스마트 공장 제품 품질 상태 분류 방안

우리 Gram 써요

백지수, 가상은, 서유진, 오정민, 이경원

모델 검증

코드

```
# Test 데이터에 대한 분류 전, Train 데이터에 대해 학습이 잘 되었는지 확인
print(VLD.score(train_x*train_x, train_y))
print('Done.')
```

모델 검증을 위해 sklearn에서 제공하는 score 함수를 사용해 검증하였다.

데이터 전처리

최빈값-결측값을 최빈값으로 대체(mode 함수)

- 하나의 행이 모두 NaN 값일 경우, 최빈값 대체 어렵다.

평균값-각 열의 평균값으로 대체(mean 함수)

- train 데이터의 X_1 열을 보면 2,48 같이 데이터의 값 차이가 클 경우 , 평균값으로 결측값을 대체하는 방법은 바람직하지 않다.

중앙값-각 열의 중앙값으로 대체(median 함수)

- 열의 값 들간의 차이가 클 때, 결측값을 채울 값으로 바람직하지 않다.

임의의 숫자-결측값을 임의의 숫자로 대체

- 결측값을 임의의 숫자로 채워넣는다.

```
train_x = train_x.fillna(-1) #빈 값을 -1값으로 처리
test_x = test_x.fillna(-1) #빈 값을 -1값으로 처리
```

➡ **각 경우들로 데이터를 학습시켰을 때, 임의의 숫자로 결측값을 채운 경우가 정확도가 가장 높았기에 임의의 숫자 -1을 결측값에 채워넣었다.**

상관관계 분석

- Train, Test시 입력에 해당하는 데이터들을 제곱하여 명확하게 파악할 수 있도록 하였다.

- Train -

```
모델.fit(train_x*train_x, train_y)
모델.score(train_x*train_x, train_y)
```

- Test -

```
preds= 모델.predict(test_x*test_x)
```

동작 속도

VotingClassifier의 **n_jobs** 매개변수

- 모델 학습시 사용할 CPU 코어의 수를 지정하는 매개변수
- 이 매개변수의 설정에 따라 모델 학습 속도가 바뀐다.

n_jobs=1 인 경우

- 하나의 CPU 코어만을 사용하여 모델 학습을 수행한다

➡ 하나의 CPU 코어를 사용하므로 모델 학습 속도가 느리다

n_jobs=-1 인 경우

- 설정 가능한 모든 CPU 코어를 사용하여 모델 학습을 수행한다

➡ 가능한 최대로 CPU 코어를 사용함으로써 모델 학습 속도를 향상시킨다

현업 적용 가능성

RandomForestClassifier, GradientBoostingClassifier, KNeighborsClassifier, AdaBoostClassifier 모델을 기반으로 학습을 진행하였다.

이후 앙상블 기법인 VotingClassifier를 이용하여 각 분류 모델의 확률값을 통한 최종 결과값을 예측하며, 한 분류 모델의 결과에만 치우치지 않은 적절한 결과값을 출력한다.

➡️ **이는 효율적인 코어 사용을 통해 학습 시간을 단축하고 다양한 학습 결과를 고려하는 앙상블 기법을 이용한 빠르고 정확한 모델로써 현업에 적용 가능하다.**