

AIAP Final Project Report

21900416 안건힐

I. Introduction

i. What is Press Processing?



Figure 1. Product by Press Processing

소성 가공(Plastic Working)은 물체의 소성을 이용하여 변형시켜 다양한 모양을 만드는 가공법으로, 압연, 압출, 인발, 프레스 등이 있다. 자동차 프레임 제작 시 사용되는 프레스 생산 기술은 철제 강판의 소성변형 성질을 이용하여 제품의 형상을 구현하며, 금형을 프레스에 장착하고 왕복 압축 운동을 통해 높은 압력으로 재료를 절단하거나 성형하는 기술이다. 프레스 방식을 사용하는 이유는 제품의 강도와 치수 정밀도가 높고, 기계적 성질이 우수하며, 금형을 사용하여 대량 생산에 적합하기 때문이다.

ii. How does the Press Processing Work?

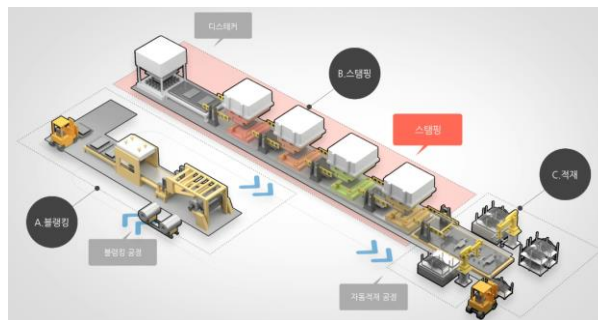


Figure 2. Hyundai 3D Manufacturing Process

Pressing 공정은 Figure 2의 Minimap과 같다. A: Blanking 과정은 코일에 말려져 있는 강판을 성형이 가능한 패널 크기로 자른다. 잘려진 강판은 디스테커로 옮겨져 컨베이어 벨트 위로 분배한다. B: Stamping 과정은 강판을 고열로 가열하여 성형이 쉬워지도록 한다. 가열된 강판은 금형 내에서 Drawing(기초 형상으로 성형), Trimming(자르기), Piercing(구멍 가공), Flanging(구부리기) 과정을 거친다. C: Stacking 과정은 자동 적재 시스템으로 성형이 완료된 부품들을 수송에 편하도록 적재한다.

iii. What is the Issue? Research Motivation

Press 공정과 같은 연속공정에 사용되는 설비 제원들은 설비의 상태(구동 모터, 감속기, 베어링 등의 마모)에 따라 에너지 소모량에 차이가 발생한다. 이에 소모되는 전력량 분석을 통해 에너지 자원의 최적화를 이루어 생산 과정에서의 비용 절감을 이루는 것을 목표로 한다.

II. Methodology

i. Data Information

데이터는 로봇 (서보 모터) 2대, 유압 펌프 4대에 설치된 전류 센서로부터 전력 실효값($RMS = \sqrt{\frac{1}{T} \int_0^T i(t) dt}$)을 얻어온다. 이는 전류가 실제로 일한 양을 의미한다. 전류 센서로부터 얻은 데이터를 수신기로 모은 후 Edge Computer를 거쳐 OPC-UA(Open Platform Communications Unified Architecture)에 저장된다.



Figure 3. Data Collection Method

ii. 데이터 전처리: Data 품질지수 검증

데이터는 완전성, 유일성, 유효성, 일관성, 정확성, 무결성을 기준으로 품질 전처리 과정을 거친다.

완전성: NULL 값이 30% 이상인 데이터 열을 삭제한다. (100%)

유일성: key 값들 개수로 정렬했을 때, 1 이상인 행 수 비율. (91%)

유효성: 개수 개수, 전류 데이터가 음수일 경우 제외. (100%)

일관성: 측정 시각에 대한 날짜와 시간 데이터의 형식 대조. (100%)

정확성: 공정 데이터(df_0)와 로봇 1,2 & 프레스 1~4 데이터 (df_1~6)의 시간 데이터 비교. (100%)

무결성: (유일성/유효성/일관성 지수 중 100% 지수 개수)/3 X 100 (66%)

iii. 데이터 시각화

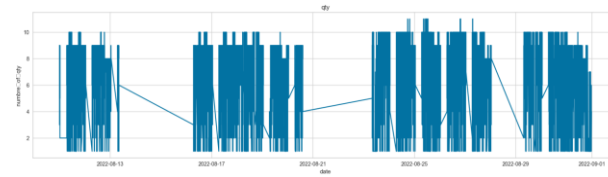


Figure 4. Process Quantity Data Visualization

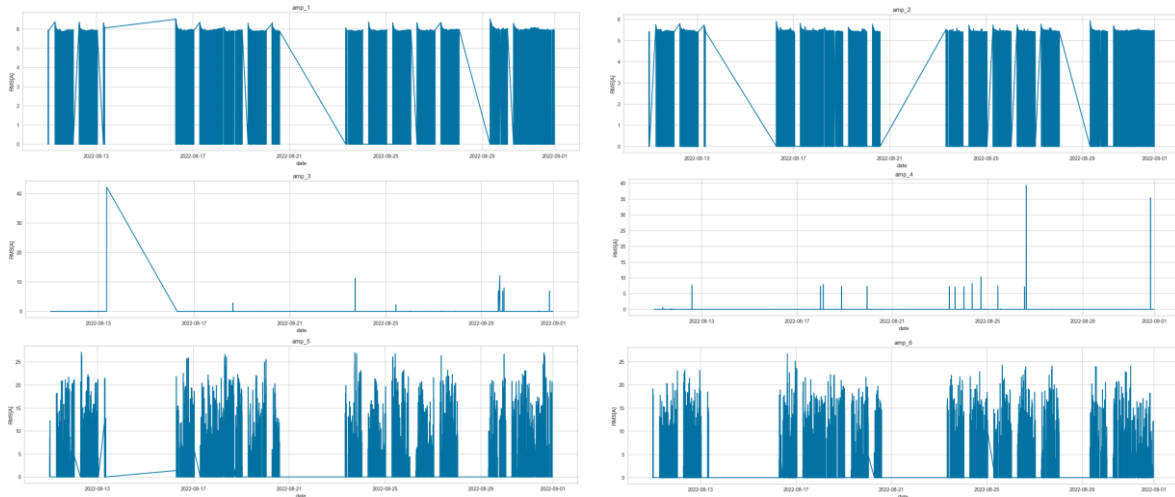


Figure 5. Current Data Plot for Visualization

iv. PyCaret을 이용한 회귀 분석

Compare Model

```
1 amp_list = ['amp_1', 'amp_2', 'amp_3', 'amp_4', 'amp_5', 'amp_6']
2
3 for amp in amp_list:
4     regi = setup(
5         df[['itemno', 'qty', amp]],
6         target = amp,
7         categorical_features = ['itemno'],
8         numeric_features = ['qty'],
9         session_id = 123,
10        log_experiment = True,
11        experiment_name = f'{amp}_best_model',
12        html = False, # silent 실행은 최신 버전에서 지원되지 않으므로 html로 대체
13        index = False
14    )
15    best_model = compare_models(
16        include = ['lr', 'rf', 'ada'], # GuideBook에서 사용한 모델
17        exclude = ['lr', 'ada', 'rf', 'xgboost', 'svm'], # 추가된 모델 (XGBOOST, SVM)
18        sort = 'MAE'
19    )
20    tuned_best_model = tune_model(best_model)
21    save_model(tuned_best_model, model_name = f'{amp}_best_model')
```

Figure 6. Comparing and Tuning Model Code

Compare_models()는 정의된 기준에 따라 최고 성능의 모델만 반환한다. sort 옵션은 모델 선택을 수행할 지표의 이름을 전달하여 순서를 변경한다. 따라서 위 코드는 'MAE' 성능이 가장 좋은 모델을 선택하도록 하였다. 'MAE' 값을 기준으로 한 것에는 다음과 같은 이유가 있다.

1. 'MAE' (Mean Absolute Error)는 실제값과 얼마나 떨어져 있는지 확인하는 것으로 각 모델이 추정한 값과의 오차값을 최대한 줄일 수 있다.
2. 'MAE'는 'MSE'와 'RMSE'보다 이상치에 덜 예민하다.

우리가 구하고자 하는 것은 각 기계에 적합한 전력 소모량 예측 모델이다. Dataset

에 주어진 Press machine 6종류는 구동 모터의 종류, 감속기, 베어링의 수명 등에 의해 서로 다른 전력 소모량을 보여준다. 이에 각 Press machine들에게 최적화된 모델을 찾아야 한다. 따라서 각 amp(RMS) 추정 비교모델군은 Linear Regression, Random Forest, AdaBoost Regression, XGBoost, SVM이다.

Model Description

Table 1. Model Comparison

Model	장점	단점
Linear Regression	<ul style="list-style-type: none"> - 단순하고 이해 및 구현이 쉬움 - 빠른 계산 속도 및 효율적인 학습 - 결과 해석이 쉬움 	<ul style="list-style-type: none"> - 독립 변수와 종속 변수 간의 선형 관계를 가정 - 이상치에 민감 - 비선형 관계를 잘 모델링하지 못함
Random Forest	<ul style="list-style-type: none"> - 비선형 관계를 잘 모델링 가능 - 높은 예측 성능 - 이상치에 강함 	<ul style="list-style-type: none"> - 많은 메모리와 계산 자원 필요 - 훈련 시간이 길어질 수 있음 - 결과 해석이 어려움 - 매우 큰 데이터셋에서는 비효율적일 수 있음
AdaBoost	<ul style="list-style-type: none"> - 약한 학습기를 결합하여 강력한 모델을 만듦 - 과적합 방지 - 다양한 약한 학습기를 사용할 수 있어 유연함 	<ul style="list-style-type: none"> - 노이즈 데이터에 민감 - 많은 부스팅 단계가 필요할 경우 훈련 시간이 길어질 수 있음 - 약한 학습기의 성능에 크게 의존함
XGBoost	<ul style="list-style-type: none"> - 높은 예측 성능 및 효율성 - 빠른 훈련 속도 - 과적합 방지 메커니즘 제공 - 다양한 데이터 전처리와 기능 지원 	<ul style="list-style-type: none"> - 많은 하이퍼파라미터로 인해 튜닝이 복잡 - 많은 메모리 사용 - 매우 큰 데이터셋에서는 비효율적일 수 있음 - 결과 해석이 어려움
SVM	<ul style="list-style-type: none"> - 고차원 데이터에 효과적 - 비선형 데이터도 커널 트릭을 통해 효과적으로 분류 가능 - 일반화 성능이 우수함 - 과적합 방지 메커니즘 제공 	<ul style="list-style-type: none"> - 큰 데이터셋에서는 비효율적이고 계산 자원 소모가 큼 - 커널과 하이퍼파라미터 선택이 어려움 - 결과 해석이 어려움 - 커널 함수의 선택에 따라 성능 차이가 큼

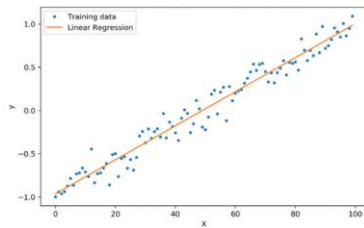


Figure 7. Linear Regression

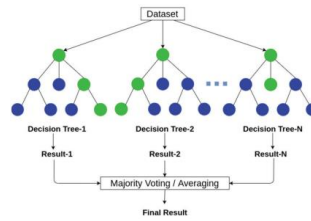


Figure 8. Random Forest

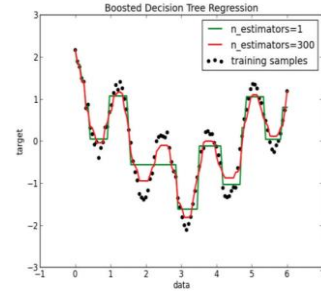


Figure 9. AdaBoost

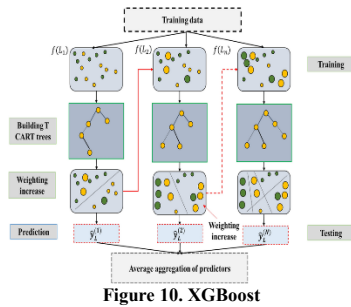


Figure 10. XGBoost

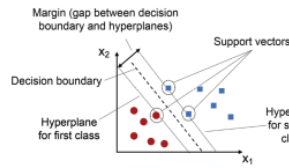


Figure 11. Support Vector Machine

KAMP Guidebook은 Linear Regression, Random Forest, AdaBoost 모델들을 후보군으로 설정하여 가장 ‘MAE’ 값이 낮은 Best Model을 선정하였다. 해당 프로젝트에서는 이 모델들 외에 두 가지 모델을 후보군에 추가하여 총 다섯 가지 모델들 중에서 Best Model을 찾고자 하였다. 추가된 모델인 XGBoost와 Support Vector Machine은 비선형 데이터 추정에 강점이 있다. 전류 데이터의 경우 위 Data Visualization에서 볼 수 있듯이 비선형 데이터인 것을 확인할 수 있다. 또한 기계의 전류를 측정하는 전류 센서는 전자기 간섭, 기계적 진동, 회로의 설계에 따라 노이즈가 검출될 가능성이 높다. 이러한 노이즈로 인해 과적합이 발생되어 정확한 예측이 이루어지지 못하는 것을 방지하기 위해 XGBoost와 Support Vector Machine이 지원하는 과적합 방지 메커니즘으로 전력 소모량을 정확하게 예측할 수 있다.

Model Comparison Results and Tunning

Table 2. Three Models Comparison

Best Model	Linear Regression	Random Forest	AdaBoost
Pump #1		0	
Pump #2		0	
Pump #3		0	
Pump #4		0	
Robot #1	0		
Robot #2	0		

Table 3. Five Models Comparison

Best Model	Linear Regression	Random Forest	AdaBoost	XGBoost	SVM
Pump #1					0
Pump #2					0
Pump #3					0
Pump #4					0
Robot #1				0	
Robot #2		0			

세 가지 모델을 비교하였을 때와 다섯 가지 모델을 비교하였을 때 결과는 위와 같다. 세 가지 모델만 비교하였을 때와 다르게 다섯 가지 모델을 비교하였을 때 기존 세 가지 모델보다 SVM 모델이 MAE 값이 가장 낮게 나와 좋은 성능을 보였다. 이에 두 모델군을 각각 사용했을 때 전력 예측 능력을 비교하고자 한다.

```
tuned_best_model = tune_model(best_model)
save_model(tuned_best_model, model_name = f'{amp}_best_model').
```

Figure 12. Tuning Model Code

Best Model로 선정된 각 모델은 `tune_model()` 함수를 통해 하이퍼 파라미터를 조정한다. 이 함수는 총 10번의 실험 점수가 있는 점수 그리드로 함수 옵션에 정의된 메트릭을 기반으로 최상의 모델을 선택한다.

III. Results (자원 최적화)

최적화 문제 설정

전류 감측량을 찾기 위해 아래와 같은 과정을 거쳤다.

① 가상의 생산계획 요구서를 작성

생산 가능 일수: 5일, 다섯 가지 item을 대상으로 임의로 생산량 할당

② 최적화 문제 변수 생성

```

item list : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
29, 30, 31, 32, 33, 34, 35]
qty list : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
92, 93, 94, 95, 96, 97, 98, 99]
machine list : ['amp_3', 'amp_2', 'amp_5', 'amp_1', 'amp_6', 'amp_4']
time list : [1, 2, 3, 4, 5]
```

Figure 13. 변수 생성 결과

I: 모든 아이템 이름 [36가지]

Q: 한번에 생산 가능한 양 [99개]

M: 생산 가능한 기계 [6가지]

T: 생산할 수 있는 날 [5일]

생산할 수 있는 모든 아이템에 대해 dictionary 형태로 저장

각 machine마다 하루 동안 사용할 수 있는 총 current에 대한 값을 할당 (80으로 설정)

③ Decision Variable 생성

```

1 model = mip.Model("KAMP_optimization")
2 xigt = {
3     (i, q, t): model.add_var(var_type=mip.BINARY, name=f"x_{i}_{str(q)}_{str(t)}" )
4     for i in I for q in Q for t in T
5 }
6 xigt
93ms 2024.06.14 16:30:47에 실행되었습니다
```

Figure 14. Decision Variable 생성 코드

`mip.Model()` 함수를 통해 model에 빈 MILP 문제 class를 생성한다. MILP (Mixed-Integer Linear Programming)란 최대화하거나 최소화하려는 선형 함수를 포함한 최적화 문제를 의미한다. 최적화 문제를 정의할 때 지정한 속성으로는 BINARY 특성이다 이는 0과 1로 결과를 도출하는데 그 이유는 특정 아이টে을 제작한 날은 1, 그렇지 않다면 0으로 표시하기 위해서이다.

④ 목적함수 생성

```
1 model.objective = mip.minimize(  
2     mip.xsum(ciqm[i, q, m]*xiqt[i, q, t] for i in I for q in Q for t in T for m in M)  
3 )  
283ms 2024.06.14 16:30:50에 실행되었습니다
```

Figure 15. 목적 함수 생성 코드

이번 프로젝트의 목표는 기계가 사용하는 총 current를 최소화하는 생산계획 최적화이다. 따라서 mip.minimize() 함수로 current를 최소화할 수 있다. 또한 mip.xsum() 함수를 사용한 이유는 그 날 기계들이 사용한 모든 전류량을 알아야하기 때문이다. 이에 생산 일자(i)에 생산량(q)을 만드는 기계의 수(m)을 각각 곱하고 합친다.

⑤ 제약조건 생성

해당 목적 함수에는 세 가지의 제약조건을 추가하여 생산 조건을 만족시켰다.

첫 번째로 생산가능한 날짜 T안에 만들어지는 모든 양을 합했을 때 생산 목표량 이상을 만들도록 제약하였다.

두 번째로 하루 생산에 사용되는 모든 기계는 할당된 전류 80을 넘지 못하도록 제한하였다.

세 번째로는 생산 가능한 날짜 T에 생산할 수 있는 생산량이 5개, 3개, 1개와 같이 복수의 생산량만큼 만들 수 없도록 제약하였다.

⑥ 최적화

```
In 114 1 model.optimize(max_seconds=30)  
4s 546ms 2024.06.14 16:30:57에 실행되었습니다  
  
Out 114 <OptimizationStatus.OPTIMAL: 0>
```

Figure 16. 최적화 코드 및 결과

마지막으로 최적화를 통해 모델에 정의된 목적 함수와 제약 조건을 기반으로 최적화 문제를 해결한다. 이 과정에서 가능한 최적의 해를 찾을 수 있도록 변수들을 조정한다. 이 후 해당 모델이 최적 모델을 찾았는지에 대해 결과를 도출한다. (0: 최적해를 찾음, 1: 최적해를 찾지 못함)

결과 확인

```
print(f'objective value : current {model.objective_value}')
```

Figure 17. 목적 함수 결과 코드

위 코드를 통해 목적함수 결과를 알아보았다. 3가지 모델을 사용했을 때 목적함수 값은 112[A]이고 5가지 모델을 사용했을 때 목적함수 값은 77[A]이었다. 이는 주어진 5일동안 생산 요구서에 대해 자원 최적화를 고려한 최소 Current를 의미하며 작을수록 생산 시 소모되는 전력이 줄어들었음을 의미한다.

다음으로는 3가지 모델을 사용했을 때와 5가지 모델을 사용했을 때 5일간의 itemno 다섯 가지에 따른 생산량을 그래프로 표시하였다.

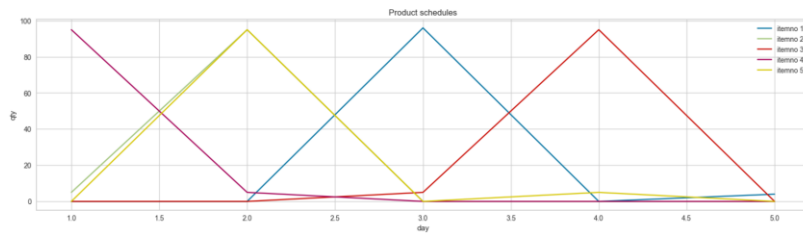


Figure 18. 3가지 모델군을 후보로 추정하였을 때 품목(itemno)별 5일간 생산량

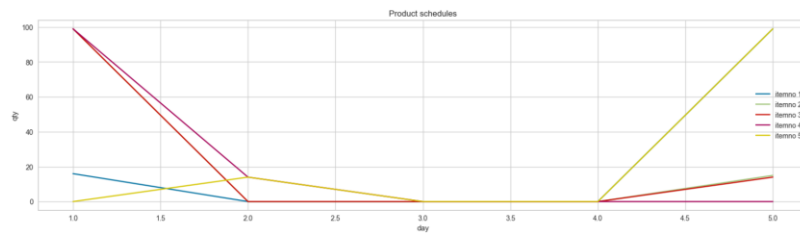


Figure 19. 5가지 모델군을 후보로 추정하였을 때 품목(item_no)별 5일간 생산량

마지막으로 최적화의 성능을 검증하기 위해 가상의 시나리오와 최적화를 한 시나리오를 비교하였다. 비교 시나리오는 위에서 Tunning을 마친 모델들로 각 기계가 100개의 생산요구량을 매일 균일하게 20개씩 생산했을 때이다. 이에 3가지 모델군으로 비교를 하였을 때와 5가지 모델군을 비교하였을 때 결과는 아래 그림과 같다.

최적화 하지 않은 생산 계획의 전류량: 276.32 [A]
 자원 최적화 생산 계획의 전류량: 146.53 [A]
 전류 감축량: 129.79 [A]
 전류 감축비율: 47.0%

Figure 20. 3가지 모델군을 후보로 추정하였을 때

최적화 하지 않은 생산 계획의 전류량: 196.61 [A]
 자원 최적화 생산 계획의 전류량: 77.22 [A]
 전류 감축량: 119.38 [A]
 전류 감축비율: 61.0%

Figure 21. 5가지 모델군을 후보로 추정하였을 때

3가지 모델군을 후보로 추정하였을 때는 생산 계획의 전류량이 146.53[A]가 도출되었고 5가지 모델군을 후보로 추정하였을 때 자원 최적화 생산 계획의 전류량은 77.22[A]로 SVM과 XGBoost 모델을 사용하여 추정하였을 때 더 높은 전류 최적화 성능을 보였다.

IV. Discussion

Table 2와 Table 3를 보면 알 수 있듯이 다섯 가지 모델군을 후보로 모델의 성능을 비교한 결과 세 가지 모델군 후보 중 유일하게 Random Forest만 프레스 4호 데이터에 좋은 성능을 보이고 나머지는 SVM 모델 또는 XGBoost 모델에서 좋은 성능을 보였다. 대부분 SVM 모델에서 좋은 성능을 보였는데 이러한 결과를 나타낼 수 있었던 이유는 SVM 모델의 비선형 데이터 분석 능력과 과적합 방지 메커니즘에 있다고 생각한다. 데이터셋을 분석하면 기계를 가동했을 때 사용한 RMS값을 알 수 있다. 해당 데이터는 전류 센서를 통해 얻은 데이터로 외부 노이즈에 의해 데이터 상에서도 비선형 구간이 많이 발생했음을 볼 수 있다. 이러한 노이즈는 다른 모델을 사용할 경우 학습 과정에서 과적합을 발생시킬 수 있다. SVM의 과적합 방지 메커니즘은 아래의 그림으로 설명할 수 있다.

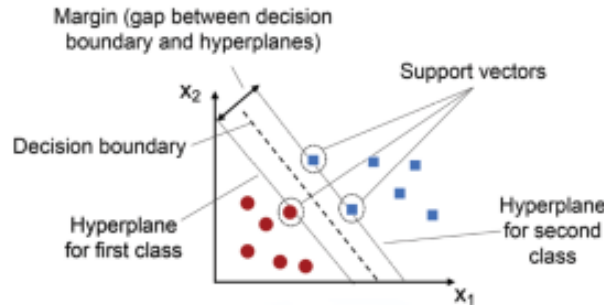


Figure 22. Support Vector Machine

SVM은 두 개의 클래스 간의 마진을 최대화하는 hyperplane을 찾는 알고리즘이다. 이 평면은 가장 가까운 데이터 포인트인 Support Vector끼리의 거리를 최대화하도록 한다. 넓은 마진은 학습 데이터의 작은 변동에도 덜 민감하게 반응되어 과적합을 방지한다.

또한 정규화(Regularization)과정을 통해 과적합을 방지한다. 정규화 과정은 아래 수식과 함께 설명할 수 있다.

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i$$

여기서 $\|\omega\|$ 는 마진의 너비를, ξ_i 는 슬랙 변수로 허용 가능한 오류를 나타낸다. 슬랙 변수란 선형적으로 분리할 수 없는 데이터셋을 처리하기 위한 개념이다. 위에서 그림과 같이 데이터셋을 분리할 때 예외적으로 일부 데이터 포인트가 경계를 넘는 것을 허용하여 유연한 모델을 구축할 수 있게 한다. C 는 모델의 복잡도를 조절한다. C 가 크면 오류를 줄이는데 중점을 두고, C 가 작으면 마진을 넓히는데 중점을 둔다.

이 두 가지 방법 외에도 교차 검증과 커널 트릭은 과적합을 방지하여 전류 데이터셋에서 좋은 결과를 기대할 수 있다.

V. Conclusion

본 연구에서는 프레스 기계의 전력 소모량을 예측하기 위해 다양한 머신러닝 모델을 비교하고 최적의 모델을 선정하는 과정을 수행하였다. 국내 전체 산업의 전기 사용량 중 약 70~80%가 공장에서 소비되고 있으며, 이는 대부분 모터 구동에 쓰인다. 이러한 맥락에서 본 연구는 전력 소모 예측과 최적화를 통해 에너지 효율성을 높이는 데 기여하고자 하였다.

세 가지 모델만을 비교했을 때와 달리 다섯 가지 모델을 비교했을 때, SVM 모델이 MAE 값이 가장 낮게 나와 최적의 성능을 보였다. 이는 SVM 모델이 비선형 데이터와 노이즈 처리에서 강점을 보였기 때문이다. SVM의 과적합 방지 메커니즘과 커널 트릭을 통해 고차원 특성을 효과적으로 다룰 수 있었으며, 슬랙 변수 도입으로 데이터의 불확실성을 허용하면서도 최적의 분류 경계를 찾을 수 있었다.

최적화 문제에서는 자원 최적화를 고려하여 전류 소모량을 최소화하는 생산 계획을 수립하였고, 이를 통해 전력 소모량을 크게 줄일 수 있음을 확인하였다. 최적화하지 않은 생산 계획과 비교했을 때, 최적화된 생산 계획은 전류 소모량을 효과적으로 줄였으며, 이는 생산 과정에서의 비용 절감으로 이어질 수 있다.

본 연구 결과는 프레스 기계의 전력 소모 예측과 최적화를 위한 머신러닝 모델 선택에 있어 중요한 참고자료가 될 수 있으며, 이와 비슷한 연속 공정을 사용하는 제조 현장에도 적용 가능하다. 특히 전류 데이터뿐만 아니라 다양한 원재료 데이터로 대체할 수 있어, 생산 시간, 노동력 비용, 원자재 비용, 원자재 종류 등 다양한 변수들을 고려한 예측 모델을 구축할 수 있다. 또한, 실제 데이터 값을 활용하여 현실성 있는 문제를 구축하고, 각 기계별 사용할 수 있는 전류 최대량과 같은 제약조건을 추가함으로써 실제 산업 현장에서 필요한 제약조건을 반영할 수 있다.

결론적으로, 본 연구는 프레스 기계의 전력 소모 예측과 최적화를 통해 에너지 효율성을 높이는 방법을 제시하였으며, 이는 향후 다양한 산업 분야에서 에너지 절감 및 비용 절감에 기여할 수 있을 것으로 기대된다.

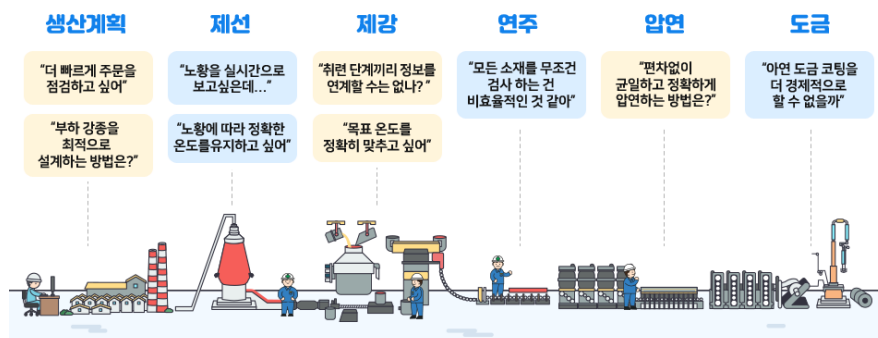


Figure 23. 포스코 스마트 제철소의 연속 공정