀분석의 경우 수치형 설명변수가 극값을 가질 경우 회귀계수가 왜곡될 수 있는데, 이 경우 로그스케일을 수행하면 이러한 문제를 완화시킬 수 있다.

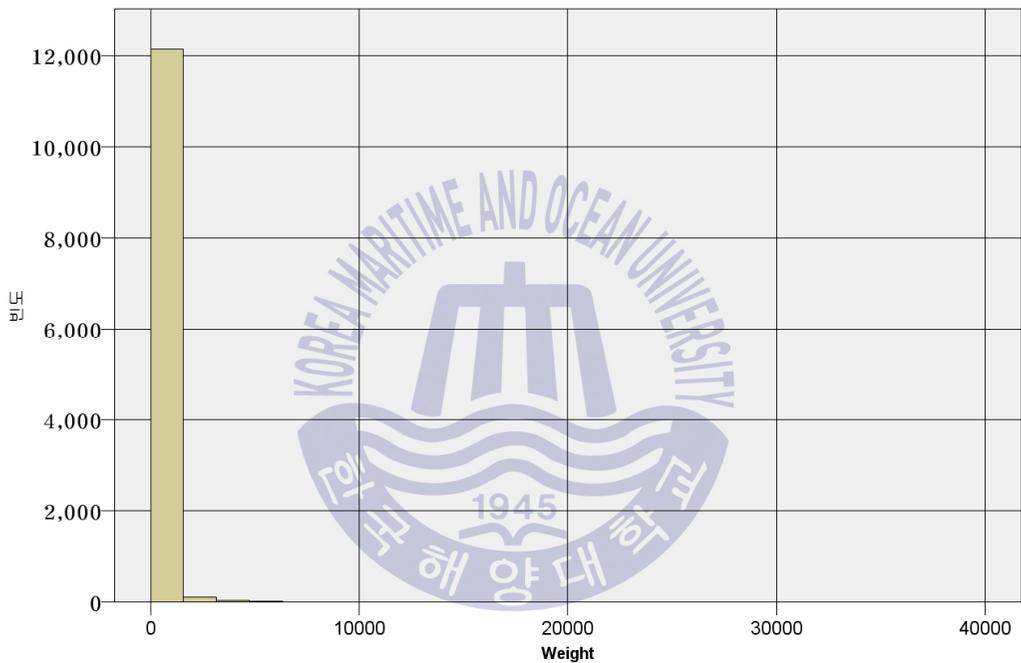


Fig. 32 Weight 히스토그램

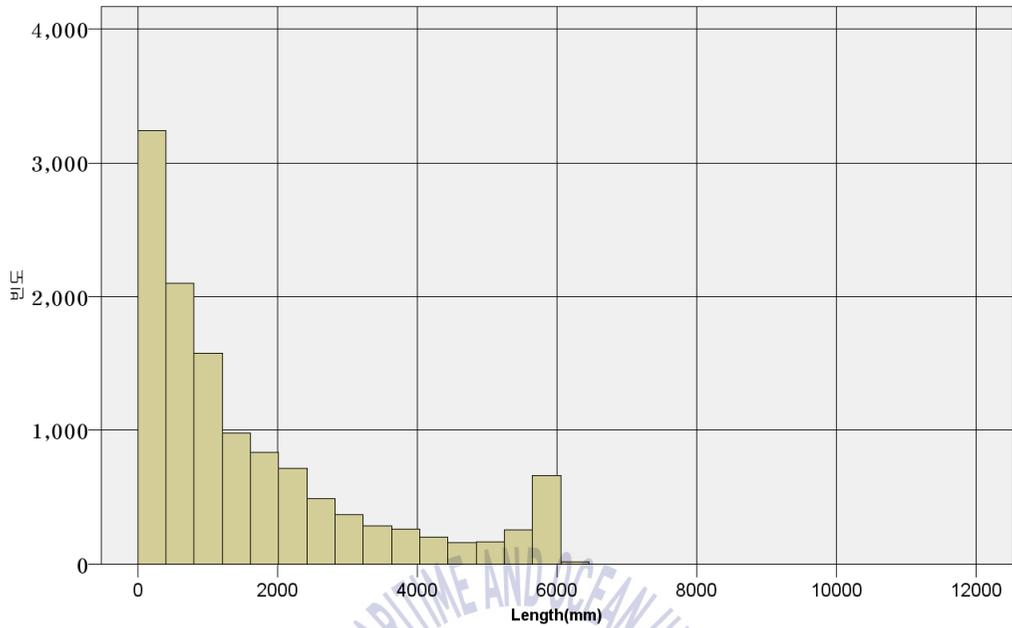


Fig. 33 Length(길이) 히스토그램

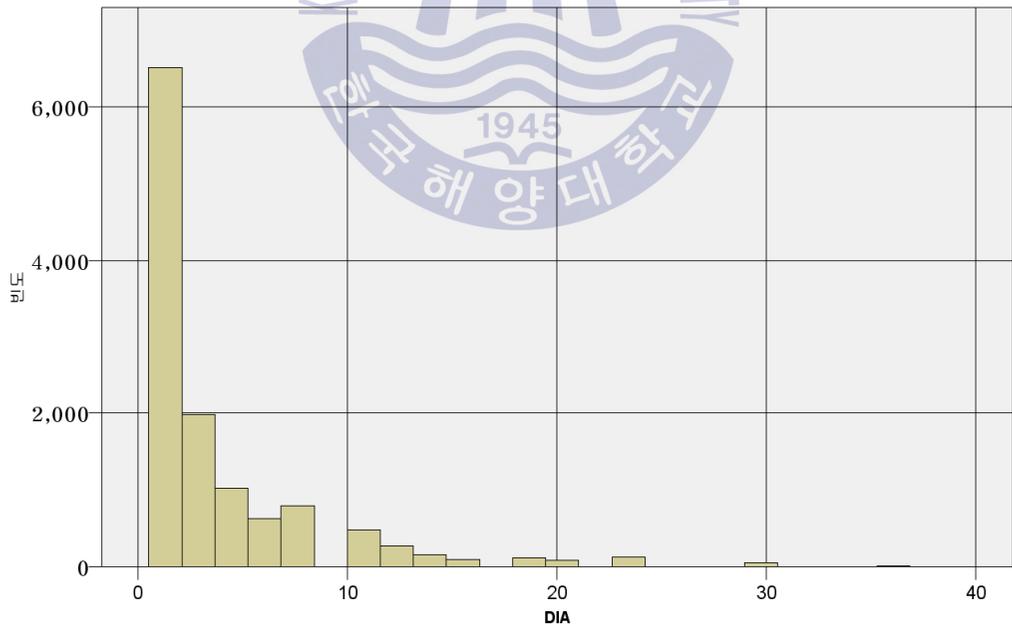


Fig. 34 DIA(직경) 히스토그램

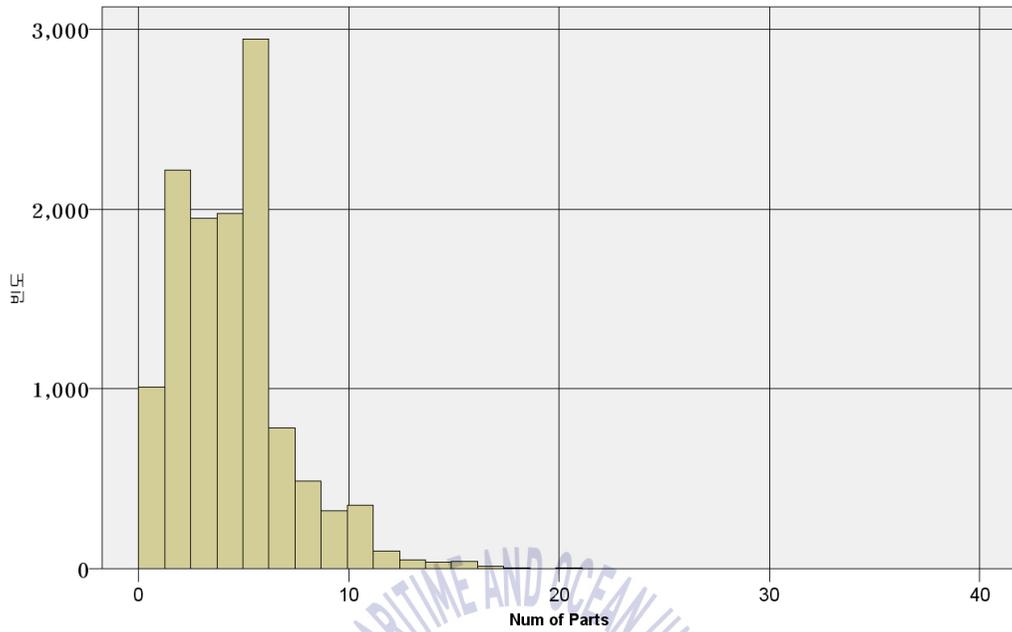


Fig. 35 Num of Parts(부품 수) 히스토그램

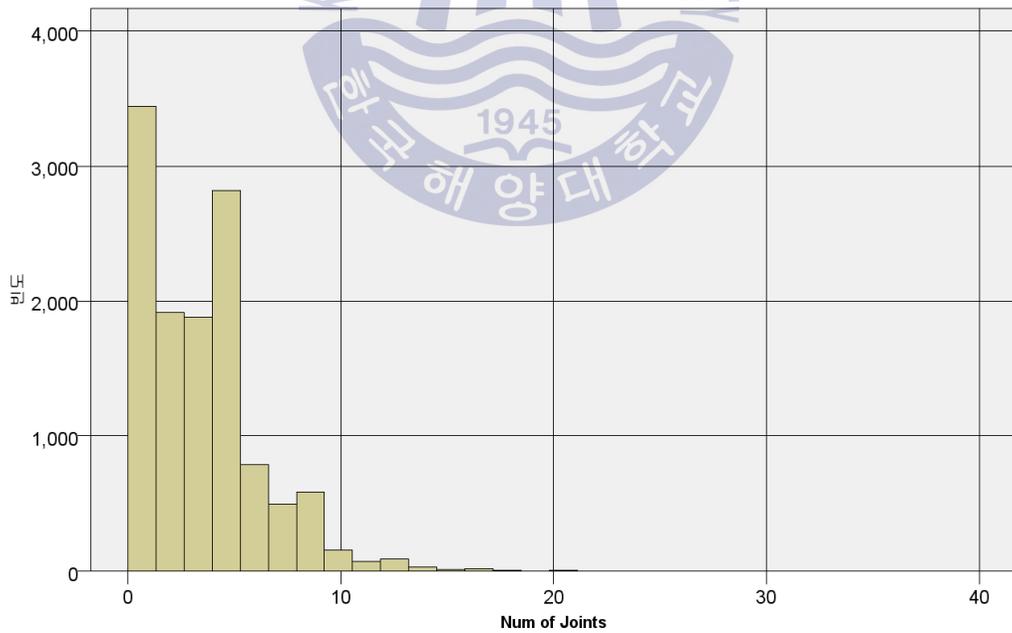


Fig. 36 Num of Joints(연결 수) 히스토그램

## A.2 범주형 설명변수 빈도분석

범주형 설명변수의 빈도분석은 데이터의 추후 평가용 데이터를 분리하거나 예측값 출력 시 변수의 이탈여부 및 극값 발생 가능성을 사전에 확인하고 데이터 활용 시 참고하기 위해 분석 전 수행하게 된다.

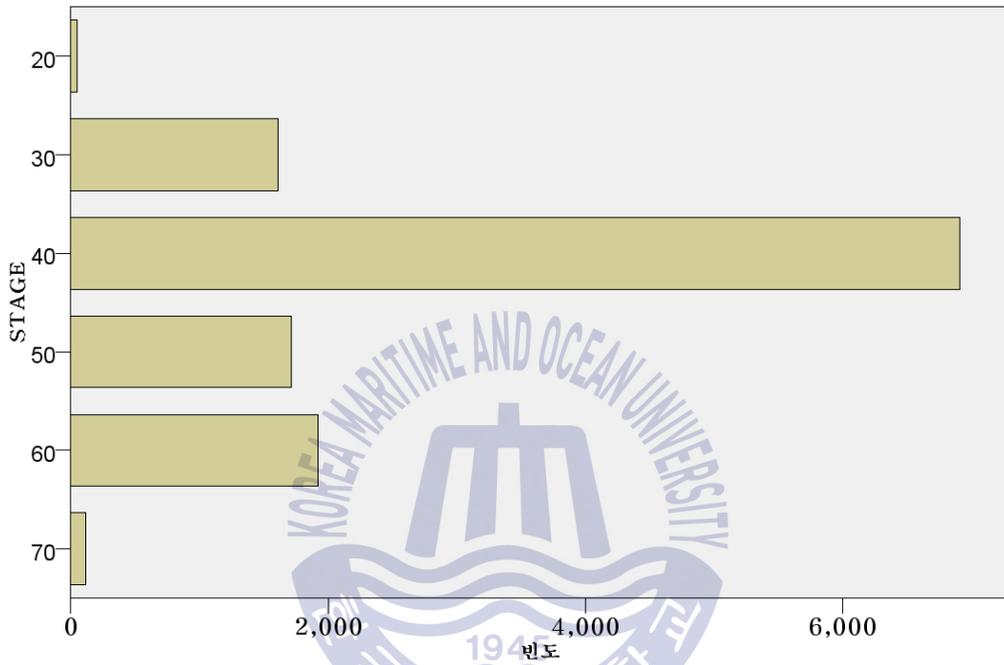


Fig. 37 Stage(작업수행 공정시점) 빈도분석

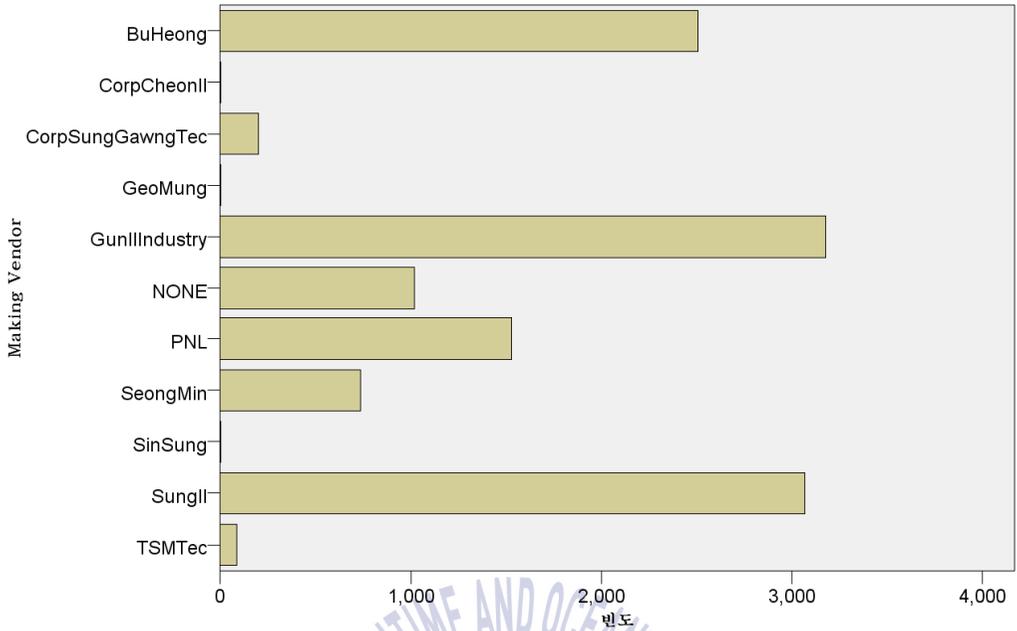


Fig. 38 Making Vendor(제작업체) 빈도분석

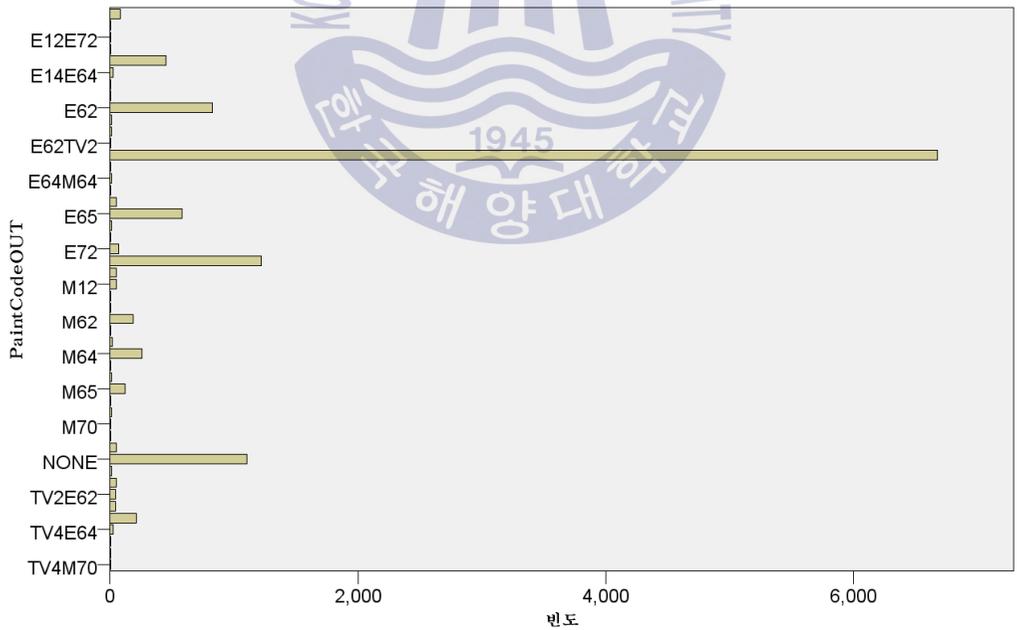


Fig. 39 Paint Code(도장 방법) 빈도분석

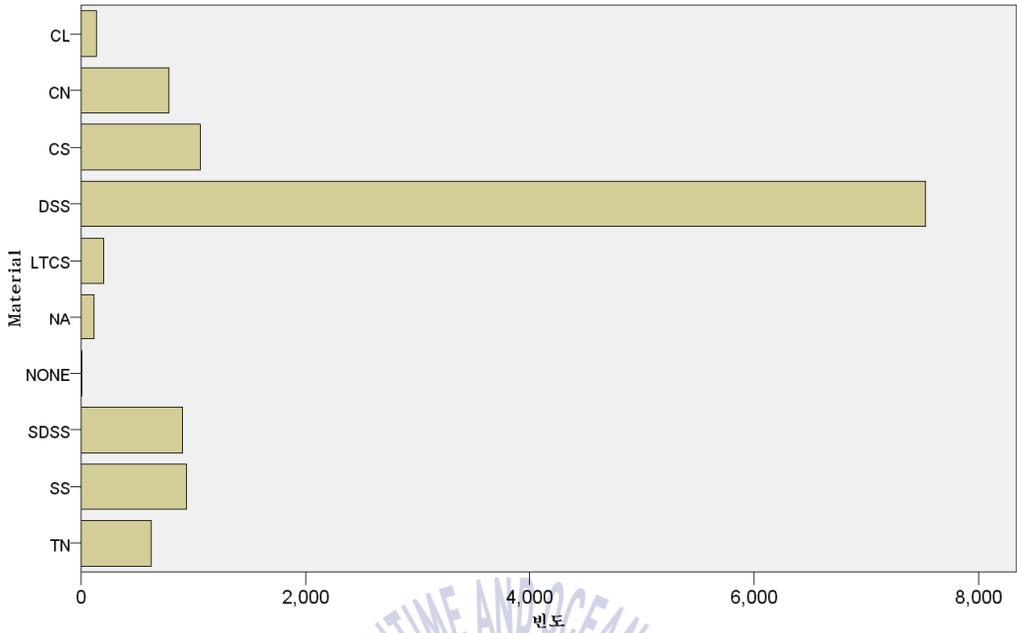


Fig. 40 Material(배관 재질) 빈도분석

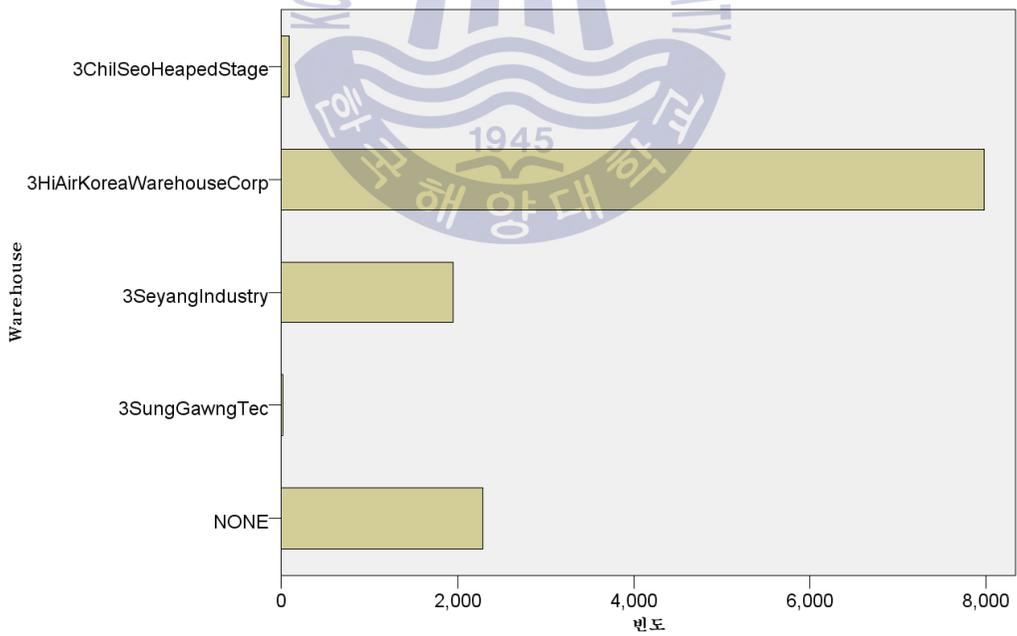


Fig. 41 Warehouse(물류업체) 빈도분석

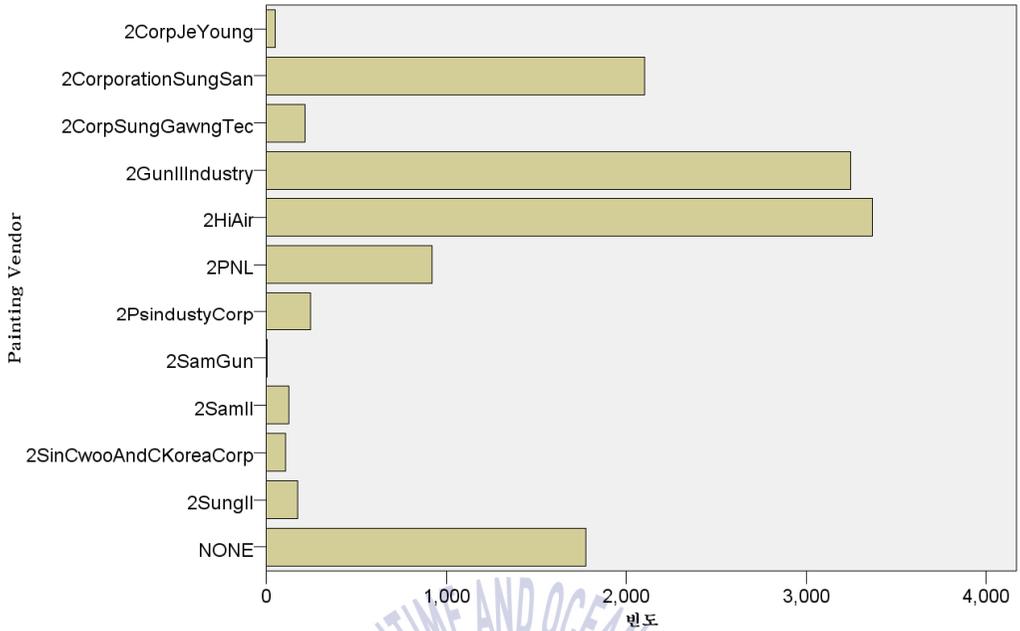


Fig. 42 Painting Vendor(도장업체) 빈도분석

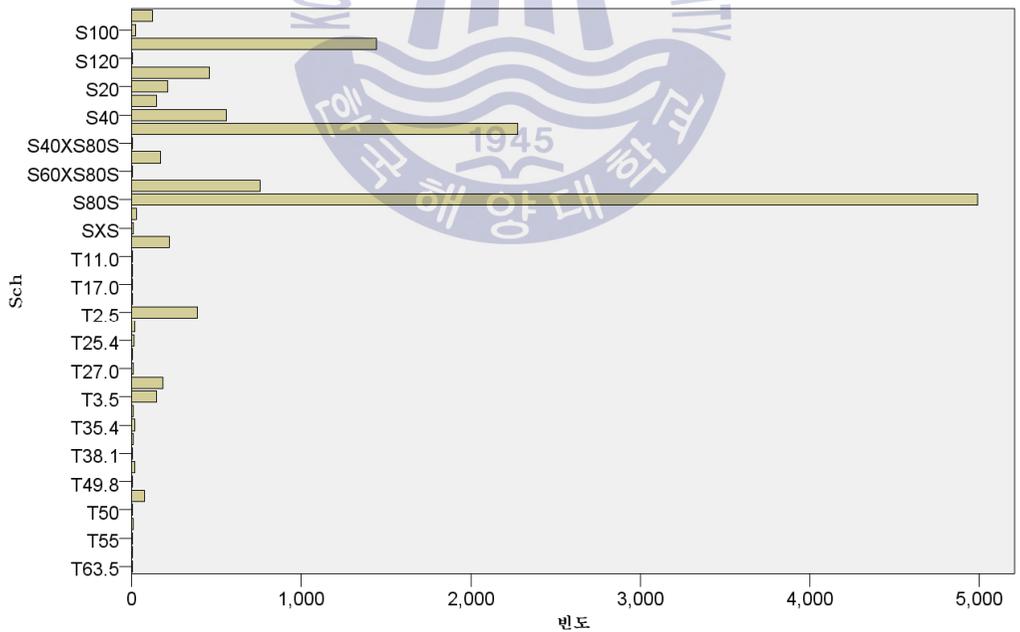


Fig. 43 Sch(배관 두께) 빈도분석

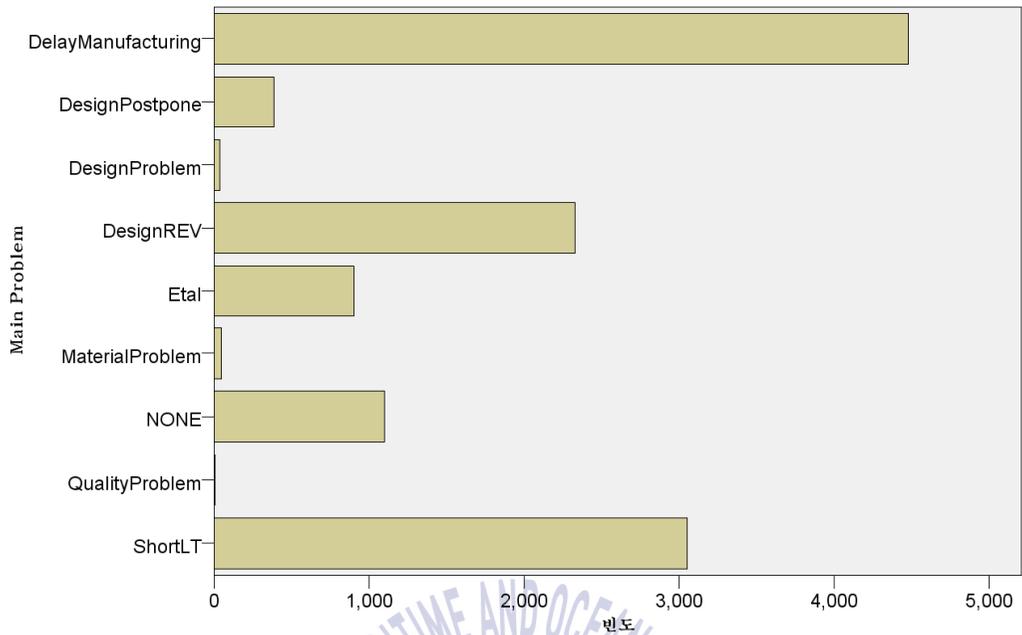
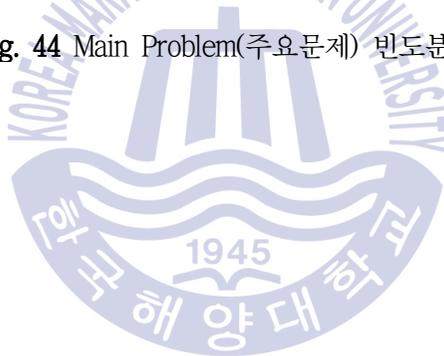


Fig. 44 Main Problem(주요문제) 빈도분석



### A.3 수치형 목표변수 히스토그램

GLM 모형은 각 목표변수의 특정분포를 가정한다. 가령, 선형회귀의 경우 정규분포, 포아송 회귀의 경우 포아송분포를 가정한다. 본 연구에서는 기존 데이터가 어떤 알고리즘에 적합한 지는 상호 비교하기 위해 목표변수의 변환과정은 수행하지 않는다.

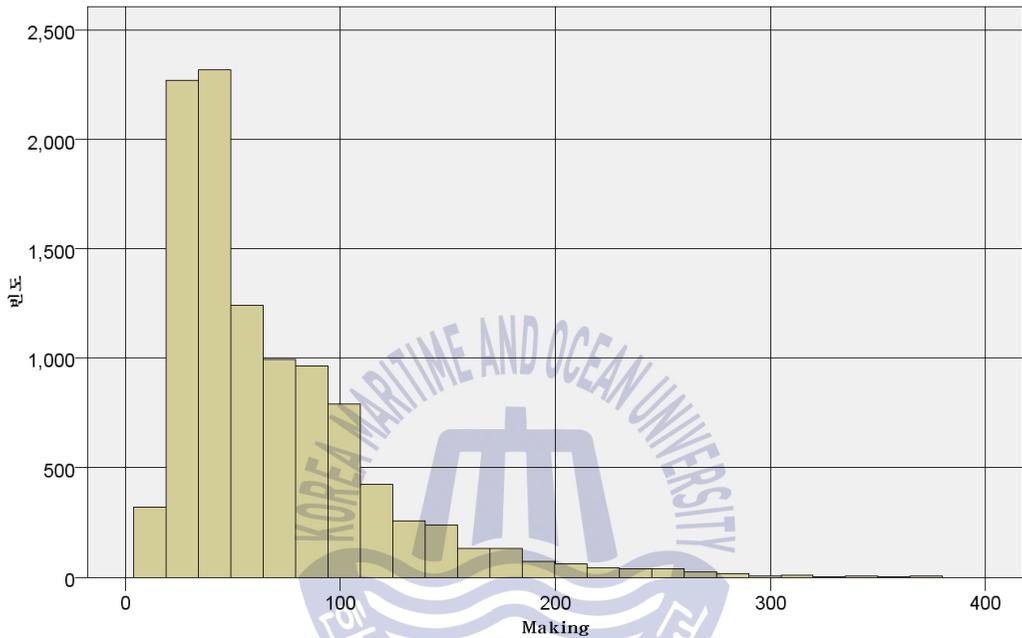


Fig. 45 Making(제작 관련 공정) 빈도분석

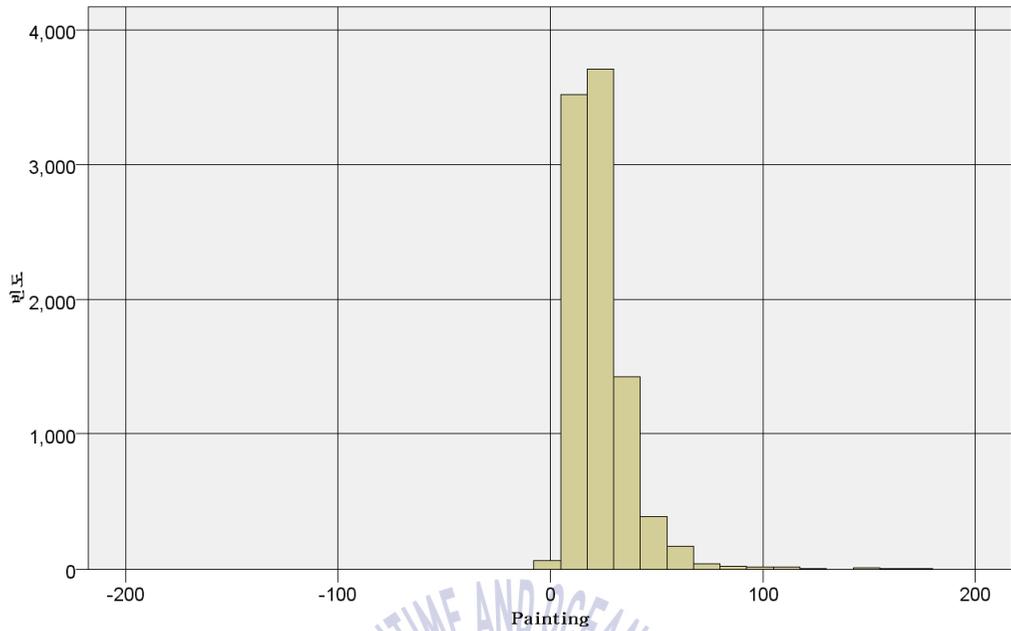


Fig. 46 Painting(도장 관련 공정) 빈도분석

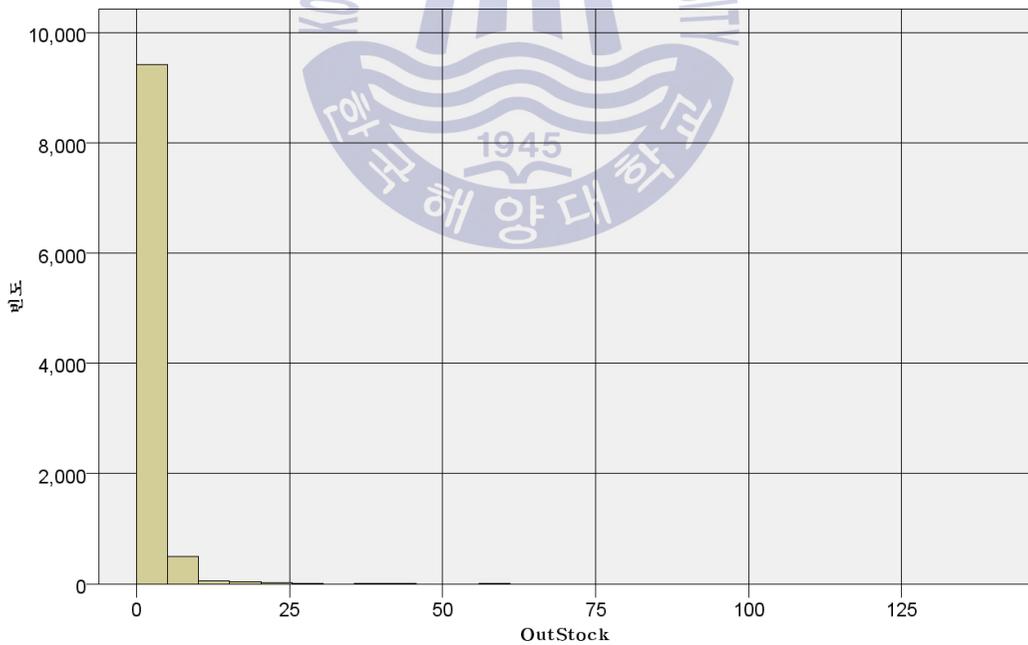


Fig. 47 Out Stock(사외재고 관련 공정) 빈도분석

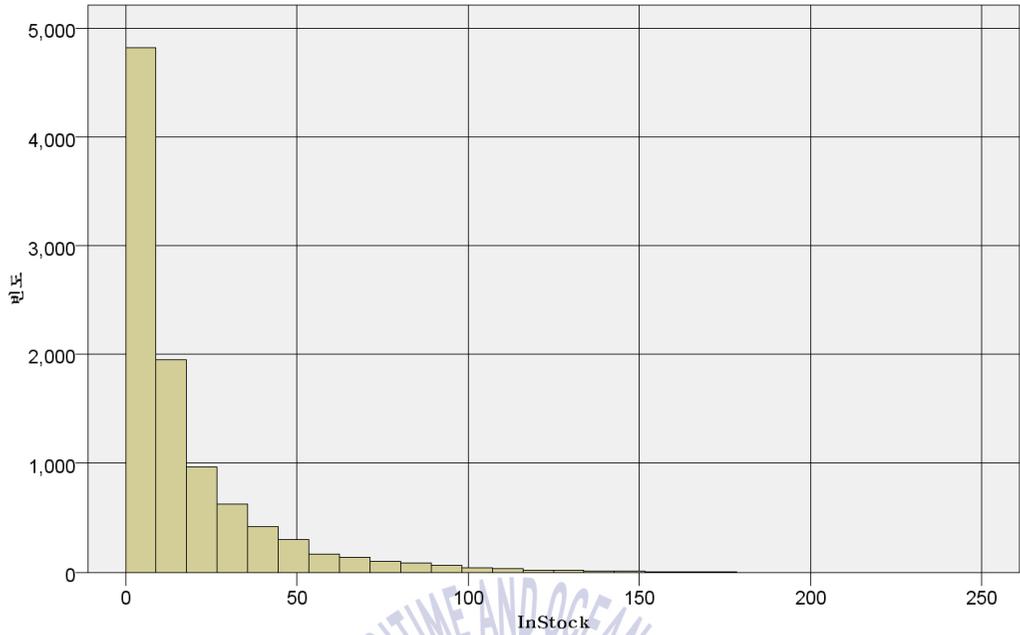


Fig. 48 In Stock(사내재고 관련 공정) 빈도분석

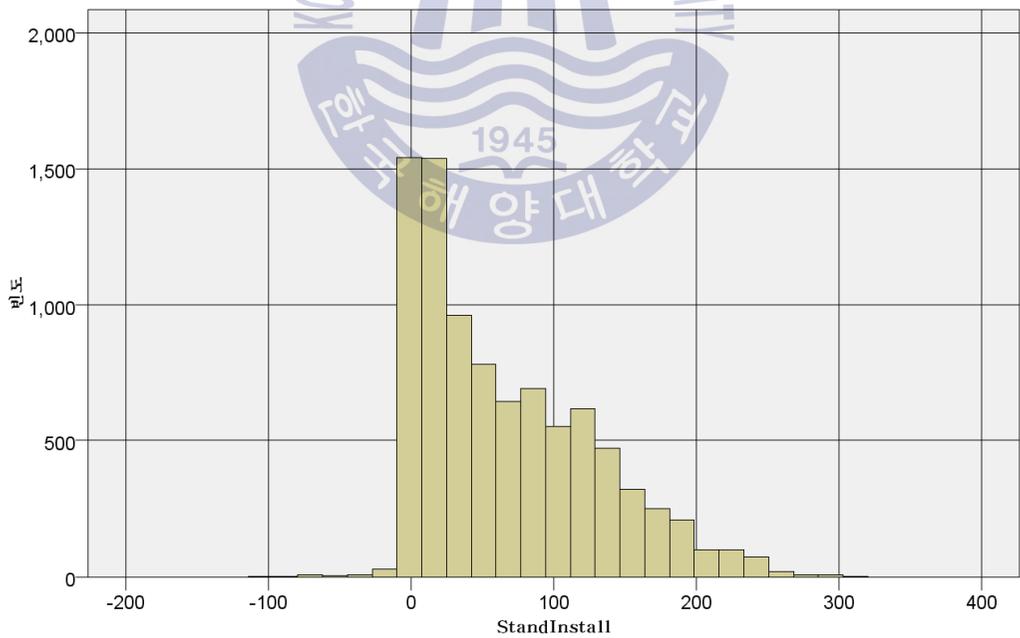


Fig. 49 Stand Install(설치대기 관련 공정) 빈도분석

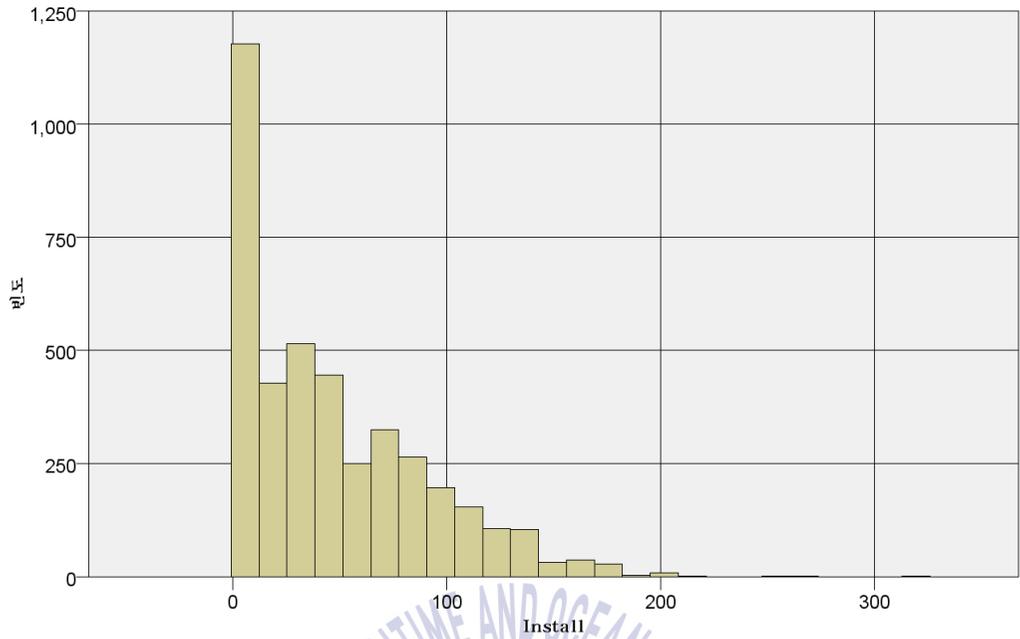


Fig. 50 Install(설치관련 공정) 빈도분석



## 부록 B

### B.1 SPSS를 사용한 예측모델의 회귀계수

Table 20 SPSS 다중선형회귀 계수(Making)

모형	비표준화 계수		표준화 계수 베타	t	유의 확률
	B	표준오차			
1 (상수)	38.091	18.671		2.040	.041
DIA	-1.510	.285	-.134	-5.301	.000
Num of Joints	2.717	.220	.152	12.335	.000
LogLength	-4.972	.640	-.118	-7.768	.000
LogWeight	7.337	.774	.232	9.486	.000
STAGE_20	25.616	9.186	.037	2.789	.005
STAGE_30	28.155	5.304	.206	5.308	.000
STAGE_40	16.218	5.130	.169	3.161	.002
STAGE_50	10.199	5.268	.072	1.936	.053
STAGE_60	9.783	5.273	.069	1.855	.064
Main_Problem_ShortLT	-27.553	8.894	-.262	-3.098	.002
Main_Problem_Etal	-4.102	9.045	-.024	-.454	.650
Main_Problem_DesignREV	8.154	8.919	.068	.914	.361
Main_Problem_DesignPostpone	-12.598	9.493	-.037	-1.327	.185
Main_Problem_DesignProblem	17.132	14.026	.015	1.221	.222
Main_Problem_MaterialProblem	34.991	12.355	.039	2.832	.005
Main_Problem_DelayManufacturing	-.396	8.826	-.004	-.045	.964
Sch_S100	-15.443	10.335	-.015	-1.494	.135
Sch_S10S	2.907	1.870	.021	1.555	.120
Sch_S160	-8.799	3.631	-.027	-2.423	.015
Sch_S20	.094	6.159	.000	.015	.988
Sch_S30	-1.120	5.703	-.002	-.196	.844
Sch_S40	9.303	5.445	.043	1.709	.088
Sch_S40S	.255	1.393	.002	.183	.855
Sch_S60	-10.846	5.097	-.025	-2.128	.033
Sch_S80	2.719	4.742	.015	.573	.566
Sch_SSTD	8.971	12.944	.007	.693	.488
Sch_SXS	-20.604	28.758	-.007	-.716	.474
Sch_SXXS	10.607	4.253	.031	2.494	.013
Sch_T18	-14.352	19.261	-.007	-.745	.456
Sch_T2.5	13.466	15.380	.053	.876	.381
Sch_T25.4	51.107	18.817	.032	2.716	.007
Sch_T26	-53.912	21.748	-.024	-2.479	.013
Sch_T3	28.929	15.544	.080	1.861	.063
Sch_T3.5	38.753	15.736	.087	2.463	.014
Sch_T31.8	89.385	26.884	.033	3.325	.001
Sch_T35.4	36.719	14.792	.030	2.482	.013
Sch_T42	8.110	15.486	.005	.524	.600
Sch_T49.8	-47.807	38.072	-.012	-1.256	.209
Sch_T5.5	31.246	16.153	.055	1.934	.053
Sch_T53.5	-80.973	37.725	-.021	-2.146	.032
Sch_T55	3.263	26.399	.001	.124	.902
Material_CL	-58.055	12.480	-.132	-4.652	.000

Material_LTCS	26.347	6.158	.064	4.278	.000
Material_NA	-4.084	27.915	-.009	-.146	.884
Material_SDSS	-35.181	3.009	-.216	-11.693	.000
Material_SS	5.850	2.352	.033	2.487	.013
Material_TN	-2.843	26.460	-.014	-.107	.914
Making_Vendor_CorpSungGawngTec	-10.699	16.315	-.032	-.656	.512
Making_Vendor_CorpCheonll	17.458	41.593	.004	.420	.675
Making_Vendor_GeoMung	18.427	40.217	.005	.458	.647
Making_Vendor_BuHeong	2.653	4.702	.023	.564	.573
Making_Vendor_Sungll	35.754	4.519	.338	7.911	.000
Making_Vendor_SinSung	-28.238	40.204	-.007	-.702	.482
Making_Vendor_PNL	59.756	5.377	.431	11.114	.000
Making_Vendor_TSMTec	33.109	8.688	.063	3.811	.000
Painting_Vendor_2CorpSungGawngTec	12.646	21.903	.038	.577	.564
Painting_Vendor_2CorpJeYoung	2.559	16.264	.004	.157	.875
Painting_Vendor_2GunllIndustry	-7.816	14.947	-.075	-.523	.601
Painting_Vendor_2Samll	-1.360	16.157	-.003	-.084	.933
Painting_Vendor_2SinCwooAndCKoreaCorp	22.487	16.240	.049	1.385	.166
Painting_Vendor_2CorporationSungSan	-16.573	15.654	-.143	-1.059	.290
Painting_Vendor_2PNL	-15.679	15.822	-.087	-.991	.322
Painting_Vendor_2PsindustryCorp	5.925	15.887	.019	.373	.709
Painting_Vendor_2HiAir	6.736	15.609	.065	.432	.666
Warehouse_3SungGawngTec	-20.445	12.970	-.017	-1.576	.115
Warehouse_3SevangIndustry	19.277	2.793	.156	6.902	.000
Warehouse_3ChilSeoHeapedStage	10.774	5.521	.022	1.951	.051
Warehouse_3HiAirKoreaWarehouseCorp	-1.876	2.669	-.016	-.703	.482
PaintCodeOUT_E64	14.744	3.296	.157	4.473	.000
PaintCodeOUT_E12	6.988	8.377	.012	.834	.404
PaintCodeOUT_E12E62	-14.807	37.590	-.004	-.394	.694
PaintCodeOUT_E12E72	54.352	41.639	.014	1.305	.192
PaintCodeOUT_E14	19.578	4.562	.063	4.292	.000
PaintCodeOUT_E14E64	62.138	17.388	.036	3.574	.000
PaintCodeOUT_E62	17.516	6.117	.097	2.863	.004
PaintCodeOUT_E62M62	44.877	16.242	.028	2.763	.006
PaintCodeOUT_E62MJ2	20.616	18.648	.015	1.106	.269
PaintCodeOUT_E62TV2	9.739	29.364	.004	.332	.740
PaintCodeOUT_E64E14	70.111	37.424	.018	1.873	.061
PaintCodeOUT_E64M64	.721	15.498	.000	.047	.963
PaintCodeOUT_E64PFP	12.462	26.435	.005	.471	.637
PaintCodeOUT_E64TV4	-7.386	7.275	-.011	-1.015	.310
PaintCodeOUT_E65	6.434	26.251	.031	.245	.806
PaintCodeOUT_E65M65	7.889	30.813	.006	.256	.798
PaintCodeOUT_E65TV0	-29.042	46.520	-.007	-.624	.532
PaintCodeOUT_E72	12.998	9.311	.021	1.396	.163
PaintCodeOUT_E74	20.437	3.723	.129	5.489	.000
PaintCodeOUT_E74TV4	77.586	8.884	.091	8.733	.000
PaintCodeOUT_M12	19.375	9.022	.028	2.147	.032
PaintCodeOUT_M12E12	-47.814	37.577	-.012	-1.272	.203
PaintCodeOUT_M62	10.030	6.761	.030	1.483	.138
PaintCodeOUT_M62TV2	-18.353	12.252	-.016	-1.498	.134
PaintCodeOUT_M64	21.371	4.366	.069	4.895	.000

PaintCodeOUT_M64E64	-16.163	26.403	-.006	-.612	.540
PaintCodeOUT_M64TV4	29.496	12.746	.023	2.314	.021
PaintCodeOUT_M65	-7.923	26.835	-.018	-.295	.768
PaintCodeOUT_M65E65	-24.257	37.340	-.009	-.650	.516
PaintCodeOUT_M65TV0	-68.015	30.274	-.055	-2.247	.025
PaintCodeOUT_M70	-7.484	26.915	-.003	-.278	.781
PaintCodeOUT_NAPFP	6.502	9.333	.008	.697	.486
PaintCodeOUT_TV0	-46.816	30.473	-.034	-1.536	.125
PaintCodeOUT_TV2	17.529	10.275	.025	1.706	.088
PaintCodeOUT_TV2E62	14.566	15.367	.018	.948	.343
PaintCodeOUT_TV2MJ2	16.286	14.879	.023	1.095	.274
PaintCodeOUT_TV4	16.626	4.802	.049	3.463	.001
PaintCodeOUT_TV4E64	1.089	11.431	.001	.095	.924
PaintCodeOUT_TV4E74	-38.879	27.423	-.014	-1.418	.156
PaintCodeOUT_TV4M64	7.254	26.406	.003	.275	.784
PaintCodeOUT_TV4M70	60.543	37.201	.016	1.627	.104

Table 21 SPSS 다중선형회귀 계수(Painting)

모형	비표준화 계수		표준화 계수 베타	t	유의확률
	B	표준오차			
1 (상수)	12.545	4.233		2.964	.003
DIA	-.363	.068	-.108	-5.339	.000
Num of Parts	.143	.074	.023	1.925	.054
LogLength	-.659	.163	-.051	-4.035	.000
STAGE_20	2.026	3.220	.009	.629	.529
STAGE_30	5.242	1.711	.123	3.064	.002
STAGE_40	5.422	1.638	.185	3.310	.001
STAGE_50	3.931	1.691	.091	2.325	.020
STAGE_60	4.220	1.694	.098	2.491	.013
Main_Problem_ShortLT	-2.734	3.112	-.084	-.878	.380
Main_Problem_Etal	-1.246	3.158	-.026	-.395	.693
Main_Problem_DesignREV	-3.904	3.120	-.105	-1.251	.211
Main_Problem_DesignPostpone	-.255	3.362	-.002	-.076	.940
Main_Problem_DesignProblem	-5.454	5.082	-.015	-1.073	.283
Main_Problem_MaterialProblem	3.206	4.346	.012	.738	.461
Main_Problem_DelayManufacturing	-1.865	3.086	-.065	-.604	.546
Sch_S100	13.931	4.591	.035	3.035	.002
Sch_S105	.085	.631	.002	.134	.893
Sch_S160	.891	1.183	.010	.753	.451
Sch_S20	2.458	2.096	.024	1.173	.241
Sch_S30	1.874	1.970	.014	.951	.341
Sch_S40	2.770	1.885	.045	1.470	.142
Sch_S40S	.468	.484	.013	.967	.334
Sch_S60	-.677	1.725	-.006	-.393	.695
Sch_S80	2.880	1.630	.054	1.766	.077
Sch_SSTD	.275	4.021	.001	.068	.945
Sch_SXS	-.792	8.172	-.001	-.097	.923
Sch_SXXS	-2.266	1.372	-.024	-1.651	.099
Sch_T18	1.710	8.949	.002	.191	.848
Sch_T25.4	-9.669	7.024	-.017	-1.376	.169
Sch_T31.8	-1.554	6.533	-.003	-.238	.812
Sch_T35.4	-8.432	4.956	-.025	-1.701	.089
Sch_T42	.355	5.241	.001	.068	.946
Sch_T55	-1.553	7.357	-.002	-.211	.833
Material_CL	-10.948	4.206	-.086	-2.603	.009
Material_LTCS	1.439	2.146	.012	.671	.503

Material_NA	5.381	9.464	.044	.569	.570
Material_SDSS	4.188	1.035	.084	4.048	.000
Material_SS	2.570	.804	.049	3.198	.001
Material_TN	2.277	9.001	.040	.253	.800
Making					
Vendor_CorpSungGawngTec	18.316	6.630	.182	2.762	.006
Making_Vendor_BuHeong	4.794	1.633	.144	2.936	.003
Making_Vendor_SungIl	2.733	1.565	.086	1.746	.081
Making_Vendor_PNL	4.694	1.852	.113	2.534	.011
Making_Vendor_TSMTec	-1.174	2.876	-.008	-.408	.683
Painting					
Vendor_2CorpSungGawngTec	-20.401	6.539	-.206	-3.120	.002
Painting					
Vendor_2CorpJeYoung	27.561	2.370	.131	11.627	.000
Painting_Vendor_2Samll	.155	2.086	.001	.074	.941
Painting					
Vendor_2SinCwooAndCKoreaCorp	3.417	2.114	.027	1.616	.106
Painting					
Vendor_2CorporationSungSan	3.743	1.587	.108	2.358	.018
Painting_Vendor_2PNL	-8.397	1.778	-.156	-4.722	.000
Painting					
Vendor_2PsindustryCorp	-.261	1.829	-.003	-.143	.886
Painting_Vendor_2HiAir	-7.094	1.542	-.232	-4.602	.000
Warehouse_3SungGawngTec	-12.038	4.835	-.032	-2.490	.013
Warehouse_3SevangIndustry	3.857	1.879	.104	2.053	.040
Warehouse_3ChilSeoHeapedStage	.272	2.528	.002	.108	.914
Warehouse_3HiAirKoreaWarehouseCorp	3.418	1.875	.095	1.823	.068
PaintCodeOUT_E64	7.723	1.186	.268	6.511	.000
PaintCodeOUT_E12	4.077	2.982	.023	1.367	.172
PaintCodeOUT_E12E62	23.905	12.790	.021	1.869	.062
PaintCodeOUT_E14	9.686	1.636	.104	5.920	.000
PaintCodeOUT_E14E64	4.896	7.402	.008	.661	.508
PaintCodeOUT_E14E74	3.523	12.641	.003	.279	.781
PaintCodeOUT_E14TV4	5.680	12.661	.005	.449	.654
PaintCodeOUT_E62	14.086	2.125	.267	6.630	.000
PaintCodeOUT_E62M62	25.258	6.610	.045	3.821	.000
PaintCodeOUT_E62MJ2	6.584	8.538	.010	.771	.441
PaintCodeOUT_E62TV2	16.845	13.359	.015	1.261	.207
PaintCodeOUT_E64E14	-.203	12.722	.000	-.016	.987
PaintCodeOUT_E64M64	4.823	5.296	.010	.911	.362
PaintCodeOUT_E64PFP	11.989	7.388	.018	1.623	.105
PaintCodeOUT_E64TV4	.727	2.631	.003	.276	.782
PaintCodeOUT_E65	2.551	8.924	.043	.286	.775
PaintCodeOUT_E65M65	-1.443	10.205	-.004	-.141	.888
PaintCodeOUT_E65TV0	5.879	15.763	.005	.373	.709
PaintCodeOUT_E72	21.936	3.308	.120	6.632	.000
PaintCodeOUT_E74	12.785	1.326	.274	9.643	.000
PaintCodeOUT_E74TV4	9.038	2.870	.040	3.149	.002
PaintCodeOUT_M12	5.166	3.111	.026	1.661	.097
PaintCodeOUT_M12E12	21.726	12.784	.019	1.699	.089
PaintCodeOUT_M62	16.650	2.357	.164	7.063	.000
PaintCodeOUT_M62E62	27.993	12.798	.025	2.187	.029
PaintCodeOUT_M62TV2	17.820	4.176	.055	4.267	.000
PaintCodeOUT_M64	9.032	1.527	.103	5.916	.000
PaintCodeOUT_M64E64	4.393	7.404	.007	.593	.553
PaintCodeOUT_M64TV4	4.246	5.758	.008	.737	.461
PaintCodeOUT_M65	4.175	9.108	.035	.458	.647
PaintCodeOUT_M65E65	5.270	11.682	.008	.451	.652
PaintCodeOUT_M65TV0	12.137	10.536	.028	1.152	.249

PaintCodeOUT_M70	7.552	12.876	.007	.587	.558
PaintCodeOUT_NAPFP	4.898	3.158	.019	1.551	.121
PaintCodeOUT_TV0	3.543	10.147	.010	.349	.727
PaintCodeOUT_TV2	10.436	3.369	.055	3.098	.002
PaintCodeOUT_TV2E62	12.384	5.083	.059	2.436	.015
PaintCodeOUT_TV2MJ2	5.680	5.034	.028	1.128	.259
PaintCodeOUT_TV4	4.371	1.692	.043	2.583	.010
PaintCodeOUT_TV4E64	2.000	4.270	.006	.468	.640
PaintCodeOUT_TV4E74	25.239	9.287	.032	2.718	.007
PaintCodeOUT_TV4M70	25.545	12.649	.023	2.020	.043

Table 22 SPSS 다중선형회귀 계수(Out Stock)

모형	비표준화 계수		표준화 계수 베타	t	유의확률
	B	표준오차			
1 (상수)	23.747	1.896		12.527	.000
LogLength	-.074	.054	-.020	-1.362	.173
LogWeight	-.084	.049	-.031	-1.716	.086
STAGE_20	.276	.913	.005	.303	.762
STAGE_30	.005	.537	.000	.010	.992
STAGE_40	.244	.520	.030	.470	.639
STAGE_50	.304	.533	.025	.570	.568
STAGE_60	.586	.532	.049	1.101	.271
Main_Problem_ShortLT	-.142	.858	-.016	-.166	.868
Main_Problem_Etal	-.037	.871	-.003	-.042	.966
Main_Problem_DesignREV	.031	.859	.003	.037	.971
Main_Problem_DesignPostpone	-.379	.926	-.012	-.409	.683
Main_Problem_DesignProblem	-1.237	1.308	-.014	-.945	.345
Main_Problem_MaterialProblem	-.103	1.229	-.001	-.084	.933
Main_Problem_DelayManufacturing	-.324	.851	-.040	-.380	.704
Sch_S100	-.576	1.114	-.006	-.517	.605
Sch_S105	.170	.171	.015	.998	.319
Sch_S160	.774	.348	.029	2.224	.026
Sch_S20	.160	.511	.006	.314	.754
Sch_S30	.159	.473	.004	.335	.737
Sch_S40	-.363	.482	-.020	-.753	.452
Sch_S40S	.035	.136	.003	.254	.800
Sch_S60	-.126	.445	-.004	-.283	.777
Sch_S80	.073	.413	.005	.176	.860
Sch_SSTD	-1.283	1.223	-.012	-1.049	.294
Sch_SXS	-2.037	1.853	-.013	-1.099	.272
Sch_SXXS	.917	.425	.031	2.157	.031
Sch_T18	-.714	2.278	-.004	-.314	.754
Sch_T2.5	-19.868	1.455	-.959	-13.651	.000
Sch_T25.4	.949	1.963	.007	.483	.629
Sch_T26	.831	2.089	.004	.398	.691
Sch_T3	-19.915	1.474	-.672	-13.515	.000
Sch_T3.5	-18.879	1.505	-.462	-12.540	.000
Sch_T31.8	2.307	2.086	.012	1.106	.269
Sch_T35.4	-1.912	1.399	-.019	-1.366	.172
Sch_T42	-1.291	1.305	-.011	-.989	.323
Sch_T5.5	-19.108	1.534	-.405	-12.454	.000
Sch_T55	26.807	2.082	.143	12.874	.000
Material_CL	-1.539	1.340	-.040	-1.149	.251
Material_LTCS	-.180	.610	-.005	-.295	.768
Material_NA	-.092	3.687	-.002	-.025	.980

Material_SDSS	-.007	.296	.000	-.023	.982
Material_SS	1.498	.228	.099	6.565	.000
Material_TN	-.170	3.586	-.010	-.048	.962
Making					
Vendor_CorpSungGawngTec	.310	1.721	.011	.180	.857
Making_Vendor_BuHeong	.745	.469	.078	1.587	.112
Making_Vendor_Sungll	.698	.454	.077	1.537	.124
Making_Vendor_PNL	1.182	.534	.099	2.214	.027
Making_Vendor_TSMTec	.571	.838	.013	.682	.496
Painting					
Vendor_2CorpSungGawngTec	-21.362	2.180	-.768	-9.799	.000
Painting					
Vendor_2CorpJeYoung	-19.308	1.538	-.330	-12.551	.000
Painting					
Vendor_2GunllIndustry	-18.058	1.393	-2.038	-12.960	.000
Painting_Vendor_2Samll	-19.720	1.511	-.535	-13.049	.000
Painting					
Vendor_2SinCwooAndCKoreaCorp	-20.760	1.522	-.512	-13.639	.000
Painting					
Vendor_2CorporationSungSan	-20.272	1.459	-2.043	-13.898	.000
Painting_Vendor_2PNL	-21.095	1.476	-1.353	-14.297	.000
Painting					
Vendor_2PsindustryCorp	-19.715	1.480	-.766	-13.321	.000
Painting_Vendor_2HiAir	-19.894	1.454	-2.273	-13.686	.000
Warehouse_3SungGawngTec	-2.827	1.474	-.026	-1.917	.055
Warehouse_3SeyangIndustry	-.321	.762	-.031	-.421	.673
Warehouse_3ChilSeoHeapedStage	-4.326	.887	-.103	-4.877	.000
Warehouse_3HiAirKoreaWarehouseCorp	-2.726	.765	-.272	-3.564	.000
PaintCodeOUT_E64	-.366	.326	-.046	-1.121	.262
PaintCodeOUT_E12	1.983	.799	.039	2.481	.013
PaintCodeOUT_E12E62	-.691	3.615	-.002	-.191	.848
PaintCodeOUT_E12E72	.501	4.075	.002	.123	.902
PaintCodeOUT_E12M12	6.968	3.611	.021	1.930	.054
PaintCodeOUT_E14	-.370	.464	-.013	-.797	.425
PaintCodeOUT_E14E64	-2.708	2.796	-.012	-.968	.333
PaintCodeOUT_E14E74	-2.743	3.582	-.008	-.766	.444
PaintCodeOUT_E14TV4	1.789	3.586	.005	.499	.618
PaintCodeOUT_E62	.679	.549	.044	1.237	.216
PaintCodeOUT_E62M62	1.466	1.543	.011	.950	.342
PaintCodeOUT_E62MJ2	1.496	2.263	.009	.661	.509
PaintCodeOUT_E62TV2	-1.549	2.879	-.007	-.538	.590
PaintCodeOUT_E64E14	-2.157	3.609	-.007	-.598	.550
PaintCodeOUT_E64M64	-1.633	1.395	-.013	-1.171	.242
PaintCodeOUT_E64PFP	5.502	2.093	.029	2.629	.009
PaintCodeOUT_E64TV4	-.981	.769	-.015	-1.275	.202
PaintCodeOUT_E65	-.658	3.570	-.039	-.184	.854
PaintCodeOUT_E65M65	-.101	4.093	-.001	-.025	.980
PaintCodeOUT_E65TV0	-12.285	4.538	-.053	-2.707	.007
PaintCodeOUT_E72	1.081	.832	.022	1.298	.194
PaintCodeOUT_E74	.324	.366	.024	.885	.376
PaintCodeOUT_E74TV4	-.202	.830	-.003	-.243	.808
PaintCodeOUT_M12	.914	.827	.016	1.105	.269
PaintCodeOUT_M12E12	.921	3.611	.003	.255	.799
PaintCodeOUT_M62	.825	.606	.029	1.362	.173
PaintCodeOUT_M62E62	.313	3.616	.001	.086	.931
PaintCodeOUT_M62TV2	-1.220	1.194	-.012	-1.022	.307
PaintCodeOUT_M64	-.483	.437	-.018	-1.105	.269
PaintCodeOUT_M64E64	1.189	2.567	.005	.463	.643

PaintCodeOUT_M64TV4	6.452	1.299	.056	4.968	.000
PaintCodeOUT_M65	-.396	3.615	-.011	-.110	.913
PaintCodeOUT_M65E65	-1.362	4.395	-.006	-.310	.757
PaintCodeOUT_M65TV0	-.503	3.874	-.005	-.130	.897
PaintCodeOUT_NA	-.016	3.605	.000	-.004	.996
PaintCodeOUT_NAPFP	2.571	.864	.041	2.976	.003
PaintCodeOUT_TV0	1.990	4.002	.014	.497	.619
PaintCodeOUT_TV2	.142	.985	.002	.144	.885
PaintCodeOUT_TV2E62	.078	1.546	.001	.050	.960
PaintCodeOUT_TV2MJ2	1.575	1.551	.025	1.015	.310
PaintCodeOUT_TV4	-.604	.484	-.020	-1.248	.212
PaintCodeOUT_TV4E64	-.967	1.213	-.009	-.798	.425
PaintCodeOUT_TV4E74	-.818	3.651	-.003	-.224	.823
PaintCodeOUT_TV4M64	2.420	2.544	.011	.951	.342
PaintCodeOUT_TV4M70	-.307	3.584	-.001	-.086	.932

Table 23 SPSS 다중선형회귀 계수(In Stock)

모형	비표준화 계수		표준화 계수	t	유의확률	
	B	표준오차	베타			
1	LogLength	-1.097	.322	-.265	-3.406	.001
	LogWeight	2.164	.293	.254	7.391	.000
	STAGE_20	-17.777	5.346	-.041	-3.325	.001
	STAGE_30	-22.242	2.968	-.274	-7.494	.000
	STAGE_40	-11.770	2.856	-.314	-4.120	.000
	STAGE_50	-13.182	2.942	-.156	-4.481	.000
	STAGE_60	-3.367	2.940	-.041	-1.145	.252
	Main_Problem_ShortLT	16.065	5.314	.284	3.023	.003
	Main_Problem_Etal	9.736	5.395	.099	1.805	.071
	Main_Problem_DesignREV	18.794	5.324	.268	3.530	.000
	Main_Problem_DesignPostpone	15.591	5.790	.061	2.693	.007
	Main_Problem_DesignProblem	34.073	8.096	.050	4.209	.000
	Main_Problem_MaterialProblem	21.283	6.961	.042	3.058	.002
	Main_Problem_DelayManufacturing	6.872	5.274	.157	1.303	.193
	Sch_S100	18.093	6.573	.025	2.753	.006
	Sch_S105	.000	1.015	.000	.000	1.000
	Sch_S160	-2.003	2.129	-.010	-.941	.347
	Sch_S20	-6.909	3.023	-.032	-2.285	.022
	Sch_S30	-6.004	2.877	-.023	-2.087	.037
	Sch_S40	-2.977	2.855	-.024	-1.043	.297
	Sch_S40S	2.125	.808	.030	2.630	.009
	Sch_S60	-14.121	2.706	-.055	-5.219	.000
	Sch_S80	-3.561	2.411	-.033	-1.477	.140
	Sch_SSTD	10.510	6.807	.014	1.544	.123
	Sch_SXS	-16.635	12.387	-.012	-1.343	.179
	Sch_SXXS	9.496	2.481	.044	3.827	.000
	Sch_T18	8.527	24.236	.004	.352	.725
	Sch_T25.4	-6.926	11.657	-.006	-.594	.552
	Sch_T26	2.226	20.997	.001	.106	.916
	Sch_T3	-5.282	2.484	-.023	-2.127	.033
	Sch_T3.5	-4.907	2.905	-.017	-1.689	.091
	Sch_T31.8	.262	14.908	.000	.018	.986
	Sch_T35.4	2.257	9.046	.003	.249	.803
	Sch_T42	7.485	9.511	.007	.787	.431
	Sch_T5.5	-16.261	3.291	-.049	-4.941	.000
	Sch_T55	14.781	14.893	.009	.993	.321
	Material_CL	16.167	7.368	.061	2.194	.028

Material_LTCS	-3.098	3.464	-.012	-.894	.371
Material_NA	.614	21.669	.002	.028	.977
Material_SDSS	-2.012	1.745	-.020	-1.153	.249
Material_SS	-5.663	1.373	-.051	-4.126	.000
Material_TN	-3.503	21.018	-.030	-.167	.868
Making_Vendor_CorpSungGawngTec	-16.589	14.730	-.078	-1.126	.260
Making_Vendor_GunllIndustry	-1.743	11.457	-.031	-.152	.879
Making_Vendor_BuHeong	-9.941	11.729	-.161	-.848	.397
Making_Vendor_Sungll	-11.059	11.733	-.197	-.943	.346
Making_Vendor_SeongMin	3.819	7.890	.035	.484	.628
Making_Vendor_PNL	3.191	11.839	.039	.270	.788
Making_Vendor_TSMTec	-4.838	12.500	-.015	-.387	.699
Painting_Vendor_2CorpSungGawngTec	8.343	12.339	.040	.676	.499
Painting_Vendor_2CorpJeYoung	-5.282	9.266	-.012	-.570	.569
Painting_Vendor_2GunllIndustry	-3.416	8.327	-.062	-.410	.682
Painting_Vendor_2Samll	22.859	9.074	.088	2.519	.012
Painting_Vendor_2SinCwooAndCKoreaCorp	9.605	9.091	.036	1.057	.291
Painting_Vendor_2CorporationSungSan	-3.879	8.785	-.060	-.442	.659
Painting_Vendor_2PNL	-3.405	8.894	-.031	-.383	.702
Painting_Vendor_2PsindustryCorp	7.695	8.899	.043	.865	.387
Painting_Vendor_2HiAir	-3.023	8.755	-.056	-.345	.730
Warehouse_3SungGawngTec	-7.811	8.147	-.011	-.959	.338
Warehouse_3SeyangIndustry	6.786	4.555	.096	1.490	.136
Warehouse_3ChilSeoHeapedStage	-7.655	5.354	-.024	-1.430	.153
Warehouse_3HiAirKoreaWarehouseCorp	24.983	4.564	.771	5.474	.000
PaintCodeOUT_E64	5.553	1.958	.137	2.835	.005
PaintCodeOUT_E12	5.309	4.882	.014	1.088	.277
PaintCodeOUT_E12E62	-3.632	21.194	-.002	-.171	.864
PaintCodeOUT_E12M12	9.898	21.167	.004	.468	.640
PaintCodeOUT_E14	6.191	2.901	.028	2.134	.033
PaintCodeOUT_E14E64	-2.257	12.329	-.002	-.183	.855
PaintCodeOUT_E14TV4	4.804	21.026	.002	.228	.819
PaintCodeOUT_E62	12.336	3.244	.113	3.803	.000
PaintCodeOUT_E62M62	52.526	9.052	.055	5.803	.000
PaintCodeOUT_E62MJ2	-9.076	10.676	-.011	-.850	.395
PaintCodeOUT_E62TV2	-8.863	16.656	-.005	-.532	.595
PaintCodeOUT_E64E14	-12.836	21.155	-.005	-.607	.544
PaintCodeOUT_E64M64	6.744	9.595	.006	.703	.482
PaintCodeOUT_E64PFP	5.596	21.136	.002	.265	.791
PaintCodeOUT_E64TV4	4.765	4.310	.011	1.106	.269
PaintCodeOUT_E65	11.774	20.922	.096	.563	.574
PaintCodeOUT_E65M65	1.855	23.572	.002	.079	.937
PaintCodeOUT_E65TV0	2.143	31.646	.001	.068	.946
PaintCodeOUT_E72	6.226	5.307	.016	1.173	.241
PaintCodeOUT_E74	8.690	2.188	.094	3.972	.000
PaintCodeOUT_E74TV4	9.694	5.156	.018	1.880	.060
PaintCodeOUT_M12	14.525	4.730	.037	3.071	.002
PaintCodeOUT_M12E12	6.341	21.167	.003	.300	.765
PaintCodeOUT_M62	15.784	3.648	.073	4.326	.000
PaintCodeOUT_M62E62	10.022	21.205	.004	.473	.637
PaintCodeOUT_M62TV2	10.121	6.211	.017	1.629	.103

PaintCodeOUT_M64	3.800	2.588	.020	1.468	.142
PaintCodeOUT_M64E64	41.461	12.315	.031	3.367	.001
PaintCodeOUT_M64TV4	4.254	7.216	.005	.590	.556
PaintCodeOUT_M65	7.825	21.239	.030	.368	.713
PaintCodeOUT_M65E65	-6.624	24.801	-.005	-.267	.789
PaintCodeOUT_M65TV0	-15.908	22.642	-.021	-.703	.482
PaintCodeOUT_M70	6.102	15.226	.004	.401	.689
PaintCodeOUT_NAPFP	-3.382	6.053	-.006	-.559	.576
PaintCodeOUT_TV0	-.791	22.777	-.001	-.035	.972
PaintCodeOUT_TV2	-.631	5.801	-.001	-.109	.913
PaintCodeOUT_TV2E62	-8.003	8.621	-.018	-.928	.353
PaintCodeOUT_TV2MJ2	-15.182	8.649	-.034	-1.755	.079
PaintCodeOUT_TV4	3.490	2.806	.017	1.244	.214
PaintCodeOUT_TV4E64	-5.088	8.170	-.006	-.623	.533
PaintCodeOUT_TV4E74	6.813	12.580	.005	.542	.588
PaintCodeOUT_TV4M64	-8.902	14.919	-.005	-.597	.551
PaintCodeOUT_TV4M70	8.776	21.009	.004	.418	.676

Table 24 SPSS 다중선형회귀 계수(Stand Install)

모형	비표준화 계수		표준화 계수	t	유의확률
	B	표준오차	베타		
1 DIA	1.967	.431	.120	4.567	.000
Num of Parts	-1.555	.808	-.083	-1.926	.054
Num of Joints	-2.008	.775	-.090	-2.591	.010
LogLength	3.906	.942	.300	4.149	.000
LogWeight	-8.487	1.160	-.314	-7.314	.000
STAGE_20	101.723	13.131	.073	7.747	.000
STAGE_30	31.892	6.949	.124	4.589	.000
STAGE_40	22.048	6.661	.187	3.310	.001
STAGE_50	26.978	6.880	.102	3.921	.000
STAGE_60	26.421	6.893	.101	3.833	.000
Main_Problem_ShortLT	-52.890	17.368	-.297	-3.045	.002
Main_Problem_Etal	-17.601	17.533	-.057	-1.004	.315
Main_Problem_DesignREV	-38.865	17.395	-.168	-2.234	.026
Main_Problem_DesignPostpone	-50.966	18.632	-.052	-2.735	.006
Main_Problem_DesignProblem	-58.672	21.765	-.031	-2.696	.007
Main_Problem_MaterialProblem	-31.385	21.000	-.018	-1.495	.135
Main_Problem_DelayManufacturing	-12.797	17.292	-.095	-.740	.459
Sch_S100	8.489	14.765	.004	.575	.565
Sch_S105	2.331	2.672	.010	.872	.383
Sch_S160	11.480	5.730	.015	2.004	.045
Sch_S20	-3.239	8.981	-.005	-.361	.718
Sch_S30	-14.484	7.794	-.017	-1.858	.063
Sch_S40	-12.851	8.250	-.032	-1.558	.119
Sch_S40S	3.952	2.016	.018	1.960	.050
Sch_S60	4.210	7.116	.005	.592	.554
Sch_S80	-14.784	7.125	-.044	-2.075	.038
Sch_SSTD	-24.771	19.298	-.009	-1.284	.199
Sch_SXS	-20.914	38.619	-.004	-.542	.588
Sch_SXXS	1.267	6.136	.002	.206	.836
Sch_T2.5	57.771	23.583	.126	2.450	.014
Sch_T25.4	-19.682	37.200	-.005	-.529	.597
Sch_T26	27.128	48.957	.004	.554	.580
Sch_T3	71.399	23.692	.102	3.014	.003
Sch_T3.5	61.498	23.826	.075	2.581	.010
Sch_T31.8	-35.991	35.541	-.007	-1.013	.311

Sch_T35.4	-441	37.276	.000	-.012	.991
Sch_T42	-19.966	28.465	-.005	-.701	.483
Sch_T5.5	88.866	24.403	.093	3.642	.000
Sch_T55	-36.897	48.887	-.005	-.755	.450
Material_CL	-34.900	21.304	-.043	-1.638	.101
Material_LTCS	-41.868	9.760	-.044	-4.290	.000
Material_NA	50.119	50.341	.062	.996	.319
Material_SDSS	3.352	4.159	.011	.806	.420
Material_SS	5.235	3.442	.015	1.521	.128
Material_TN	49.457	48.905	.139	1.011	.312
Making					
Vendor_CorpSungGawngTec	39.495	38.945	.062	1.014	.311
Making Vendor_GunllIndustry	29.592	29.132	.164	1.016	.310
Making Vendor_BuHeong	42.786	29.642	.220	1.443	.149
Making Vendor_Sungll	28.624	29.661	.162	.965	.335
Making Vendor_PNL	.592	29.852	.002	.020	.984
Making Vendor_TSMTec	1.671	31.203	.002	.054	.957
Painting					
Vendor_2CorpSungGawngTec	-38.069	31.273	-.061	-1.217	.224
Painting					
Vendor_2CorpJeYoung	-25.769	20.417	-.019	-1.262	.207
Painting					
Vendor_2GunllIndustry	5.663	18.068	.032	.313	.754
Painting Vendor_2Samll	-30.643	19.923	-.037	-1.538	.124
Painting					
Vendor_2SinCwooAndCKoreaCorp	-34.705	19.983	-.041	-1.737	.082
Painting					
Vendor_2CorporationSungSan	11.588	19.132	.057	.606	.545
Painting Vendor_2PNL	13.845	19.399	.041	.714	.475
Painting					
Vendor_2PsindustryCorp	-15.937	19.444	-.028	-.820	.412
Painting Vendor_2HiAir	4.184	19.041	.025	.220	.826
Warehouse_3SungGawngTec	.829	21.173	.000	.039	.969
Warehouse_3SeyangIndustry	1.146	12.205	.005	.094	.925
Warehouse_3ChilSeoHeapedStage	-8.691	14.019	-.008	-.620	.535
Warehouse_3HiAirKoreaWarehouseCorp	38.300	12.246	.378	3.127	.002
PaintCodeOUT_E64	6.229	4.678	.048	1.332	.183
PaintCodeOUT_E12	46.917	13.224	.037	3.548	.000
PaintCodeOUT_E12E62	109.161	49.529	.015	2.204	.028
PaintCodeOUT_E12E72	70.731	61.292	.010	1.154	.249
PaintCodeOUT_E12M12	13.357	49.568	.002	.269	.788
PaintCodeOUT_E14	.788	6.853	.001	.115	.908
PaintCodeOUT_E14E64	-52.905	49.199	-.007	-1.075	.282
PaintCodeOUT_E62	30.001	8.963	.089	3.347	.001
PaintCodeOUT_E62M62	-54.864	19.282	-.022	-2.845	.004
PaintCodeOUT_E62MJ2	11.644	28.913	.004	.403	.687
PaintCodeOUT_E62TV2	43.741	40.986	.009	1.067	.286
PaintCodeOUT_E64E14	-52.103	49.244	-.007	-1.058	.290
PaintCodeOUT_E64M64	3.787	22.301	.001	.170	.865
PaintCodeOUT_E64PFP	85.038	34.867	.017	2.439	.015
PaintCodeOUT_E64TV4	14.210	10.244	.011	1.387	.165
PaintCodeOUT_E65	-20.906	48.664	-.056	-.430	.668
PaintCodeOUT_E65M65	39.294	54.669	.012	.719	.472
PaintCodeOUT_E65TV0	12.413	61.809	.002	.201	.841
PaintCodeOUT_E72	46.699	14.608	.030	3.197	.001
PaintCodeOUT_E74	-2.733	5.285	-.009	-.517	.605
PaintCodeOUT_E74TV4	-7.783	11.945	-.005	-.652	.515
PaintCodeOUT_M12	28.489	12.976	.021	2.196	.028

PaintCodeOUT_M62	37.581	9.861	.059	3.811	.000
PaintCodeOUT_M62E62	-52.848	49.579	-.007	-1.066	.286
PaintCodeOUT_M62TV2	-17.888	15.309	-.010	-1.168	.243
PaintCodeOUT_M64	19.730	6.066	.035	3.253	.001
PaintCodeOUT_M64E64	69.405	35.003	.014	1.983	.047
PaintCodeOUT_M64TV4	-12.556	16.891	-.005	-.743	.457
PaintCodeOUT_M65	.211	49.219	.000	.004	.997
PaintCodeOUT_M65E65	8.343	56.789	.002	.147	.883
PaintCodeOUT_M65TV0	-22.176	52.733	-.009	-.421	.674
PaintCodeOUT_M70	-83.549	50.109	-.012	-1.667	.095
PaintCodeOUT_NA	-95.907	49.074	-.013	-1.954	.051
PaintCodeOUT_NAPFP	5.118	12.758	.003	.401	.688
PaintCodeOUT_TV0	-45.642	54.409	-.014	-.839	.402
PaintCodeOUT_TV2	21.298	13.729	.016	1.551	.121
PaintCodeOUT_TV2E62	44.902	24.174	.033	1.857	.063
PaintCodeOUT_TV2MJ2	16.480	23.874	.014	.690	.490
PaintCodeOUT_TV4	17.848	7.112	.025	2.510	.012
PaintCodeOUT_TV4E64	12.987	19.243	.005	.675	.500
PaintCodeOUT_TV4E74	12.695	29.225	.003	.434	.664
PaintCodeOUT_TV4M64	1.672	34.732	.000	.048	.962

Table 25 SPSS 다중선형회귀 계수(Install)

모형	비표준화 계수		표준화 계수	t	유의확률	
	B	표준오차	베타			
1	STAGE_20	54.490	13.777	.069	3.955	.000
	STAGE_30	47.758	8.739	.306	5.465	.000
	STAGE_40	26.234	8.512	.336	3.082	.002
	STAGE_50	28.211	8.783	.151	3.212	.001
	STAGE_60	20.293	8.987	.084	2.258	.024
	Main_Problem_ShortLT	-10.120	2.182	-.078	-4.638	.000
	Main_Problem_Etal	-9.149	3.577	-.046	-2.558	.011
	Main_Problem_DesignREV	-11.735	2.718	-.069	-4.318	.000
	Main	-35.078	9.170	-.048	-3.825	.000
	Problem_DesignPostpone					
	Main_Problem_DesignProblem	-26.417	19.717	-.016	-1.340	.180
	Main					
	Problem_MaterialProblem	3.726	15.147	.003	.246	.806
	Sch_S10S	9.654	39.496	.059	.244	.807
	Sch_S160	8.385	39.980	.016	.210	.834
	Sch_S20	10.793	40.960	.022	.264	.792
	Sch_S30	.645	40.881	.001	.016	.987
	Sch_S40	7.850	40.631	.033	.193	.847
	Sch_S40S	17.381	39.474	.114	.440	.660
	Sch_S60	43.927	40.747	.057	1.078	.281
	Sch_S80	11.611	40.499	.058	.287	.774
	Sch_S80S	21.351	39.425	.214	.542	.588
	Sch_SSTD	29.855	55.820	.009	.535	.593
	Sch_SXS	33.306	46.874	.017	.711	.477
	Sch_SXXS	21.645	40.262	.038	.538	.591
	Sch_T15	-.562	55.544	.000	-.010	.992
	Sch_T2.5	-10.247	15.056	-.031	-6.81	.496
	Sch_T25.4	2.390	61.111	.001	.039	.969
	Sch_T27.0	85.480	68.865	.026	1.241	.215
	Sch_T3	-11.228	15.808	-.022	-.710	.478
	Sch_T3.5	5.098	17.399	.006	.293	.770
	Sch_T5.5	4.084	16.540	.006	.247	.805
	Material_CL	-22.784	18.441	-.051	-1.236	.217
	Material_LTCS	3.976	13.156	.004	.302	.763
	Material_NA	-24.391	13.745	-.062	-1.775	.076

Material_SDSS	4.357	5.651	.024	.771	.441
Material_SS	9.635	5.112	.029	1.885	.060
Making					
Vendor_CorpSungGawngTec	-97.738	73.698	-.200	-1.326	.185
Making_Vendor_CorpCheonll	-64.629	52.731	-.027	-1.226	.220
Making_Vendor_GeoMung	-25.014	57.189	-.007	-.437	.662
Making_Vendor_GunllIndustry	-43.004	69.667	-.327	-.617	.537
Making_Vendor_BuHeong	-45.192	69.211	-.337	-.653	.514
Making_Vendor_Sungll	-37.753	69.306	-.335	-.545	.586
Making_Vendor_SinSung	-72.318	57.180	-.022	-1.265	.206
Making_Vendor_PNL	-46.837	69.481	-.283	-.674	.500
Making_Vendor_TSMTec	-51.718	69.824	-.111	-.741	.459
Painting					
Vendor_2CorpSungGawngTec	53.781	62.258	.114	.864	.388
Painting					
Vendor_2CorpJeYoung	-21.586	59.060	-.020	-.365	.715
Painting					
Vendor_2GunllIndustry	-4.695	57.728	-.036	-.081	.935
Painting_Vendor_2Samll	-10.277	57.708	-.020	-.178	.859
Painting					
Vendor_2SinCwooAndCKoreaCorp	-5.930	57.656	-.010	-.103	.918
Painting					
Vendor_2CorporationSungSan	1.713	57.371	.014	.030	.976
Painting_Vendor_2PNL	-3.866	57.463	-.016	-.067	.946
Painting					
Vendor_2PsindustryCorp	-3.944	57.456	-.011	-.069	.945
Painting_Vendor_2HiAir	5.931	57.269	.052	.104	.918
Warehouse_3SungGawngTec	40.599	41.703	.012	.974	.330
Warehouse_3SevangIndustry	3.067	11.783	.016	.260	.795
Warehouse_3ChilSeoHeapedStage	6.273	16.280	.006	.385	.700
Warehouse_3HiAirKoreaWarehouseCorp	21.993	11.847	.331	1.856	.063
PaintCodeOUT_E64	34.104	4.794	.366	7.114	.000
PaintCodeOUT_E12	34.405	14.224	.048	2.419	.016
PaintCodeOUT_E12M12	-19.036	40.812	-.006	-.466	.641
PaintCodeOUT_E14	42.265	9.284	.069	4.552	.000
PaintCodeOUT_E14E64	14.984	40.671	.006	.368	.713
PaintCodeOUT_E62	22.009	10.995	.095	2.002	.045
PaintCodeOUT_E62M62	-15.657	17.633	-.013	-.888	.375
PaintCodeOUT_E62MJ2	15.775	34.379	.007	.459	.646
PaintCodeOUT_E62TV2	36.190	34.400	.015	1.052	.293
PaintCodeOUT_E64M64	5.759	16.913	.004	.340	.734
PaintCodeOUT_E64TV4	28.905	13.672	.027	2.114	.035
PaintCodeOUT_E65	39.613	7.424	.110	5.336	.000
PaintCodeOUT_E65M65	16.984	18.580	.015	.914	.361
PaintCodeOUT_E65TV0	52.146	41.949	.022	1.243	.214
PaintCodeOUT_E72	-3.845	26.904	-.002	-.143	.886
PaintCodeOUT_E74	26.751	5.650	.136	4.735	.000
PaintCodeOUT_E74TV4	18.450	10.612	.024	1.739	.082
PaintCodeOUT_M12	34.208	13.158	.053	2.600	.009
PaintCodeOUT_M12E12	11.964	40.812	.004	.293	.769
PaintCodeOUT_M62	8.929	11.502	.026	.776	.438
PaintCodeOUT_M62E62	-22.769	41.199	-.007	-.553	.581
PaintCodeOUT_M62TV2	5.418	18.300	.004	.296	.767
PaintCodeOUT_M64	15.152	6.125	.044	2.474	.013
PaintCodeOUT_M64E64	9.776	28.446	.004	.344	.731
PaintCodeOUT_M64TV4	14.074	23.488	.007	.599	.549
PaintCodeOUT_M65	42.128	9.489	.109	4.439	.000
PaintCodeOUT_M65E65	40.205	41.850	.012	.961	.337

PaintCodeOUT_M65TV0	82.023	19.316	.069	4.246	.000
PaintCodeOUT_M70	73.210	40.147	.022	1.824	.068
PaintCodeOUT_NA	39.425	40.048	.012	.984	.325
PaintCodeOUT_NAPFP	46.672	22.614	.028	2.064	.039
PaintCodeOUT_TV0	60.785	30.712	.026	1.979	.048
PaintCodeOUT_TV2	21.056	14.621	.030	1.440	.150
PaintCodeOUT_TV2E62	33.562	21.872	.047	1.534	.125
PaintCodeOUT_TV2MJ2	10.606	21.946	.015	.483	.629
PaintCodeOUT_TV4	25.616	6.801	.068	3.767	.000
PaintCodeOUT_TV4E64	26.415	18.620	.021	1.419	.156
PaintCodeOUT_TV4E74	44.145	24.021	.023	1.838	.066
PaintCodeOUT_TV4M70	67.242	39.623	.020	1.697	.090

Table 26 SPSS PLS회귀 계수(Making)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	Making		Making
(Constant)	46.202	[Painting_Vendor=2CorpSungGawngTec]	-1.332
[STAGE=20]	17.143	[Painting_Vendor=2GunllIndustry]	-10.673
[STAGE=30]	11.837	[Painting_Vendor=2HiAir]	11.080
[STAGE=40]	.984	[Painting_Vendor=2PNL]	4.184
[STAGE=50]	-5.728	[Painting_Vendor=2PsinustryCorp]	10.408
[STAGE=60]	-6.180	[Painting_Vendor=2Samll]	4.281
[Main_Problem=DelayManufactu]	24.878	[Painting_Vendor=2SinCwooAndCKoreaCorp]	26.831
[Main_Problem=DesignPostpone]	7.064	[Warehouse=3ChilSeoHeapedStage]	3.752
[Main_Problem=DesignProblem]	38.385	[Warehouse=3HiAirKoreaWarehouseCorp]	-7.568
[Main_Problem=DesignREV]	34.731	[Warehouse=3SeyangIndustry]	13.956
[Main_Problem=Etal]	19.875	[Warehouse=3SungGawngTec]	-40.470
[Main_Problem=MaterialProble]	69.283	[PaintCodeOUT=E12]	-4.445
[Main_Problem=NONE]	21.088	[PaintCodeOUT=E12E62]	-33.565
[Sch=S100]	-11.698	[PaintCodeOUT=E12E72]	56.514
[Sch=S10S]	3.855	[PaintCodeOUT=E14]	7.650
[Sch=S160]	-4.095	[PaintCodeOUT=E14E64]	52.741
[Sch=S20]	-5.549	[PaintCodeOUT=E62]	6.785
[Sch=S30]	-5.780	[PaintCodeOUT=E62M62]	50.806
[Sch=S40]	8.475	[PaintCodeOUT=E62MJ2]	-14.063
[Sch=S40S]	-1.647	[PaintCodeOUT=E62TV2]	-24.867
[Sch=S60]	-10.725	[PaintCodeOUT=E64]	1.163
[Sch=S80]	-3.257	[PaintCodeOUT=E64E14]	61.331
[Sch=S80S]	-2.239	[PaintCodeOUT=E64M64]	-17.715

[Sch=SSTD]	6.489	[PaintCodeOUT=E64PFP]	1.915
[Sch=SXS]	-65.733	[PaintCodeOUT=E64TV4]	-28.175
[Sch=SXXS]	9.488	[PaintCodeOUT=E65]	-8.423
[Sch=T18]	-10.070	[PaintCodeOUT=E65M65]	21.554
[Sch=T2.5]	-11.729	[PaintCodeOUT=E65TV0]	-31.250
[Sch=T25.4]	55.765	[PaintCodeOUT=E72]	-.089
[Sch=T26]	-41.294	[PaintCodeOUT=E74]	9.863
[Sch=T3]	12.313	[PaintCodeOUT=E74TV4]	77.857
[Sch=T3.5]	25.566	[PaintCodeOUT=M12]	.816
[Sch=T31.8]	95.534	[PaintCodeOUT=M12 / E12]	-73.443
[Sch=T35.4]	43.168	[PaintCodeOUT=M62]	.536
[Sch=T42]	27.324	[PaintCodeOUT=M62TV2]	-23.565
[Sch=T49.8]	-82.676	[PaintCodeOUT=M64]	8.039
[Sch=T5.5]	23.669	[PaintCodeOUT=M64E64]	-28.524
[Sch=T53.5]	-105.898	[PaintCodeOUT=M64TV4]	16.339
[Material=CL]	-26.697	[PaintCodeOUT=M65]	-18.101
[Material=CN]	3.733	[PaintCodeOUT=M65E65]	-38.178
[Material=CS]	1.638	[PaintCodeOUT=M65TV0]	-68.300
[Material=DSS]	4.482	[PaintCodeOUT=M70]	-41.589
[Material=LTCS]	32.416	[PaintCodeOUT=NAPFP]	-8.859
[Material=NA]	3.063	[PaintCodeOUT=NONE]	-12.409
[Material=SDSS]	-18.622	[PaintCodeOUT=TV0]	-61.203
[Material=SS]	11.097	[PaintCodeOUT=TV2]	2.657
[Making_Vendor=BuHeong]	-14.585	[PaintCodeOUT=TV2E62]	-20.824
[Making_Vendor=CorpCheonI]	-8.485	[PaintCodeOUT=TV2MJ2]	-18.985
[Making_Vendor=CorpSungGawngTec]	-6.080	[PaintCodeOUT=TV4]	3.576
[Making_Vendor=GeoMung]	-25.766	[PaintCodeOUT=TV4E64]	-11.728
[Making_Vendor=GunIIndustry]	-12.746	[PaintCodeOUT=TV4E74]	-42.339
[Making_Vendor=PNL]	20.120	[PaintCodeOUT=TV4M64]	-7.293
[Making_Vendor=SeongMin]	3.733	DIA	-.264
[Making_Vendor=SinSung]	-69.975	Num_of_Parts	1.414
[Making_Vendor=SungI]	13.870	Num_of_Joints	1.974
[Painting_Vendor=2CorpJeYoung]	.052	LogLength	-2.564
[Painting_Vendor=2CorporationSungSan]	-7.131	LogWeight	3.060

Table 27 SPSS PLS회귀 계수(Painting)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	Painting		Painting
(Constant)	27.471	[Warehouse=3ChilSeoHeapedStage]	-3.027
[STAGE=20]	-4.134	[Warehouse=3HiAirKoreaWarehouseCorp]	.195
[STAGE=30]	.577	[Warehouse=3SeyangIndustry]	.809
[STAGE=40]	.954	[Warehouse=3SungGawngTec]	-15.802
[STAGE=50]	-.656	[PaintCodeOUT=E12]	-7.030
[STAGE=60]	-.350	[PaintCodeOUT=E12E62]	13.374
[Main_Problem=DelayManufactu]	.916	[PaintCodeOUT=E14]	1.623
[Main_Problem=DesignPostpone]	2.707	[PaintCodeOUT=E14E64]	-4.814
[Main_Problem=DesignProblem]	-2.683	[PaintCodeOUT=E14E74]	-4.935
[Main_Problem=DesignREV]	-1.184	[PaintCodeOUT=E14TV4]	-1.570
[Main_Problem=EtaI]	1.107	[PaintCodeOUT=E62]	2.664
[Main_Problem=MaterialProble]	7.303	[PaintCodeOUT=E62M62]	15.205
[Main_Problem=NONE]	4.120	[PaintCodeOUT=E62MJ2]	-6.952
[Sch=S100]	15.074	[PaintCodeOUT=E62TV2]	4.432
[Sch=S10S]	-.437	[PaintCodeOUT=E64]	-.573
[Sch=S160]	.456	[PaintCodeOUT=E64E14]	-10.557
[Sch=S20]	1.036	[PaintCodeOUT=E64M64]	-3.911
[Sch=S30]	.657	[PaintCodeOUT=E64PPF]	5.239
[Sch=S40]	1.349	[PaintCodeOUT=E64TV4]	-8.508
[Sch=S40S]	.215	[PaintCodeOUT=E65]	-4.048
[Sch=S60]	-2.247	[PaintCodeOUT=E65M65]	-9.880
[Sch=S80]	1.302	[PaintCodeOUT=E65TV0]	-3.471
[Sch=S80S]	-.436	[PaintCodeOUT=E72]	9.245
[Sch=SSTD]	-.304	[PaintCodeOUT=E74]	4.834
[Sch=SXS]	-3.943	[PaintCodeOUT=E74TV4]	.217
[Sch=SXXS]	-1.959	[PaintCodeOUT=M12]	-6.768
[Sch=T18]	2.223	[PaintCodeOUT=M12 / E12]	11.754
[Sch=T25.4]	-6.606	[PaintCodeOUT=M62]	5.239
[Sch=T31.8]	-3.713	[PaintCodeOUT=M62E62]	18.088
[Sch=T35.4]	-5.805	[PaintCodeOUT=M62TV2]	6.417
[Sch=T42]	.840	[PaintCodeOUT=M64]	.859
[Material=CL]	-5.509	[PaintCodeOUT=M64E64]	-5.052

[Material=CS]	2.918	[PaintCodeOUT=M64TV4]	-4.189
[Material=DSS]	-1.002	[PaintCodeOUT=M65]	-1.481
[Material=LTCSS]	2.424	[PaintCodeOUT=M65E65]	- .442
[Material=NA]	1.399	[PaintCodeOUT=M65TV0]	4.226
[Material=SDSS]	2.106	[PaintCodeOUT=M70]	-4.554
[Material=SS]	1.183	[PaintCodeOUT=NAPFP]	-2.788
[Making_Vendor=BuHeong]	.899	[PaintCodeOUT=NONE]	-7.583
[Making_Vendor=CorpSungGawngTec]	-.091	[PaintCodeOUT=TV0]	1.224
[Making_Vendor=GunllIndustry]	-.362	[PaintCodeOUT=TV2]	-1.291
[Making_Vendor=PNL]	1.379	[PaintCodeOUT=TV2E62]	-.654
[Making_Vendor=Sungll]	-.714	[PaintCodeOUT=TV2MJ2]	-7.831
[Painting_Vendor=2CorpJeYoung]	29.327	[PaintCodeOUT=TV4]	-3.107
[Painting_Vendor=2CorporationSungSan ]	6.347	[PaintCodeOUT=TV4E64]	-8.006
[Painting_Vendor=2CorpSungGawngTec ]	-1.856	[PaintCodeOUT=TV4E74]	13.009
[Painting_Vendor=2GunllIndustry]	-.505	DIA	-.246
[Painting_Vendor=2HiAir]	-4.782	Num_of_Parts	.137
[Painting_Vendor=2PNL]	-5.868	Num_of_Joints	.035
[Painting_Vendor=2PsindustryCorp]	2.548	LogLength	-.462
[Painting_Vendor=2Samll]	1.624	LogWeight	-.326

Table 28 SPSS PLS회귀 계수(Out Stock)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	Out Stock		Out Stock
(Constant)	3.655	[Painting_Vendor=2Samll]	.138
[STAGE=20]	.001	[Painting_Vendor=2SinCwooAndCKoreaCorp]	-.961
[STAGE=30]	-.241	[Warehouse=3ChilSeoHeapedStage]	-2.720
[STAGE=40]	-.024	[Warehouse=3HiAirKoreaWarehouseCorp]	-1.181
[STAGE=50]	.013	[Warehouse=3SeyangIndustry]	1.307
[STAGE=60]	.323	[Warehouse=3SungGawngTec]	-1.147
[Main_Problem=DelayManufactu]	-.159	[PaintCodeOUT=E12]	1.440
[Main_Problem=DesignPostpone]	-.292	[PaintCodeOUT=E12E62]	-1.148
[Main_Problem=DesignProblem]	-1.072	[PaintCodeOUT=E12E72]	-.309
[Main_Problem=DesignREV]	.203	[PaintCodeOUT=E12M12]	6.621
[Main_Problem=Etal]	.075	[PaintCodeOUT=E14]	-470

[Main_Problem=MaterialProble]	-.129	[PaintCodeOUT=E14E64]	-2.526
[Main_Problem=NONE]	.202	[PaintCodeOUT=E14E74]	-2.958
[Sch=S100]	-.192	[PaintCodeOUT=E14TV4]	1.827
[Sch=S10S]	.019	[PaintCodeOUT=E62]	.093
[Sch=S160]	.471	[PaintCodeOUT=E62M62]	.902
[Sch=S20]	.086	[PaintCodeOUT=E62MJ2]	.371
[Sch=S30]	.254	[PaintCodeOUT=E62TV2]	-2.981
[Sch=S40]	-.253	[PaintCodeOUT=E64]	-.124
[Sch=S40S]	-.066	[PaintCodeOUT=E64E14]	-2.033
[Sch=S60]	-.568	[PaintCodeOUT=E64M64]	-1.638
[Sch=S80]	.152	[PaintCodeOUT=E64PPF]	5.948
[Sch=S80S]	-.101	[PaintCodeOUT=E64TV4]	-1.157
[Sch=SSTD]	-1.654	[PaintCodeOUT=E65]	-.707
[Sch=SXS]	-2.114	[PaintCodeOUT=E65M65]	-.384
[Sch=SXXS]	.668	[PaintCodeOUT=E65TV0]	-1.862
[Sch=T18]	-1.087	[PaintCodeOUT=E72]	.445
[Sch=T2.5]	-.441	[PaintCodeOUT=E74]	.531
[Sch=T25.4]	1.195	[PaintCodeOUT=E74TV4]	-.053
[Sch=T26]	.053	[PaintCodeOUT=M12]	.346
[Sch=T3]	-.472	[PaintCodeOUT=M12 / E12]	.303
[Sch=T3.5]	.517	[PaintCodeOUT=M62]	.299
[Sch=T31.8]	2.300	[PaintCodeOUT=M62E62]	-.116
[Sch=T35.4]	-1.598	[PaintCodeOUT=M62TV2]	-1.833
[Sch=T42]	-.727	[PaintCodeOUT=M64]	-.287
[Sch=T5.5]	.282	[PaintCodeOUT=M64E64]	.770
[Material=CL]	-.593	[PaintCodeOUT=M64TV4]	6.855
[Material=CN]	-.263	[PaintCodeOUT=M65]	-.363
[Material=CS]	.227	[PaintCodeOUT=M65E65]	-1.525
[Material=DSS]	-.220	[PaintCodeOUT=M65TV0]	-.830
[Material=LTCS]	.244	[PaintCodeOUT=NA]	-.076
[Material=NA]	-.340	[PaintCodeOUT=NAPFP]	7.359
[Material=SDSS]	-.082	[PaintCodeOUT=NONE]	-.002
[Material=SS]	1.502	[PaintCodeOUT=TV0]	2.177
[Making_Vendor=BuHeong]	-.120	[PaintCodeOUT=TV2]	-.747
[Making_Vendor=CorpSungGawngTec]	-.891	[PaintCodeOUT=TV2E62]	-1.047
[Making_Vendor=GunllIndustry]	.533	[PaintCodeOUT=TV2MJ2]	.427
[Making_Vendor=PNL]	-.063	[PaintCodeOUT=TV4]	-.331

[Making_Vendor=SeongMin]	-.263	[PaintCodeOUT=TV4E64]	-1.314
[Making_Vendor=SungIl]	-.168	[PaintCodeOUT=TV4E74]	-.917
[Painting_Vendor=2CorpJeYoung]	-.775	[PaintCodeOUT=TV4M64]	2.689
[Painting_Vendor=2CorporationSungSan]	-.232	DIA	-.012
[Painting_Vendor=2CorpSungGawngTec]	-.940	Num_of_Parts	-.013
[Painting_Vendor=2GunIlIndustry]	.452	Num_of_Joints	.000
[Painting_Vendor=2HiAir]	-.076	LogLength	-.127
[Painting_Vendor=2PNL]	-.901	LogWeight	-.006
[Painting_Vendor=2PsindustryCorp]	.178		

Table 29 SPSS PLS회귀 계수(In Stock)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	In Stock		In Stock
(Constant)	10.761	[Painting_Vendor=2PsindustryCorp]	7.852
[STAGE=20]	-4.609	[Painting_Vendor=2SamIl]	24.787
[STAGE=30]	-10.605	[Painting_Vendor=2SinCwooAndCKoreaCorp]	10.195
[STAGE=40]	-.067	[Warehouse=3ChilSeoHeapedStage]	-22.336
[STAGE=50]	-.912	[Warehouse=3HiAirKoreaWarehouseCorp]	9.598
[STAGE=60]	9.260	[Warehouse=3SeyangIndustry]	-7.507
[Main_Problem=DelayManufactu]	-7.607	[Warehouse=3SungGawngTec]	-23.183
[Main_Problem=DesignPostpone]	2.026	[PaintCodeOUT=E12]	-6.417
[Main_Problem=DesignProblem]	20.664	[PaintCodeOUT=E12E62]	-17.688
[Main_Problem=DesignREV]	3.691	[PaintCodeOUT=E12M12]	-5.637
[Main_Problem=Etal]	-4.465	[PaintCodeOUT=E14]	1.687
[Main_Problem=MaterialProble]	8.388	[PaintCodeOUT=E14E64]	-1.966
[Main_Problem=NONE]	-13.497	[PaintCodeOUT=E14TV4]	-2.845
[Sch=S100]	23.214	[PaintCodeOUT=E62]	1.141
[Sch=S10S]	-.969	[PaintCodeOUT=E62M62]	47.795
[Sch=S160]	-.753	[PaintCodeOUT=E62MJ2]	-2.252
[Sch=S20]	-2.127	[PaintCodeOUT=E62TV2]	-3.029
[Sch=S30]	-4.747	[PaintCodeOUT=E64]	-.620
[Sch=S40]	1.348	[PaintCodeOUT=E64E14]	-19.951
[Sch=S40S]	1.152	[PaintCodeOUT=E64M64]	3.980
[Sch=S60]	-10.738	[PaintCodeOUT=E64PFP]	-3.955

[Sch=S80]	-.256	[PaintCodeOUT=E64TV4]	.722
[Sch=S80S]	-.532	[PaintCodeOUT=E65]	1.799
[Sch=SSTD]	16.008	[PaintCodeOUT=E65M65]	-1.044
[Sch=SXS]	-12.036	[PaintCodeOUT=E65TV0]	-17.168
[Sch=SXXS]	9.402	[PaintCodeOUT=E72]	-4.117
[Sch=T18]	3.042	[PaintCodeOUT=E74]	2.390
[Sch=T2.5]	2.724	[PaintCodeOUT=E74TV4]	3.789
[Sch=T25.4]	-6.611	[PaintCodeOUT=M12]	2.180
[Sch=T26]	11.929	[PaintCodeOUT=M12 / E12]	-7.537
[Sch=T3]	-.756	[PaintCodeOUT=M62]	5.257
[Sch=T3.5]	-.783	[PaintCodeOUT=M62E62]	-.989
[Sch=T31.8]	2.076	[PaintCodeOUT=M62TV2]	2.850
[Sch=T35.4]	2.161	[PaintCodeOUT=M64]	-2.478
[Sch=T42]	11.742	[PaintCodeOUT=M64E64]	35.346
[Sch=T5.5]	-8.307	[PaintCodeOUT=M64TV4]	.029
[Material=CL]	1.873	[PaintCodeOUT=M65]	.151
[Material=CN]	.261	[PaintCodeOUT=M65E65]	-8.451
[Material=CS]	.618	[PaintCodeOUT=M65TV0]	-19.853
[Material=DSS]	.319	[PaintCodeOUT=M70]	-4.466
[Material=LTCS]	1.037	[PaintCodeOUT=NAPFP]	-8.376
[Material=NA]	-.929	[PaintCodeOUT=NONE]	-1.376
[Material=SDSS]	1.499	[PaintCodeOUT=TV0]	-14.228
[Material=SS]	-5.244	[PaintCodeOUT=TV2]	-5.491
[Making_Vendor=BuHeong]	-1.938	[PaintCodeOUT=TV2E62]	-2.707
[Making_Vendor=CorpSungGawngTec]	-.901	[PaintCodeOUT=TV2MJ2]	-8.570
[Making_Vendor=GunllIndustry]	1.898	[PaintCodeOUT=TV4]	-2.080
[Making_Vendor=PNL]	5.859	[PaintCodeOUT=TV4E64]	-9.046
[Making_Vendor=SeongMin]	.261	[PaintCodeOUT=TV4E74]	8.506
[Making_Vendor=Sungll]	-3.559	[PaintCodeOUT=TV4M64]	-17.779
[Painting_Vendor=2CorpJeYoung]	-.182	DIA	.029
[Painting_Vendor=2CorporationSungSan ]	-1.880	Num_of_Parts	.199
[Painting_Vendor=2CorpSungGawngTec ]	.901	Num_of_Joints	.461
[Painting_Vendor=2GunllIndustry]	.969	LogLength	-.155
[Painting_Vendor=2HiAir]	-2.992	LogWeight	1.054
[Painting_Vendor=2PNL]	1.150		

Table 30 SPSS PLS회귀 계수(Stand Install)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	StandInstall		StandInstall
(Constant)	58.263	[Painting_Vendor=2PsindustryCorp]	-19.255
[STAGE=20]	75.416	[Painting_Vendor=2Samll]	-41.098
[STAGE=30]	4.085	[Painting_Vendor=2SinCwooAndCKoreaCorp]	-41.934
[STAGE=40]	-2.521	[Warehouse=3ChilSeoHeapedStage]	-31.142
[STAGE=50]	1.847	[Warehouse=3HiAirKoreaWarehouseCorp]	21.231
[STAGE=60]	2.250	[Warehouse=3SeyangIndustry]	-19.538
[Main_Problem=DelayManufactu]	34.475	[Warehouse=3SungGawngTec]	-1.842
[Main_Problem=DesignPostpone]	-7.278	[PaintCodeOUT=E12]	17.960
[Main_Problem=DesignProblem]	-10.688	[PaintCodeOUT=E12E62]	96.280
[Main_Problem=DesignREV]	9.444	[PaintCodeOUT=E12E72]	40.611
[Main_Problem=Etal]	29.559	[PaintCodeOUT=E12M12]	-15.381
[Main_Problem=MaterialProble]	10.593	[PaintCodeOUT=E14]	-8.578
[Main_Problem=NONE]	45.352	[PaintCodeOUT=E14E64]	-69.308
[Sch=S100]	3.255	[PaintCodeOUT=E62]	3.043
[Sch=S10S]	3.483	[PaintCodeOUT=E62M62]	-96.513
[Sch=S160]	3.527	[PaintCodeOUT=E62MJ2]	-33.160
[Sch=S20]	14.306	[PaintCodeOUT=E62TV2]	14.220
[Sch=S30]	-5.798	[PaintCodeOUT=E64]	-0.883
[Sch=S40]	-1.459	[PaintCodeOUT=E64E14]	-56.578
[Sch=S40S]	1.410	[PaintCodeOUT=E64M64]	-5.699
[Sch=S60]	6.796	[PaintCodeOUT=E64PFP]	100.724
[Sch=S80]	-4.184	[PaintCodeOUT=E64TV4]	12.847
[Sch=S80S]	-4.260	[PaintCodeOUT=E65]	14.851
[Sch=SSTD]	-25.602	[PaintCodeOUT=E65M65]	43.524
[Sch=SXS]	45.138	[PaintCodeOUT=E65TV0]	-5.269
[Sch=SXXS]	-9.898	[PaintCodeOUT=E72]	23.357
[Sch=T2.5]	-.370	[PaintCodeOUT=E74]	-11.983
[Sch=T25.4]	-20.839	[PaintCodeOUT=E74TV4]	-24.059
[Sch=T26]	8.502	[PaintCodeOUT=M12]	-0.534
[Sch=T3]	15.315	[PaintCodeOUT=M62]	10.356
[Sch=T3.5]	2.174	[PaintCodeOUT=M62E62]	-91.306

[Sch=T31.8]	-.995	[PaintCodeOUT=M62TV2]	-60.426
[Sch=T35.4]	7.628	[PaintCodeOUT=M64]	12.421
[Sch=T42]	-34.785	[PaintCodeOUT=M64E64]	71.427
[Sch=T5.5]	36.750	[PaintCodeOUT=M64TV4]	-31.458
[Material=CL]	-19.958	[PaintCodeOUT=M65]	29.608
[Material=CN]	8.334	[PaintCodeOUT=M65E65]	9.580
[Material=CS]	6.128	[PaintCodeOUT=M65TV0]	-29.286
[Material=DSS]	-3.830	[PaintCodeOUT=M70]	-106.600
[Material=LTCS]	-28.457	[PaintCodeOUT=NA]	-120.526
[Material=NA]	-6.539	[PaintCodeOUT=NAPFP]	-2.327
[Material=SDSS]	-14.585	[PaintCodeOUT=NONE]	-2.887
[Material=SS]	2.873	[PaintCodeOUT=TV0]	-15.780
[Making_Vendor=BuHeong]	12.145	[PaintCodeOUT=TV2]	-11.321
[Making_Vendor=CorpSungGawngTec]	-16.147	[PaintCodeOUT=TV2E62]	1.119
[Making_Vendor=GunllIndustry]	.297	[PaintCodeOUT=TV2MJ2]	-29.241
[Making_Vendor=PNL]	-18.486	[PaintCodeOUT=TV4]	11.216
[Making_Vendor=SeongMin]	8.334	[PaintCodeOUT=TV4E64]	.872
[Making_Vendor=Sungll]	1.059	[PaintCodeOUT=TV4E74]	-32.752
[Painting_Vendor=2CorpJeYoung]	-35.826	DIA	.103
[Painting_Vendor=2CorporationSungSan ]	3.643	Num_of_Parts	-1.862
[Painting_Vendor=2CorpSungGawngTec ]	-19.084	Num_of_Joints	-1.949
[Painting_Vendor=2GunllIndustry]	1.544	LogLength	.786
[Painting_Vendor=2HiAir]	2.287	LogWeight	-4.478
[Painting_Vendor=2PNL]	1.199		

Table 31 SPSS PLS회귀 계수(Install)

Independent Variables	Dependent Variables	Independent Variables	Dependent Variables
	Install		Install
(Constant)	31.178	[Painting_Vendor=2PsindustryCorp]	-7.060
[STAGE=20]	25.338	[Painting_Vendor=2Samll]	-10.122
[STAGE=30]	16.716	[Painting_Vendor=2SinCwooAndCKoreaCorp]	-8.645
[STAGE=40]	-3.648	[Warehouse=3ChilSeoHeapedStage]	-8.488
[STAGE=50]	-1.758	[Warehouse=3HiAirKoreaWarehouseCorp]	10.229
[STAGE=60]	-10.066	[Warehouse=3SeyangIndustry]	-9.210

[Main_Problem=DelayManufactu]	8.712	[Warehouse=3SungGawngTec]	28.145
[Main_Problem=DesignPostpone]	-28.467	[PaintCodeOUT=E12]	9.770
[Main_Problem=DesignProblem]	-35.449	[PaintCodeOUT=E12M12]	-48.976
[Main_Problem=DesignREV]	-2.103	[PaintCodeOUT=E14]	27.745
[Main_Problem=Etal]	-.445	[PaintCodeOUT=E14E64]	4.014
[Main_Problem=MaterialProble]	14.178	[PaintCodeOUT=E62]	-.100
[Sch=S10S]	-3.192	[PaintCodeOUT=E62M62]	-37.416
[Sch=S160]	-5.719	[PaintCodeOUT=E62MJ2]	-17.157
[Sch=S20]	-3.249	[PaintCodeOUT=E62TV2]	8.875
[Sch=S30]	-16.184	[PaintCodeOUT=E64]	7.800
[Sch=S40]	-6.938	[PaintCodeOUT=E64M64]	-22.017
[Sch=S40S]	1.075	[PaintCodeOUT=E64TV4]	6.553
[Sch=S60]	34.158	[PaintCodeOUT=E65]	11.293
[Sch=S80]	-4.743	[PaintCodeOUT=E65M65]	-35.118
[Sch=S80S]	3.098	[PaintCodeOUT=E65TV0]	-4.011
[Sch=SSTD]	23.382	[PaintCodeOUT=E72]	3.450
[Sch=SXS]	38.857	[PaintCodeOUT=E74]	-1.043
[Sch=SXXS]	8.261	[PaintCodeOUT=E74TV4]	-10.154
[Sch=T2.5]	-7.395	[PaintCodeOUT=M12]	9.602
[Sch=T25.4]	-42.909	[PaintCodeOUT=M12 / E12]	-14.084
[Sch=T3]	-4.492	[PaintCodeOUT=M62]	-13.950
[Sch=T3.5]	10.555	[PaintCodeOUT=M62E62]	-45.847
[Material=CL]	-13.701	[PaintCodeOUT=M62TV2]	-17.093
[Material=CN]	3.758	[PaintCodeOUT=M64]	-12.474
[Material=CS]	-3.669	[PaintCodeOUT=M64E64]	-13.348
[Material=DSS]	.451	[PaintCodeOUT=M64TV4]	-12.205
[Material=LTCS]	1.793	[PaintCodeOUT=M65]	4.467
[Material=NA]	-11.021	[PaintCodeOUT=M65E65]	-.663
[Material=SDSS]	-4.443	[PaintCodeOUT=M65TV0]	34.513
[Material=SS]	9.912	[PaintCodeOUT=NA]	49.513
[Making_Vendor=BuHeong]	-1.234	[PaintCodeOUT=NAPFP]	18.822
[Making_Vendor=CorpCheonll]	-24.716	[PaintCodeOUT=NONE]	-11.813
[Making_Vendor=CorpSungGawngTec]	-4.530	[PaintCodeOUT=TV0]	13.866
[Making_Vendor=GeoMung]	21.055	[PaintCodeOUT=TV2]	-4.011
[Making_Vendor=GunllIndustry]	-3.778	[PaintCodeOUT=TV2E62]	2.207
[Making_Vendor=PNL]	-3.045	[PaintCodeOUT=TV2MJ2]	-22.438
[Making_Vendor=SeongMin]	3.758	[PaintCodeOUT=TV4]	-.818

[Making_Vendor=SinSung]	-28.649	[PaintCodeOUT=TV4E64]	6.147
[Making_Vendor=SungIl]	5.612	[PaintCodeOUT=TV4E74]	14.432
[Painting_Vendor=2CorpJeYoung]	-20.901	DIA	-.478
[Painting_Vendor=2CorporationSungSan]	1.293	Num_of_Parts	.442
[Painting_Vendor=2CorpSungGawngTec]	4.882	Num_of_Joints	-.171
[Painting_Vendor=2GunIlIndustry]	-3.745	LogLength	1.012
[Painting_Vendor=2HiAir]	4.541	LogWeight	-1.578
[Painting_Vendor=2PNL]	-4.956		



## 부록 C

### C.1 2차 데이터전처리 R 스크립트 전문

Table 32 2차 데이터전처리 R 스크립트 전문

---

```
#원본데이터 로딩
totalSpoolDataMaking <- read.csv("101_1제작_10000.csv", header=TRUE) # 제작데이터 로딩
totalSpoolDataPaint <- read.csv("101_2도장_10000.csv", header=TRUE) # 도장데이터 로딩
totalSpoolDataOutStock <- read.csv("101_3사외재고_10000.csv", header=TRUE) #
사외재고데이터 로딩
totalSpoolDataInStock <- read.csv("101_4사내재고_10000.csv", header=TRUE) #
사내재고데이터 로딩
totalSpoolDataStandInstall <- read.csv("101_5설치대기_10000.csv", header=TRUE) #
설치대기데이터 로딩
totalSpoolDataInstall <- read.csv("101_6설치_10000.csv", header=TRUE) # 설치데이터 로딩

## 데이터 타입 변형
totalSpoolDataMaking$DIA <- as.numeric(totalSpoolDataMaking$DIA)
totalSpoolDataMaking$LogLength <- as.numeric(totalSpoolDataMaking$LogLength)
totalSpoolDataMaking$LogWeight <- as.numeric(totalSpoolDataMaking$LogWeight)
totalSpoolDataMaking$Num.of.Parts <- as.numeric(totalSpoolDataMaking$Num.of.Parts)
totalSpoolDataMaking$Num.of.Joints <- as.numeric(totalSpoolDataMaking$Num.of.Joints)
totalSpoolDataMaking$STAGE<- as.factor(totalSpoolDataMaking$STAGE)

totalSpoolDataPaint$DIA <- as.numeric(totalSpoolDataPaint$DIA)
totalSpoolDataPaint$LogLength <- as.numeric(totalSpoolDataPaint$LogLength)
totalSpoolDataPaint$LogWeight <- as.numeric(totalSpoolDataPaint$LogWeight)
totalSpoolDataPaint$Num.of.Parts <- as.numeric(totalSpoolDataPaint$Num.of.Parts)
totalSpoolDataPaint$Num.of.Joints <- as.numeric(totalSpoolDataPaint$Num.of.Joints)
totalSpoolDataPaint$STAGE<- as.factor(totalSpoolDataPaint$STAGE)

totalSpoolDataOutStock$DIA <- as.numeric(totalSpoolDataOutStock$DIA)
totalSpoolDataOutStock$LogLength <- as.numeric(totalSpoolDataOutStock$LogLength)
totalSpoolDataOutStock$LogWeight <- as.numeric(totalSpoolDataOutStock$LogWeight)
totalSpoolDataOutStock$Num.of.Parts <- as.numeric(totalSpoolDataOutStock$Num.of.Parts)
totalSpoolDataOutStock$Num.of.Joints <- as.numeric(totalSpoolDataOutStock$Num.of.Joints)
totalSpoolDataOutStock$STAGE<- as.factor(totalSpoolDataOutStock$STAGE)

totalSpoolDataInStock$DIA <- as.numeric(totalSpoolDataInStock$DIA)
totalSpoolDataInStock$LogLength <- as.numeric(totalSpoolDataInStock$LogLength)
totalSpoolDataInStock$LogWeight <- as.numeric(totalSpoolDataInStock$LogWeight)
totalSpoolDataInStock$Num.of.Parts <- as.numeric(totalSpoolDataInStock$Num.of.Parts)
totalSpoolDataInStock$Num.of.Joints <- as.numeric(totalSpoolDataInStock$Num.of.Joints)
totalSpoolDataInStock$STAGE<- as.factor(totalSpoolDataInStock$STAGE)
```

---

---

```
totalSpoolDataStandInstall$DIA <- as.numeric(totalSpoolDataStandInstall$DIA)
totalSpoolDataStandInstall$LogLength <- as.numeric(totalSpoolDataStandInstall$LogLength)
totalSpoolDataStandInstall$LogWeight <- as.numeric(totalSpoolDataStandInstall$LogWeight)
totalSpoolDataStandInstall$Num.of.Parts <- as.numeric(totalSpoolDataStandInstall$Num.of.Parts)
totalSpoolDataStandInstall$Num.of.Joints <- as.numeric(totalSpoolDataStandInstall$Num.of.Joints)
totalSpoolDataStandInstall$STAGE<- as.factor(totalSpoolDataStandInstall$STAGE)
```

```
totalSpoolDataInstall$DIA <- as.numeric(totalSpoolDataInstall$DIA)
totalSpoolDataInstall$LogLength <- as.numeric(totalSpoolDataInstall$LogLength)
totalSpoolDataInstall$LogWeight <- as.numeric(totalSpoolDataInstall$LogWeight)
totalSpoolDataInstall$Num.of.Parts <- as.numeric(totalSpoolDataInstall$Num.of.Parts)
totalSpoolDataInstall$Num.of.Joints <- as.numeric(totalSpoolDataInstall$Num.of.Joints)
totalSpoolDataInstall$STAGE<- as.factor(totalSpoolDataInstall$STAGE)
```

##칼럼 병합 및 삭제

###제작

```
totalSpoolDataMakingCB <-
cbind(totalSpoolDataMaking["STAGE"],totalSpoolDataMaking["Main.Problem"],totalSpoolDataMaking["DIA"],totalSpoolDataMaking["Sch"],totalSpoolDataMaking["Material"],
totalSpoolDataMaking["Num.of.Parts"],totalSpoolDataMaking["Num.of.Joints"],totalSpoolDataMaking["Making.Vendor"],totalSpoolDataMaking["Painting.Vendor"],totalSpoolDataMaking["Warehouse"],totalSpoolDataMaking["PaintCodeOUT"], totalSpoolDataMaking["LogLength"],
totalSpoolDataMaking["LogWeight"], totalSpoolDataMaking["Making"])
names(totalSpoolDataMakingCB)
```

###도장

```
totalSpoolDataPaintCB <-
cbind(totalSpoolDataPaint["STAGE"],totalSpoolDataPaint["Main.Problem"],totalSpoolDataPaint["DIA"],totalSpoolDataPaint["Sch"],totalSpoolDataPaint["Material"],
totalSpoolDataPaint["Num.of.Parts"],totalSpoolDataPaint["Num.of.Joints"],totalSpoolDataPaint["Making.Vendor"], totalSpoolDataPaint["Painting.Vendor"], totalSpoolDataPaint["Warehouse"],
totalSpoolDataPaint["PaintCodeOUT"], totalSpoolDataPaint["LogLength"],
totalSpoolDataPaint["LogWeight"], totalSpoolDataPaint["Painting"])
```

###사외재고

```
totalSpoolDataOutStockCB <-
cbind(totalSpoolDataOutStock["STAGE"],totalSpoolDataOutStock["Main.Problem"],totalSpoolDataOutStock["DIA"],totalSpoolDataOutStock["Sch"],totalSpoolDataOutStock["Material"],
totalSpoolDataOutStock["Num.of.Parts"],totalSpoolDataOutStock["Num.of.Joints"],totalSpoolDataOutStock["Making.Vendor"], totalSpoolDataOutStock["Painting.Vendor"],
totalSpoolDataOutStock["Warehouse"], totalSpoolDataOutStock["PaintCodeOUT"],
totalSpoolDataOutStock["LogLength"], totalSpoolDataOutStock["LogWeight"],
totalSpoolDataOutStock["OutStock"])
```

###사내재고

```
totalSpoolDataInStockCB <-
cbind(totalSpoolDataInStock["STAGE"],totalSpoolDataInStock["Main.Problem"],totalSpoolDataInStock
```

---

---

```
ck["DIA"],totalSpoolDataInStock["Sch"],totalSpoolDataInStock["Material"],
totalSpoolDataInStock["Num.of.Parts"],totalSpoolDataInStock["Num.of.Joints"],totalSpoolDataInSto
ck["Making.Vendor"], totalSpoolDataInStock["Painting.Vendor"],
totalSpoolDataInStock["Warehouse"], totalSpoolDataInStock["PaintCodeOUT"],
totalSpoolDataInStock["LogLength"], totalSpoolDataInStock["LogWeight"],
totalSpoolDataInStock["InStock"])
```

```
###설치대
```

```
totalSpoolDataStandInstallCB <-
cbind(totalSpoolDataStandInstall["STAGE"],totalSpoolDataStandInstall["Main.Problem"],totalSpoolD
ataStandInstall["DIA"],totalSpoolDataStandInstall["Sch"],totalSpoolDataStandInstall["Material"],
totalSpoolDataStandInstall["Num.of.Parts"],totalSpoolDataStandInstall["Num.of.Joints"],totalSpoolD
ataStandInstall["Making.Vendor"], totalSpoolDataStandInstall["Painting.Vendor"],
totalSpoolDataStandInstall["Warehouse"], totalSpoolDataStandInstall["PaintCodeOUT"],
totalSpoolDataStandInstall["LogLength"], totalSpoolDataStandInstall["LogWeight"],
totalSpoolDataStandInstall["StandInstall"])
totalSpoolDataStandInstallCB <-
totalSpoolDataStandInstallCB[totalSpoolDataStandInstallCB$StandInstall >=0,]
```

```
###설치
```

```
totalSpoolDataInstallCB <-
cbind(totalSpoolDataInstall["STAGE"],totalSpoolDataInstall["Main.Problem"],totalSpoolDataInstall["
DIA"],totalSpoolDataInstall["Sch"],totalSpoolDataInstall["Material"],
totalSpoolDataInstall["Num.of.Parts"],totalSpoolDataInstall["Num.of.Joints"],totalSpoolDataInstall["
Making.Vendor"], totalSpoolDataInstall["Painting.Vendor"], totalSpoolDataInstall["Warehouse"],
totalSpoolDataInstall["PaintCodeOUT"], totalSpoolDataInstall["LogLength"],
totalSpoolDataInstall["LogWeight"], totalSpoolDataInstall["Install"])
```

```
#학습-평가데이터 분리
```

```
##제작
```

```
smp_size <- floor(0.7*nrow(totalSpoolDataMakingCB))
MakingTrain_ind <- sample(seq_len(nrow(totalSpoolDataMakingCB)),size = smp_size)
set.seed(1)
MakingTrain <- totalSpoolDataMakingCB[MakingTrain_ind,]
MakingTest <- totalSpoolDataMakingCB[-MakingTrain_ind,]
```

```
##도장
```

```
smp_size <- floor(0.7*nrow(totalSpoolDataPaintCB))
PaintTrain_ind <- sample(seq_len(nrow(totalSpoolDataPaintCB)),size = smp_size)
PaintTrain <- totalSpoolDataPaintCB[PaintTrain_ind,]
PaintTest <- totalSpoolDataPaintCB[-PaintTrain_ind,]
```

```
##사외재고
```

```
smp_size <- floor(0.7*nrow(totalSpoolDataOutStockCB))
OutStockTrain_ind <- sample(seq_len(nrow(totalSpoolDataOutStockCB)),size = smp_size)
OutStockTrain <- totalSpoolDataOutStockCB[PaintTrain_ind,]
OutStockTest <- totalSpoolDataOutStockCB[-PaintTrain_ind,]
```

---

---

```
##사내재고
smp_size <- floor(0.7*nrow(totalSpoolDataInStockCB))
InStockTrain_ind <- sample(seq_len(nrow(totalSpoolDataInStockCB)),size = smp_size)
InStockTrain <- totalSpoolDataInStockCB[InStockTrain_ind,]
InStockTest <- totalSpoolDataInStockCB[-InStockTrain_ind,]

##설치대기
smp_size <- floor(0.7*nrow(totalSpoolDataStandInstallCB))
StandInstallTrain_ind <- sample(seq_len(nrow(totalSpoolDataStandInstallCB)),size = smp_size)
StandInstallTrain <- totalSpoolDataStandInstallCB[StandInstallTrain_ind,]
StandInstallTest <- totalSpoolDataStandInstallCB[-StandInstallTrain_ind,]

##설치
smp_size <- floor(0.7*nrow(totalSpoolDataInstallCB))
InstallTrain_ind <- sample(seq_len(nrow(totalSpoolDataInstallCB)),size = smp_size)
InstallTrain <- totalSpoolDataInstallCB[InstallTrain_ind,]
InstallTest <- totalSpoolDataInstallCB[-InstallTrain_ind,]
```

