

# 生産管理 最終レポート

学籍番号:22D7103022I

氏名:小泉 日南太

## 1. 目的

本課題では、VM 工程における品質指標である OV を予測する回帰モデルを構築し、末尾 500 レコードを評価区間として、予測精度を RMSE により評価する。あわせて実務上の監視・保守コストを想定し、説明変数 X1～X83 のうち、利用する変数数を可能な限り削減することを目的とする。

また未来情報リークを防ぐため、先頭ロットの process\_end\_time (2016-11-28 07:26) を境界時刻 cutoff と定義し、学習には final\_mes\_time < cutoff を満たすデータのみを使用する。説明変数に欠損はないため、欠損補完は行わない。

最終モデルは StandardScaler による標準化の後に Ridge 回帰を適用し、予測値は不良品数として負値が不適切なため  $\hat{y} = \max(\hat{y}, 0)$  により 0 下限クリップを行う。特徴量は時間項( $t, t^2$ )と 4 変数(X32, X36, X27, X83)を用いる(定義は 4.2 節)。

ここで  $t$  は process\_end\_time を 2000-01-01 からの経過日数に変換した実数値、 $t^2$  はその二乗であり、相互作用項は導入しない。以上により、少数変数による運用性と汎化性能の両立を狙った。

## 2. データ利用方針と評価設定

### 2.1 学習データと評価データの分離

評価区間は末尾 500 レコードとした。未来情報リーク防止のため、評価区間先頭ロットの process\_end\_time (2016-11-28 07:26) を境界時刻 cutoff とし、学習データは final\_mes\_time < cutoff を満たすレコードに限定した(final\_mes\_time は計測完了時刻)。これにより、予測時点以後に得られる情報が学習に混入しない条件で評価できる。

- 学習データ条件: final\_mes\_time < cutoff
- cutoff: 2016-11-28 07:26 (評価区間先頭ロットの process\_end\_time)

ここで final\_mes\_time は各ロットの計測完了時刻であり、この条件は予測時点より後に計測が完了するデータを学習から排除することを意味する。したがって、評価対象ロットの品質を予測する時点では本来観測できない情報が学習に混入しないようにし、時系列予測としてリークが起こらないように分離を行った。

### 2.2 学習データの妥当性確認

学習開始時に、学習データに含まれる final\_mes\_time の最大値と境界時刻を比較し、学習データが観測可能範囲に収まっていることを確認した。確認結果を表 1 に示す。

表 1:学習データ時刻の確認結果

| 項目        | 時刻               |
|-----------|------------------|
| 評価開始境界    | 2016-11-28 07:26 |
| 学習データ内の最大 | 2016-11-27 20:32 |
| 余裕時間      | 0 日 10 時間 54 分   |

以上より、本実験において学習データと評価データは時間的に明確に分離されている。

### 2.3 該当コード

```
def apply_dataguard(train_df: pd.DataFrame, cutoff_time: pd.Timestamp) -> dict:
    if len(train_df) == 0:
        return {"result": "EMPTY", "margin": None}
    max_mes_time = train_df["final_mes_time"].max()
    # マージンの計算
    margin = cutoff_time - max_mes_time

    # ここで厳密に停止させる
    assert max_mes_time < cutoff_time, f"CRITICAL LEAKAGE: Found measure time {max_mes_time} >= cutoff {cutoff_time}"

    return {"result": "PASSED", "margin": margin}
```

## 3. 結果概要

最終的に採用したモデル(Final)の性能を、明示的に定義したベースライン(Baseline-0/1)と比較して表 2 に示す。Final は 4 変数(X32, X36, X27, X83)に限定しつつ、Baseline-1(全変数)より低い Test RMSE 59.50 を達成した。特に、最終モデルと同一の前処理・評価条件で全変数、つまり( $t, t^2 + X1 \sim X83$ )を用いる Baseline-1 と比較しても大きく改善しており、変数数削減が汎化性能の向上にも寄与した可能性が示唆される。

表 2:ベースラインと最終モデルの性能比較

| 指標                      | Baseline-0(授業サンプル) | Baseline-1(全変数 Ridge) | Final(4 変数 Ridge) |
|-------------------------|--------------------|-----------------------|-------------------|
| Test RMSE               | 97.17              | 136.62                | 59.50             |
| 改善率(vs Baseline-1)      | -                  | -                     | 56.4%             |
| 使用説明変数数                 | 83                 | 83                    | 4                 |
| Rolling CV 平均 RMSE      | -                  | -                     | 51.89             |
| Rolling CV 最終 Fold RMSE | -                  | -                     | 31.14             |

(注) 時間項  $t, t^2$  は全モデル共通のため、表 2 の変数数は X のみを数える。

## 4. 最終モデルの設計

### 4.1 モデル設定

今回は Ridge 回帰を最終モデルとして採用した。モデル設定を表 3 に示す。正則化係数 alpha は複数候補を Rolling TimeSplit による交差検証で比較し、平均 RMSE が最小となる値を選定した。また、推論時は 0 下限クリップ ( $\hat{y} = \max(\hat{y}, 0)$ ) を適用した(理由は 1 章)。

表 3: 最終モデル設定

| 項目     | 内容                           |
|--------|------------------------------|
| モデル    | Ridge 回帰                     |
| 正則化係数  | alpha = 500.0                |
| 入力特徴量数 | 6                            |
| 予測値の補正 | $\hat{y} = \max(\hat{y}, 0)$ |

### 4.2 特徴量設計

最終モデルで使用した入力特徴量の内訳を表 4 に示す。時間項として  $t$ ,  $t^2$  を用い、説明変数は X32, X36, X27, X83 の 4 変数に限定した。ここで  $t$  は process\_end\_time を 2000-01-01 からの経過日数に変換した実数値であり、 $t^2$  はその二乗である。

また、時間項と説明変数の交互作用を含む構成も検討したが、評価区間において誤差が増加する傾向が確認されたため、最終モデルでは交互作用項を採用せず、相互作用なしの線形構造に単純化した。

表 4: 使用特徴量の内訳

| 種別    | 内容                 |
|-------|--------------------|
| 時間項   | $t, t^2$           |
| 説明変数  | X32, X36, X27, X83 |
| 交互作用項 | 使用しない              |

### 4.3 モデル係数の確認

StandardScaler により各特徴量を標準化した上で学習した Ridge 回帰の係数を表 5 に示す。標準化後の係数は、他の特徴量を一定としたときに、当該特徴量が 1 標準偏差増加した場合の予測 OV の変化量に対応する。係数の絶対値を見ると、本データでは時間項( $t, t^2$ )よりも、特定のプロセス変数(特に X32, X27, X83)が OV 予測に強く寄与している傾向が読み取れる。ただし、係数の大きさや符号の解釈は、対象データの分布・ドリフト・モデル構成に依存するため、一般化には注意が必要である。

表 5: 標準化係数

| 特徴量 | 係数     | 絶対値   |
|-----|--------|-------|
| X32 | -19.44 | 19.44 |
| X27 | 11.67  | 11.67 |
| X83 | 10.56  | 10.56 |
| X36 | 4.27   | 4.27  |

|       |       |      |
|-------|-------|------|
| t     | -3.86 | 3.86 |
| $t^2$ | -3.84 | 3.84 |

#### 4.4 該当コード

```
def build_matrix(df_slice, features, t_mode="abs", ref_time=None, interactions=False):
    # 時間項 t, t^2 の計算部分は省略

    # Stack: [t, t^2, X]
    if interactions:
        # t * X for each feature (Phase 1: Drift Trap)
        t_X = X_raw * t_vals
        X_out = np.hstack([t_vals, t_sq, X_raw, t_X])
    else:
        #単純な結合のみ。つまり相互作用なし
        X_out = np.hstack([t_vals, t_sq, X_raw])
        col_names = ["t", "t^2"] + features

    return X_out, col_names
```

## 5. 検証方法と検証結果

### 5.1 検証方法

内部検証には Rolling TimeSplit (5 分割) を用いた。各分割において、検証区間より前に観測されたデータのみを学習に使用し、時間順序を保った評価を行った。

### 5.2 検証結果

Fold ごとの RMSE と学習データ数を表 6 に示す。初期 Fold では学習データ数が少なく、推定が不安定になりやすい。一方、後半 Fold では学習データ数が増え、RMSE が安定する傾向が見られる。

表 6: Rolling TimeSplit による検証結果

| Fold | RMSE  | 学習データ数 | 検証開始日      |
|------|-------|--------|------------|
| 1    | 61.06 | 24     | 2016-09-29 |
| 2    | 86.38 | 90     | 2016-10-12 |
| 3    | 41.63 | 563    | 2016-11-06 |
| 4    | 39.23 | 719    | 2016-11-10 |
| 5    | 31.14 | 1080   | 2016-11-19 |

### 5.3 該当コード

```
# Rolling TimeSplit CV Loop
for fold_i, (train_idx, val_idx) in enumerate(tscv.split(full_train_candidates)):
```

```
# データ分割

# Validation Start Point acts as the "Reference Time" (Cutoff) for this fold
val_start_time = val_fold["process_end_time"].iloc[0]

# 学習データも検証データ開始時刻より前のものに限定
mask = tr_fold["final_mes_time"] < val_start_time
tr_filtered = tr_fold[mask]
```

## 6. 評価区間にに対する性能確認

本節では、評価区間 RMSE という数値指標に加えて、予測が極端に破綻していないかを確認するために可視化を用いる。各図は、確認したい観点ごとに目的を分けて配置する。

### 6.1 時系列上の追随性

ここでは評価区間において、予測値が実測値の大まかな変動傾向を捉えられているか、また予測が特定方向に外れ続けて破綻していないかを確認する。図 1 に、評価区間における OV の実測値と予測値の時系列比較、および残差の推移を示す。図 1 より、予測値は実測値の水準変化や増減の方向性を概ね捉えている一方で、実測値に見られる急峻なスパイクについては十分に追随できていない。残差については、時間の経過とともに一方向へ拡大し続けるような明確な傾向は確認されず、評価区間を通じて予測が特定方向に大きく外れ続ける挙動は限定的である。

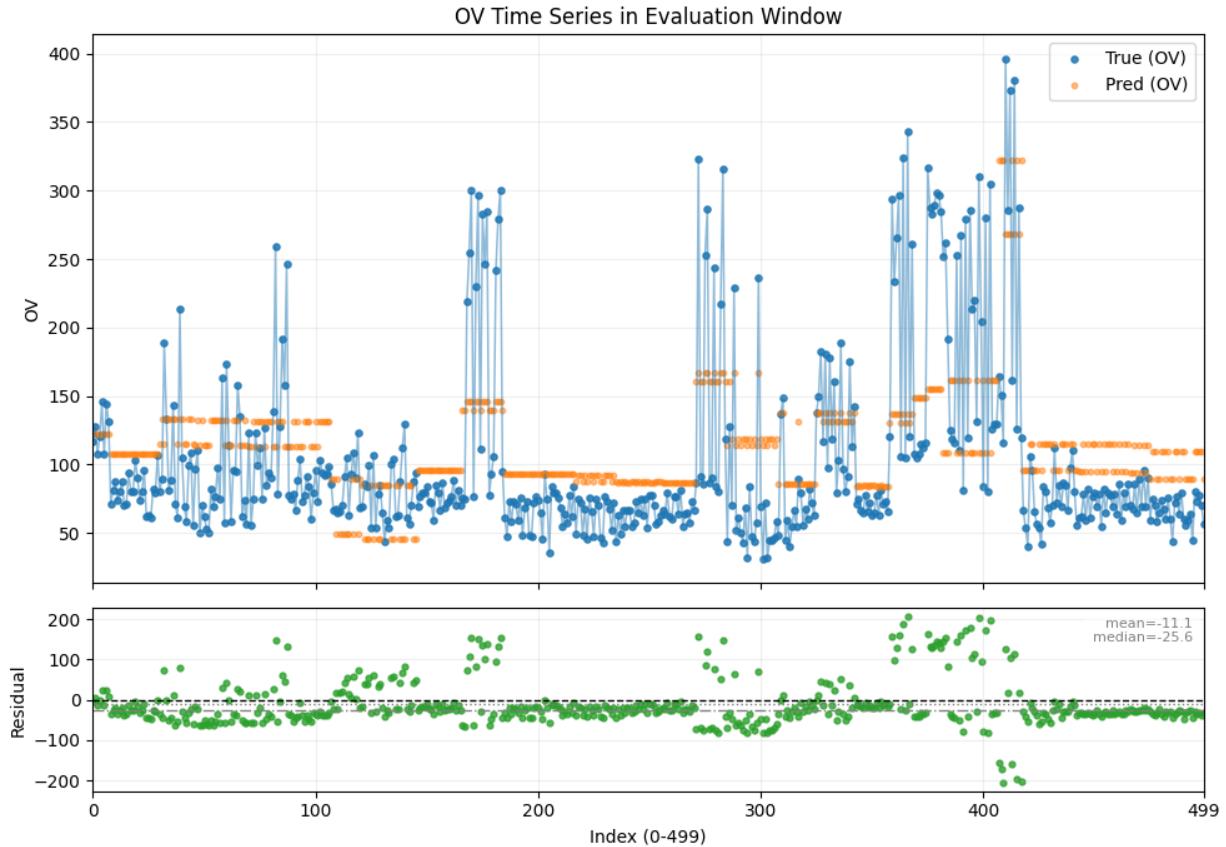


図 1: OV の実測値と予測値の時系列比較、および残差の推移

## 6.2 予測値と実測値の対応関係

ここでは評価区間における予測値と実測値の対応を確認するため、Pred vs True の散布図を用いる。散布図では、点が  $y=x$  付近に集まるほど予測と実測の一一致が良い。また、点が理想線より下側に偏る場合は過小予測、上側に偏る場合は過大予測を示す。

以下の図 2 を見ると、特に OV が大きい領域で点が理想線の下側に多く分布しており、高 OV を系統的に過小予測する傾向が見られる。これは線形モデルでは高 OV 側の上振れを表現しきれない可能性を示唆しており、今後は高 OV 領域の扱い、つまり重み付けや非線形性の導入等での改善余地がある。

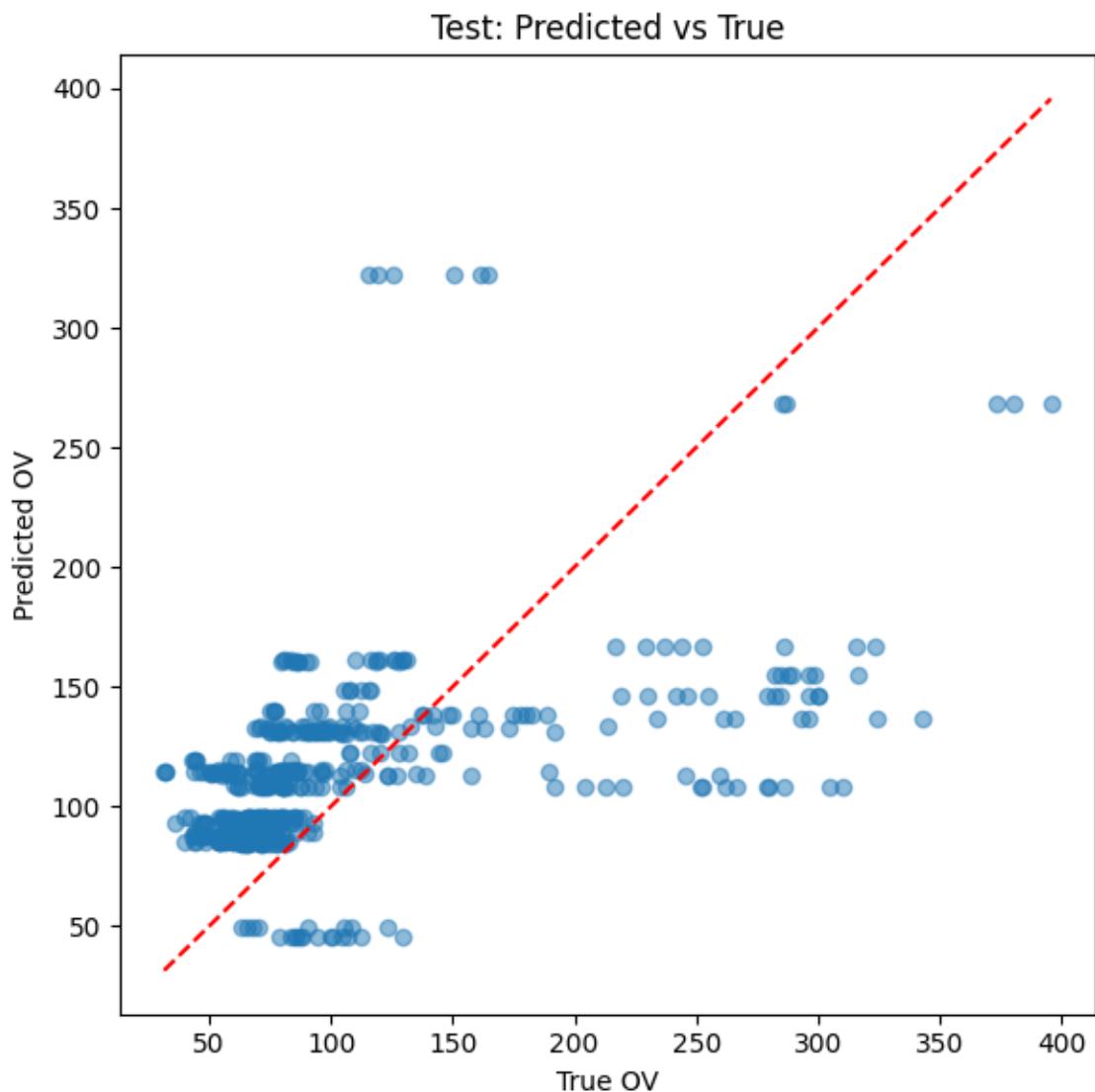


図 2:評価区間における予測値と実測値の散布図

### 6.3 残差の時間的偏り

目的は、残差が特定の符号に偏り続けるなど、時間経過に伴う系統誤差が疑われる挙動がないかを確認することである。残差が正負の両側に分布し続ける場合、少なくとも目視の範囲では一方向の偏りが強いとは断定しにくい。

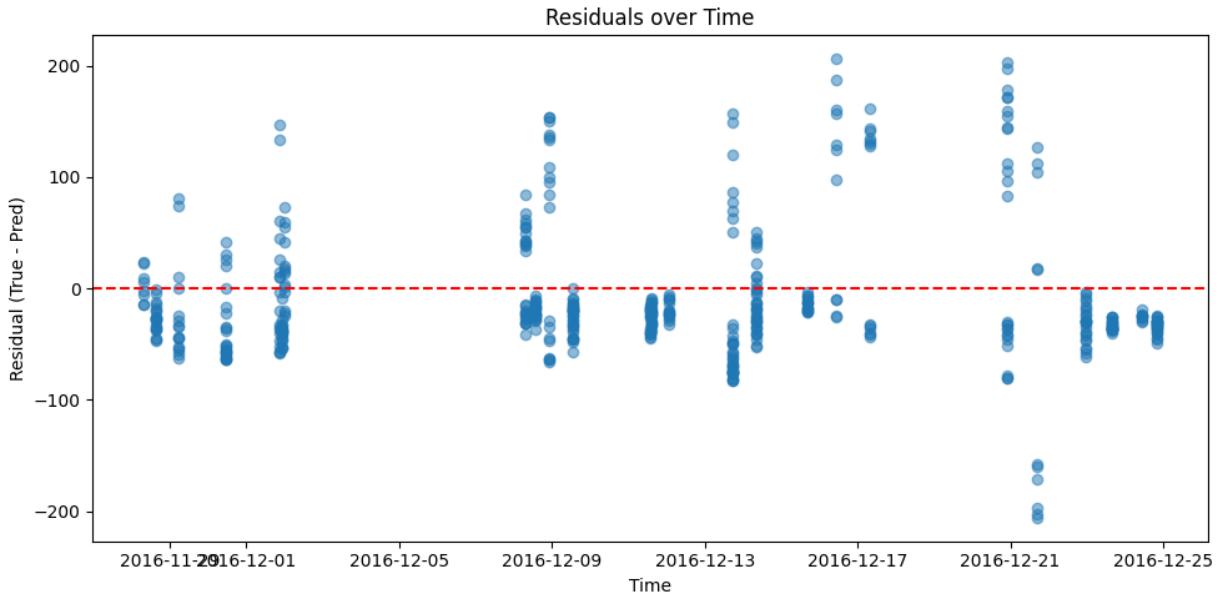


図 3: 残差の時系列、評価区間を明示

## 7. 試行錯誤と最終モデル選択の根拠

時間とプロセス変数の交互作用を含むモデルも検討したが、評価区間で誤差が著しく増大するケースが確認された。そのため、本課題ではモデル構造を単純化し、時間項とプロセス変数を独立に扱う構成を採用した。説明変数の選択は、1変数ずつ除外するアブレーション実験により、性能と安定性の両面から検討した。

### 7.1 説明変数の Ablation

説明変数を削除した場合の評価区間 RMSE を表 7 に示す。X32 を除外した場合に RMSE は改善したが、検証結果のばらつきが大きく安定性に課題が見られたため、総合的に 4 变数を用いた最終モデルを採用した。

表 7:Ablation 実験結果

| 条件    | 除外変数 | Test RMSE |
|-------|------|-----------|
| 最終モデル | なし   | 59.50     |
| -X32  | X32  | 58.04     |
| -X36  | X36  | 63.27     |
| -X27  | X27  | 69.47     |
| -X83  | X83  | 69.97     |

### 7.2 ドリフト対策の比較

追加のドリフト対策を含めた比較を表 8 に示す。検証指標が改善して見える手法でも評価区間で悪化する場合があるため、検証と評価の整合性を重視し、Base 構成を最終採用とした。

表 8:ドリフト対策の比較

| 手法            | 設定           | 最終 Fold RMSE | 評価区間 RMSE |
|---------------|--------------|--------------|-----------|
| Base          | 絶対時間         | 31.14        | 59.50     |
| Relative Time | 相対時間         | 42.28        | 65.61     |
| Decay Weight  | k=0.05       | 58.35        | 76.70     |
| Log Target    | log1p        | 30.31        | 71.38     |
| Interaction   | 時間と説明変数の交互作用 | 38.11        | 62.93     |

## 8. まとめ

本課題では、評価区間(末尾 500 レコード)の先頭ロットの process\_end\_time を境界時刻 cutoff とし、未来情報リークを防ぐため final\_mes\_time < cutoff を満たす観測可能データのみで OV 予測モデルを構築した。最終モデルは StandardScaler → Ridge(alpha=500)を採用し、説明変数を 4 つ(X32, X36, X27, X83)に削減したうえで、評価区間の Test RMSE 59.50 を達成した。Rolling TimeSplit による内部検証、説明変数のアブレーション、ドリフト対策の比較実験により、最終構成の妥当性を数値で確認した。さらに、時系列・散布図・残差の可視化により予測の挙動を確認し、(残差 = 実測 - 予測と定義すると) 残差平均 -11.08 / 中央値 -25.57 とわずかな過大予測傾向はあるものの、時間方向の系統的な偏りは小さいことを確認した。