

2022

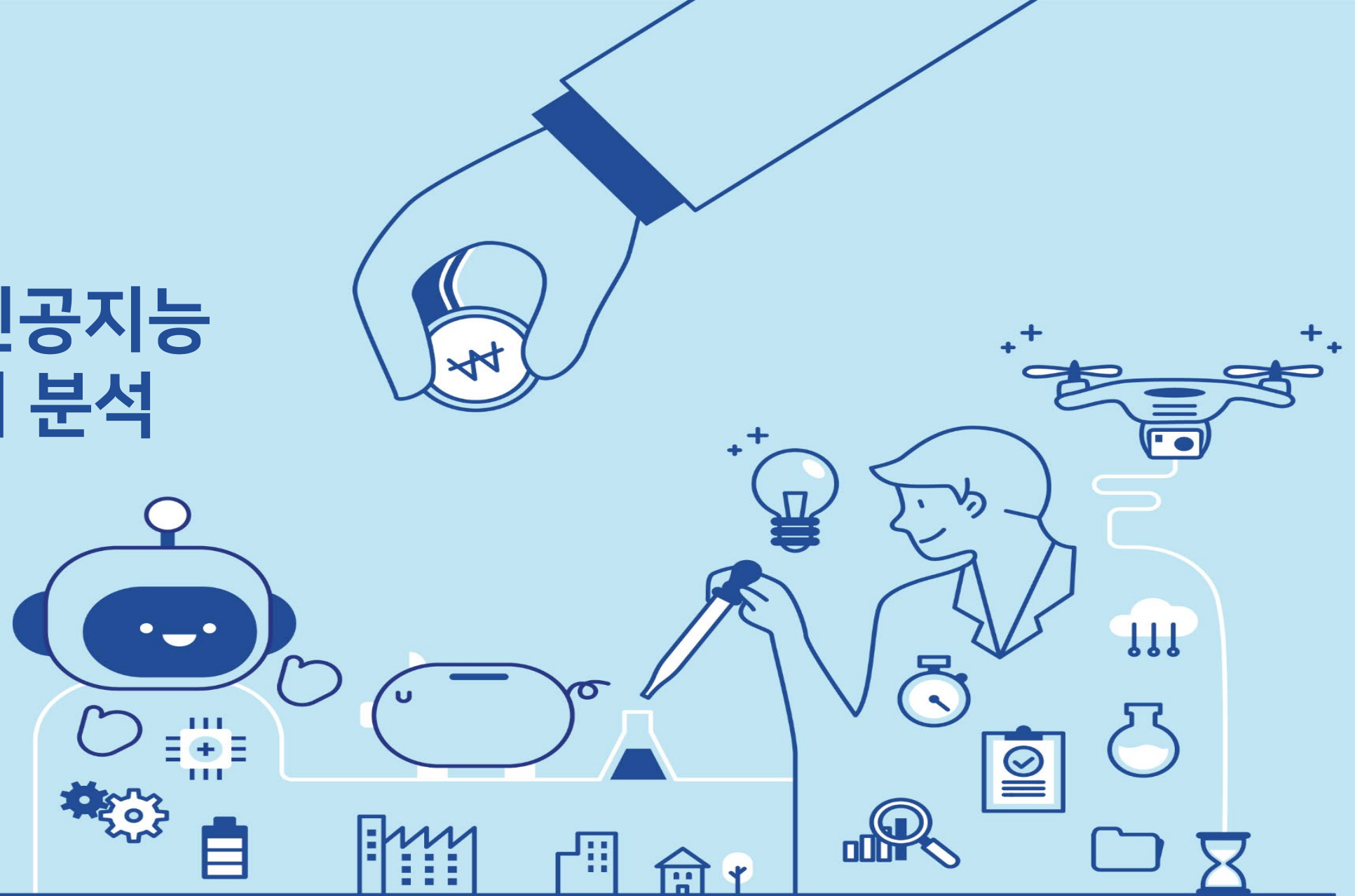
제2회 K-인공지능 제조데이터 분석 경진대회

팀명 : 마지막 불씨

팀장 : 서현범

팀원 : 부도현

팀원 : 박지환



원료의 특성을 고려한 설명가능한 용해 공정 불량 예측 최적화 모델

목차



분석 목적



EDA



분석모델 개발



분석 결과



분석모델 확장성

01

분석 목적

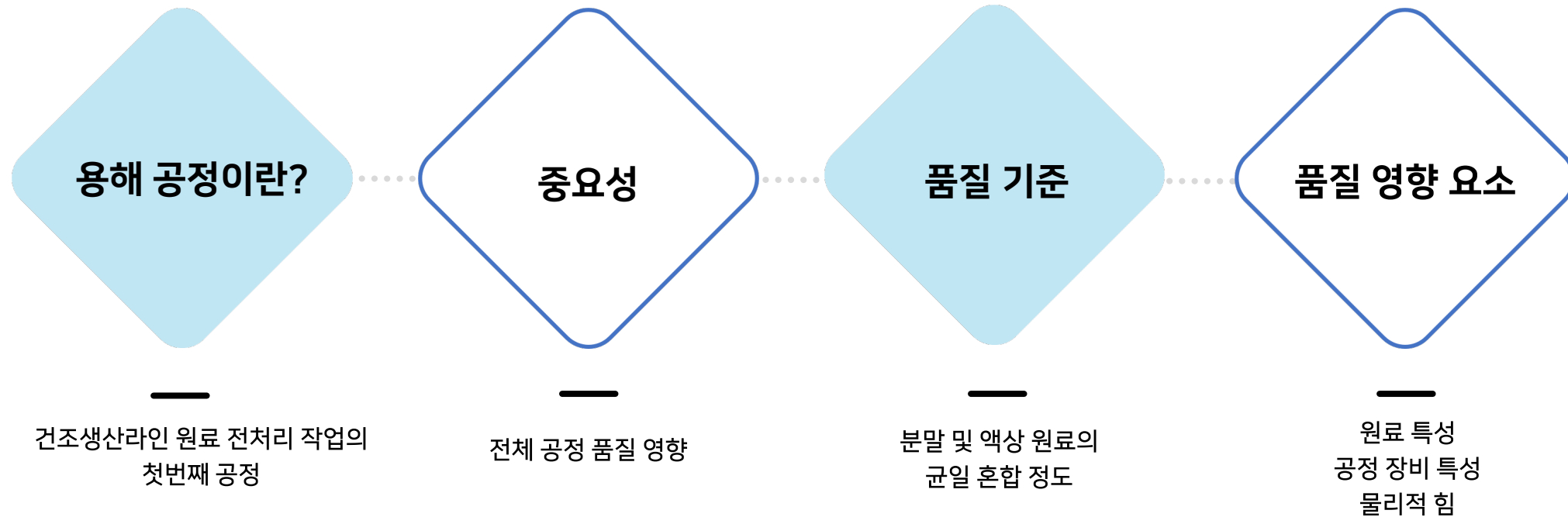
- Target
- Problem
- Goal





- Target

식품제조업의 용해 공정



- Problem

투입 원료의 다양성
인력 부족



- Goal

투입 원료의 특성 반영한 수율 예측
특성들이 품질 지표에 미치는 영향력 수치화



EDA

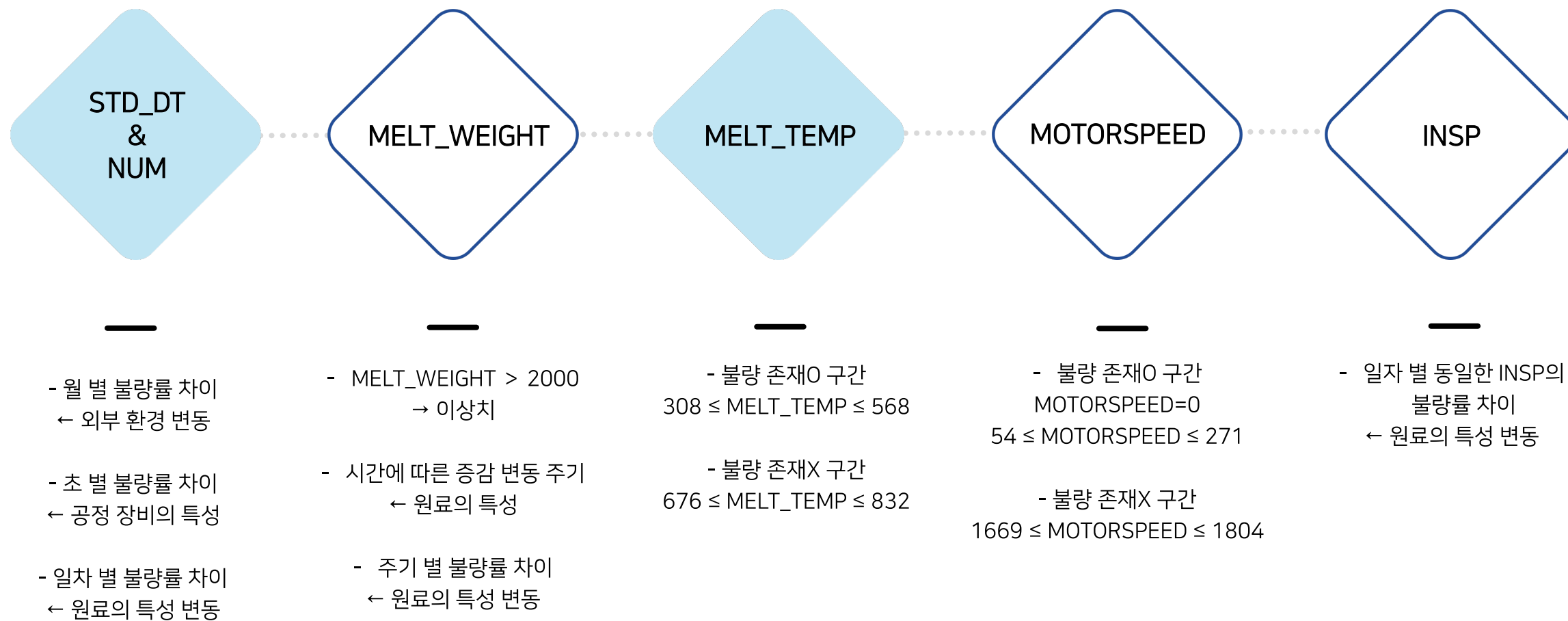
- EDA
- Data Preprocessing





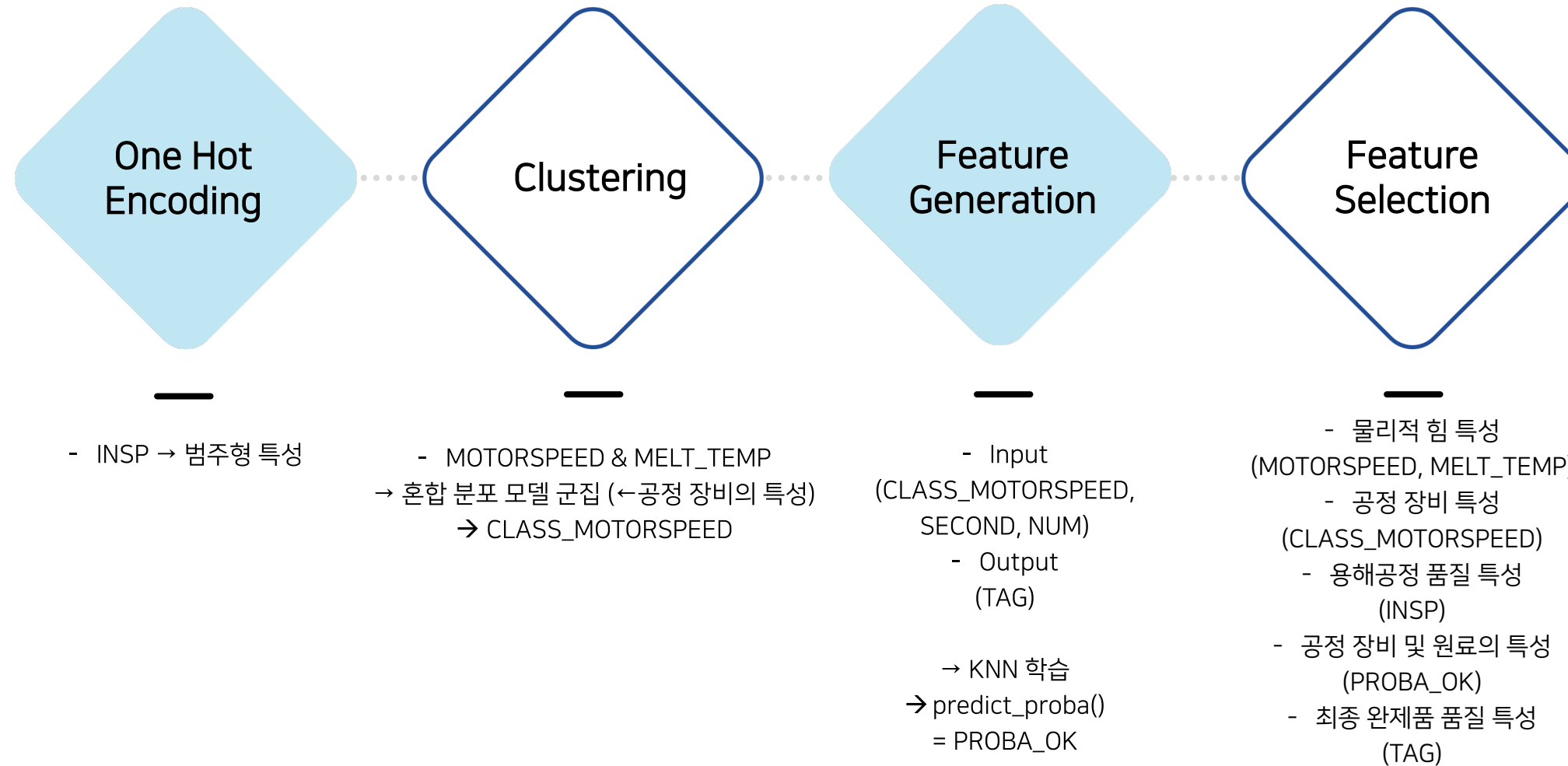
- EDA

경진대회용 용해탱크 데이터셋.csv





- Data Preprocessing



03

분석모델 개발

- AutoML
- Bayesian Optimization
- XAI
- Data Split
- Model Architecture





- AutoML

Feature → Pycaret AutoML → Model selection

```
best_models = compare_models(sort="Accuracy", n_select=3)
```

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|----------|---------------------------------|----------|--------|--------|--------|--------|--------|--------|----------|
| xgboost | Extreme Gradient Boosting | 0.8413 | 0.9018 | 0.9452 | 0.8657 | 0.9037 | 0.4580 | 0.4727 | 11.8120 |
| lightgbm | Light Gradient Boosting Machine | 0.8389 | 0.8986 | 0.9554 | 0.8567 | 0.9033 | 0.4292 | 0.4533 | 0.5140 |
| rf | Random Forest Classifier | 0.8269 | 0.8856 | 0.9097 | 0.8755 | 0.8922 | 0.4531 | 0.4554 | 7.3580 |
| gbc | Gradient Boosting Classifier | 0.8208 | 0.8725 | 0.9638 | 0.8344 | 0.8944 | 0.3212 | 0.3626 | 10.8710 |
| et | Extra Trees Classifier | 0.8138 | 0.8590 | 0.8959 | 0.8714 | 0.8835 | 0.4209 | 0.4220 | 9.2690 |

→ XGBoost Model

- Bayesian Optimization

Feature → XGBoost → Bayesian Optimization → Hyper-parameter optimization

```
{'target': 0.988855962643678, 'params': {'colsample': 0.885030598969056, 'gamma': 16.63782564380123, 'l1_reg': 566.0874092148872, 'l2_reg': 273.793769867424, 'lr': 0.05740236271365283, 'max_depth': 9.736006499859627, 'min_child_weight': 5.986727556541564, 'n_trees': 746.5351321102312, 'subsample': 0.8861015002631764}}}
```

- XAI

- XGBoost → get_booster().get_score() → Feature Weight → Feature Importance
- SHAP → Feature Shapley Value

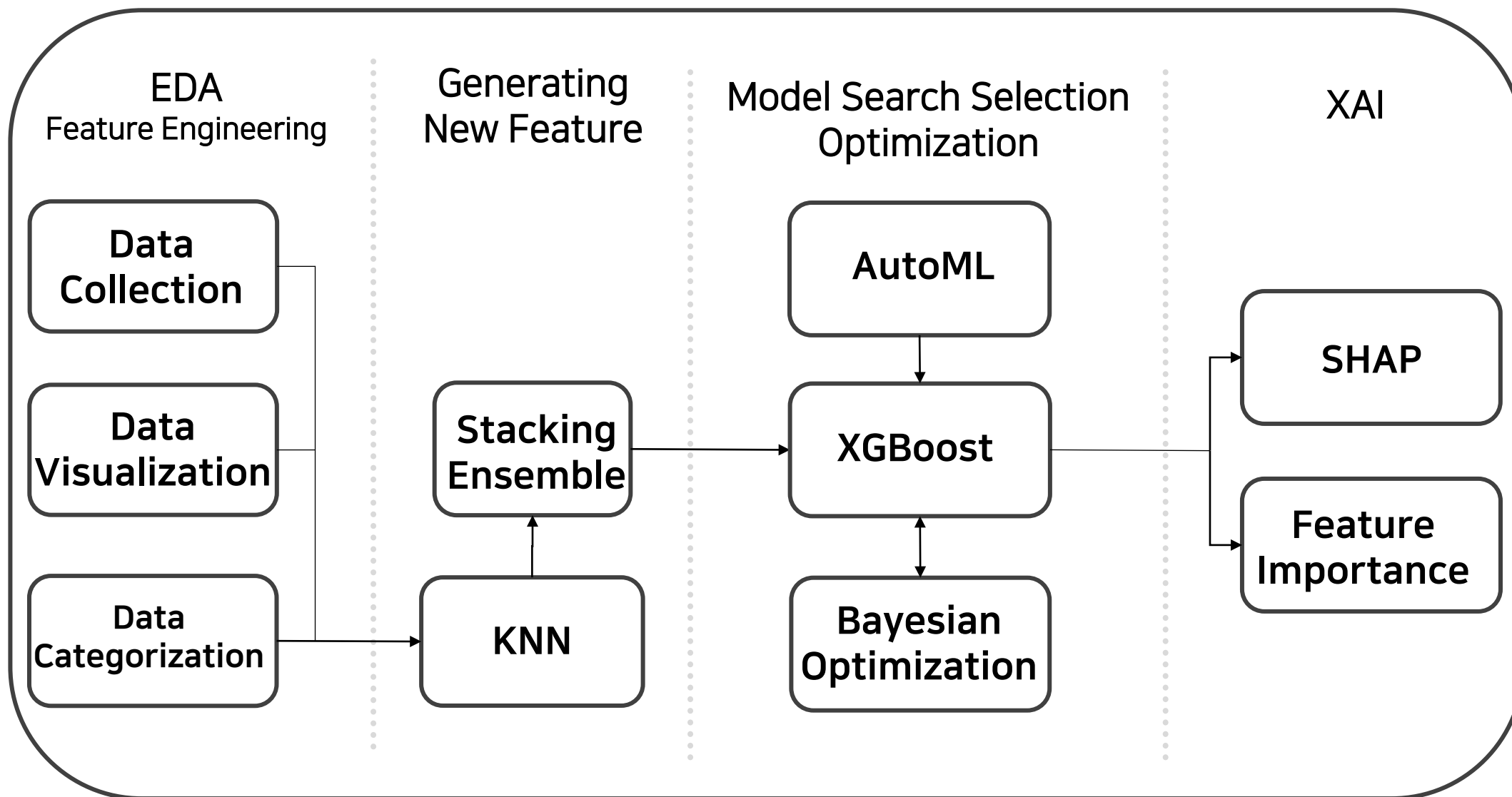


- Data Split

| | | |
|-------------------------|-----|--------------------------|
| KNN_Train_Data | 40% | Train KNN |
| KNN_Validation_Data | 10% | Optimize KNN |
| XGBoost_train_Data | 32% | Train XGBoost |
| XGBoost_Validation_Data | 8% | Optimize XGBoost |
| Test_Data | 10% | Architecture Performance |



- Model Architecture



04

분석 결과

- Architecture Performance
- Feature Importance
- SHAP





Architecture Performance

XGBoost Train Data

```
from sklearn.metrics import classification_report
y_pred_train = model.predict(X_train)
print(classification_report(y_train, y_pred_train))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.99 | 0.99 | 0.99 | 56662 |
| 1 | 1.00 | 1.00 | 1.00 | 210602 |
| accuracy | | | 1.00 | 267264 |
| macro avg | 0.99 | 1.00 | 1.00 | 267264 |
| weighted avg | 1.00 | 1.00 | 1.00 | 267264 |

XGBoost Validation Data

```
y_pred_val = model.predict(X_val)
print(classification_report(y_val, y_pred_val))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.98 | 0.98 | 0.98 | 14165 |
| 1 | 0.99 | 0.99 | 0.99 | 52651 |
| accuracy | | | 0.99 | 66816 |
| macro avg | 0.99 | 0.99 | 0.99 | 66816 |
| weighted avg | 0.99 | 0.99 | 0.99 | 66816 |

Test Data

```
y_pred_test = model.predict(test_x)
print(classification_report(test_y, y_pred_test))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.98 | 0.98 | 0.98 | 17707 |
| 1 | 0.99 | 0.99 | 0.99 | 65813 |
| accuracy | | | 0.99 | 83520 |
| macro avg | 0.99 | 0.99 | 0.99 | 83520 |
| weighted avg | 0.99 | 0.99 | 0.99 | 83520 |

Performance Comparison

KNN-XGBoost model

```
precision: 0.9948
recall: 0.9944
f1-score: 0.9946
accuracy: 0.9915
```

KNN-DNN model

```
precision: 0.9677
recall: 0.9902
f1-score: 0.9788
accuracy: 0.9662
```

Feature → AutoKeras → DNN Selection

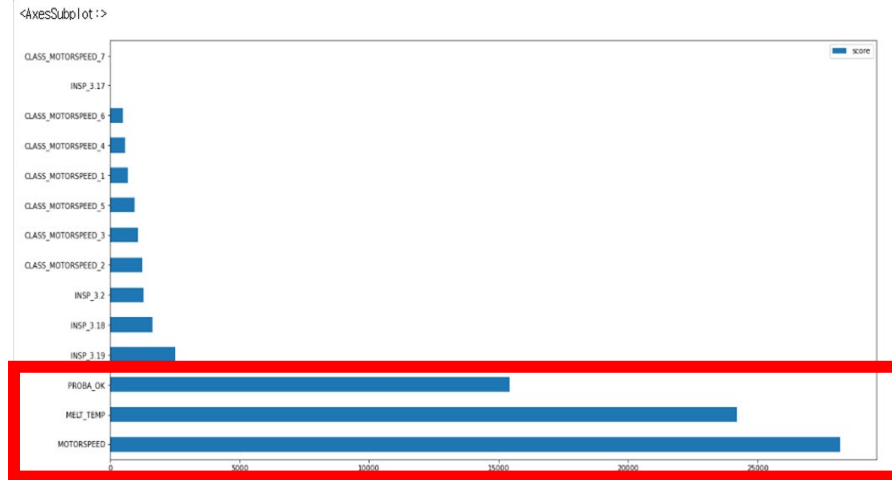
LSTM model

```
precision: 0.9961
recall: 0.8063
f1-score: 0.8912
accuracy: 0.8055
```

GuideBook



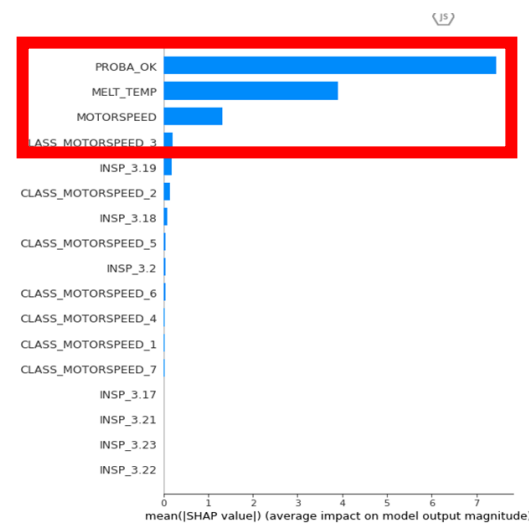
- Feature Importance



최종 품질에 대한 영향력



- SHAP



최종 품질에 대한 영향력 + 선형 관계성



05

분석모델 확장성

- 용해 공정에서의 발전 가능성
- 타 공정으로의 확장 가능성





- 용해 공정에서의 발전 가능성

수율 예측
정확도 향상

원료 조합 Labeling
→ 원료 조합별 PROBA_OK
→ 원료 교체 시, 초기값 설정
→ 초기 수율 예측 정확도 ↑

품질 개선

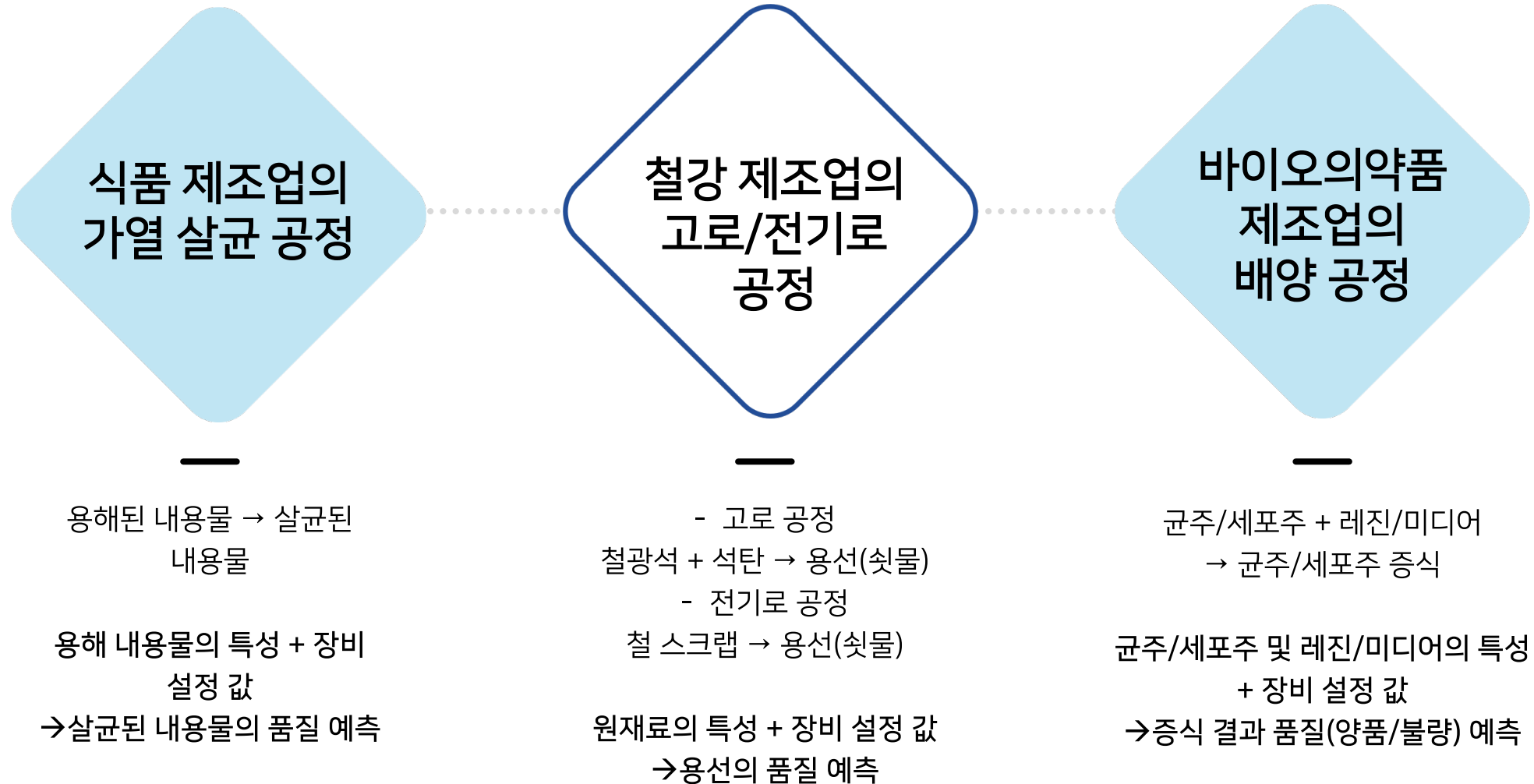
Feature Importance & Shapley Value
→ 용해온도 & 모터속도 기준 실험 계획
→ 용해온도 & 모터속도 최적 구간 도출
→ 품질 개선

공정 최적화

장비 설정 값 및 불량에 따른 비용
최소화
→ 공정 최적화



- 타 공정으로의 확장 가능성



Thank You
감사합니다

