EDA

June 21, 2025

[3]:
```
import pandas as pd
import numpy as np
```

0.1 About Dataset

This dataset describes the actual pharmaceutical product manufacturing for all relevant process steps. In particular, the published dataset includes the data on incoming raw materials, compression process time series and final product quality for the selected product. Reference to publication will follow ("Cholesterol-Lowering Drug Process and Quality Data" Authors: Janja Zagar, Jurij Mihelic).

0.1.1 Detail About the Dataset:

Dataset scope. The product or product family in the scope of the research has several product sub-families, which are defined by product code. Product sub-families differ in strength and manufacturing batch size. There are four different strengths and nine different batch sizes present in the research dataset. Products of different strengths within the scope have proportional or semi-proportional formulations and only differ in the weight of the final tablet, keeping formulation ratios the same. In order to account for the differences between product sub-families, categorical data are also included in the research dataset. The data collected for the present research range from November 2018 to April 2021. The time interval exceeding one year ensures that seasonal variation, changes in incoming raw materials, the impact of operator shift work, holidays, and other common process and equipment variability, are all taken into account. It is thus safe to assume that the presented dataset is robust and representative of the selected product. #### Data sources. The primary data sources are laboratory analysis results of incoming raw materials (excipients and API), of the intermediate product (tablet cores), and of the final product. The analyses were performed by trained laboratory technicians specialized in corresponding test. Devices used for analysis ranged from HPLC (high-performance liquid chromatography), GC (gas chromatography), moisture analyzer and particle size analyzer to automatic tablet cores analyzer. The second primary source of data are the tablet compression process time series. Time series output, such as tablet press speed, compaction force, fill depth, etc., is generated by tablet press sensors (Table 2). Time series output is generated for every second of the process and is stored in the tablet press SQL database. From there, time series are uploaded to a server that allows for visualization or extraction of the data by domain experts. This data is semi-structured and requires cleaning and organizing before use. #### Data collection methods. Before accessing and exporting securely the stored laboratory and process data, the so-called batch genealogy was performed. All laboratory and process data in the above-mentioned databases are stored using batch identifiers. In order to extract the relevant data from databases, it was necessary to determine the corresponding raw

material batches that entered into each of the 1,005 final product batches included in this data descriptor study. Only after this initial information was known, did the process of data collection begin. We exported the data by product material code (i.e., product sub-family), which groups all the batches that have been manufactured under that particular code. The export filter settings, therefore, included the time interval, product code, and laboratory analysis range. The process time series export was more challenging compared to the laboratory data, due to the quantity of the data. The tablet compression process typically runs between 2hours and 20hours, depending on product sub-family (i.e., product code), which defines the batch size (i.e., the target number of tablets produced).

0.2 Process Dataset

This dataset includes an example of new feature creation from the original time-series datasets provided. These were obtained based on expert knowledge of the compression process and impact on product quality.

[11]:
```
df_process = pd.read_csv('Process.csv', sep=';')
```

[13]:
```
df_process.head()
```

[13]:

	batch	code	tbl_speed_mean	tbl_speed_change	tbl_speed_0_duration	\
0	1	25	99.864656	5.416667	149.583333	
1	2	25	99.936342	2.500000	128.333333	
2	3	25	99.985984	2.500000	83.333333	
3	4	25	99.976868	2.916667	76.250000	
4	5	25	99.968284	2.500000	121.250000	

	total_waste	startup_waste	weekend	fom_mean	fom_change	…	\
0	2125.416667	5085	no	49.961446	12	…	
1	887.500000	2115	no	49.962040	5	…	
2	796.250000	1895	no	49.961176	6	…	
3	695.833333	1645	no	49.960900	9	…	
4	829.166667	1971	no	50.000000	5	…	

	ejection_min	Startup_tbl_fill_maxDifference	Startup_main_CompForce_mean	\
0	196	0.38	4.587500	
1	194	0.18	4.390909	
2	184	0.12	4.430000	
3	197	0.24	4.500000	
4	205	0.19	3.960000	

	Startup_tbl_fill_mean	Drug release average (%)	Drug release min (%)	\
0	5.466667	93.83	86.0	
1	5.315455	99.67	92.0	
2	5.242000	97.33	92.0	
3	5.221250	94.50	89.0	
4	5.233000	92.00	88.0	

```
   Residual solvent  Total impurities  Impurity 0  Impurity L
0              0.06              0.33        0.05        0.16
1              0.04              0.34        0.06        0.16
2              0.03              0.28        0.05        0.16
3              0.03              0.30        0.05        0.18
4              0.04              0.31        0.05        0.18

[5 rows x 35 columns]
```

[15]:
```
df_process.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 35 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   batch                            1005 non-null   int64
 1   code                             1005 non-null   int64
 2   tbl_speed_mean                   1005 non-null   float64
 3   tbl_speed_change                 1005 non-null   float64
 4   tbl_speed_0_duration             1005 non-null   float64
 5   total_waste                      1005 non-null   float64
 6   startup_waste                    1005 non-null   int64
 7   weekend                          1005 non-null   object
 8   fom_mean                         1005 non-null   float64
 9   fom_change                       1005 non-null   int64
 10  SREL_startup_mean                1005 non-null   float64
 11  SREL_production_mean             1005 non-null   float64
 12  SREL_production_max              1005 non-null   float64
 13  main_CompForce mean              1005 non-null   float64
 14  main_CompForce_sd                1005 non-null   float64
 15  main_CompForce_median            1005 non-null   float64
 16  pre_CompForce_mean               1005 non-null   float64
 17  tbl_fill_mean                    1005 non-null   float64
 18  tbl_fill_sd                      1005 non-null   float64
 19  cyl_height_mean                  1005 non-null   float64
 20  stiffness_mean                   1005 non-null   float64
 21  stiffness_max                    1005 non-null   int64
 22  stiffness_min                    1005 non-null   int64
 23  ejection_mean                    1005 non-null   float64
 24  ejection_max                     1005 non-null   int64
 25  ejection_min                     1005 non-null   int64
 26  Startup_tbl_fill_maxDifference   1005 non-null   float64
 27  Startup_main_CompForce_mean      1005 non-null   float64
 28  Startup_tbl_fill_mean            1005 non-null   float64
 29  Drug release average (%)         987 non-null    float64
 30  Drug release min (%)             987 non-null    float64
 31  Residual solvent                 987 non-null    float64
```

```
 32  Total impurities                987 non-null    float64
 33  Impurity 0                      987 non-null    float64
 34  Impurity L                      987 non-null    float64
dtypes: float64(26), int64(8), object(1)
memory usage: 274.9+ KB
```

0.3 Normalisation Dataset

Considering different batch sizes of the product family included in presented datasets, normalisation factors needed to be applied for the more accutrate feature extraction from original time-series data.

[17]:
```
df_nor = pd.read_csv('Normalization.csv', sep=';')
```

[19]:
```
df_nor.head()
```

[19]:

	Product code	Batch Size (tablets)	Normalisation factor
0	1	240000	2.40
1	2	1920000	19.20
2	3	960000	9.60
3	4	583000	5.83
4	5	2400000	24.00

[21]:
```
df_nor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Product code          25 non-null     int64
 1   Batch Size (tablets)  25 non-null     int64
 2   Normalisation factor  25 non-null     float64
dtypes: float64(1), int64(2)
memory usage: 732.0 bytes
```

0.4 Laboratory Dataset

Laboratory analysed data are gathered in this dataset for selected cholestrerol-lowering film coated tablet medicine. The file includes data collected for 1005 production batches manufactured between 2018 and 2021. Besides critical quality attributes (CQAs), intermediate product attributes, excipient and entering API batches' analysis results are included for each final product batch.

The laboratory data includes the results from the incoming raw material analysis (independent variables), intermediate product quality (independent variables), and final product quality (dependent variables). Product quality parameters included in the dataset are final product impurities, residual solvents and drug release results.

[23]:
```
df_lab = pd.read_csv('Laboratory.csv', sep=';')
```

[25]:
```
df_lab.head()
```

[25]:

	batch	code	strength	size	start	api_code	api_batch	smcc_batch	\
0	1	25	5MG	240000	nov.18	5	2	1	
1	2	25	5MG	240000	nov.18	5	2	1	
2	3	25	5MG	240000	nov.18	5	2	1	
3	4	25	5MG	240000	nov.18	5	2	1	
4	5	25	5MG	240000	nov.18	5	2	1	

	lactose_batch	starch_batch	…	tbl_tensile	fct_tensile	tbl_yield	\
0	2	1	…	1.412698	1.926183	95.785	
1	2	1	…	1.412698	1.986377	98.467	
2	2	1	…	1.412698	2.016473	98.496	
3	2	1	…	1.474120	1.956280	97.736	
4	2	1	…	1.443409	1.926183	98.106	

	batch_yield	dissolution_av	dissolution_min	resodual_solvent	\
0	94.697	93.83	86	0.06	
1	97.348	99.67	92	0.04	
2	99.242	97.33	92	0.03	
3	98.106	94.50	89	0.03	
4	98.106	92.00	88	0.04	

	impurities_total	impurity_o	impurity_l
0	0.33	0.05	0.16
1	0.34	0.06	0.16
2	0.28	0.05	0.16
3	0.30	0.05	0.18
4	0.31	0.05	0.18

[5 rows x 55 columns]

[27]:
```
df_lab.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 55 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   batch                 1005 non-null   int64
 1   code                  1005 non-null   int64
 2   strength              1005 non-null   object
 3   size                  1005 non-null   int64
 4   start                 1005 non-null   object
 5   api_code              1005 non-null   int64
 6   api_batch             1005 non-null   int64
 7   smcc_batch            1005 non-null   int64
 8   lactose_batch         1005 non-null   int64
```

```
 9   starch_batch          1005 non-null   int64
 10  api_water             1005 non-null   object
 11  api_total_impurities  1000 non-null   object
 12  api_l_impurity        996 non-null    object
 13  api_content           1003 non-null   float64
 14  api_ps01              1005 non-null   object
 15  api_ps05              1005 non-null   object
 16  api_ps09              1005 non-null   object
 17  lactose_water         1005 non-null   float64
 18  lactose_sieve0045     1005 non-null   int64
 19  lactose_sieve015      1005 non-null   int64
 20  lactose_sieve025      1005 non-null   int64
 21  smcc_water            1005 non-null   float64
 22  smcc_td               1005 non-null   float64
 23  smcc_bd               1005 non-null   float64
 24  smcc_ps01             1005 non-null   float64
 25  smcc_ps05             1005 non-null   float64
 26  smcc_ps09             1005 non-null   float64
 27  starch_ph             1005 non-null   float64
 28  starch_water          1005 non-null   float64
 29  tbl_min_thickness     1005 non-null   float64
 30  tbl_max_thickness     1005 non-null   float64
 31  fct_min_thickness     1005 non-null   float64
 32  fct_max_thickness     1005 non-null   float64
 33  tbl_min_weight        995 non-null    float64
 34  tbl_max_weight        995 non-null    float64
 35  tbl_rsd_weight        1005 non-null   float64
 36  fct_rsd_weight        1005 non-null   float64
 37  tbl_min_hardness      1005 non-null   float64
 38  tbl_max_hardness      1005 non-null   float64
 39  tbl_av_hardness       1005 non-null   int64
 40  fct_min_hardness      1005 non-null   float64
 41  fct_max_hardness      1005 non-null   float64
 42  fct_av_hardness       1005 non-null   float64
 43  tbl_max_diameter      1005 non-null   float64
 44  fct_max_diameter      1005 non-null   float64
 45  tbl_tensile           1005 non-null   float64
 46  fct_tensile           1005 non-null   float64
 47  tbl_yield             1005 non-null   float64
 48  batch_yield           1005 non-null   float64
 49  dissolution_av        1005 non-null   float64
 50  dissolution_min       1005 non-null   int64
 51  resodual_solvent      1005 non-null   float64
 52  impurities_total      1005 non-null   float64
 53  impurity_o            1005 non-null   float64
 54  impurity_l            1005 non-null   float64
dtypes: float64(34), int64(13), object(8)
memory usage: 432.0+ KB
```

0.5 Process time series Dataset

The time series data files are arranged by product codes, i.e., product sub-families. Each product code combines all final product batches manufactured in the selected period. The process time series includes the most relevant tablet compression process parameters based on product history and expert knowldge.

0.5.1 Process time series/1.csv ... 25.csv

Consist of 1-25 such CSV Files

[53]:
```
df_raw_2 = pd.read_csv('2.csv', sep=';')
```

[55]:
```
df_raw_2.head()
```

[55]:

	timestamp	campaign	batch	code	tbl_speed	fom	main_comp	\
0	2018-11-18 22:34:33	5	16	2	0.0	0	0.0	
1	2018-11-18 22:34:43	5	16	2	0.0	0	0.0	
2	2018-11-18 22:34:53	5	16	2	0.0	0	0.0	
3	2018-11-18 22:35:03	5	16	2	0.0	0	0.0	
4	2018-11-18 22:35:13	5	16	2	0.0	0	0.0	

	tbl_fill	SREL	pre_comp	produced	waste	cyl_main	cyl_pre	stiffness	\
0	3.85	0.0	0.0	0	0	1.25	5.0	0	
1	3.85	0.0	0.0	0	0	1.25	5.0	0	
2	3.85	0.0	0.0	0	0	1.25	5.0	0	
3	3.85	0.0	0.0	0	0	1.25	5.0	0	
4	3.85	0.0	0.0	0	0	1.25	5.0	0	

	ejection
0	0
1	0
2	0
3	0
4	0

[65]:
```
df_raw_2["timestamp"]=pd.to_datetime(df_raw_2["timestamp"])
```

[71]:
```
df_raw_2.head()
```

[71]:

	timestamp	campaign	batch	code	tbl_speed	fom	main_comp	\
0	2018-11-18 22:34:33	5	16	2	0.0	0	0.0	
1	2018-11-18 22:34:43	5	16	2	0.0	0	0.0	
2	2018-11-18 22:34:53	5	16	2	0.0	0	0.0	
3	2018-11-18 22:35:03	5	16	2	0.0	0	0.0	
4	2018-11-18 22:35:13	5	16	2	0.0	0	0.0	

	tbl_fill	SREL	pre_comp	produced	waste	cyl_main	cyl_pre	stiffness	\
0	3.85	0.0	0.0	0	0	1.25	5.0	0	

```
1      3.85   0.0       0.0         0      0      1.25      5.0          0
2      3.85   0.0       0.0         0      0      1.25      5.0          0
3      3.85   0.0       0.0         0      0      1.25      5.0          0
4      3.85   0.0       0.0         0      0      1.25      5.0          0

   ejection
0         0
1         0
2         0
3         0
4         0
```

[73]:
```
df_raw_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 160513 entries, 0 to 160512
Data columns (total 16 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   timestamp   160513 non-null  datetime64[ns]
 1   campaign    160513 non-null  int64
 2   batch       160513 non-null  int64
 3   code        160513 non-null  int64
 4   tbl_speed   160513 non-null  float64
 5   fom         160513 non-null  int64
 6   main_comp   160513 non-null  float64
 7   tbl_fill    160513 non-null  float64
 8   SREL        160513 non-null  float64
 9   pre_comp    160513 non-null  float64
 10  produced    160513 non-null  int64
 11  waste       160513 non-null  int64
 12  cyl_main    160513 non-null  float64
 13  cyl_pre     160513 non-null  float64
 14  stiffness   160513 non-null  int64
 15  ejection    160513 non-null  int64
dtypes: datetime64[ns](1), float64(7), int64(8)
memory usage: 19.6 MB
```

[75]:
```
df_raw_2.describe()
```

[75]:
```
                           timestamp       campaign          batch      code  \
count                         160513  160513.000000  160513.000000  160513.0
mean   2019-04-26 12:17:24.158841088      48.539533     210.701202       2.0
min              2018-11-18 22:34:33       5.000000      16.000000       2.0
25%              2019-04-16 17:54:57      41.000000     162.000000       2.0
50%              2019-06-22 15:05:14      68.000000     277.000000       2.0
75%              2019-07-04 23:55:26      69.000000     321.000000       2.0
max              2019-08-07 23:01:22      69.000000     324.000000       2.0
```

```
std                           NaN      25.460581     110.799096       0.0

           tbl_speed            fom      main_comp       tbl_fill  \
count  160513.000000  160513.000000  160513.000000  160513.000000
mean       72.451878      21.613776       4.099085       5.613922
min         0.000000       0.000000       0.000000       3.850000
25%         0.000000       0.000000       3.600000       5.360000
50%       120.000000      20.000000       4.000000       5.420000
75%       120.000000      40.000000       4.700000       6.000000
max       126.400000     100.000000      11.200000       6.840000
std        58.673945      19.850420       0.634645       0.347789

                SREL       pre_comp       produced          waste  \
count  160513.000000  160513.000000  160513.000000  160513.000000
mean        3.645918       0.011736    1021.499523   13719.140998
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000     654.000000    5935.000000
50%         4.800000       0.000000    1122.000000   10646.000000
75%         5.700000       0.000000    1329.000000   25139.000000
max       163.600000       0.300000    1919.000000   42529.000000
std         2.879827       0.040405     501.836924    9280.074012

            cyl_main        cyl_pre      stiffness       ejection
count  160513.000000  160513.000000  160513.000000  160513.000000
mean        1.715494       5.015349     203.739448     146.908344
min         0.650000       5.000000       0.000000       0.000000
25%         1.590000       5.000000      42.000000     142.000000
50%         1.740000       5.000000      76.000000     169.000000
75%         1.800000       5.000000     553.000000     173.000000
max         8.000000       7.980000     781.000000     373.000000
std         0.191395       0.082099     233.962469      51.521278
```

[77]:
```
df_raw_2.isnull().sum()
```

[77]:
```
timestamp    0
campaign     0
batch        0
code         0
tbl_speed    0
fom          0
main_comp    0
tbl_fill     0
SREL         0
pre_comp     0
produced     0
waste        0
cyl_main     0
```

```
cyl_pre      0
stiffness    0
ejection     0
dtype: int64
```

[ ]: