DECEMBER 2023

Artificial Intelligence for Decision Making in Industrial Engineering

HIGH SPEED MACHINING REPORT

ARTIFICIAL INTELLIGENCE FOR DECISION MAKING IN INDUSTRIAL ENGINEERING

ANALYSIS OF HSM PERFORMANCE REPORT

A PROJECT REPORT

Submitted by

Nithish Sallustre PERTHUS SALLUSTRE
Karthikeyan SELVARAJAN

Under the guidance Of

CATHERINE DA CUNHA
Professor

ÉCOLE CENTRALE DE NANTES

DECEMBER 2023

— AI4IE: DATA MINING ON REAL INDUSTRIAL DATASET —

The objective is to have a global experience of Data Mining on a real industrial dataset. To do so, we will follow the steps of CRISP-DM: the Cross Industry Standard Process for Data Mining.

1) Business objectives

The "business objective" is to exploit the collected data and provide KPI (Key Performance Indicator) concerning productivity and the industrial performance. • For productivity, the OEE (Overall Equipment Efficiency) will be computed. To do so, it is necessary to know how long the machine-tool has been machining. And, if time allows, to detect faulty parts. • For the industrial performance, the average cutting time per workpiece is interesting. Thus, the number of blanks (raw parts) should be detected automatically. We will start with this 2nd objective, following a data-driven approach. Since the answer is unknown, the dataset is unlabeled and it should be determined by unsupervised machine learning. Lately, a model-based approach will be tried, through the combination of data analytics and knowledge integration (with business rules).

2) Data import and understanding

The dataset were collected by EmmaTools device on a 5-axes machine-tool of Five Machining in an aeronautic company that manufactures structural parts in aluminum alloy. The matrix consists in 72 variables (columns), measured every tenth of a second (rows), during one day of industrial production.

Data is presented in data777.CSV file. To facilitate the use, it will be store in a Panda dataFrame.

[3]:
```
# Import of the needed libraires
#graphical librairies

import matplotlib as mpl
from matplotlib import pyplot
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import figure, subplot, hist, xlim, show, plot
%matplotlib inline

#data librairies

import pandas as pd
import pylab as pl
import numpy as np
```

```
from pandas.plotting import scatter_matrix
from pandas.plotting import boxplot
from pandas.plotting import parallel_coordinates

from scipy.io import loadmat
```

[4]:
```
#data import from data777.CSV and creation of panda object

HSM_data = pd.read_csv("data777.csv")
```

[6]:
```
print(HSM_data)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
0       5560779  4105603  190312004        31     69     2         0        0
1       5560780  4105603  190312005        31     69     2         0        0
2       5560781  4105603  190312006        31     69     2         0        0
3       5560782  4105603  190312007        31     69     2         0        0
4       5560783  4105603  190312008        31     69     2         0        0
...         ...      ...        ...       ...    ...   ...       ...
862569  6423348  4802871  192409145        36     74     2        22        6
862570  6423349  4802872  192409152        36     74     2        22        6
862571  6423350  4802873  192409153        36     74     2        22        6
862572  6423351  4802874  192409154        36     74     2        22        6
862573  6423352  4802875  192409155        36     74     2        22        6

        usure outil  nligne  ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
0                20       0  ...     0.000   0.000   0.000   0.000     0.000
1                20       0  ...     0.000   0.000   0.000   0.000     0.000
2                20       0  ...     0.000   0.000   0.000   0.000     0.000
3                20       0  ...     0.000   0.000   0.000   0.000     0.000
4                20       0  ...     0.000   0.000   0.000   0.000     0.000
...             ...     ...  ...       ...     ...     ...     ...       ...
862569          300     402  ...  2372.390   4.629   3.148   3.145  1186.195
862570          300     402  ...  2371.731   4.594   3.160   2.945  1185.866
862571          300     402  ...  2767.105   4.492   3.418   3.098  1185.902
862572          300     402  ...  2767.960   4.605   3.074   3.027  1186.268
862573          300     402  ...  2372.061   4.328   3.293   3.094  1186.030

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
0          0.000     0.000   0.000   0.000   0.000
1          0.000     0.000   0.000   0.000   0.000
2          0.000     0.000   0.000   0.000   0.000
3          0.000     0.000   0.000   0.000   0.000
4          0.000     0.000   0.000   0.000   0.000
...          ...       ...     ...     ...
862569  2372.390  1583.138   5.457   4.285   1.906
```

```
862570  2371.731  1582.698  5.453    4.113    1.660
862571  2371.804  1582.747  5.340    4.461    1.746
862572  2372.537  1583.236  5.473    4.250    1.820
862573  2372.061  2767.404  5.145    4.277    1.656

[862574 rows x 72 columns]
```

Check that import went well: - Display the variable names, - Check the size of the dataset in the "variable explorer" - And visualize the production schedule that day, through the sequence of part programs 'id_ProgP'.

[9]:
```python
# Visualise the variable names (labels) and the first lines values

HSM_data.head()
```

[9]:
```
      tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
0  5560779  4105603  190312004        31     69     2         0        0
1  5560780  4105603  190312005        31     69     2         0        0
2  5560781  4105603  190312006        31     69     2         0        0
3  5560782  4105603  190312007        31     69     2         0        0
4  5560783  4105603  190312008        31     69     2         0        0

   usure  outil  nligne  ...  FFT_15 FFT_16  FFT_17  FFT_18 FFT_19  FFT_20  \
0            20       0  ...     0.0    0.0     0.0     0.0    0.0     0.0
1            20       0  ...     0.0    0.0     0.0     0.0    0.0     0.0
2            20       0  ...     0.0    0.0     0.0     0.0    0.0     0.0
3            20       0  ...     0.0    0.0     0.0     0.0    0.0     0.0
4            20       0  ...     0.0    0.0     0.0     0.0    0.0     0.0

   FFT_21  FFT_22  FFT_23  FFT_24
0     0.0     0.0     0.0     0.0
1     0.0     0.0     0.0     0.0
2     0.0     0.0     0.0     0.0
3     0.0     0.0     0.0     0.0
4     0.0     0.0     0.0     0.0

[5 rows x 72 columns]
```

[10]:
```python
# print the list of the variable names

print(HSM_data.keys())
```

```
Index(['tpsT', 'tps B', 'date', 'id_ProgP', 'id pc', 'mode', 'id_outil',
       'n outil', 'usure outil', 'nligne', 'nbloc', 'Abloc', 'Cbloc', 'Temp_1',
       'Temp_2', 'Temp_3', 'Temp_4', 'Arms_1', 'Arms_2', 'Arms_3', 'Arms_4',
       'Apic_1', 'Apic_2', 'Apic_3', 'Apic_4', 'Vrms_1', 'Vrms_2', 'Vrms_3',
       'Vrms_4', 'Vpic_1', 'Vpic_2', 'Vpic_3', 'Vpic_4', 'PosX', 'PosY',
       'PosZ', 'PosA', 'PosC', 'VitX', 'VitY', 'VitZ', 'VitA', 'VitC', 'Vf',
```

```
       'N', 'P', '%Vf', '%N', 'FFT_1', 'FFT_2', 'FFT_3', 'FFT_4', 'FFT_5',
       'FFT_6', 'FFT_7', 'FFT_8', 'FFT_9', 'FFT_10', 'FFT_11', 'FFT_12',
       'FFT_13', 'FFT_14', 'FFT_15', 'FFT_16', 'FFT_17', 'FFT_18', 'FFT_19',
       'FFT_20', 'FFT_21', 'FFT_22', 'FFT_23', 'FFT_24'],
      dtype='object')
```

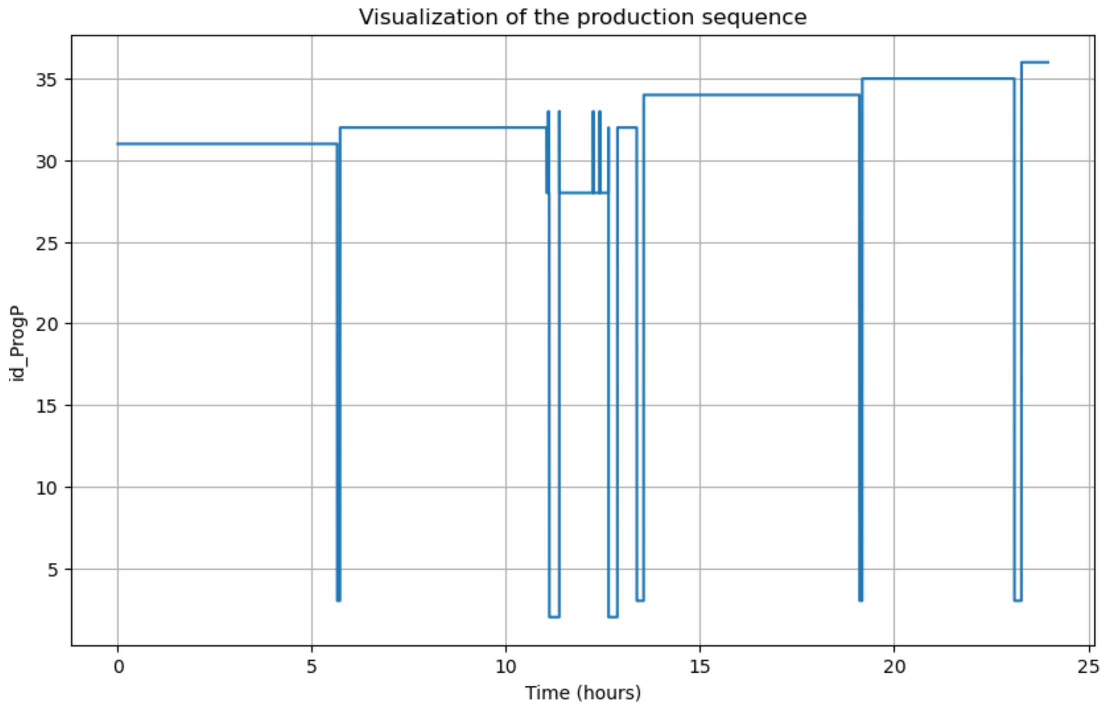
[14]:

```
# length of the dataset?

# Find the length of the dataset
dataset_length = len(HSM_data)

# Print the length of the dataset
print("Length of the Dataset:", dataset_length)
```

```
Length of the Dataset: 862574
```

[11]:

```
## Visualisation of the production sequence:

## There are 862574 length of data

#tmp=np.arange(0,nb_specimen*0.1,0.1)
#tmpH=tmp/3600
#tmpH # can be imported in DataFrame for abscissa x=..

# Define the number of specimens
nb_specimen = 862574

# Create a time array with 0.1 second intervals
tmp = np.arange(0, nb_specimen * 0.1, 0.1)

# Convert time to hours
tmpH = tmp / 3600

# Assuming you have a DataFrame named 'data' with a column named 'id_ProgP'
id_ProgP = HSM_data["id_ProgP"]

# Create a plot
plt.figure(figsize=(10, 6))
plt.plot(tmpH, id_ProgP)
plt.xlabel("Time (hours)")
plt.ylabel("id_ProgP")
plt.title("Visualization of the production sequence")
plt.grid(True)
plt.show()

#data_panda.plot(y='VariableName_XXX')
```



[13]:
```
# Production Sequence of 'id_ProgP'

print(HSM_data['id_ProgP'])
```

```
0         31
1         31
2         31
3         31
4         31
          ..
862569    36
862570    36
862571    36
862572    36
862573    36
Name: id_ProgP, Length: 862574, dtype: int64
```

1 Objective 1: Number of parts machined

The objective is to determine how many parts were machined on each pallet, by unsupervised machine learning. Clustering will be performed on data set of machine-tool motion. In this section, 2 variables will be used as input of the Machine Learning, the output consists in determining the K number of clusters.

1.1 Data Selection

Firstly, select a subset of data: during id_ProgP=32, and then for X & Y variables of current position ('PosX',...).

1.2 Visualization

Visualization enables to better understand the data and to verify the need of pre-treatments. Here df.plot can be used and we focus on Program n°32.

[14]:
```python
#  Data  Selection:

filtered_df = HSM_data[HSM_data['id_ProgP'] == 32]
print(filtered_df)
```

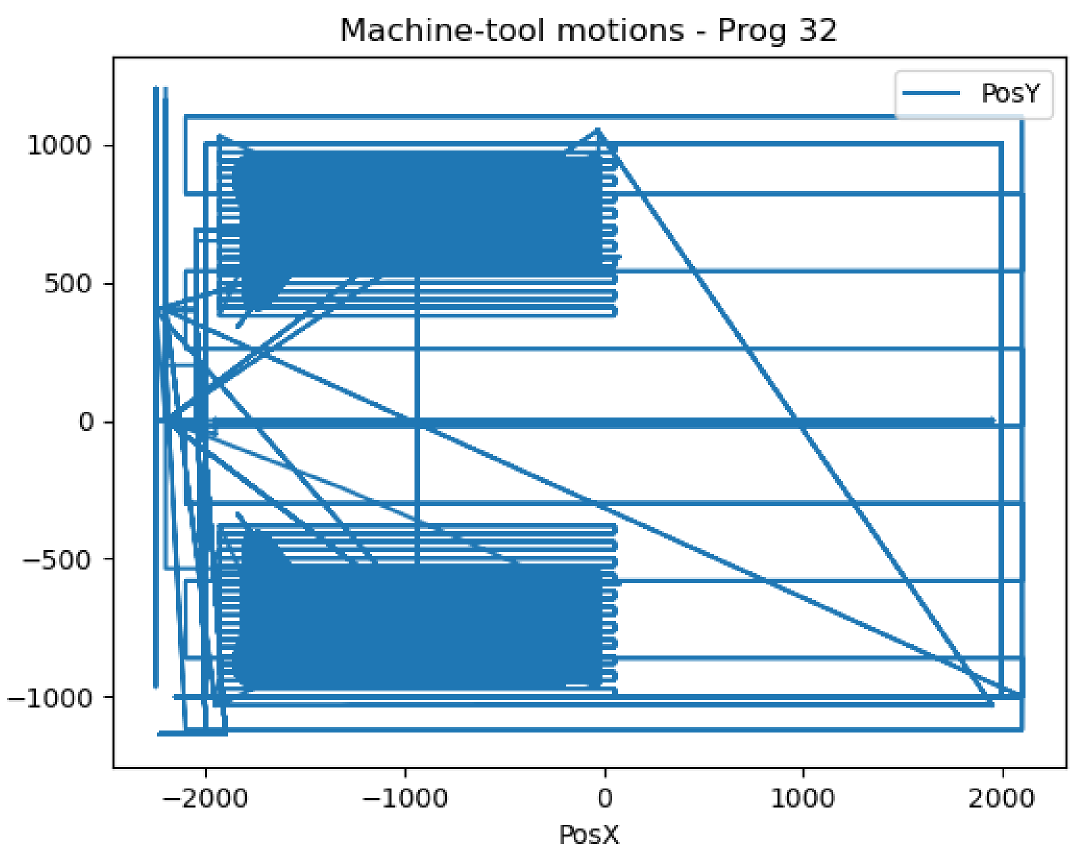
```
            tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
206071  5766850  4292619  191212597        32     70     2         0        0
206072  5766851  4292619  191212598        32     70     2         0        0
206073  5766852  4292619  191212599        32     70     2         0        0
206074  5766853  4292619  191212600        32     70     2         0        0
206075  5766854  4292619  191212601        32     70     2         0        0
...          ...      ...        ...       ...    ...   ...       ...
481254  6042033  4473434  191717433        32     70     2         7        0
481255  6042034  4473434  191717440        32     70     2         7        0
481256  6042035  4473434  191717441        32     70     2         7        0
481257  6042036  4473434  191717442        32     70     2         7        0
481258  6042037  4473434  191717443        32     70     2         7        0

        usure outil  nligne  ...  FFT_15 FFT_16  FFT_17  FFT_18  FFT_19  \
206071           20       0  ...     0.0    0.0     0.0     0.0     0.0
206072           20       0  ...     0.0    0.0     0.0     0.0     0.0
206073           20       0  ...     0.0    0.0     0.0     0.0     0.0
206074           20       0  ...     0.0    0.0     0.0     0.0     0.0
206075           20       0  ...     0.0    0.0     0.0     0.0     0.0
...             ...     ...  ...     ...    ...     ...     ...
481254           20       0  ...     0.0    0.0     0.0     0.0     0.0
481255           20       0  ...     0.0    0.0     0.0     0.0     0.0
481256           20       0  ...     0.0    0.0     0.0     0.0     0.0
481257           20       0  ...     0.0    0.0     0.0     0.0     0.0
481258           20       0  ...     0.0    0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22  FFT_23 FFT_24
206071     0.0     0.0     0.0     0.0    0.0
206072     0.0     0.0     0.0     0.0    0.0
206073     0.0     0.0     0.0     0.0    0.0
206074     0.0     0.0     0.0     0.0    0.0
206075     0.0     0.0     0.0     0.0    0.0
...        ...     ...     ...     ...    ...
481254     0.0     0.0     0.0     0.0    0.0
```

481255	0.0	0.0	0.0	0.0	0.0
481256	0.0	0.0	0.0	0.0	0.0
481257	0.0	0.0	0.0	0.0	0.0
481258	0.0	0.0	0.0	0.0	0.0

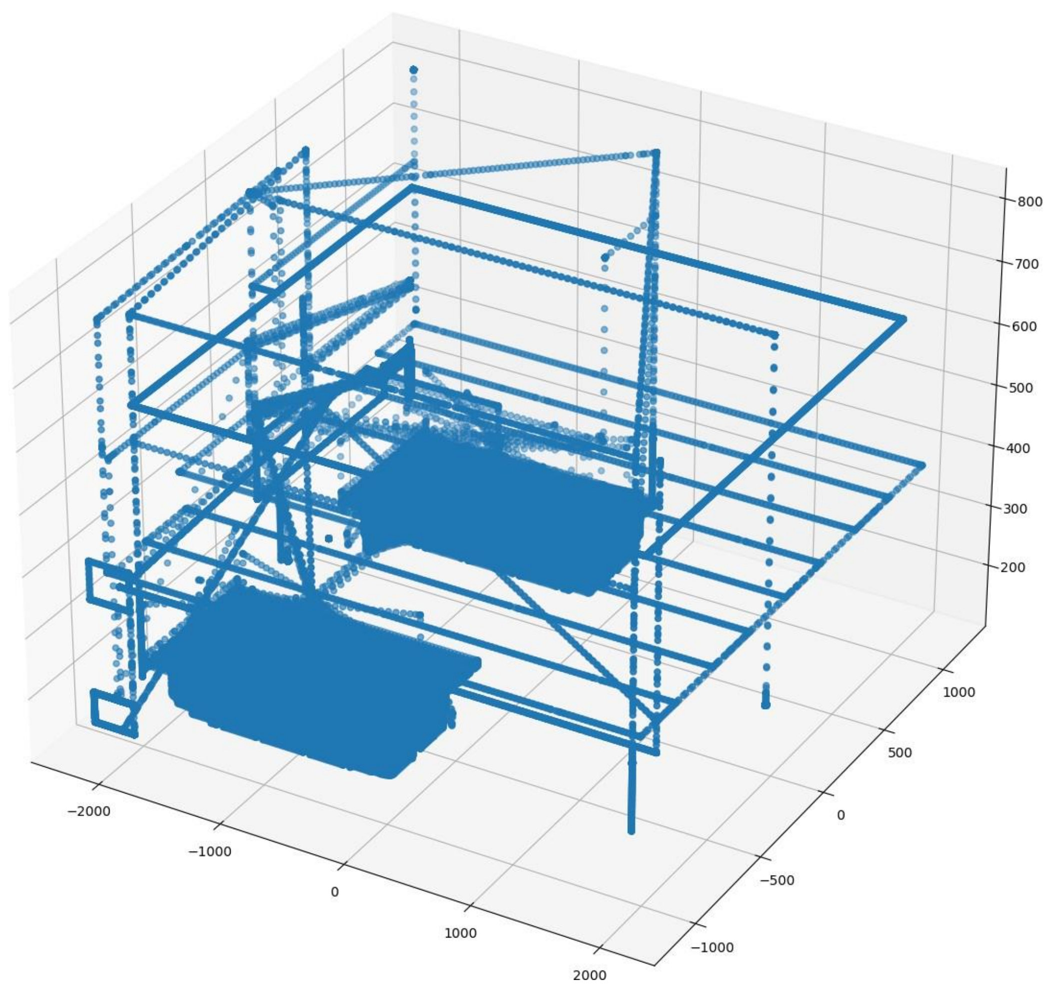
[209585 rows x 72 columns]

[ ]:
```
# To facilitate futur use, we can create a set with the variables labels,
# Input data for Machine Learning.
Input_cols = ["PosX","PosY","PosZ"]
```

[15]:
```
# Visualize the machine-tool motions with df.plot:
# df.plot(x='var1', y='var2')
# plt.title('Machine-tool motions - Prog 32')
# plt.show()

# Note that lately, scatter is more suitable for cluster visualization.
#df.plot(kind="scatter", x='var1', y='var2')

# Visualize the machine-tool motions with df.plot:
filtered_df.plot(x="PosX", y="PosY")
plt.title("Machine-tool motions - Prog 32")
plt.show()
```



[17]:
```python
# If you want to make a 3D plot, 'PosZ' should be added to the data selection

fig = plt.figure(figsize=(15, 15))
ax = plt.axes(projection="3d")
ax.scatter3D(filtered_df["PosX"], filtered_df["PosY"], filtered_df["PosZ"])
plt.show()
```



[19]:
```python
# Selected Data

filtered_df = HSM_data[HSM_data["id_ProgP"] == 32]
```

```
# Select only two specific columns
selected_columns = ['PosX', 'PosY','PosZ']
filtered_dff =filtered_df[selected_columns]
print(filtered_dff)
```

```
            PosX      PosY     PosZ
206071 -2200.028  1199.989  800.002
206072 -2200.028  1199.989  800.002
206073 -2200.028  1199.989  800.002
206074 -2200.028  1199.989  800.002
206075 -2200.028  1199.989  800.002
...          ...       ...      ...
481254 -2200.028   199.993  800.002
481255 -2200.028   199.993  800.002
481256 -2200.028   199.993  800.002
481257 -2200.028   199.993  800.002
481258 -2200.028   199.993  800.002

[209585 rows x 3 columns]
```

1.3 K-Means

In order to determine how many parts were machined on each pallet, by unsupervised machine learning, a clustering will be performed on data set of machine-tool motion. 2 variables will be used as input of the Machine Learning, the output consists in different k number of clusters. k should be optimized to determine the probable number of workpieces (clusters in the dataset). A common technic is k-means, where k is the number of cluster. The centroid is the center of the cluster.

The algorithm is: -Initialization with k centroids (randomly) -WHILE clustering is unstable DO: - Affect each observation to the cluster of which the center is the closest - Compute new cluster centers (average position)

The 'inertia' refers to the intra-cluster variance (related to the sum of the distances between a centroid and all the points belonging to its cluster).

The probable number of clusters K* can then be determined by the elbow method:

1.3.1 Initialization

Try the k-means algorithm of ScikitLearn on the selected subdataset, with for example 3 clusters. https://scikit-learn.org/stable/modules/clustering.html#clustering

[32]:
```
from sklearn import cluster
from sklearn.cluster import KMeans
from sklearn.metrics import completeness_score, homogeneity_score
```

[33]:
```
#definition of the colors used for visualization
color_dict_cluster={ 1:'r',2:'g' ,3:'b',4:'y',5:'c',6:'m',7:'k',8:'orange',0:
↪'teal'}
```

[34]:
```
## First tests of KMeans, progressively:

# define the cluster model (with max_iter=50,init='random')
kmeans = KMeans(n_clusters=3, max_iter=50, init='random')

# train the kmeans model (centroids) from the dataset
kmeans.fit(filtered_dff)

# where are the centroids positions? (kmeans.cluster_centers_)
centroids = kmeans.cluster_centers_

# compute the inertia = intra-cluster variance (kmeans.inertia_)
inertia = kmeans.inertia_

# prediction: affect each observation of the dataset, to the closest centroid␣
↪(kmeans.predict)
predictions = kmeans.predict(filtered_dff)

# from the cluster label of each point in the dataset (array), make a dataFrame␣
↪and concatenate to the dataset
#pred = pd.dataframe(VarXXX)
#pred.columns = 'predicted_cluster'
#df = pd.concat([df,pred], axis = 1)
cluster_labels = pd.DataFrame({'predicted_cluster': predictions})
filtered_df = pd.concat([filtered_dff, cluster_labels], axis=1)

# similarly, make a dataFrame with the centroid positions
centroids_df = pd.DataFrame(centroids)
```

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

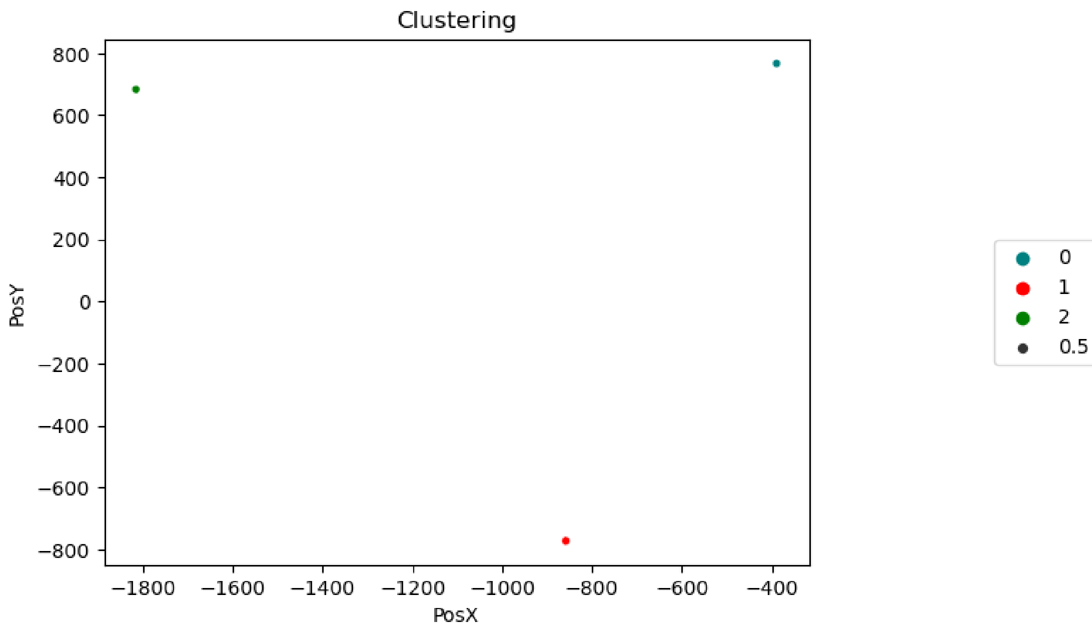
[35]:
```
centroids_df.columns=[ 'PosX', 'PosY','PosZ']
print(centroids_df)
```

	PosX	PosY	PosZ
0	-389.177642	767.404273	236.253804
1	-858.591772	-772.225175	255.577254
2	-1815.327023	684.009951	323.513455

[36]:
```
print(predictions)
```

[2 2 2 ... 2 2 2]

[39]:
```
## Visualize the results of clustering:

sns.scatterplot( data=centroids_df, x="PosX", y="PosY", hue=centroids_df.index,␣
↪size=0.5, palette=color_dict_cluster)

#sns.scatterplot(x='PosX', y='PosY', hue='predicted_cluster', size=5,␣
↪palette=c'predicted_cluster'olor_dict_cluster,  data=df_centroids)
plt.legend(loc="center left", bbox_to_anchor=(1.25, 0.5), ncol=1)
plt.title("Clustering")
plt.show()
```



1.3.2 Normalisation

To garanty that the use of Euclidan distances will not favor one of the characteristics, we need to work on normalized data.

[42]:
```
import copy

# Reset the index
Norm_32 = centroids_df.reset_index()

# Print column names
print(Norm_32.keys())

# Define Input_cols (excluding 'index')
```

```python
Input_cols = Norm_32.columns.difference(["index"])

# Normalization
Norm_32[Input_cols] = (Norm_32[Input_cols] - Norm_32[Input_cols].min()) /
 ↪(Norm_32[Input_cols].max() - Norm_32[Input_cols].min())

# Print normalized values
print(Norm_32[Input_cols])
```

```
Index(['index', 'PosX', 'PosY', 'PosZ'], dtype='object')
       PosX      PosY      PosZ
0  1.000000  1.000000  0.000000
1  0.670852  0.000000  0.221448
2  0.000000  0.945835  1.000000
```

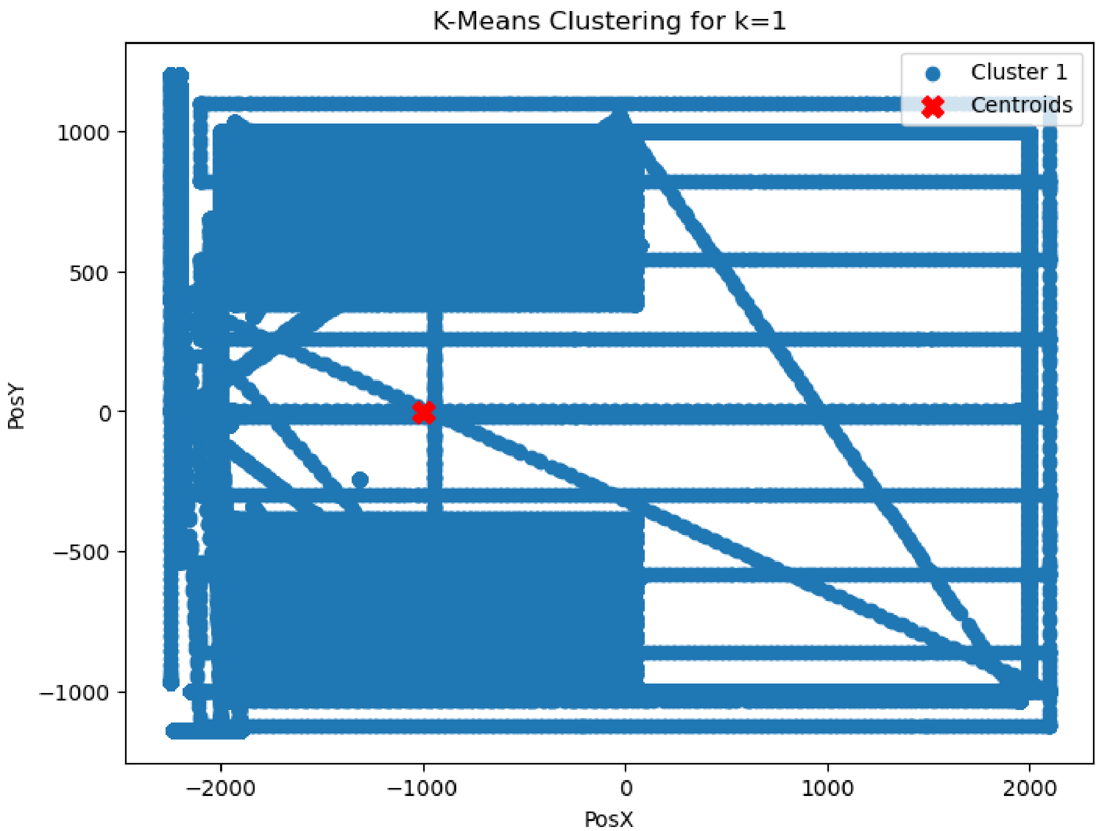
1.3.3 Elbow method

Make a FOR loop (for k in range(0,max_clusters)), to compute automatically k-means, make a new plot for each clustering, and finally use the elbow method (based on the intra-cluster variance) to determine the probable number of clusters.

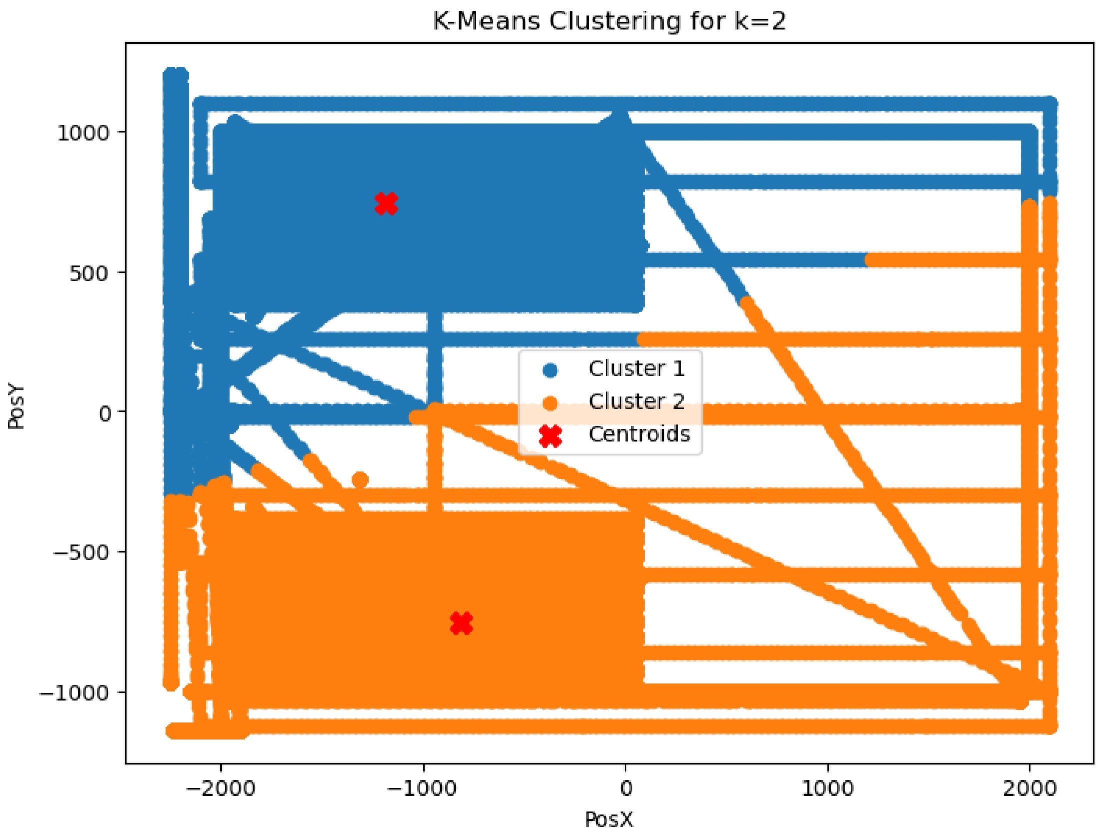
[45]:
```python
def find_optimal_clusters(range_n_clusters, ssd):
    deltas = np.diff(ssd, 2)
    elbow_index = np.argmax(deltas) + 2
    optimal_clusters = range_n_clusters[elbow_index - 1]
    return optimal_clusters

def My_function_kmeans_elbow(max_clusters, df):
    ssd = []
    range_n_clusters = np.arange(1, max_clusters + 1, 1)
    print(range_n_clusters)
    for num_clusters in range_n_clusters:

        # Launch the clustering
        kmeans = KMeans(n_clusters=num_clusters)
        kmeans.fit(df)
        ssd.append(kmeans.inertia_)

        # Plotting the clustering
        plt.figure(figsize=(8, 6))
        for i in range(num_clusters):
            cluster_indices = np.where(kmeans.labels_ == i)[0]
            plt.scatter(df.iloc[cluster_indices, 0], df.iloc[cluster_indices,
 ↪1], label=f"Cluster {i + 1}")
        plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,
 ↪1], s=100, c="red", marker="X", label="Centroids")
        plt.title(f"K-Means Clustering for k={num_clusters}")
```

```
        plt.xlabel("PosX")
        plt.ylabel("PosY")
        plt.legend()
        plt.show()

    # Plotting Elbow Curve for Optimal Clusters
    plt.plot(range_n_clusters, ssd, marker='o')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia (Sum of Squared Distances)')
    plt.title('Elbow Curve for Optimal Clusters')
    plt.show()

    # Find the optimal number of clusters
    optimal_clusters = find_optimal_clusters(range_n_clusters, ssd)
    print("Optimal number of clusters:", optimal_clusters)
    return optimal_clusters

# Example usage
# Assuming filtered_dff is defined somewhere in your code
optimal_clusters = My_function_kmeans_elbow(9, filtered_dff)
print("Optimal number of clusters:", optimal_clusters)
```

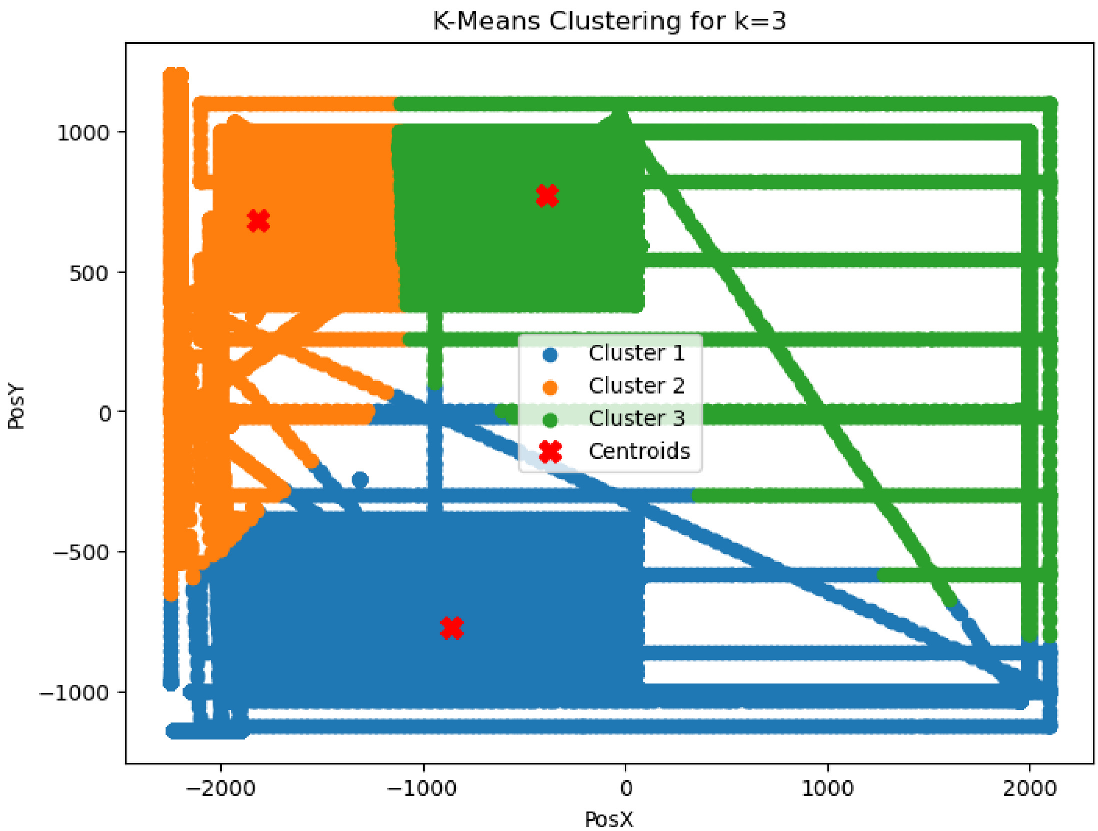
```
[1 2 3 4 5 6 7 8 9]

C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)
```



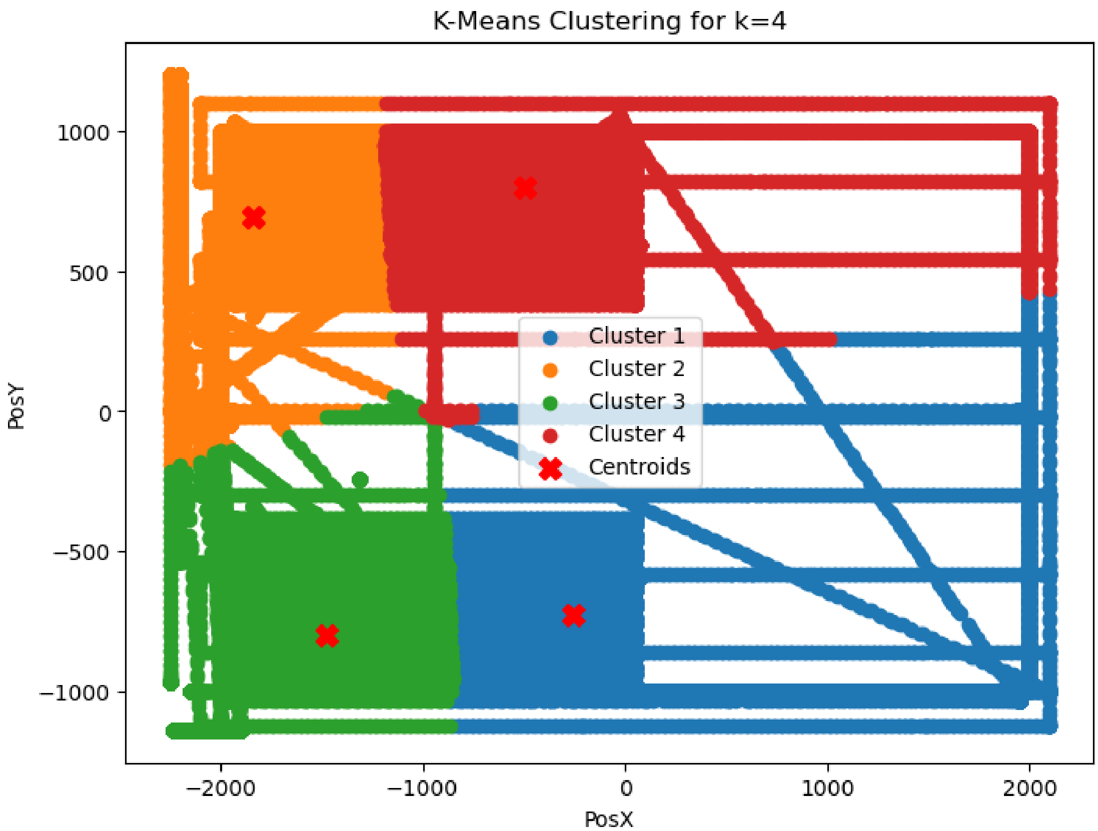
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



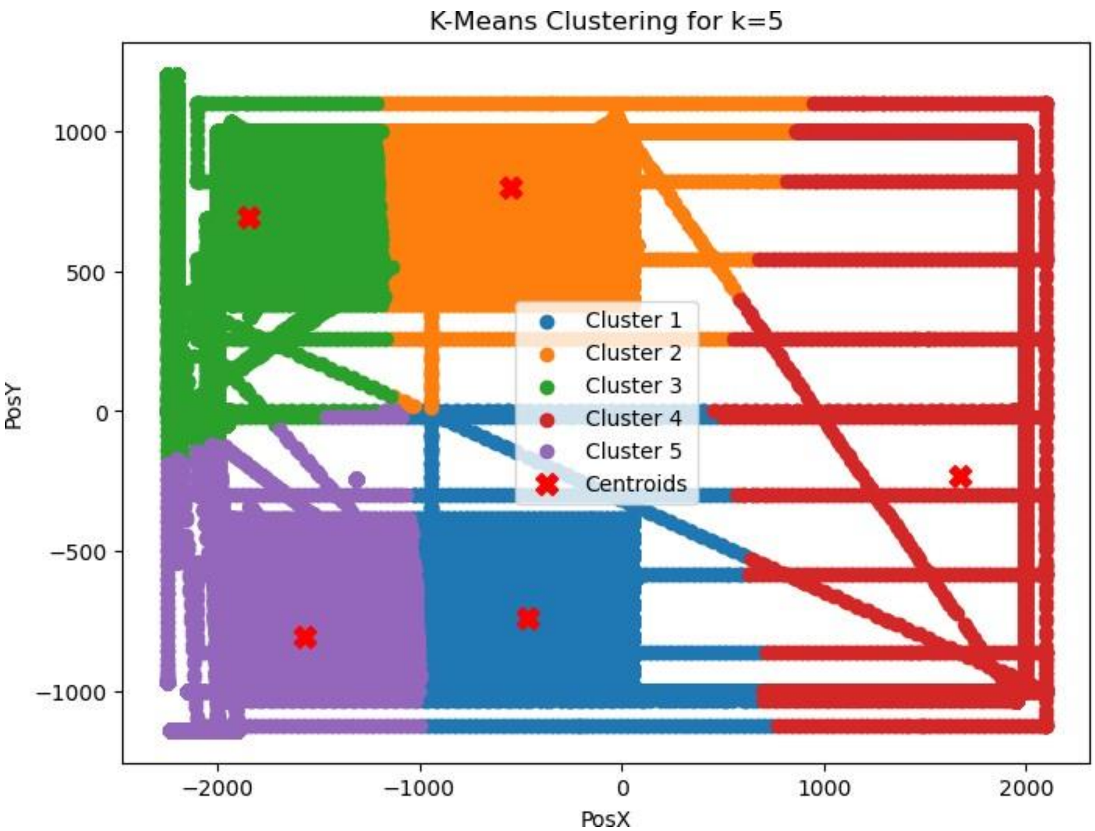
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



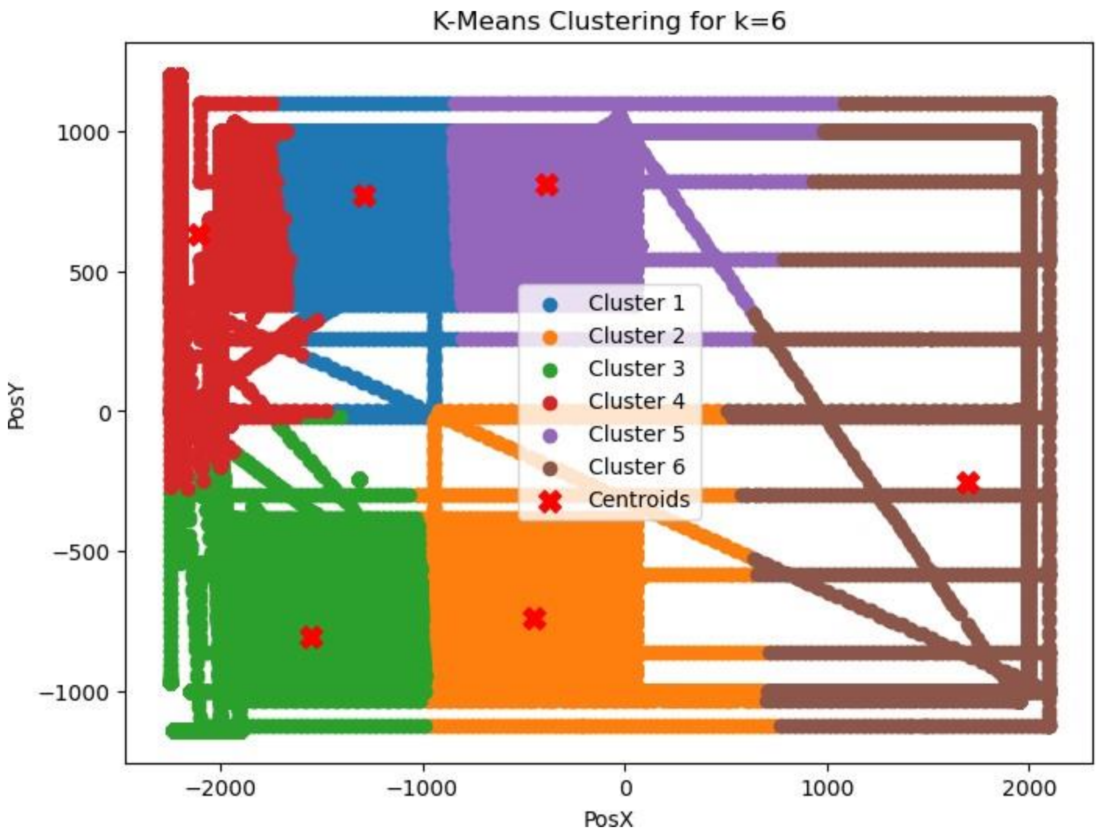
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



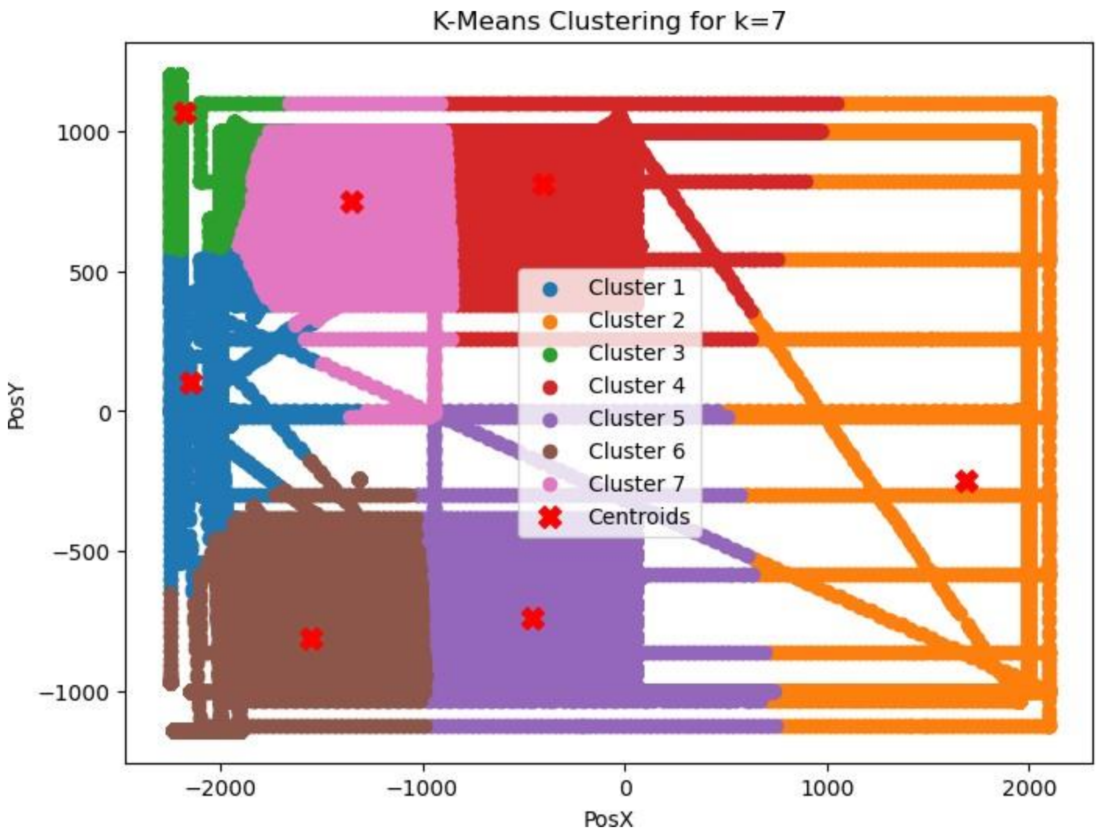
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



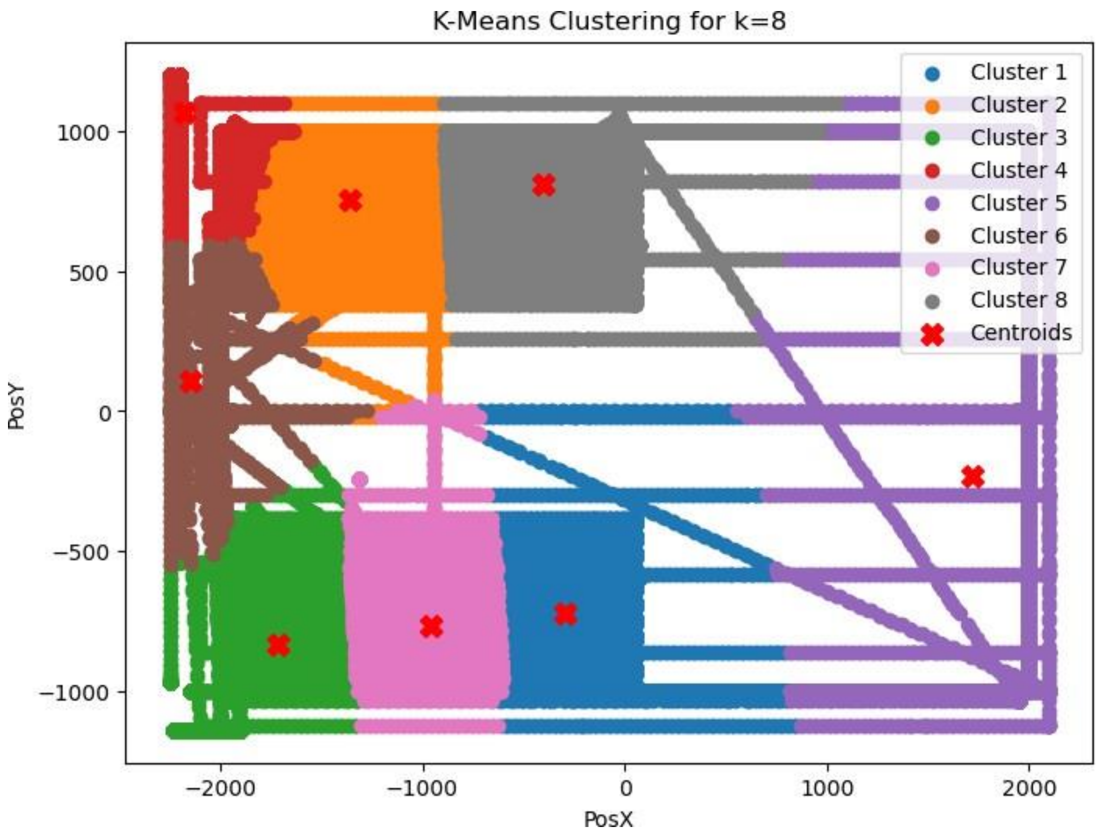
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



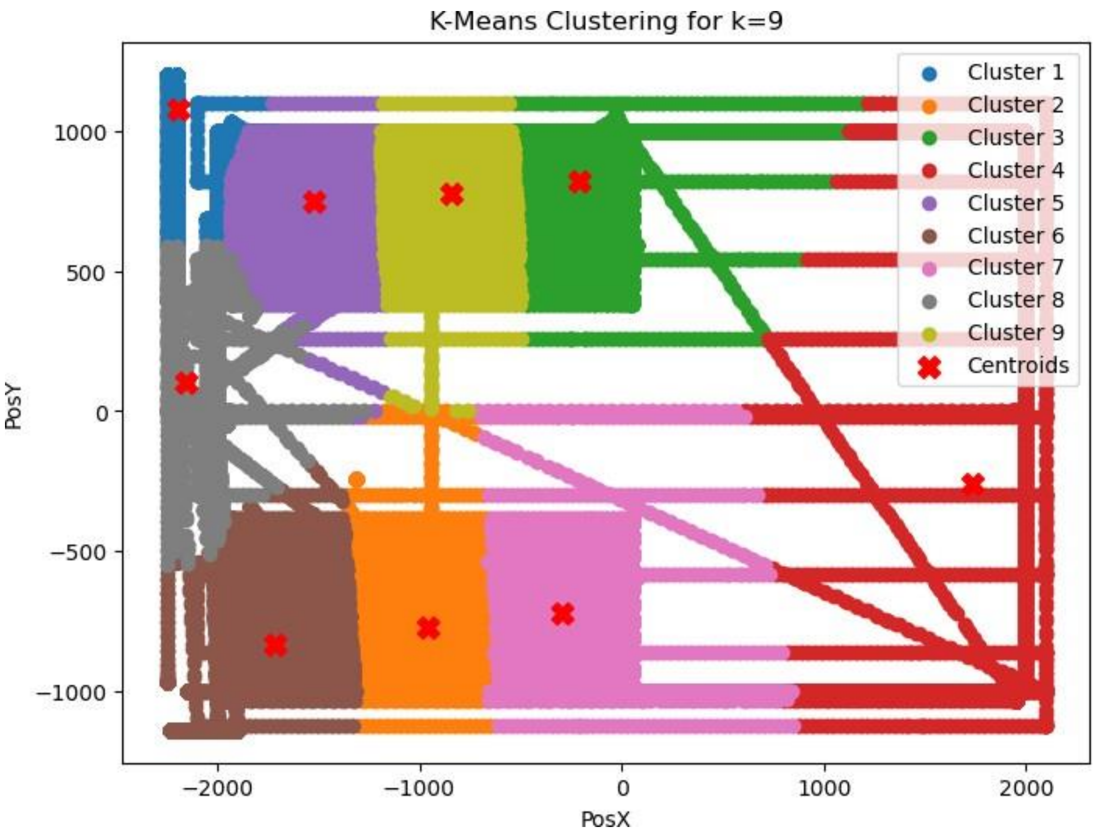
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)

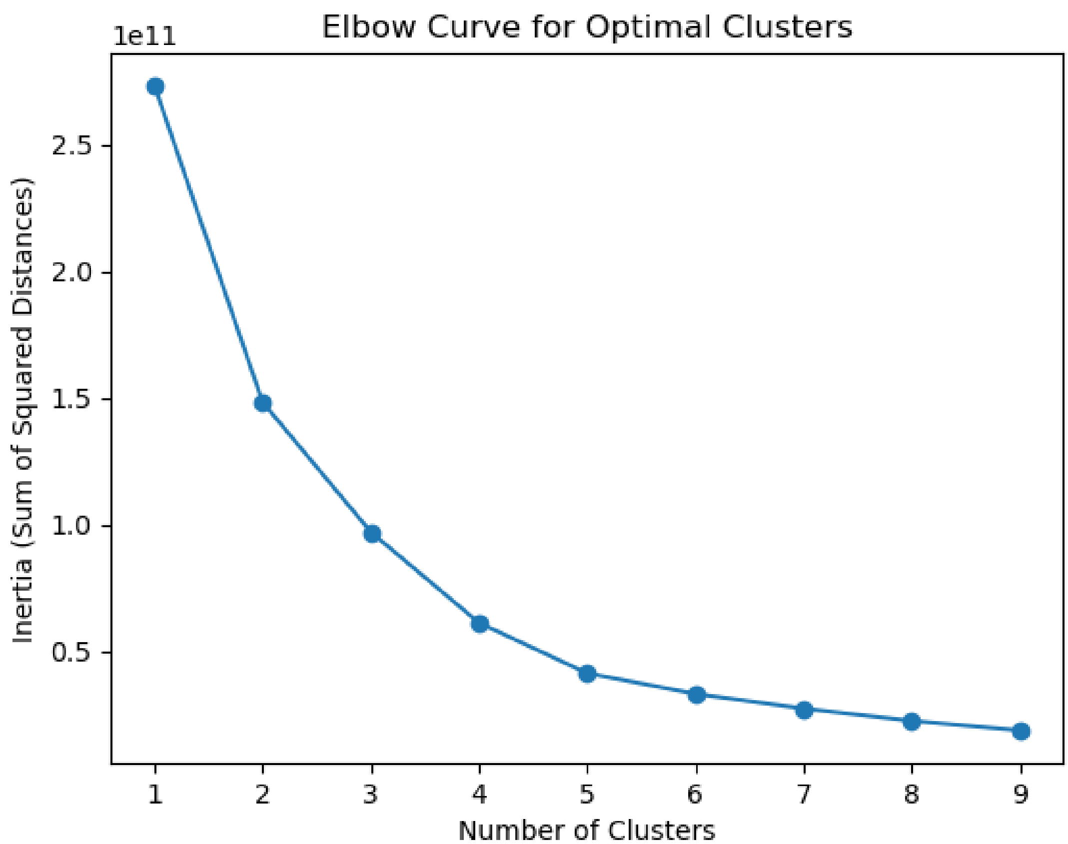


C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)



C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\nithi\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)


K-Means Clustering for k=9
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Centroids
PosY
PosX
1000
500
0
−500
−1000
−2000
−1000
0
1000
2000




```
Optimal number of clusters: 2
Optimal number of clusters: 2
```

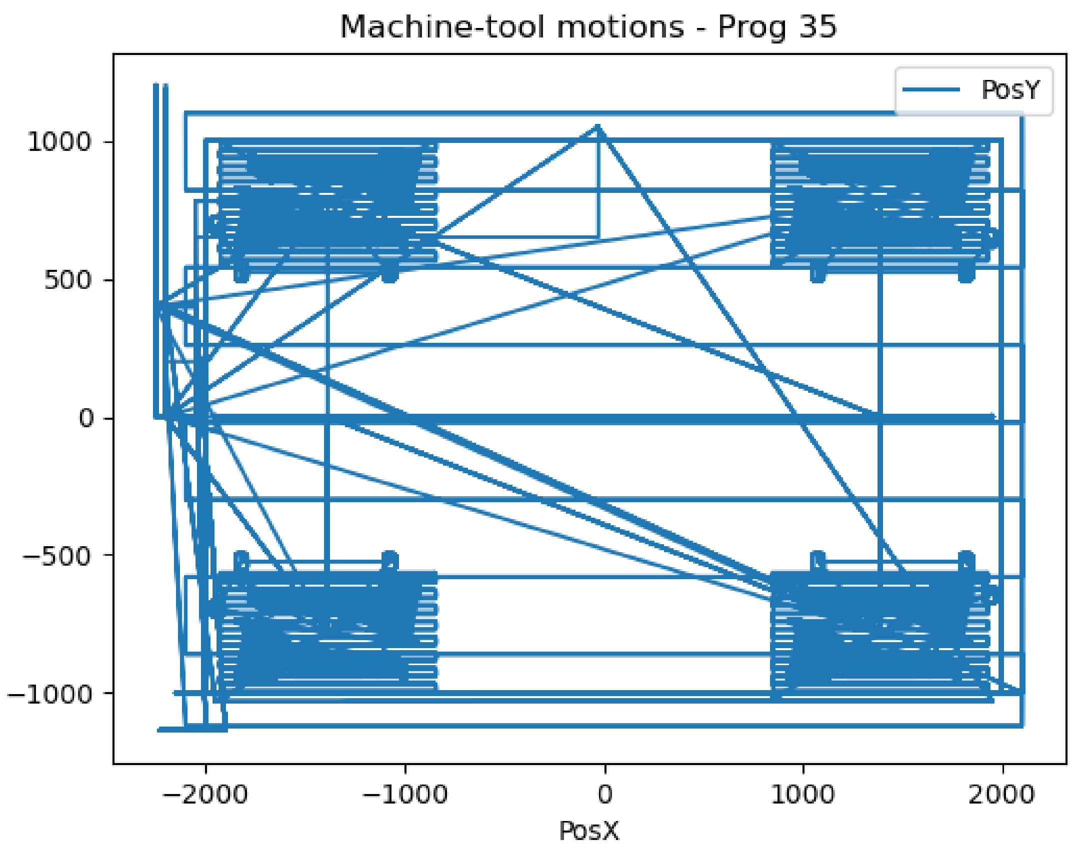
How many clusters are suggestes by the Elbow?

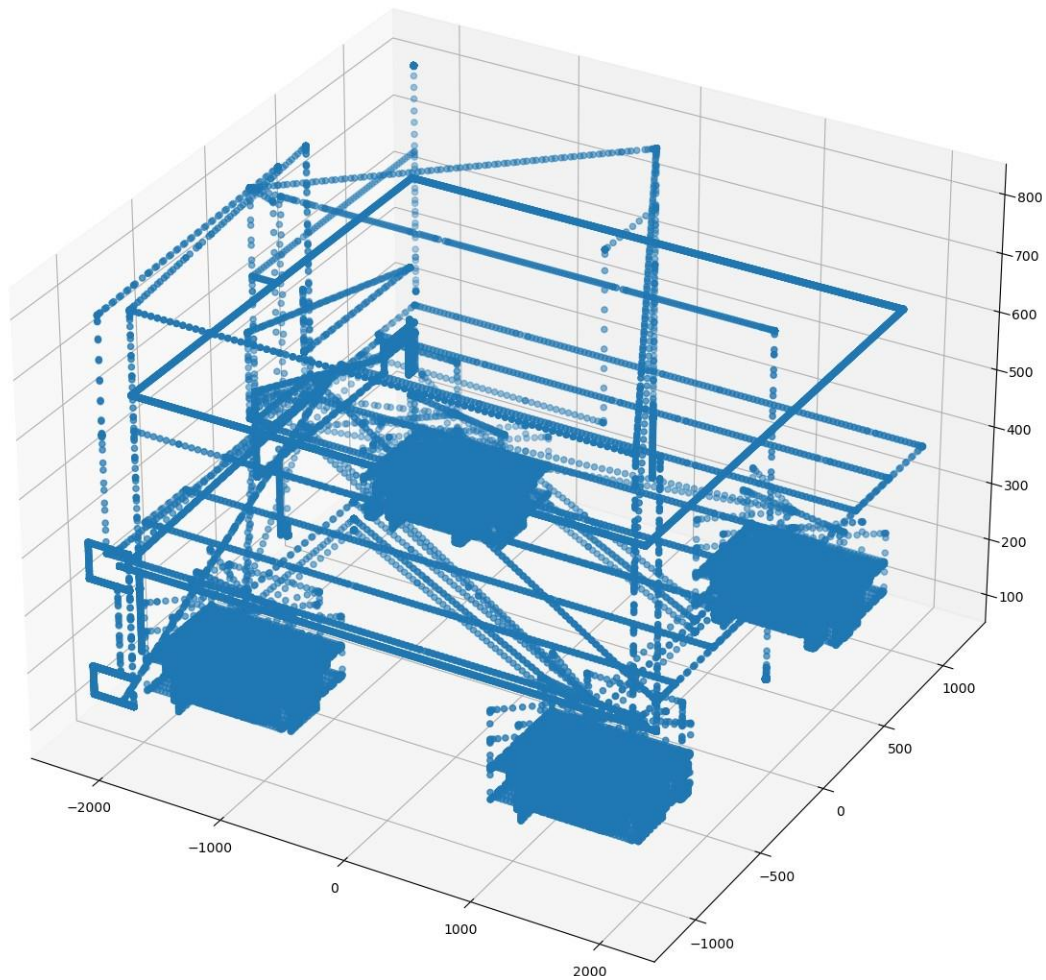
Lets try to apply the trained cluster model for program 35 and determine if it makes sense.

1.3.4 Application on Prog 35

[48]:
```
# select the new subdataset, corresponding to Program n°35.
filtered_data = HSM_data[HSM_data["id_ProgP"] == 35]
Input_cols = ["PosX","PosY","PosZ"]

# Visualize the machine-tool motions with df.plot:
filtered_data.plot(x="PosX", y="PosY")
plt.title("Machine-tool motions - Prog 35")
plt.show()

# If you want to make a 3D plot, 'PosZ' should be added to the data selection
fig = plt.figure(figsize=(15, 15))
ax = plt.axes(projection="3d")
```

```
ax.scatter3D(filtered_data['PosX'], filtered_data['PosY'],
    filtered_data['PosZ'])
plt.show()
```





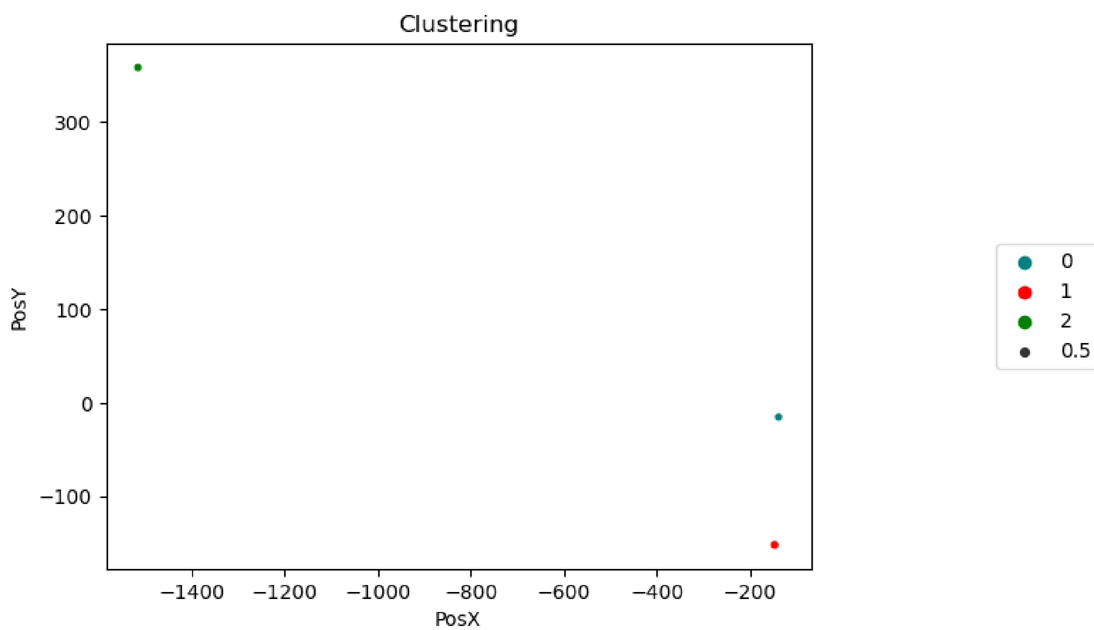
1.4 Clustering for a 2nd program : n°35

[49]:
```
# Clustering on Program 35:

# Appply the kmean model previously trained on this new dataset

# Define the cluster model (with max_iter=50,init='random')
kmeans = KMeans(n_clusters=3, max_iter=50, init='random')

# Train the kmeans model (centroids) from the dataset
kmeans.fit(filtered_data)

# Where are the centroids positions? (kmeans.cluster_centers_)
```

```python
centroids = kmeans.cluster_centers_

# Compute the inertia = intra-cluster variance (kmeans.inertia_)
inertia = kmeans.inertia_

# Prediction: affect each observation of the dataset, to the closest centroid␣
↪(kmeans.predict)
predictions = kmeans.predict(filtered_data)

# From the cluster label of each point in the dataset (array), make a DataFrame␣
↪and concatenate to the dataset
cluster_labels = pd.DataFrame({"predicted_cluster": predictions})
filtered_dff = pd.concat([filtered_data, cluster_labels], axis=1)

# Similarly, make a DataFrame with the centroid positions
centroids_dff = pd.DataFrame(centroids)

centroids_dff.columns=["tpsT", "tps B", "date", "id_ProgP", "id pc", "mode",␣
↪'id_outil',
"n outil", "usure outil", "nligne", "nbloc", "Abloc", "Cbloc", "Temp_1",
"Temp_2", "Temp_3", "Temp_4", "Arms_1", "Arms_2", "Arms_3", "Arms_4",
"Apic_1", "Apic_2", "Apic_3", "Apic_4", "Vrms_1", "Vrms_2", "Vrms_3",
"Vrms_4", "Vpic_1", "Vpic_2", "Vpic_3", "Vpic_4", "PosX", "PosY",
"PosZ", "PosA", "PosC", "VitX", "VitY", "VitZ", "VitA", "VitC", "Vf",
"N", "P", "%Vf", "%N", "FFT_1", "FFT_2", "FFT_3", "FFT_4", "FFT_5",
"FFT_6", "FFT_7", "FFT_8", "FFT_9", "FFT_10", "FFT_11", "FFT_12",
"FFT_13", "FFT_14", "FFT_15", "FFT_16", "FFT_17", "FFT_18", "FFT_19",
"FFT_20", "FFT_21", "FFT_22", "FFT_23", "FFT_24"]
print(centroids_dff)

# Visualize the ressults. Are they good?
sns.scatterplot( data=centroids_dff, x="PosX", y="PosY", hue=centroids_dff.
↪index, size=0.5, palette=color_dict_cluster)

#sns.scatterplot(x='PosX', y='PosY', hue='predicted_cluster', size=5,␣
↪palette=c'predicted_cluster'olor_dict_cluster, data=df_centroids)
plt.legend(loc="center left", bbox_to_anchor=(1.25, 0.5), ncol=1)
plt.title("Clustering")
plt.show()
```

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)

           tpsT         tps B          date  id_ProgP      id pc  mode  \
0  6.371834e+06  4.765310e+06  1.923176e+08      35.0  64.629486   2.0
```

```
1  6.275676e+06  4.680242e+06  1.921425e+08     35.0  56.372229  2.0
2  6.325864e+06  4.725865e+06  1.922334e+08     35.0  46.141769  2.0

    id_outil   n_outil  usure outil        nligne  ...      FFT_15    FFT_16  \
0  10.547548  3.275541   164.521233  23042.897319  ...  2512.255414  2.741424
1  16.473502  2.469923   225.458372   4110.116423  ...  2105.873216  1.879161
2  19.563834  3.016425   147.808214   9028.464270  ...  2358.302162  1.054745

     FFT_17    FFT_18       FFT_19       FFT_20       FFT_21    FFT_22  \
0  1.733755  1.234312  1312.041034  1524.936743  1495.850467  3.729713
1  1.289136  1.025263  1150.353093  1546.092871  1563.560600  2.228963
2  0.669084  0.533204  1522.364905  1833.640563  1822.309933  1.208282

     FFT_23    FFT_24
0  1.711569  1.387358
1  1.566943  1.134792
2  0.724953  0.592700

[3 rows x 72 columns]
```



[50]:
```python
import copy
Norm_32 = centroids_dff.reset_index()
print(Norm_32.keys())

#  Normalisation
```

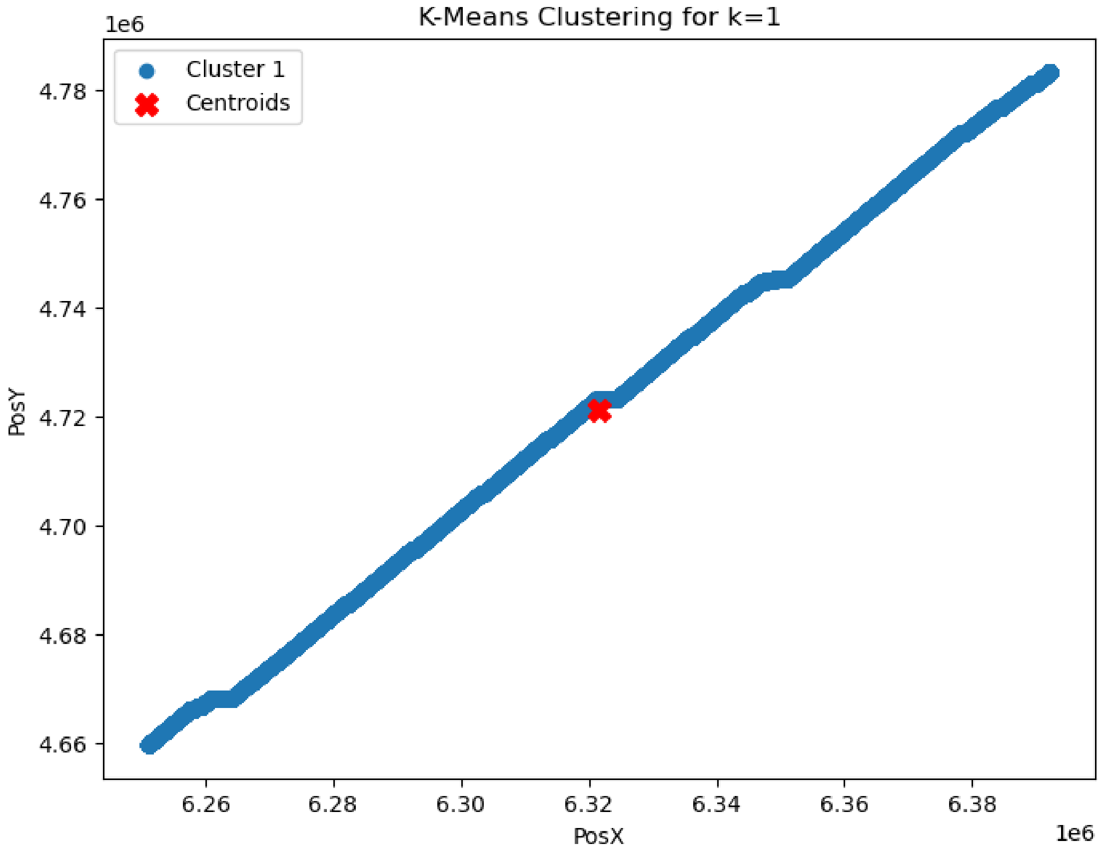
```
Norm_32[Input_cols]=(Norm_32[Input_cols]-Norm_32[Input_cols].min())/
  ↪(Norm_32[Input_cols].max()-Norm_32[Input_cols].min())
print(Norm_32[Input_cols])
```

```
Index(['index', 'tpsT', 'tps B', 'date', 'id_ProgP', 'id pc', 'mode',
       'id_outil', 'n outil', 'usure outil', 'nligne', 'nbloc', 'Abloc',
       'Cbloc', 'Temp_1', 'Temp_2', 'Temp_3', 'Temp_4', 'Arms_1', 'Arms_2',
       'Arms_3', 'Arms_4', 'Apic_1', 'Apic_2', 'Apic_3', 'Apic_4', 'Vrms_1',
       'Vrms_2', 'Vrms_3', 'Vrms_4', 'Vpic_1', 'Vpic_2', 'Vpic_3', 'Vpic_4',
       'PosX', 'PosY', 'PosZ', 'PosA', 'PosC', 'VitX', 'VitY', 'VitZ', 'VitA',
       'VitC', 'Vf', 'N', 'P', '%Vf', '%N', 'FFT_1', 'FFT_2', 'FFT_3', 'FFT_4',
       'FFT_5', 'FFT_6', 'FFT_7', 'FFT_8', 'FFT_9', 'FFT_10', 'FFT_11',
       'FFT_12', 'FFT_13', 'FFT_14', 'FFT_15', 'FFT_16', 'FFT_17', 'FFT_18',
       'FFT_19', 'FFT_20', 'FFT_21', 'FFT_22', 'FFT_23', 'FFT_24'],
      dtype='object')
       PosX      PosY      PosZ
0  1.000000  0.268017  0.000000
1  0.993687  0.000000  0.737122
2  0.000000  1.000000  1.000000
```

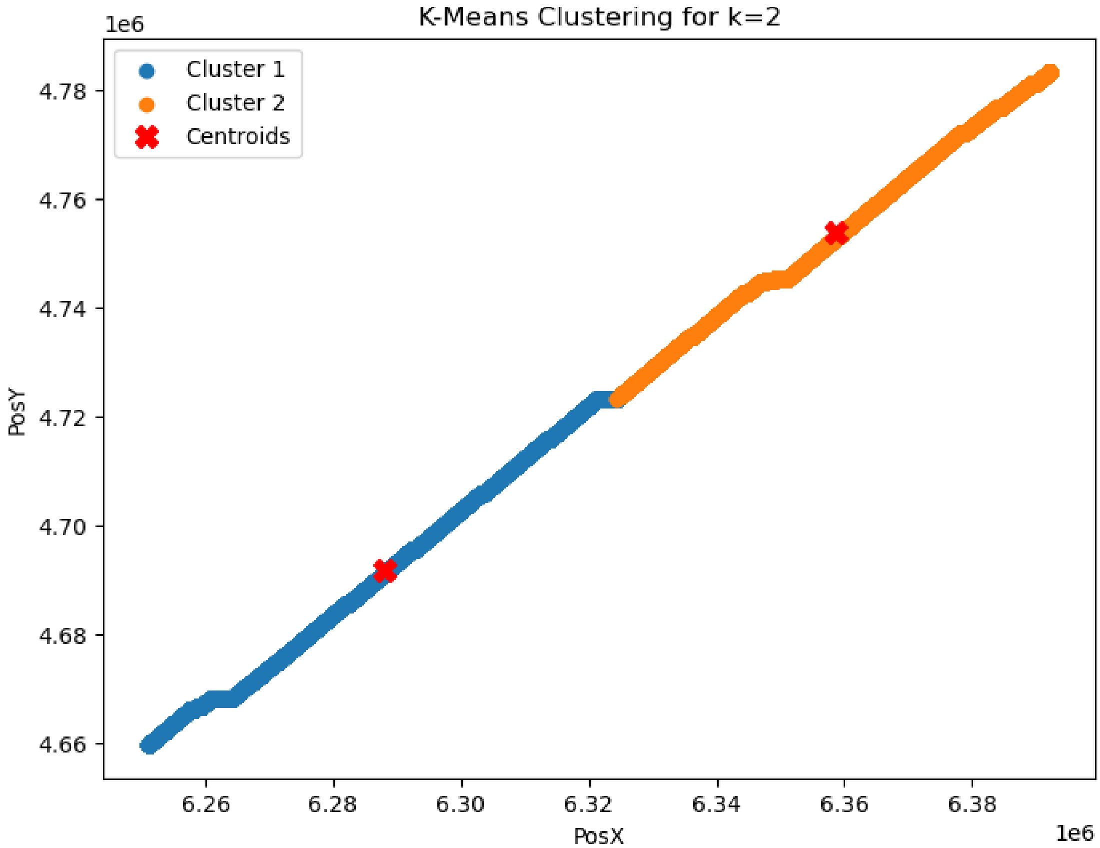
[52]:
```
def find_optimal_clusters(range_n_clusters, ssd):
    deltas = np.diff(ssd, 2)
    elbow_index = np.argmax(deltas) + 2
    optimal_clusters = range_n_clusters[elbow_index - 1]
    return optimal_clusters

def My_function_kmeans_elbow(max_clusters, df):
    ssd = []
    range_n_clusters = np.arange(1, max_clusters + 1, 1)
    print(range_n_clusters)
    for num_clusters in range_n_clusters:
        # Launch the clustering
        kmeans = KMeans(n_clusters=num_clusters)
        kmeans.fit(df)
        ssd.append(kmeans.inertia_)

        # Plotting the clustering
        plt.figure(figsize=(8, 6))
        for i in range(num_clusters):
            cluster_indices = np.where(kmeans.labels_ == i)[0]
            plt.scatter(df.iloc[cluster_indices, 0], df.iloc[cluster_indices,␣
  ↪1], label=f"Cluster {i + 1}")
        plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,␣
  ↪1], s=100, c="red", marker="X", label="Centroids")
        plt.title(f"K-Means Clustering for k={num_clusters}")
        plt.xlabel("PosX")
        plt.ylabel("PosY")
```

```
        plt.legend()
        plt.show()

    # Plotting Elbow Curve for Optimal Clusters
    plt.plot(range_n_clusters, ssd, marker='o')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia (Sum of Squared Distances)')
    plt.title('Elbow Curve for Optimal Clusters')
    plt.show()

    # Find the optimal number of clusters
    optimal_clusters = find_optimal_clusters(range_n_clusters, ssd)
    print("Optimal number of clusters:", optimal_clusters)
    return optimal_clusters

# Example usage
# Assuming filtered_dff is defined somewhere in your code
optimal_clusters = My_function_kmeans_elbow(9, filtered_data)
print("Optimal number of clusters:", optimal_clusters)
```

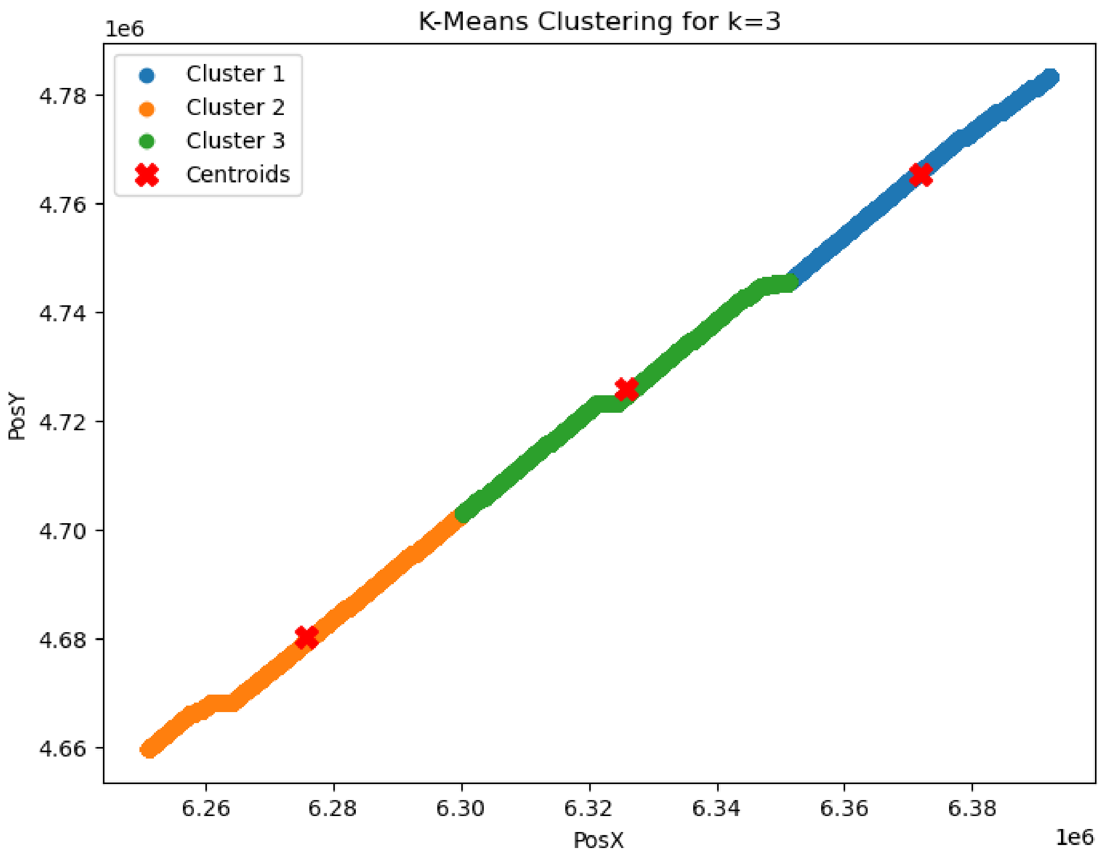
```
[1 2 3 4 5 6 7 8 9]

C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```



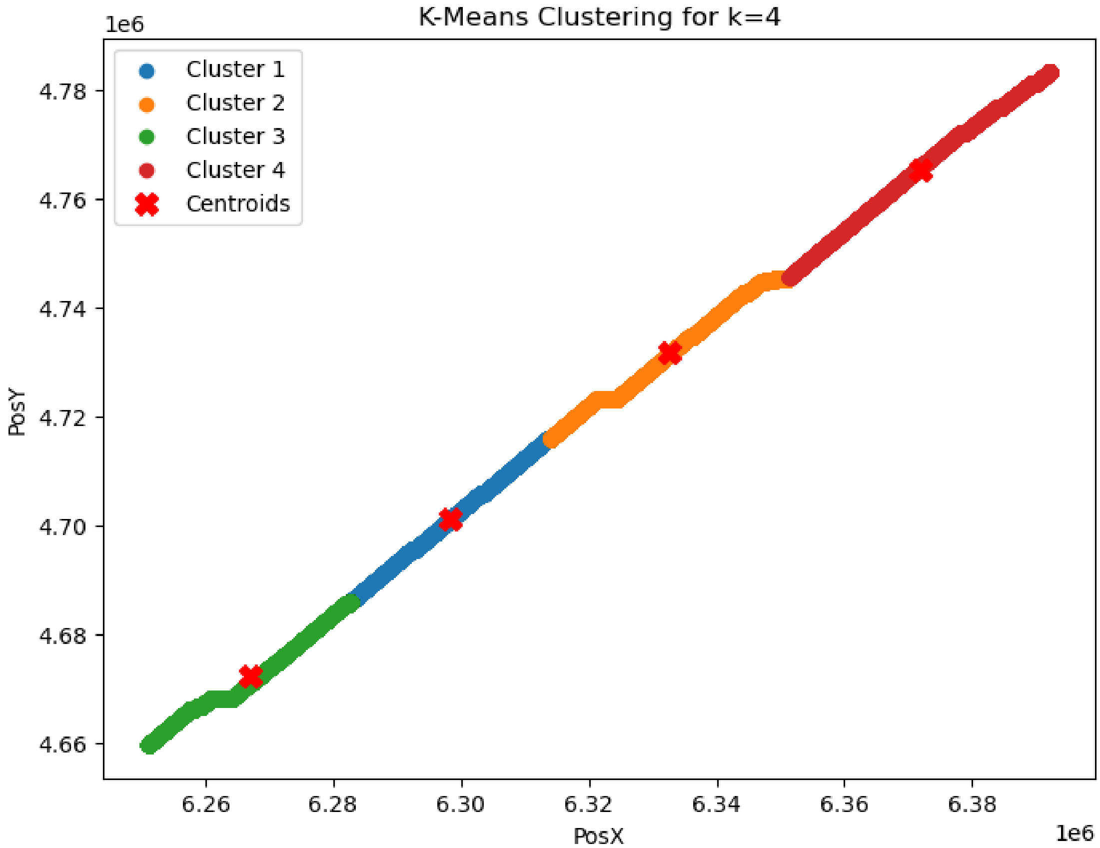
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



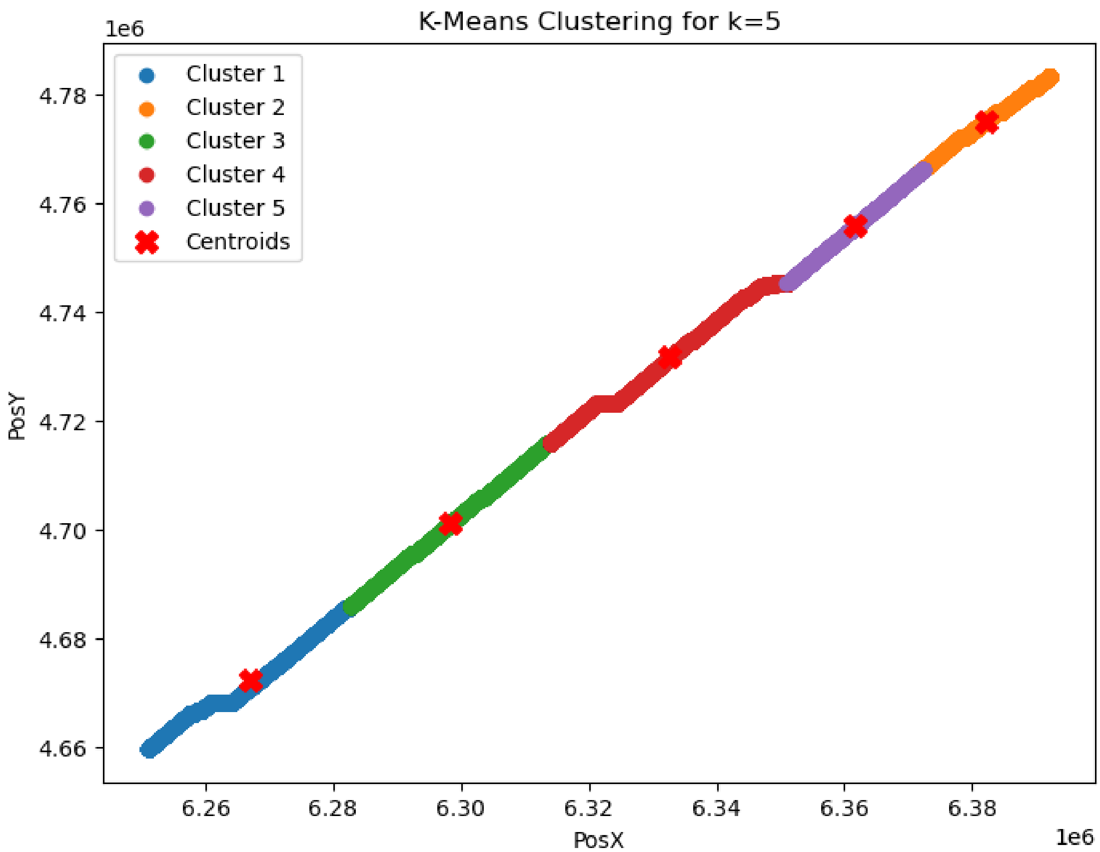
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



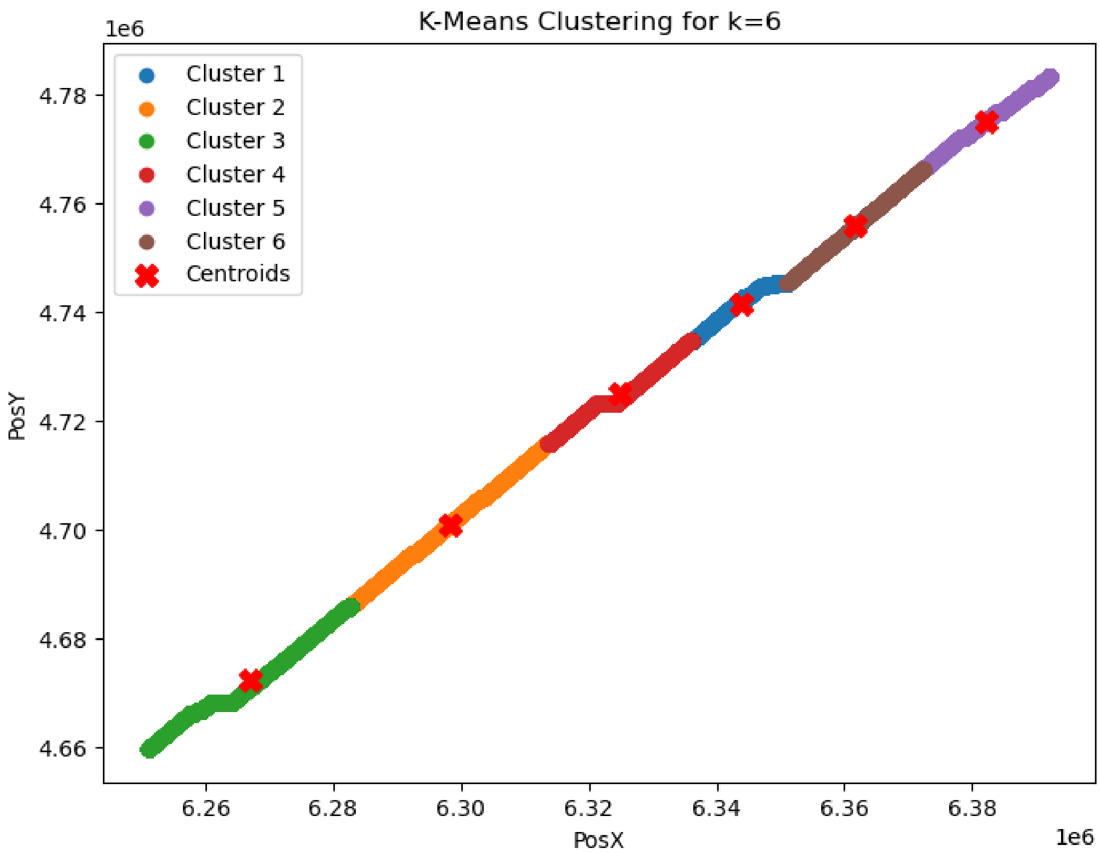
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



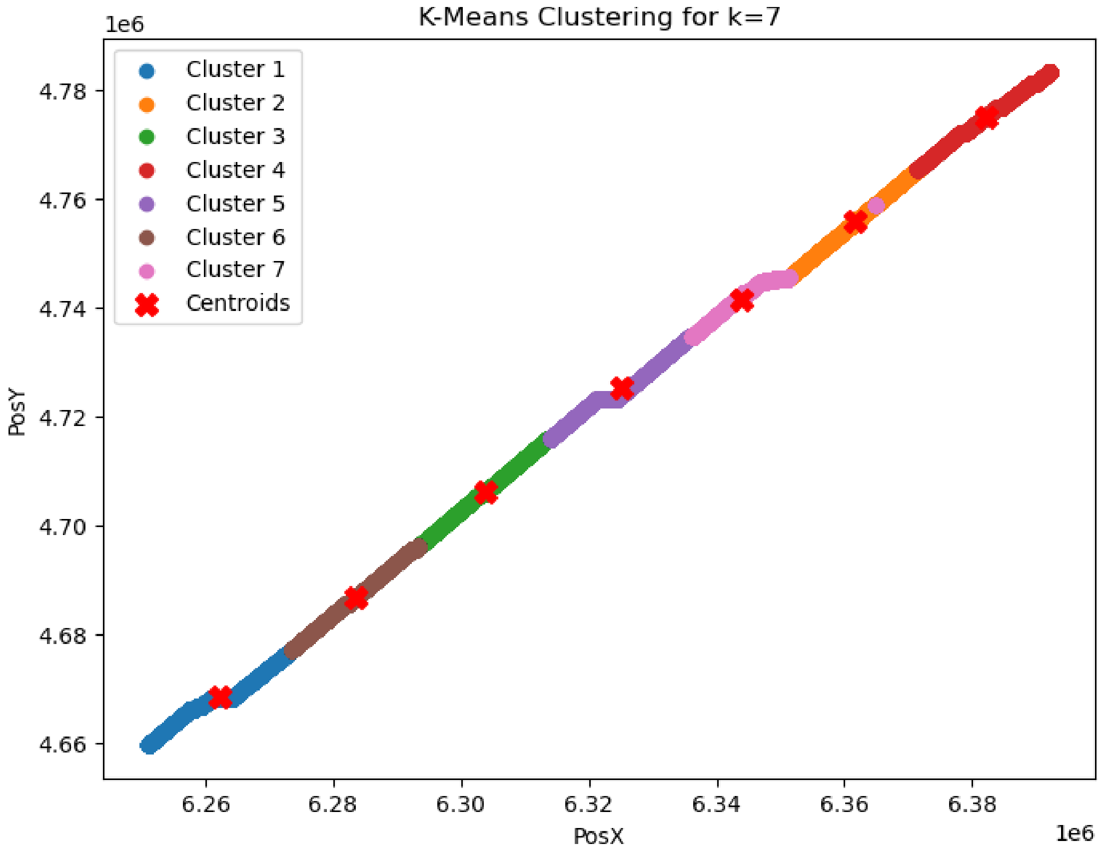
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



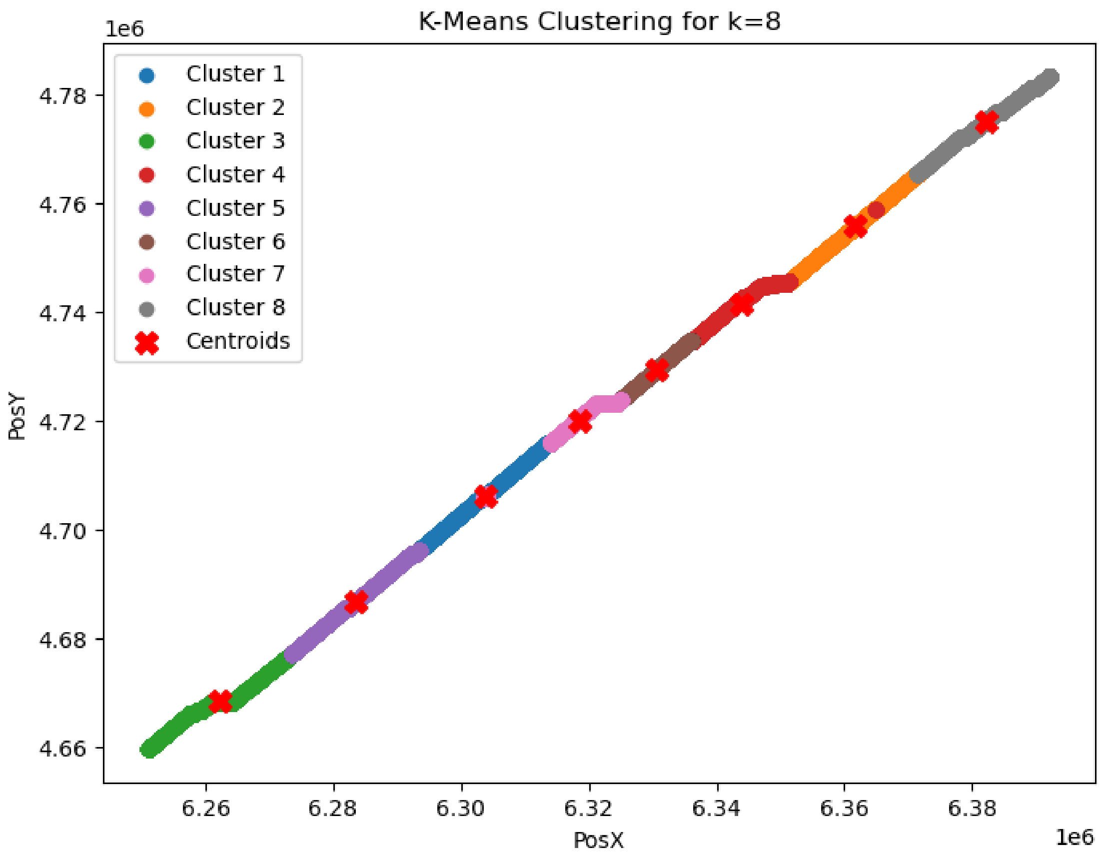
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



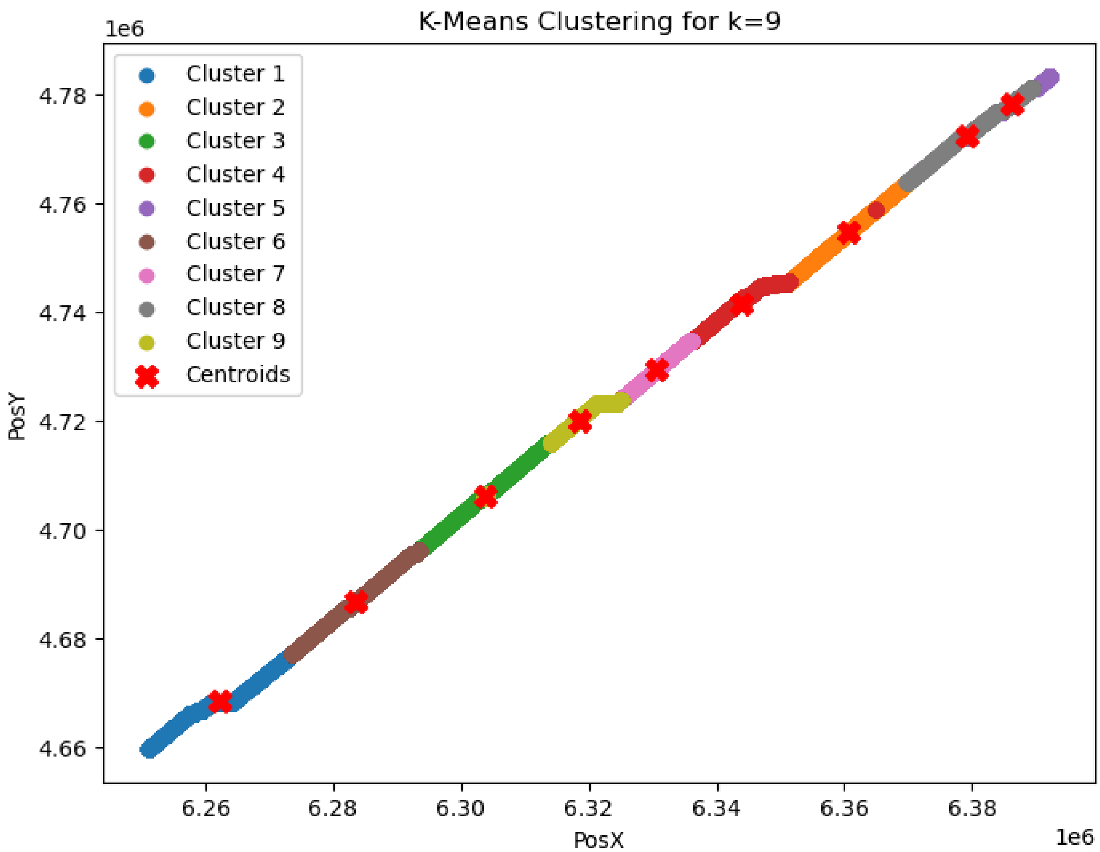
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)

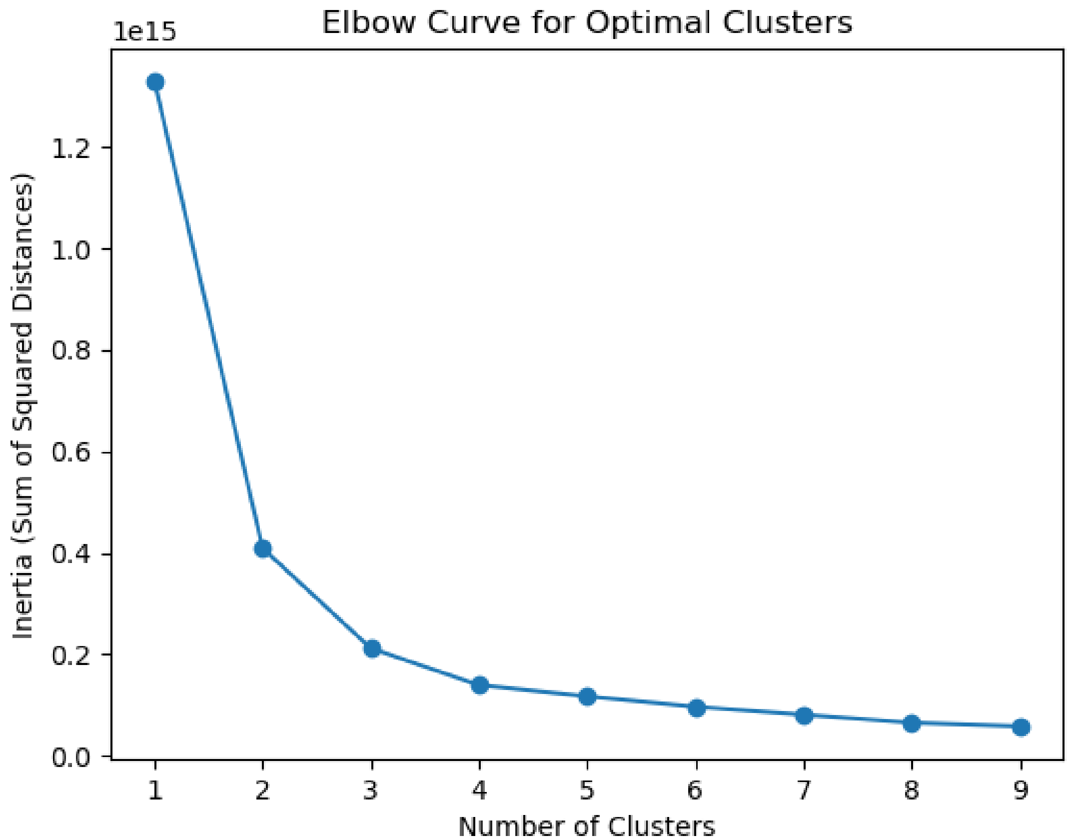


C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)


K-Means Clustering for k=9
1e6
4.78
4.76
4.74
4.72
4.70
4.68
4.66
PosY
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Centroids
6.26
6.28
6.30
6.32
6.34
6.36
6.38
PosX
1e6




```
Optimal number of clusters: 2
Optimal number of clusters: 2
```

The clustering cannot be used per se, yet the method can be replicated

2 GMM

An alternative to distance-based technics of clustering is statistical ones, such as Gaussian Mixture Model (GMM).

[64]:
```python
from sklearn import mixture
from sklearn.datasets import make_blobs
from sklearn.mixture import GaussianMixture
```

[65]:
```python
#funtion to draw mutlivariate Gaussian
def multivariate_gaussian(pos, mu, Sigma):
    """Return the multivariate Gaussian distribution on array pos.
    pos is an array constructed by packing the meshed arrays of variables
    x_1, x_2, x_3, ..., x_k into its _last_ dimension.
    """
```

```
    n = mu.shape[0]
    Sigma_det = np.linalg.det(Sigma)
    Sigma_inv = np.linalg.inv(Sigma)
    N = np.sqrt((2*np.pi)**n * Sigma_det)

    # This einsum call calculates (x-mu)T.Sigma-1.(x-mu) in a vectorized
    # way across all the input variables.
    fac = np.einsum('...k,kl,...l->...', pos-mu, Sigma_inv, pos-mu)

    return np.exp(-fac / 2) / N
```

[66]:
```
# Generate sample data (replace this with your actual data)

filtered_dff, _ = make_blobs(n_samples=300, centers=3, random_state=42)
```

[67]:
```
# Tests of the GMM functions:

# define the GMM model
nb_GMM=3
gmm = GaussianMixture(n_components=nb_GMM, covariance_type="full")

# learning of the GMM model by EM algo (Expectation Maximisation)
gmm.fit(filtered_dff)

# result? m=gmm.means_
cov=gmm.covariances_
w=gmm.weights_
```

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=2.
  warnings.warn(
```

[68]:
```
# apply the GMM model, as a classifier:
gmm.predict(filtered_dff)
```

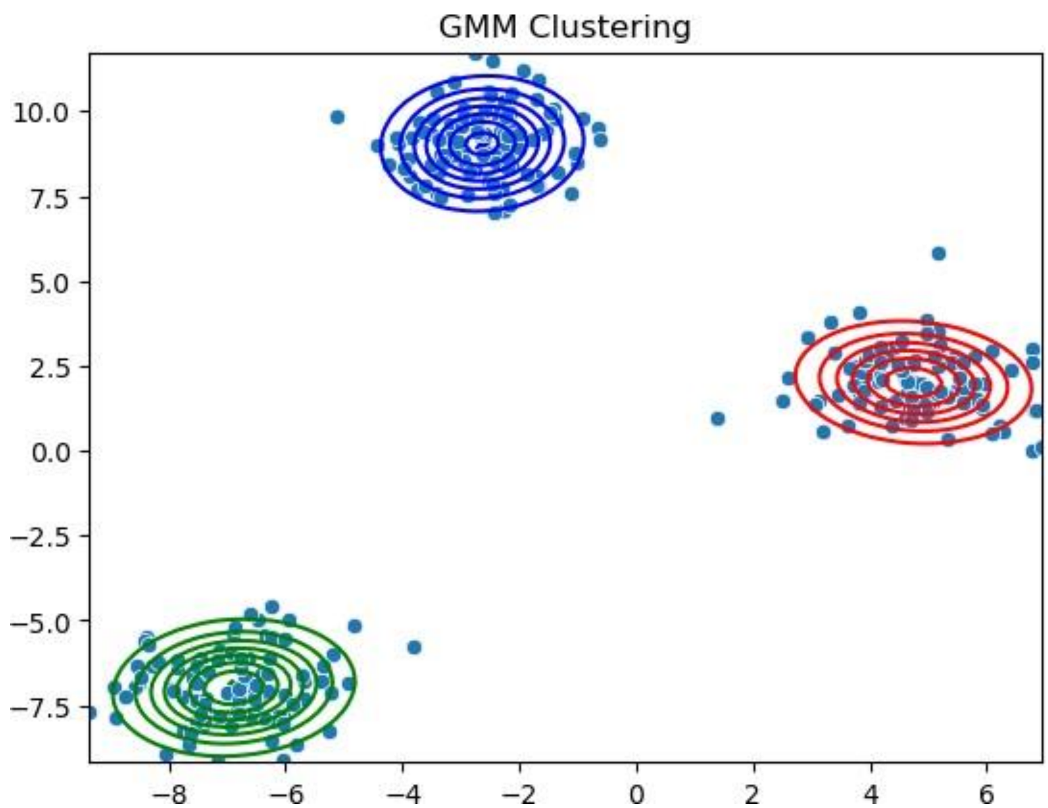
[68]:
```
array([1, 1, 2, 0, 1, 0, 2, 0, 2, 2, 2, 0, 2, 2, 1, 2, 1, 0, 2, 2, 2, 2,
       0, 1, 2, 1, 1, 0, 0, 2, 2, 2, 1, 2, 1, 2, 1, 0, 1, 0, 0, 2, 1, 0,
       2, 2, 1, 0, 1, 0, 0, 1, 1, 2, 1, 0, 1, 2, 0, 2, 1, 0, 0, 1, 1, 0,
       0, 1, 1, 2, 0, 1, 1, 2, 2, 1, 1, 0, 2, 0, 2, 2, 1, 2, 0, 1, 1, 2,
       0, 2, 1, 2, 1, 2, 2, 1, 1, 2, 1, 1, 0, 2, 0, 2, 2, 2, 2, 2, 0, 1,
       0, 2, 2, 2, 2, 0, 1, 0, 1, 0, 0, 0, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2,
       2, 2, 2, 0, 0, 1, 2, 1, 2, 2, 1, 2, 0, 0, 0, 2, 0, 2, 2, 1, 0, 1,
       2, 0, 0, 1, 1, 2, 2, 1, 1, 1, 2, 1, 0, 2, 2, 2, 2, 2, 0, 2, 0, 0,
       0, 2, 0, 0, 1, 2, 1, 0, 0, 1, 0, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
```

```
       2, 1, 2, 2, 0, 0, 2, 0, 1, 1, 0, 2, 2, 1, 0, 0, 1, 1, 1, 1, 2, 1,
       1, 0, 1, 1, 2, 0, 1, 1, 0, 2, 2, 1, 2, 1, 0, 0, 1, 0, 1, 1, 1, 0,
       0, 2, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 2, 1, 2, 2, 2, 1, 2,
       0, 0, 1, 0, 0, 2, 2, 0, 0, 0, 1, 1, 1, 2, 2, 2, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 1, 2, 0, 0, 2, 1, 2, 0, 2, 1, 1], dtype=int64)
```

[70]:

```
# Define colors for clusters (replace with your actual color choices)
color_dict_cluster = ['red', 'green', 'blue']

# Plotting of GMM
x = np.linspace(filtered_dff[:, 0].min(), filtered_dff[:, 0].max(), 100)
y = np.linspace(filtered_dff[:, 1].min(), filtered_dff[:, 1].max(), 100)
X, Y = np.meshgrid(x, y)
pos = np.empty(X.shape + (2,))
pos[:, :, 0] = X
pos[:, :, 1] = Y
for i in range(nb_GMM):
 mu_broadcast = np.expand_dims(m[i], axis=(0, 1))
 cov_inv = np.linalg.inv(cov[i])

 # Calculate the squared Mahalanobis distance directly
 diff = pos - mu_broadcast
 fac = np.sum(diff @ cov_inv * diff, axis=-1)
 Z = np.exp(-fac / 2) / np.sqrt((2 * np.pi)**2 * np.linalg.det(cov[i]))
 plt.contour(X, Y, Z, colors=color_dict_cluster[i])
 plt.scatter(m[i, 0], m[i, 1], marker="X", c=color_dict_cluster[i], s=30)

# Scatter plot of the original data
sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1],␣
 ↪palette=color_dict_cluster)
plt.title("GMM Clustering")
plt.show()
```

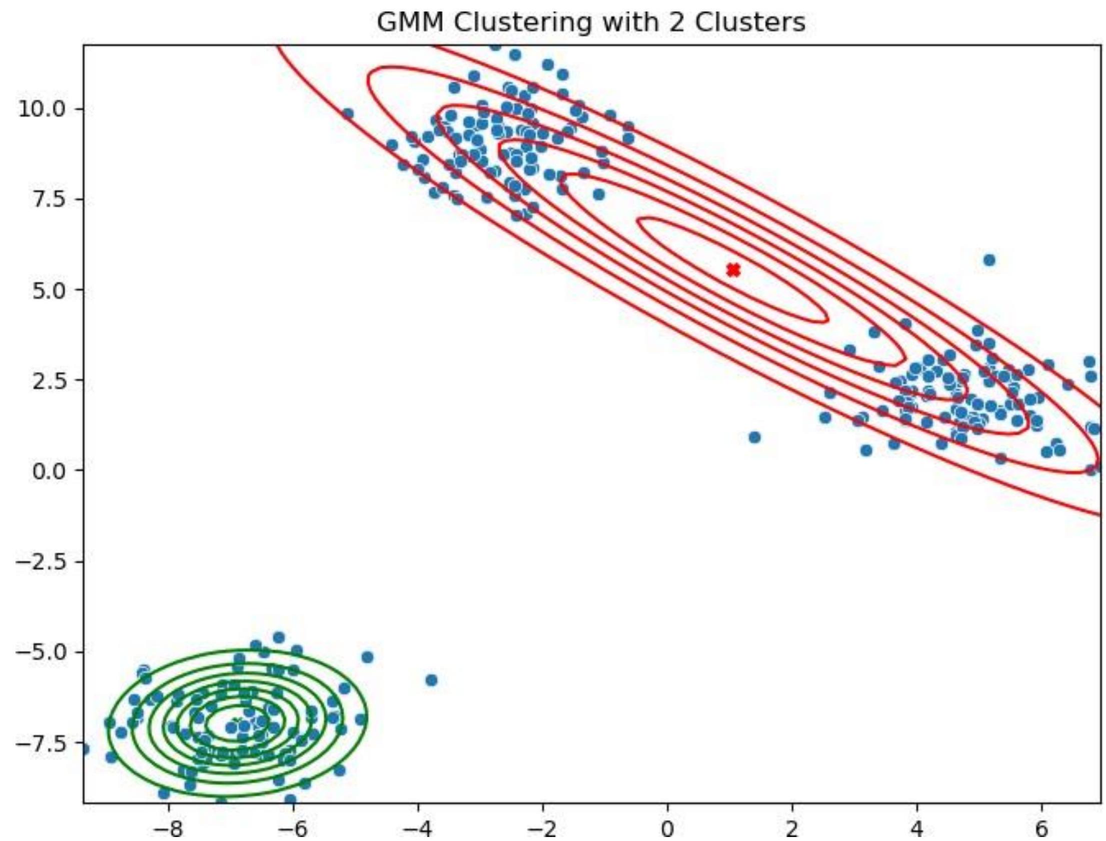
```
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\95453402.py:23: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1],
palette=color_dict_cluster)
```



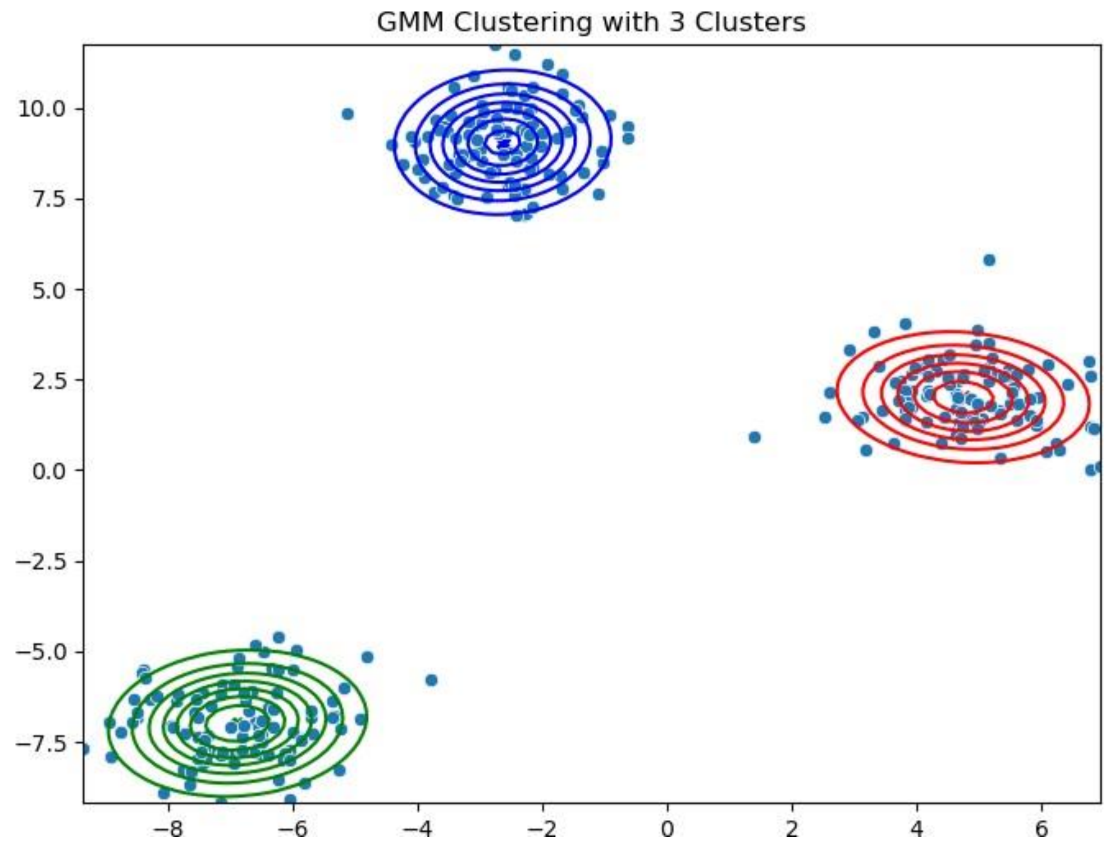
[73]:
```
# Program a FOR loop for clustering with GMM and associated visualisations

# Optimise the clusters number
# Generate sample data (replace this with your actual data)
filtered_dff, _ = make_blobs(n_samples=300, centers=3, random_state=42)

# Define the range of components for GMM
min_clusters = 2
max_clusters = 6

# Plotting of GMM for different numbers of clusters
for nb_GMM in range(min_clusters, max_clusters + 1):

    # Fit the GMM model
    gmm = GaussianMixture(n_components=nb_GMM, covariance_type="full")
    gmm.fit(filtered_dff)

    # Get the GMM parameters
    m = gmm.means_
    cov = gmm.covariances_
    w = gmm.weights_
```

```python
# Define colors for clusters (replace with your actual color choices)
color_dict_cluster = ['red', 'green', 'blue']

# Adjust the colors if there are fewer colors than clusters
color_dict_cluster *= (nb_GMM // len(color_dict_cluster)) + 1

# Plotting of GMM
x = np.linspace(filtered_dff[:, 0].min(), filtered_dff[:, 0].max(), 100)
y = np.linspace(filtered_dff[:, 1].min(), filtered_dff[:, 1].max(), 100)
X, Y = np.meshgrid(x, y)
pos = np.empty(X.shape + (2,))
pos[:, :, 0] = X
pos[:, :, 1] = Y
plt.figure(figsize=(8, 6))
for i in range(nb_GMM):
    mu_broadcast = np.expand_dims(m[i], axis=(0, 1))
    cov_inv = np.linalg.inv(cov[i])

    # Calculate the squared Mahalanobis distance directly
    diff = pos - mu_broadcast
    fac = np.sum(diff @ cov_inv * diff, axis=-1)
    Z = multivariate_gaussian(pos, m[i], cov[i])
    plt.contour(X, Y, Z, colors=color_dict_cluster[i])
    plt.scatter(m[i, 0], m[i, 1], marker="X", c=color_dict_cluster[i], s=30)

# Scatter plot of the original data
sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1],
↪palette=color_dict_cluster)
plt.title(f"GMM Clustering with {nb_GMM} Clusters")
plt.show()
```

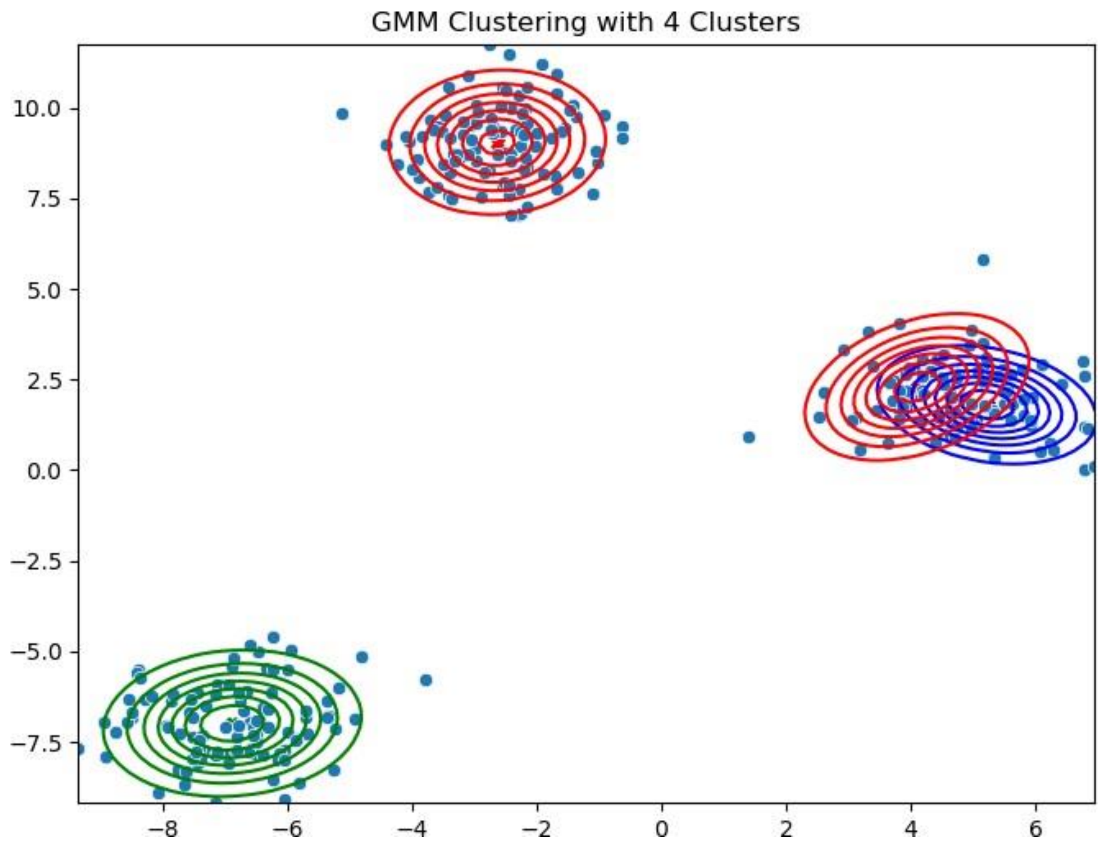
```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1167243084.py:40: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1],
palette=color_dict_cluster)
```



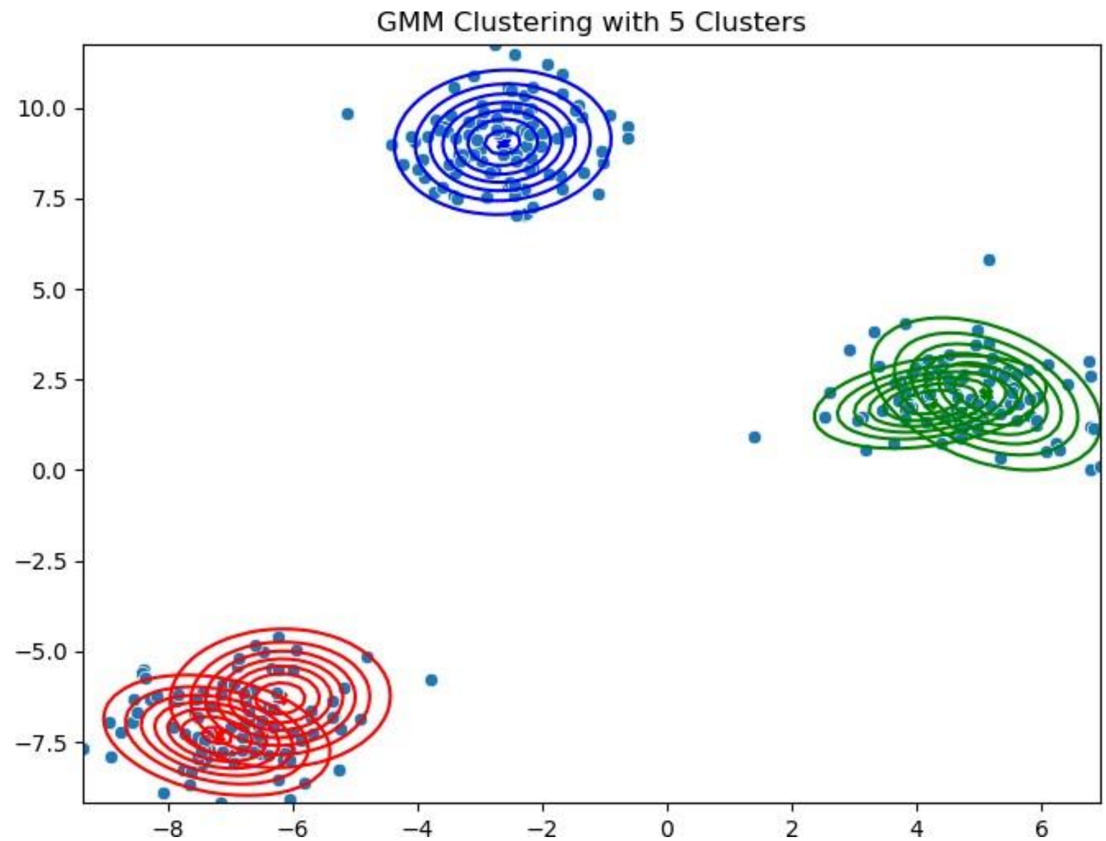
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1167243084.py:40: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1], palette=color_dict_cluster)



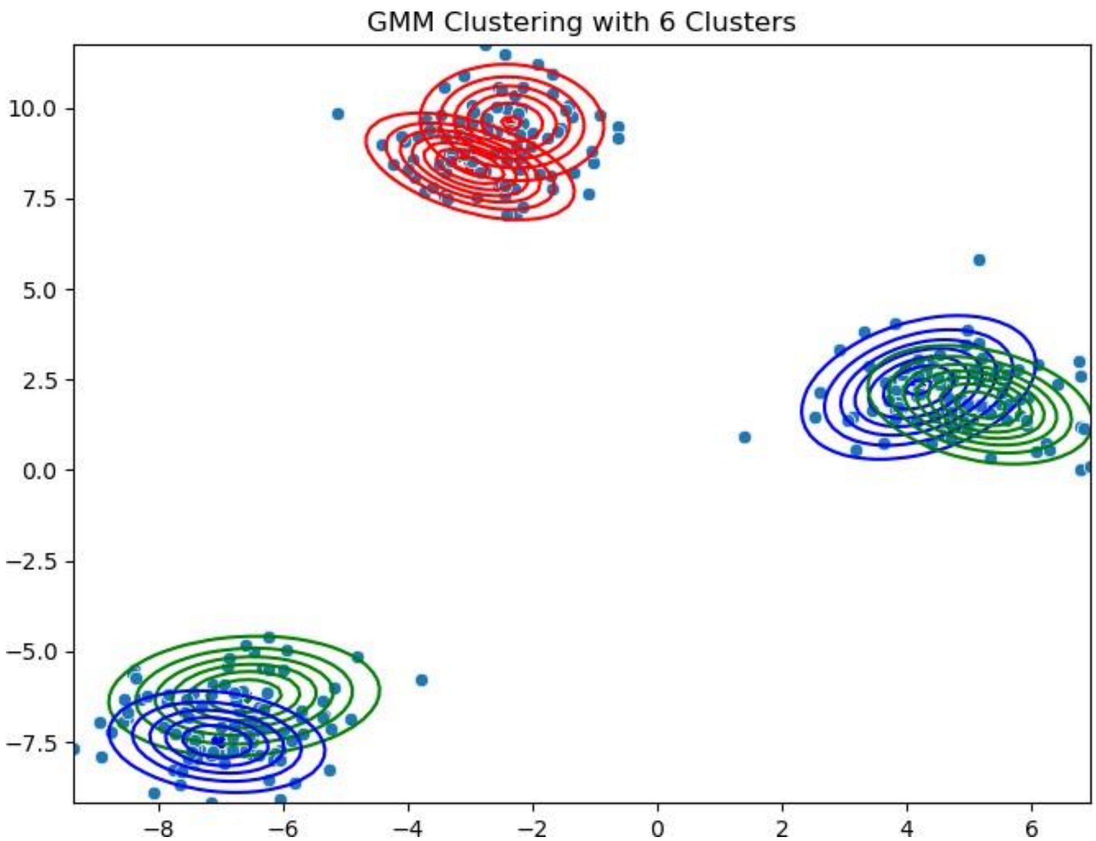
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1167243084.py:40: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1], palette=color_dict_cluster)




```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1167243084.py:40: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1], palette=color_dict_cluster)
```



C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1167243084.py:40: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=filtered_dff[:, 0], y=filtered_dff[:, 1], palette=color_dict_cluster)



2.1 Hierarchical Ascendant Classification

(optional)

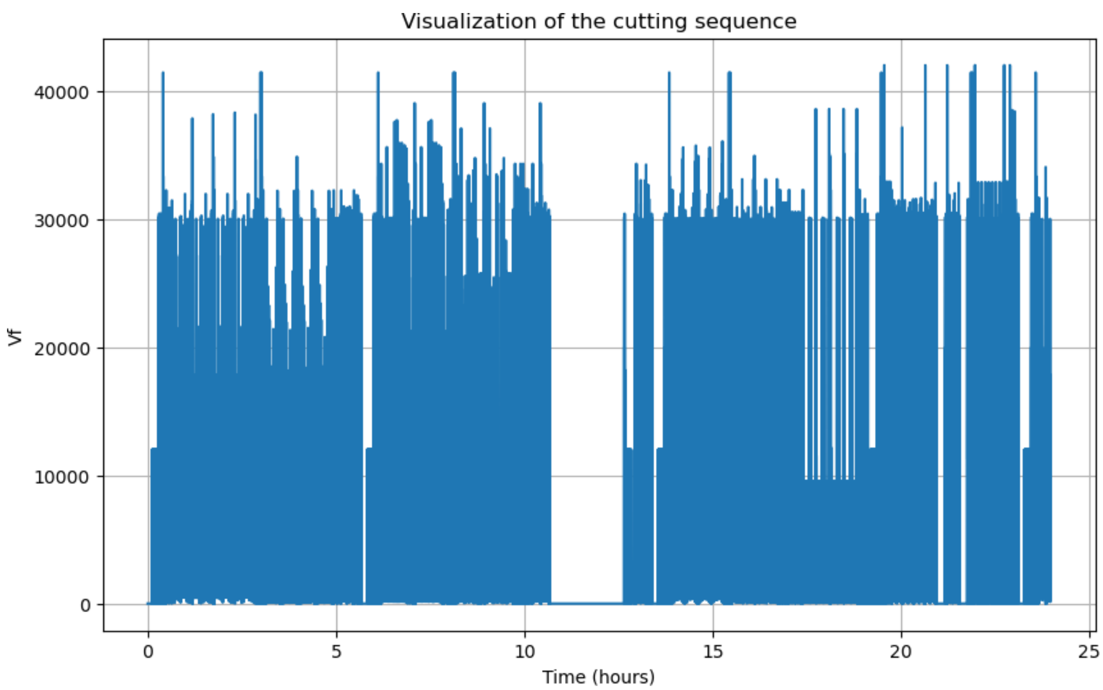
3 Objective 2: productivity

The objective is to determine how long the machine-tool has been cutting, by unsupervised machine learning, and computation of the OEE. In this section, several variables will be used as input of the Machine Learning, the output consists in 2 clusters (k=2), corresponding to: the machine-tool is machining, or not. The performance of data-driven approach will be compared with a knowledge-based approach (that combines data and knowledge integration through business rules).

[74]:
```
#select a sub-dataset associated the cutting process.

# Data Selection:

# Select only two specific columns
selected_columns = ['Vf', 'N']
cuttingdata =HSM_data[selected_columns]
print(cuttingdata)
```

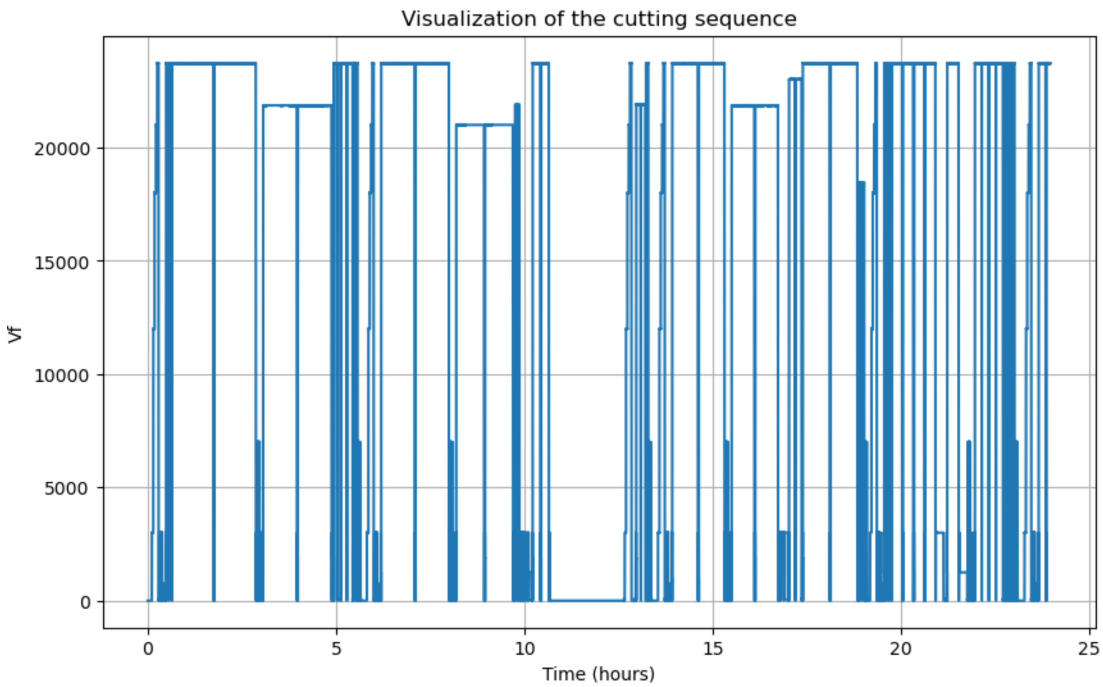
```
             Vf          N
0         0.000      0.000
1         0.000      0.000
2         0.000      0.000
3         0.000      0.000
4         0.000      0.000
...         ...        ...
862569  17789.728  23723.903
862570  17789.728  23717.312
862571  17789.728  23718.044
862572  17789.728  23725.368
862573  17789.728  23720.608

[862574 rows x 2 columns]
```

[75]:
```
## There are 862574 length of data

## Visualisation of the production sequence:
#tmp=np.arange(0,nb_specimen*0.1,0.1)
#tmpH=tmp/3600
#tmpH # can be imported in DataFrame for abscissa x=..
#data_panda.plot(y='VariableName_XXX')

# Define the number of specimens
nb_specimen = 862574

# Create a time array with 0.1 second intervals
tmp = np.arange(0, nb_specimen * 0.1, 0.1)

# Convert time to hours
tmpH = tmp / 3600

# Assuming you have a DataFrame named 'data' with a column named 'id_ProgP'
Vf = HSM_data["Vf"]

# Create a plot
plt.figure(figsize=(10, 6))
plt.plot(tmpH, Vf)
plt.xlabel("Time (hours)")
plt.ylabel("Vf")
plt.title("Visualization of the cutting sequence")
plt.grid(True)
plt.show()
```



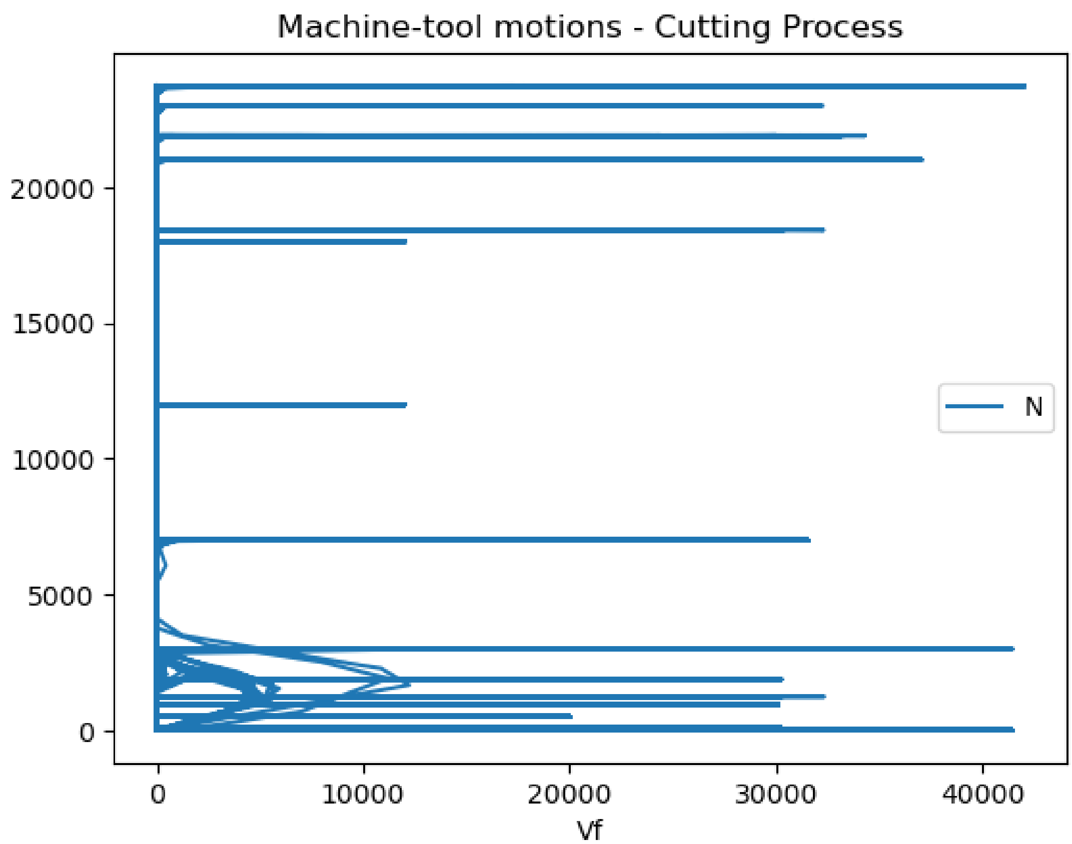
[76]:
```
# There are 862574 length of data

## Visualisation of the production sequence:
#tmp=np.arange(0,nb_specimen*0.1,0.1)
#tmpH=tmp/3600
#tmpH # can be imported in DataFrame for abscissa x=..
#data_panda.plot(y='VariableName_XXX')

# Define the number of specimens
nb_specimen = 862574

# Create a time array with 0.1 second intervals
tmp = np.arange(0, nb_specimen * 0.1, 0.1)

# Convert time to hours
tmpH = tmp / 3600

# Assuming you have a DataFrame named 'data' with a column named 'id_ProgP'
N = HSM_data["N"]
# Create a plot
plt.figure(figsize=(10, 6))
plt.plot(tmpH, N)
plt.xlabel("Time (hours)")
plt.ylabel("Vf")
```

```
plt.title('Visualization of the cutting sequence')
plt.grid(True)
plt.show()
```



[77]:
```
# To facilitate futur use, we can create a set with the variables labels,

Input_cols = ['Vf','N']
```

[78]:
```
# plots

# Visualize the machine-tool motions with df.plot:
cuttingdata.plot(x='Vf', y='N')
plt.title('Machine-tool motions - Cutting Process')
plt.show()
```



3.1 Unsupervised machine learning with Kmeans

[79]:
```
## First tests of KMeans, progressively:

# Define the cluster model (with max_iter=50,init='random')
kmeans = KMeans(n_clusters=3, max_iter=50, init='random')

# Train the kmeans model (centroids) from the dataset
kmeans.fit(cuttingdata)

# Where are the centroids positions? (kmeans.cluster_centers_)
centroids_cutting= kmeans.cluster_centers_

# Compute the inertia = intra-cluster variance (kmeans.inertia_)
inertia_cutting = kmeans.inertia_

# Prediction: affect each observation of the dataset, to the closest centroid␣
↪(kmeans.predict)
predictions_cutting = kmeans.predict(cuttingdata)
```

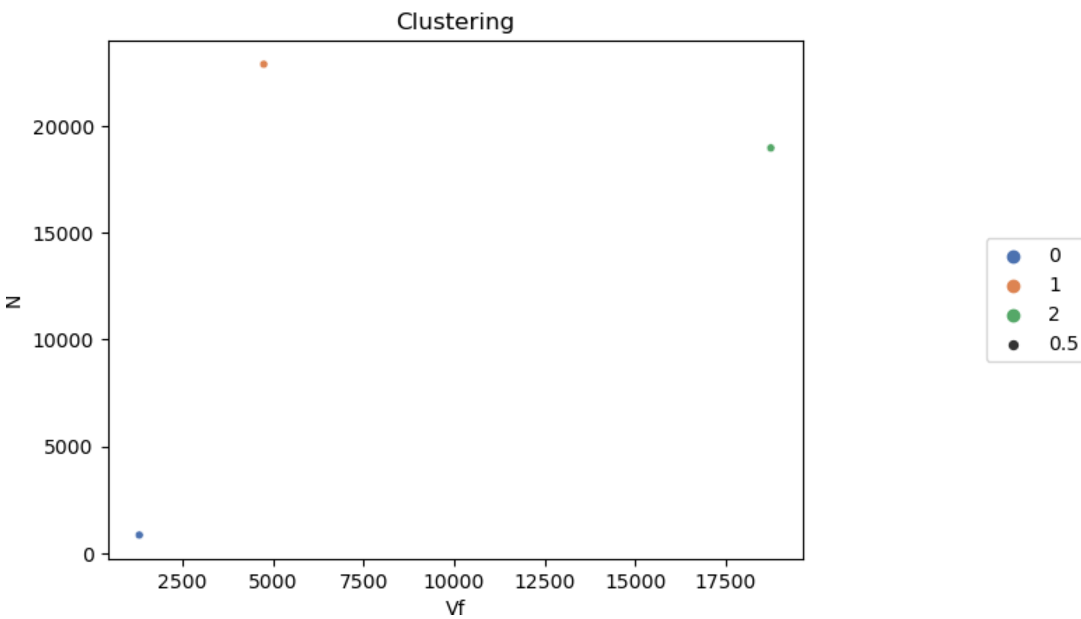
```
# From the cluster label of each point in the dataset (array), make a DataFrame␣
↪and concatenate to the dataset
cluster_labels = pd.DataFrame({"predicted_cluster": predictions_cutting})
cuttingdf = pd.concat([cuttingdata, cluster_labels], axis=1)

# Similarly, make a DataFrame with the centroid positions
centroids_cutting = pd.DataFrame(centroids_cutting)
```

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

[80]:
```
centroids_cutting.columns=[ "Vf", "N"]
print(centroids_cutting)
```

```
             Vf             N
0   1298.508587    859.341054
1   4738.521140  22883.788794
2  18751.309910  18967.638641
```

[81]:
```
## Visualize the results of clustering

sns.scatterplot( data=centroids_cutting, x="Vf", y="N", hue=centroids_cutting.
↪index, size=0.5, palette="deep")
plt.legend(loc="center left", bbox_to_anchor=(1.25, 0.5), ncol=1)
plt.title("Clustering")
plt.show()
```



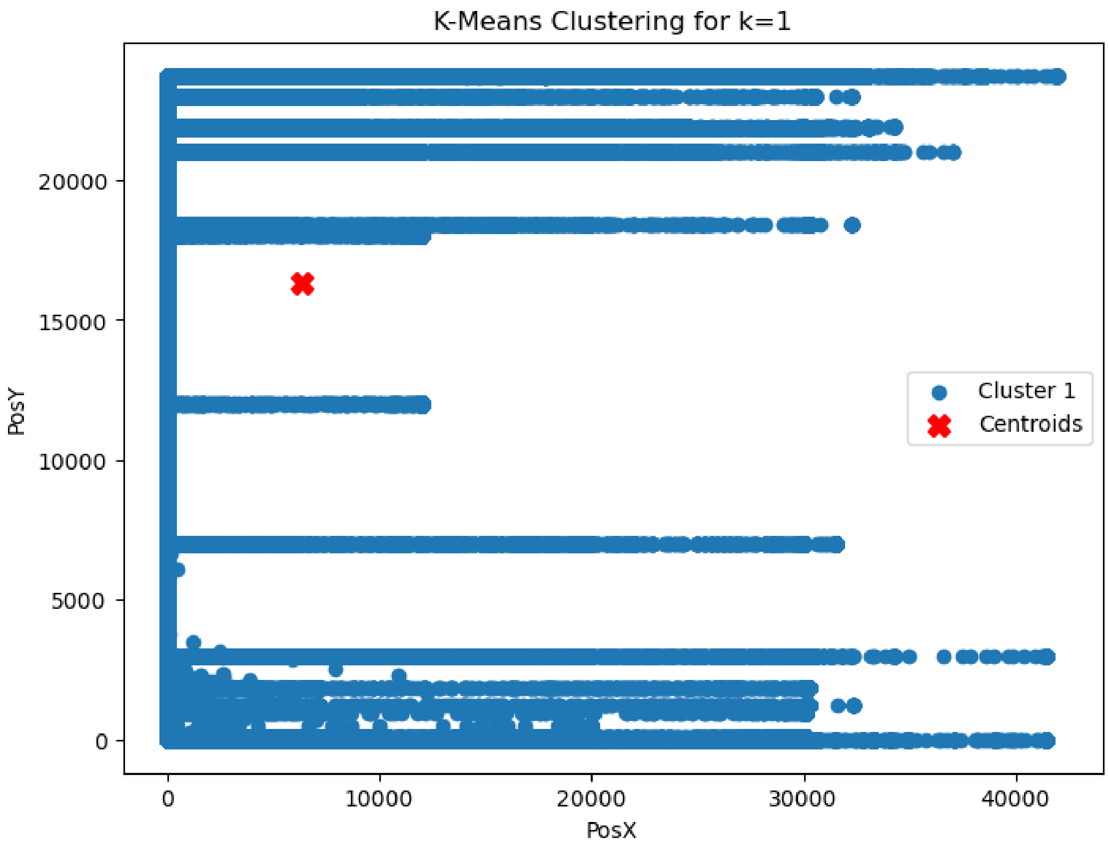
[82]:
```python
import copy
Norm_cutting = centroids_cutting.reset_index()
print(Norm_cutting.keys())

#  Normalisation
Norm_cutting[Input_cols]=(Norm_cutting[Input_cols]-Norm_cutting[Input_cols].
↪min())/(Norm_cutting[Input_cols].max()-Norm_cutting[Input_cols].min())
print(Norm_cutting[Input_cols])
```

```
Index(['index', 'Vf', 'N'], dtype='object')
         Vf         N
0  0.000000  0.000000
1  0.197104  1.000000
2  1.000000  0.822191
```

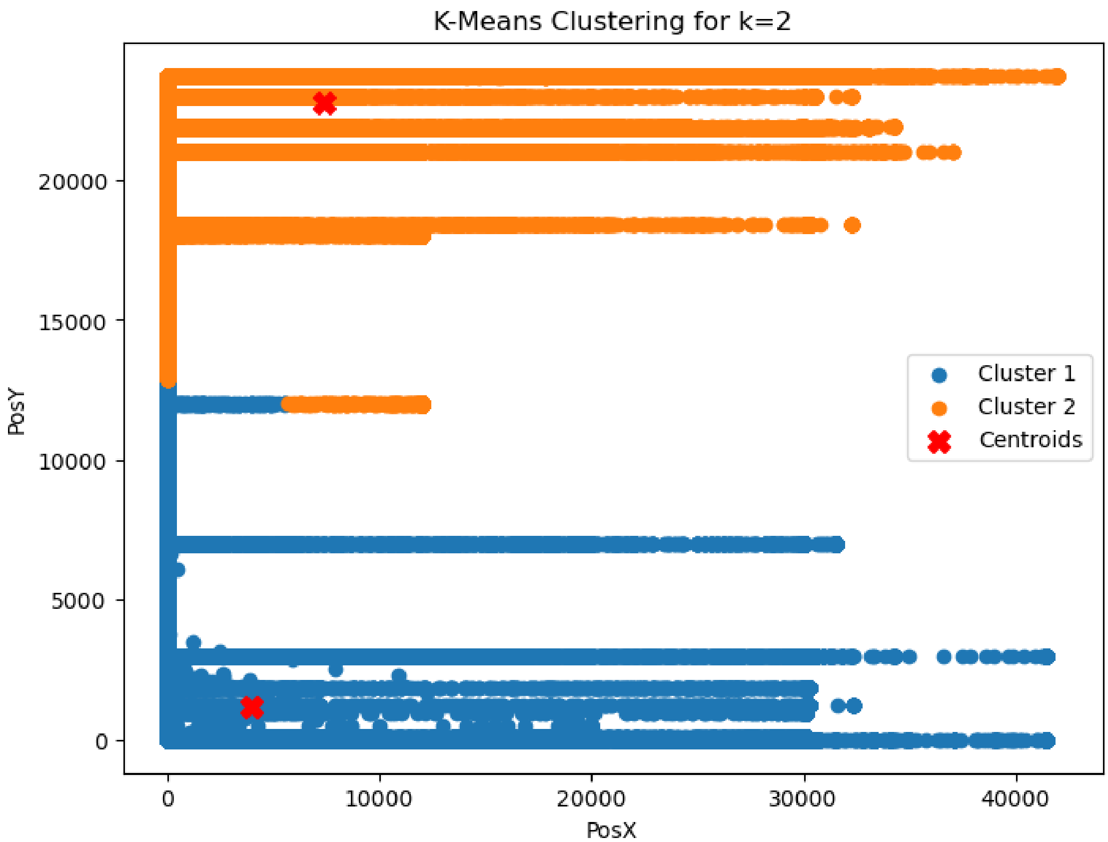
[84]:
```python
def find_optimal_clusters(range_n_clusters, ssd):
    deltas = np.diff(ssd, 2)
    elbow_index = np.argmax(deltas) + 2
    optimal_clusters = range_n_clusters[elbow_index - 1]
    return optimal_clusters

def My_function_kmeans_elbow(max_clusters, df):
    ssd = []
    range_n_clusters = np.arange(1, max_clusters + 1, 1)
    print(range_n_clusters)
    for num_clusters in range_n_clusters:

        # Launch the clustering
        kmeans = KMeans(n_clusters=num_clusters)
        kmeans.fit(df)
        ssd.append(kmeans.inertia_)

        # Plotting the clustering
        plt.figure(figsize=(8, 6))
        for i in range(num_clusters):
            cluster_indices = np.where(kmeans.labels_ == i)[0]
            plt.scatter(df.iloc[cluster_indices, 0], df.iloc[cluster_indices,␣
↪1], label=f"Cluster {i + 1}")
        plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,␣
↪1], s=100, c="red", marker="X", label="Centroids")
        plt.title(f"K-Means Clustering for k={num_clusters}")
        plt.xlabel("PosX")
        plt.ylabel("PosY")
        plt.legend()
        plt.show()
```

```python
    # Plotting Elbow Curve for Optimal Clusters
    plt.plot(range_n_clusters, ssd, marker='o')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia (Sum of Squared Distances)')
    plt.title('Elbow Curve for Optimal Clusters')
    plt.show()

    # Find the optimal number of clusters
    optimal_clusters = find_optimal_clusters(range_n_clusters, ssd)
    print("Optimal number of clusters:", optimal_clusters)
    return optimal_clusters

# Example usage
# Assuming cuttingdata is defined somewhere in your code
optimal_clusters = My_function_kmeans_elbow(9, cuttingdata)
print("Optimal number of clusters:", optimal_clusters)
```

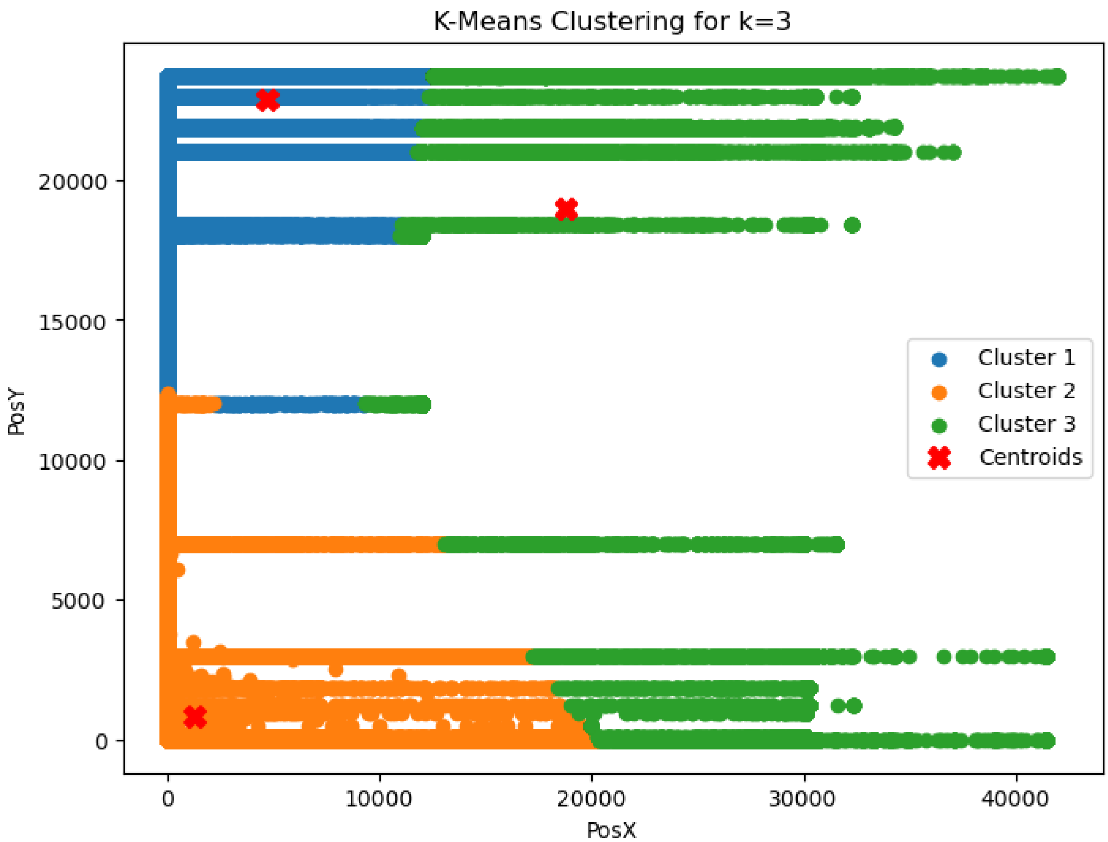
```
[1 2 3 4 5 6 7 8 9]

C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```



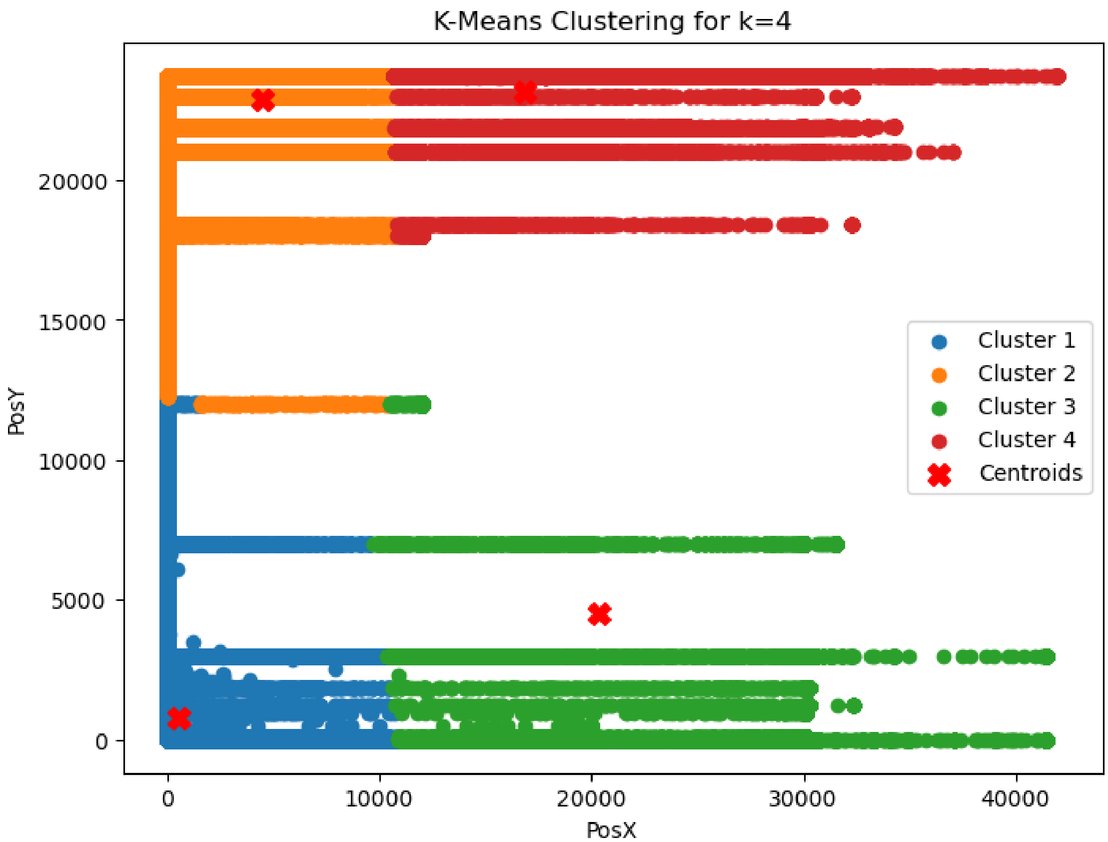
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



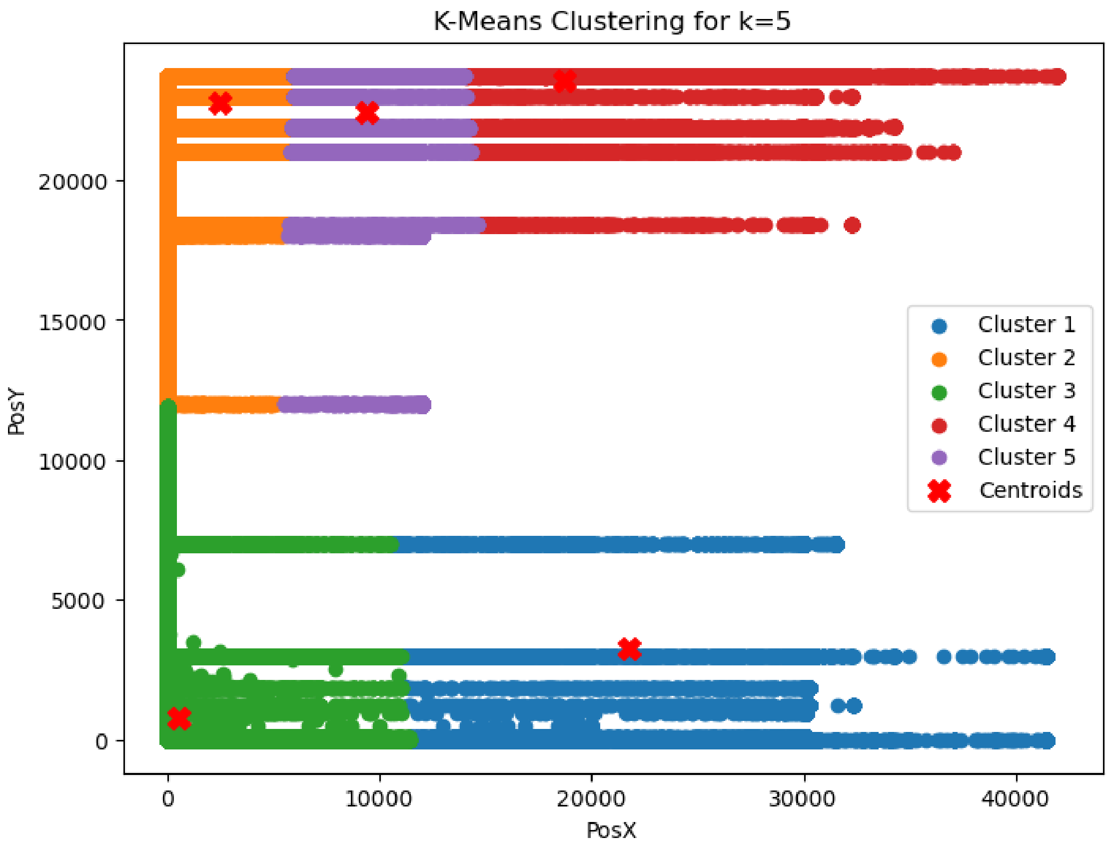
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



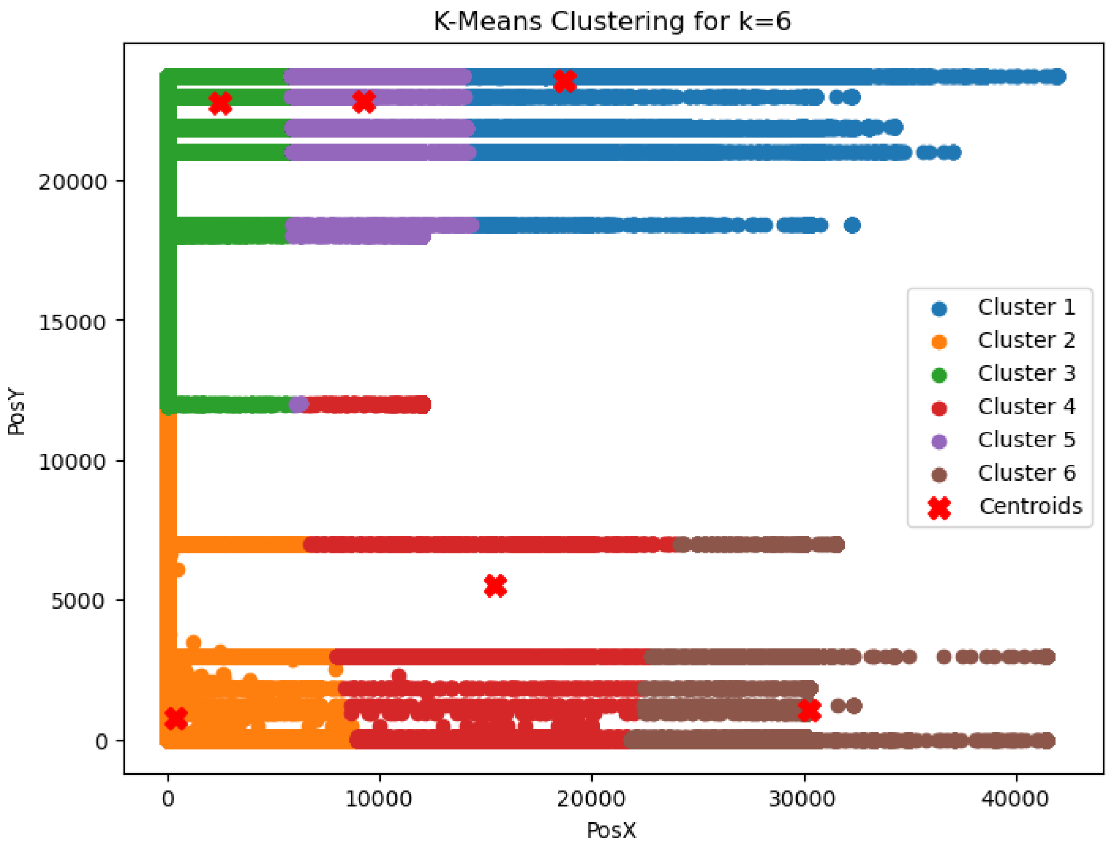
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



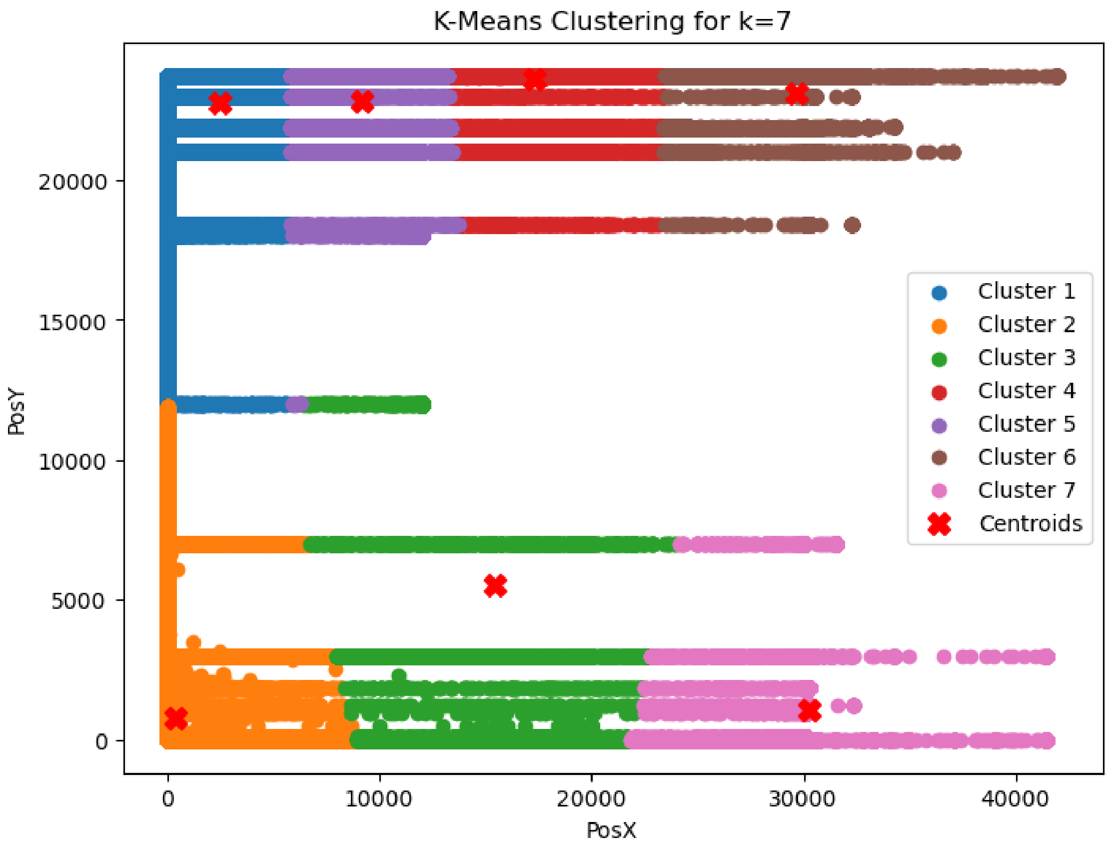
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



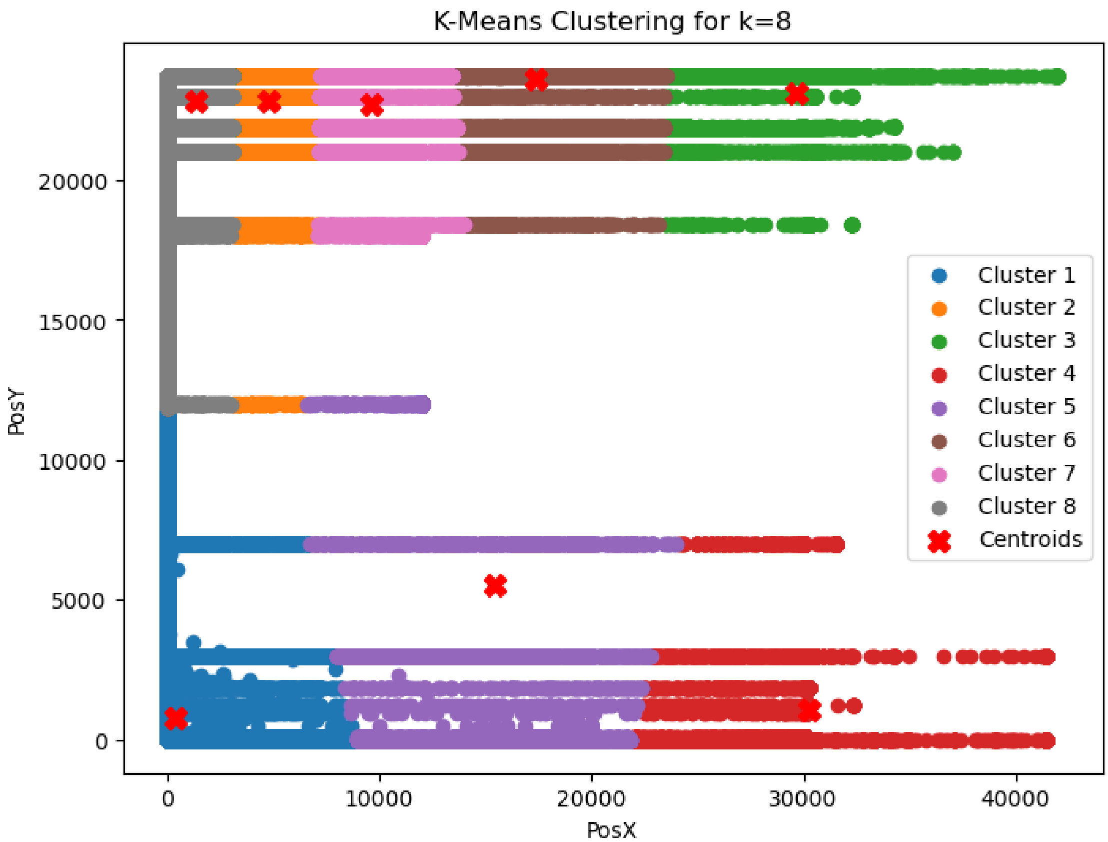
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)



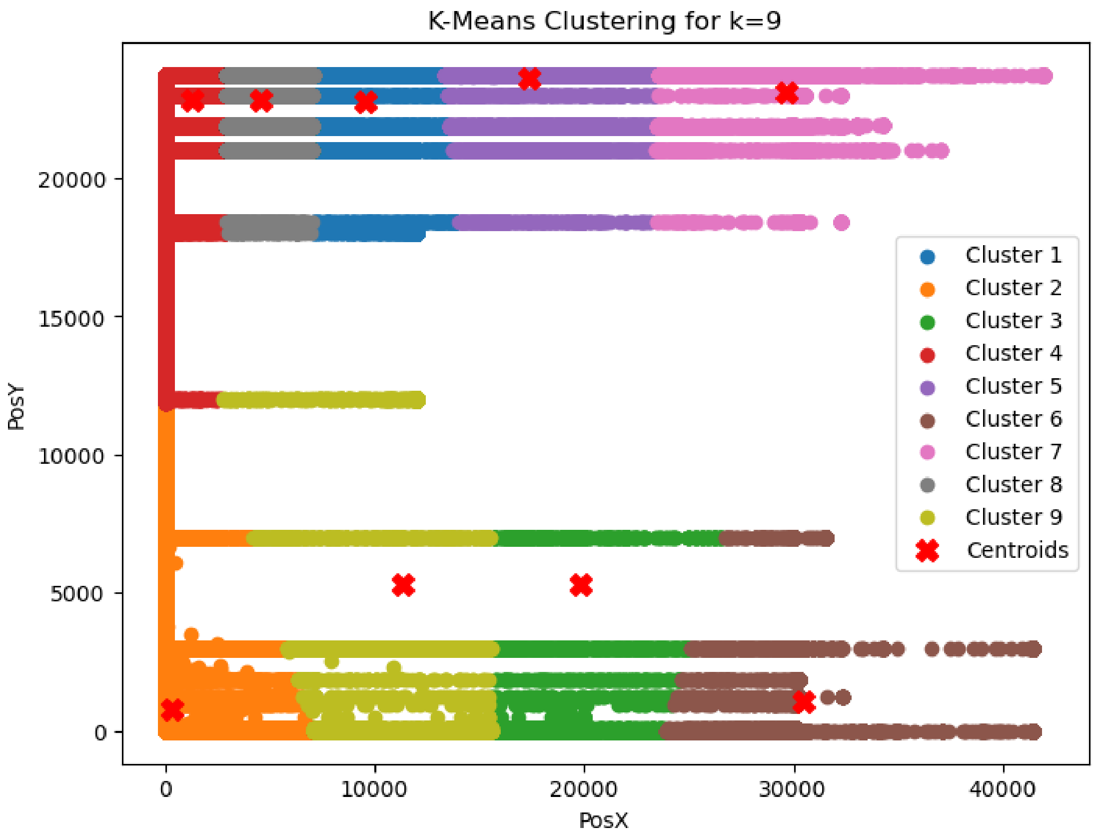

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

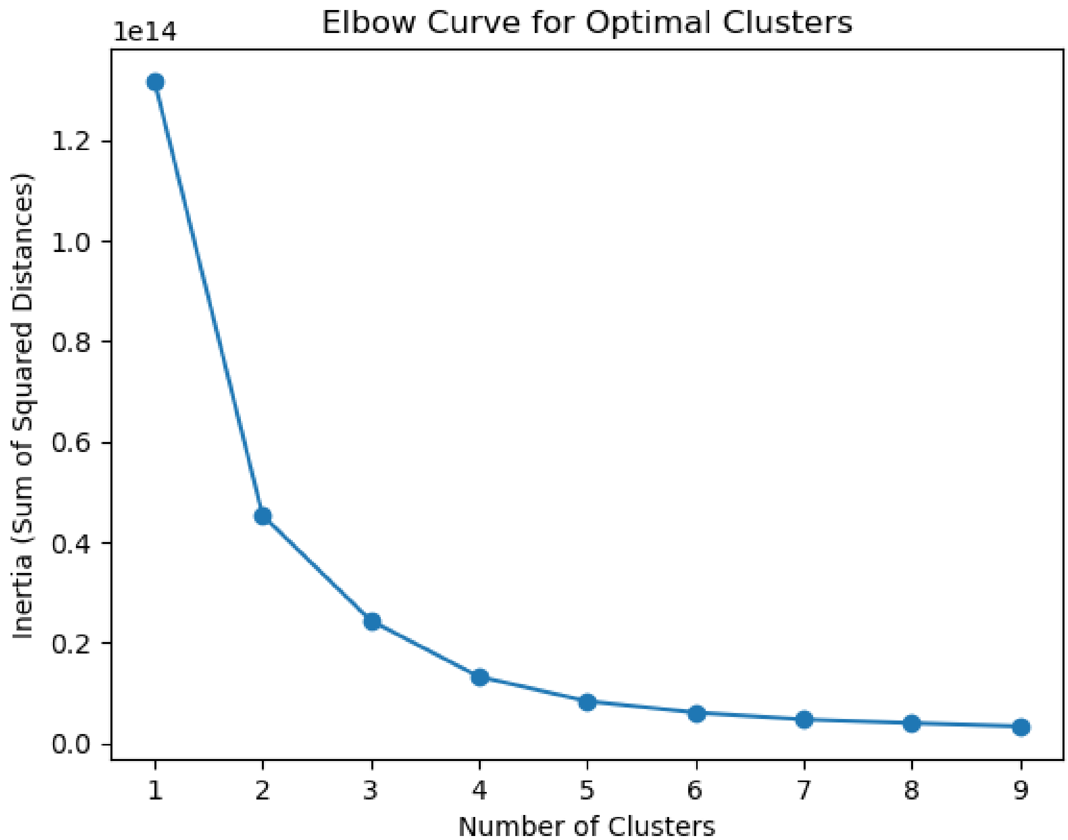



```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```



C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)


K-Means Clustering for k=9
PosY
PosX
0
5000
10000
15000
20000
0
10000
20000
30000
40000
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Centroids




```
Optimal number of clusters: 2
Optimal number of clusters: 2
```

3.2 Unsupervised machine learning with GMM

[85]:
```
#funtion to draw mutlivariate Gaussian

def multivariate_gaussian(pos, mu, Sigma):
    """Return the multivariate Gaussian distribution on array pos.
    pos is an array constructed by packing the meshed arrays of variables
    x_1, x_2, x_3, ..., x_k into its _last_ dimension.
    """
    n = mu.shape[0]
    Sigma_det = np.linalg.det(Sigma)
    Sigma_inv = np.linalg.inv(Sigma)
    N = np.sqrt((2*np.pi)**n * Sigma_det)

    # This einsum call calculates (x-mu)T.Sigma-1.(x-mu) in a vectorized
    # way across all the input variables.
    fac = np.einsum('...k,kl,...l->...', pos-mu, Sigma_inv, pos-mu)
```

```python
    return np.exp(-fac / 2) / N
```

[86]:
```python
# Generate sample data (replace this with your actual data)

cuttingdata, _ = make_blobs(n_samples=300, centers=3, random_state=42)
```

[87]:
```python
# Tests of the GMM functions:
# define the GMM model
nb_GMM=3
gmm = GaussianMixture(n_components=nb_GMM, covariance_type="full")

# learning of the GMM model by EM algo (Expectation Maximisation)
gmm.fit(cuttingdata)

# result? m=gmm.means_
cov=gmm.covariances_
w=gmm.weights_
```

```
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=2.
  warnings.warn(
```

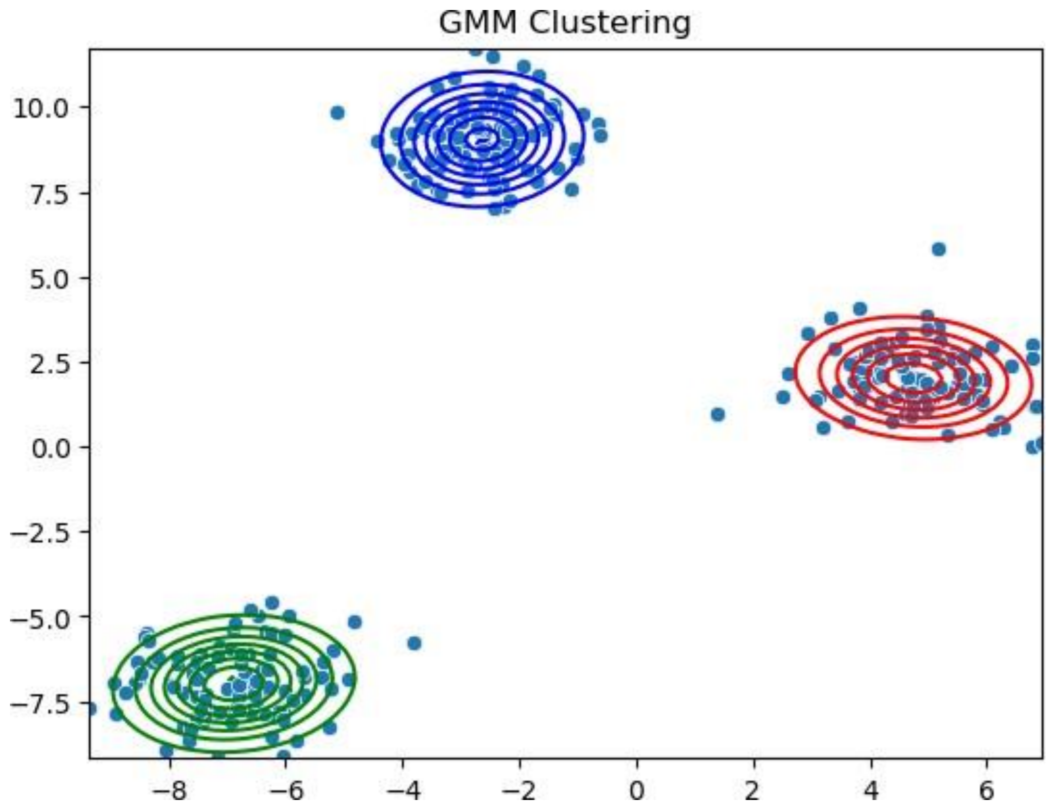
[88]:
```python
# apply the GMM model, as a classifier:
gmm.predict(cuttingdata)
```

[88]:
```
array([1, 1, 2, 0, 1, 0, 2, 0, 2, 2, 2, 0, 2, 2, 1, 2, 1, 0, 2, 2, 2, 2,
       0, 1, 2, 1, 1, 0, 0, 2, 2, 2, 1, 2, 1, 2, 1, 0, 1, 0, 0, 2, 1, 0,
       2, 2, 1, 0, 1, 0, 0, 1, 1, 2, 1, 0, 1, 2, 0, 2, 1, 0, 0, 1, 1, 0,
       0, 1, 1, 2, 0, 1, 1, 2, 2, 1, 1, 0, 2, 0, 2, 2, 1, 2, 0, 1, 1, 2,
       0, 2, 1, 2, 1, 2, 2, 1, 1, 2, 1, 1, 0, 2, 0, 2, 2, 2, 2, 2, 0, 1,
       0, 2, 2, 2, 2, 0, 1, 0, 1, 0, 0, 0, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2,
       2, 2, 2, 0, 0, 1, 2, 1, 2, 2, 1, 2, 0, 0, 0, 2, 0, 2, 2, 1, 0, 1,
       2, 0, 0, 1, 1, 2, 2, 1, 1, 1, 2, 1, 0, 2, 2, 2, 2, 2, 0, 2, 0, 0,
       0, 2, 0, 0, 1, 2, 1, 0, 0, 1, 0, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       2, 1, 2, 2, 0, 0, 2, 0, 1, 1, 0, 2, 2, 1, 0, 0, 1, 1, 1, 1, 2, 1,
       1, 0, 1, 1, 2, 0, 1, 1, 0, 2, 2, 1, 2, 1, 0, 0, 1, 0, 1, 1, 1, 0,
       0, 2, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 2, 1, 2, 2, 2, 1, 2,
       0, 0, 1, 0, 0, 2, 2, 0, 0, 0, 1, 1, 1, 2, 2, 2, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 1, 2, 0, 0, 2, 1, 2, 0, 2, 1, 1], dtype=int64)
```

[89]:
```python
# Define colors for clusters (replace with your actual color choices)
color_dict_cluster = ['red', 'green', 'blue']

# Plotting of GMM
```

```python
x = np.linspace(cuttingdata[:, 0].min(), cuttingdata[:, 0].max(), 100)
y = np.linspace(cuttingdata[:, 1].min(), cuttingdata[:, 1].max(), 100)
X, Y = np.meshgrid(x, y)
pos = np.empty(X.shape + (2,))
pos[:, :, 0] = X
pos[:, :, 1] = Y
for i in range(nb_GMM):
 mu_broadcast = np.expand_dims(m[i], axis=(0, 1))
 cov_inv = np.linalg.inv(cov[i])

 # Calculate the squared Mahalanobis distance directly
 diff = pos - mu_broadcast
 fac = np.sum(diff @ cov_inv * diff, axis=-1)
 Z = np.exp(-fac / 2) / np.sqrt((2 * np.pi)**2 * np.linalg.det(cov[i]))
 plt.contour(X, Y, Z, colors=color_dict_cluster[i])
 plt.scatter(m[i, 0], m[i, 1], marker="X", c=color_dict_cluster[i], s=30)

# Scatter plot of the original data
sns.scatterplot(x=cuttingdata[:, 0], y=cuttingdata[:, 1],
 ↪palette=color_dict_cluster)
plt.title("GMM Clustering")
plt.show()
```

```
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\2699856353.py:23: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  sns.scatterplot(x=cuttingdata[:, 0], y=cuttingdata[:, 1],
palette=color_dict_cluster)
```



3.3 Knowledge integration

[90]:
```python
#   print(HSM_data)

# Select only two specific columns
selected_columns = ['PosX','PosY','PosZ','Vf', 'N','P',]
KI_data =HSM_data[selected_columns]
print(KI_data)
```

```
            PosX      PosY     PosZ        Vf          N       P
0      -2200.028  1199.989  800.002      0.000      0.000   0.000
1      -2200.028  1199.989  800.002      0.000      0.000   0.000
2      -2200.028  1199.989  800.002      0.000      0.000   0.000
3      -2200.028  1199.989  800.002      0.000      0.000   0.000
4      -2200.028  1199.989  800.002      0.000      0.000   0.000
...          ...       ...      ...        ...        ...     ...
862569  -588.007   153.989  188.004  17789.728  23723.903  20.392
862570  -557.987   153.989  188.004  17789.728  23717.312  20.392
862571  -527.005   153.989  188.004  17789.728  23718.044  20.392
862572  -496.985   153.989  188.004  17789.728  23725.368  20.392
862573  -467.034   153.989  188.004  17789.728  23720.608  20.392
```

```
[862574 rows x 6 columns]
```

[91]:
```python
# Calculate actual event
Actual_Event = KI_data.apply(lambda row: row["PosX"] == 6 and row["Vf"] > 85,
 ↪axis=1)
print(Actual_Event)

# Add the actual event to the main KI_data
KI_data["Actual_Event"] = Actual_Event
print(KI_data)
```

```
0         False
1         False
2         False
3         False
4         False
          ...
862569    False
862570    False
862571    False
862572    False
862573    False
Length: 862574, dtype: bool
            PosX      PosY     PosZ         Vf         N       P  \
0       -2200.028  1199.989  800.002      0.000      0.000   0.000
1       -2200.028  1199.989  800.002      0.000      0.000   0.000
2       -2200.028  1199.989  800.002      0.000      0.000   0.000
3       -2200.028  1199.989  800.002      0.000      0.000   0.000
4       -2200.028  1199.989  800.002      0.000      0.000   0.000
...           ...       ...      ...        ...        ...     ...
862569   -588.007   153.989  188.004  17789.728  23723.903  20.392
862570   -557.987   153.989  188.004  17789.728  23717.312  20.392
862571   -527.005   153.989  188.004  17789.728  23718.044  20.392
862572   -496.985   153.989  188.004  17789.728  23725.368  20.392
862573   -467.034   153.989  188.004  17789.728  23720.608  20.392

        Actual_Event
0              False
1              False
2              False
3              False
4              False
...              ...
862569         False
862570         False
862571         False
862572         False
```

```
862573          False

[862574 rows x 7 columns]
```

```
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\1015651257.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  KI_data['Actual_Event'] = Actual_Event
```

[94]:

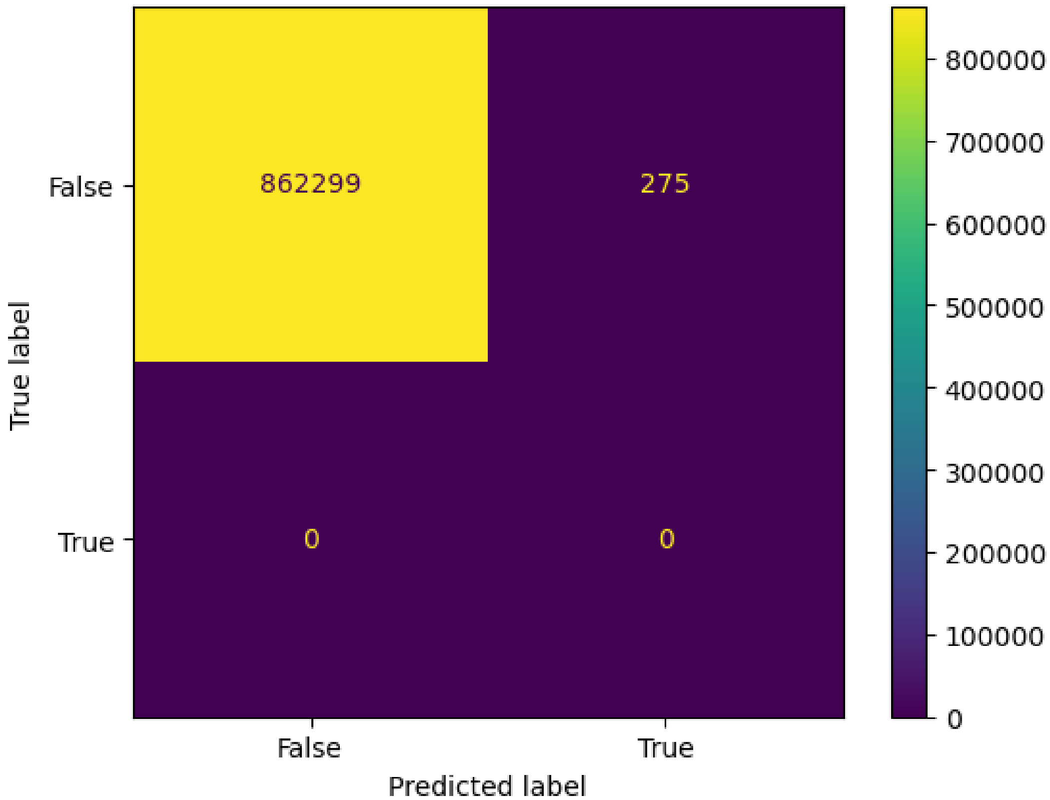
```python
from sklearn.metrics import confusion_matrix, recall_score, precision_score, f1_score, accuracy_score

# Define business rules
def apply_business_rules(row):
    if (row["PosX"] >= 6) and (row["PosX"] <= 7) and (row["Vf"] > 85):
        return 1  # Event occurred (TRUE)
    else:
        return 0  # Event did not occur (FALSE)

# Apply business rules to create a new column 'Predicted_Event'
KI_data['Predicted_Event'] = KI_data.apply(apply_business_rules, axis=1)
print(KI_data['Predicted_Event'])

# Compute confusion matrix
conf_matrix = confusion_matrix(KI_data['Actual_Event'], KI_data['Predicted_Event'])

#  Calculate repeatability
recall = recall_score(KI_data['Actual_Event'], KI_data['Predicted_Event'])

# Display results
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nRepeatability:", recall)

# Calculate evaluation metrics
accuracy = accuracy_score(KI_data['Actual_Event'], KI_data['Predicted_Event'])
recall = recall_score(KI_data['Actual_Event'], KI_data['Predicted_Event'])
precision = precision_score(KI_data['Actual_Event'], KI_data['Predicted_Event'])
f1 = f1_score(KI_data['Actual_Event'], KI_data['Predicted_Event'])

# Display results
print("\nEvaluation Metrics:")
print("Accuracy:", accuracy)
```

```python
print("Recall:", recall)
print("Precision:", precision)
print("F1:", f1)
```

```
C:\Users\nithi\AppData\Local\Temp\ipykernel_65388\3909810651.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  KI_data['Predicted_Event'] = KI_data.apply(apply_business_rules, axis=1)
0         0
1         0
2         0
3         0
4         0
         ..
862569    0
862570    0
862571    0
862572    0
862573    0
Name: Predicted_Event, Length: 862574, dtype: int64
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 due to no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))


Confusion Matrix:
[[862299    275]
 [     0      0]]

Repeatability: 0.0
C:\Users\nithi\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 due to no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))


Evaluation Metrics:
Accuracy: 0.9996811867735406
Recall: 0.0
Precision: 0.0
```

F1: 0.0

[95]:
```
import matplotlib.pyplot as plt
from sklearn import metrics

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = conf_matrix ,␣
 ↪display_labels = [False, True])
cm_display.plot()
plt.show()
```



4 Objective 3: productivity

4.1 Global OEE

[97]:
```
# Select only two specific columns

selected_columns = ["tpsT","tps B","PosX","PosY","PosZ","Vf", "N","P",]
KI_data2 = HSM_data[selected_columns]
print(KI_data2)
```

```
        tpsT    tps B     PosX     PosY    PosZ        Vf          N  \
```

```
0       5560779  4105603 -2200.028  1199.989  800.002      0.000      0.000
1       5560780  4105603 -2200.028  1199.989  800.002      0.000      0.000
2       5560781  4105603 -2200.028  1199.989  800.002      0.000      0.000
3       5560782  4105603 -2200.028  1199.989  800.002      0.000      0.000
4       5560783  4105603 -2200.028  1199.989  800.002      0.000      0.000
...         ...      ...       ...       ...      ...        ...        ...
862569  6423348  4802871  -588.007   153.989  188.004  17789.728  23723.903
862570  6423349  4802872  -557.987   153.989  188.004  17789.728  23717.312
862571  6423350  4802873  -527.005   153.989  188.004  17789.728  23718.044
862572  6423351  4802874  -496.985   153.989  188.004  17789.728  23725.368
862573  6423352  4802875  -467.034   153.989  188.004  17789.728  23720.608

             P
0        0.000
1        0.000
2        0.000
3        0.000
4        0.000
...        ...
862569  20.392
862570  20.392
862571  20.392
862572  20.392
862573  20.392

[862574 rows x 8 columns]
```

[98]:

```python
# Calculate the 90th percentile for each column
threshold_X = KI_data2["PosX"].quantile(0.9)
threshold_Y = KI_data2["PosY"].quantile(0.9)
threshold_Z  =  KI_data2["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (KI_data2["PosX"] > threshold_X) & (KI_data2["PosY"] >
  ↪threshold_Y) & (KI_data2["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = KI_data2[malfunction_time]
print(malfunctioned_data )
```

```
           tpsT    tps B      PosX      PosY   PosZ         Vf          N  \
4229    5565008  4105934  1042.973  1000.013  650.0  11999.693   3000.782
4230    5565009  4105935  1063.032  1000.013  650.0  11999.693   3001.148
4231    5565010  4105936  1083.985  1000.013  650.0  11999.693   3000.050
4232    5565011  4105937  1103.975  1000.013  650.0  11999.693   3001.148
4233    5565012  4105938  1123.966  1000.013  650.0  11999.693   3001.148
...         ...      ...       ...       ...    ...        ...        ...
844104  6404883  4789368  1999.015  1000.013  650.0      0.000  23724.636
```

```
844105  6404884  4789369  1999.977  997.990  650.0   4859.998  23713.649
844106  6404885  4789370  1999.977  985.009  650.0  11960.783  23715.480
844107  6404886  4789371  1999.977  964.011  650.0  11999.693  23717.678
844108  6404887  4789372  1999.977  944.006  650.0  11999.693  23724.270

            P
4229    0.000
4230    0.000
4231    0.000
4232    0.000
4233    0.000
…       …
844104  1.569
844105  1.569
844106  0.784
844107  0.392
844108  1.176

[3244 rows x 8 columns]
```

[99]:
```python
scheduled_production_time = (KI_data2["tps B"] - KI_data2["tpsT"]) / 60
idle_time = (KI_data2["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(KI_data2) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(KI_data2)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
```

```
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
Availability:   0         -3243.0
1         -3243.0
2         -3243.0
3         -3243.0
4         -3243.0
            ...
862569   -3243.0
862570   -3243.0
862571   -3243.0
862572   -3243.0
862573   -3243.0
Length: 862574, dtype: float64
Performance: 0         100.0
1          100.0
2          100.0
3          100.0
4          100.0
            ...
862569    100.0
862570    100.0
862571    100.0
862572    100.0
862573    100.0
Length: 862574, dtype: float64
Quality: 99.62391632486025
OEE: 0       -3.230804e+07
1        -3.230804e+07
2        -3.230804e+07
3        -3.230804e+07
4        -3.230804e+07
              ...
862569   -3.230804e+07
862570   -3.230804e+07
862571   -3.230804e+07
862572   -3.230804e+07
862573   -3.230804e+07
Length: 862574, dtype: float64
```

4.2 OEE per program

[100]:
```python
# Assuming you have a DataFrame named 'df' with a column named 'your_column'
# Replace 'your_column' and 'your_data.csv' with your actual column name and␣
↪data source
# list of values of 'idProgP'
#31,3,32,2,28,34,35,36

# Load the data into a DataFrame
HSM_data = pd.read_csv("data777.csv")

# Get the unique values in the 'is_malfunctioned' column
unique_values = HSM_data ["id_ProgP"].unique()
print("Unique values in the "id_ProgP" column:", unique_values)
```

```
Unique values in the 'id_ProgP' column: [31  3 32 28 33  2 34 35 36]
```

[102]:
```python
# Data Selection
OEE_data_31 = HSM_data[HSM_data["id_ProgP"] == 31]
print(OEE_data_31)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_31["PosX"].quantile(0.9)
threshold_Y = OEE_data_31["PosY"].quantile(0.9)
threshold_Z = OEE_data_31["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_31["PosX"] > threshold_X) & (OEE_data_31["PosY"] >␣
↪threshold_Y) & (OEE_data_31["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_31[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_31["tps B"] - OEE_data_31["tpsT"]) / 60
idle_time = (OEE_data_31["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
```

```
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_31) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_31)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT     tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
0       5560779   4105603  190312004        31     69     2         0        0
1       5560780   4105603  190312005        31     69     2         0        0
2       5560781   4105603  190312006        31     69     2         0        0
3       5560782   4105603  190312007        31     69     2         0        0
4       5560783   4105603  190312008        31     69     2         0        0
...         ...       ...        ...       ...    ...   ...       ...      ...
203418  5764197   4292619  191208034        31     69     2         7        0
203419  5764198   4292619  191208035        31     69     2         7        0
203420  5764199   4292619  191208036        31     69     2         7        0
203421  5764200   4292619  191208037        31     69     2         7        0
203422  5764201   4292619  191208038        31     69     2         7        0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
0                20       0  ...     0.0     0.0     0.0     0.0     0.0
1                20       0  ...     0.0     0.0     0.0     0.0     0.0
2                20       0  ...     0.0     0.0     0.0     0.0     0.0
3                20       0  ...     0.0     0.0     0.0     0.0     0.0
4                20       0  ...     0.0     0.0     0.0     0.0     0.0
...             ...     ...  ...     ...     ...     ...     ...
203418           20       0  ...     0.0     0.0     0.0     0.0     0.0
203419           20       0  ...     0.0     0.0     0.0     0.0     0.0
203420           20       0  ...     0.0     0.0     0.0     0.0     0.0
203421           20       0  ...     0.0     0.0     0.0     0.0     0.0
203422           20       0  ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22 FFT_23  FFT_24
0          0.0     0.0     0.0    0.0     0.0
1          0.0     0.0     0.0    0.0     0.0
```

```
2          0.0    0.0    0.0    0.0    0.0
3          0.0    0.0    0.0    0.0    0.0
4          0.0    0.0    0.0    0.0    0.0
...        ...    ...    ...    ...    ...
203418     0.0    0.0    0.0    0.0    0.0
203419     0.0    0.0    0.0    0.0    0.0
203420     0.0    0.0    0.0    0.0    0.0
203421     0.0    0.0    0.0    0.0    0.0
203422     0.0    0.0    0.0    0.0    0.0

[203423 rows x 72 columns]
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
4212    5564991  4105917  190847574        31     39     2         0        1
4213    5564992  4105918  190847575        31     39     2         0        1
4214    5564993  4105919  190847576        31     39     2         0        1
4215    5564994  4105920  190847577        31     39     2         0        1
4216    5564995  4105921  190847584        31     39     2         0        1
...         ...      ...        ...       ...    ...   ...       ...
202105  5762884  4291385  191205801        31     32     2         7        1
202106  5762885  4291386  191205808        31     32     2         7        1
202107  5762886  4291387  191205809        31     32     2         7        1
202108  5762887  4291388  191205810        31     32     2         7        1
202109  5762888  4291389  191205811        31     32     2         7        1

        usure outil  nligne  ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
4212             20      21  ...  2825.242   0.266   0.191   0.180  3732.288
4213             20      21  ...  2857.818   0.246   0.176   0.168  3731.963
4214             20      21  ...  3328.828   0.215   0.168   0.168  3730.789
4215             20      21  ...  2862.547   0.281   0.176   0.168  3732.288
4216             20      21  ...  2860.943   0.293   0.281   0.180  3732.549
...             ...     ...  ...       ...     ...     ...     ...       ...
202105           20      34  ...  3946.457   0.336   0.187   0.172   116.233
202106           20      34  ...  3395.375   0.352   0.324   0.187   117.145
202107           20      34  ...  2636.525   0.391   0.371   0.176  3950.230
202108           20      34  ...  2633.515   0.406   0.277   0.258   116.640
202109           20      34  ...  3432.599   0.379   0.172   0.152   117.533

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
4212    3799.086  2830.320   0.281   0.246   0.187
4213    3800.527  2844.340   0.293   0.234   0.172
4214    3800.355  2838.737   0.270   0.207   0.187
4215    3800.454  1855.304   0.281   0.270   0.211
4216    3799.355  3330.930   0.285   0.281   0.238
...          ...       ...     ...     ...     ...
202105  1481.403  3236.295   0.344   0.262   0.234
202106  3946.912  3236.295   0.332   0.266   0.262
202107   117.108  1481.393   0.348   0.336   0.301
202108  3948.906  2635.793   0.363   0.305   0.289
```

```
202109  1481.917  3236.711    0.352    0.309    0.234

[827 rows x 72 columns]
Availability:  0          -826.0
1          -826.0
2          -826.0
3          -826.0
4          -826.0
            ...
203418   -826.0
203419   -826.0
203420   -826.0
203421   -826.0
203422   -826.0
Length: 203423, dtype: float64
Performance: 0            100.0
1            100.0
2            100.0
3            100.0
4            100.0
            ...
203418     100.0
203419     100.0
203420     100.0
203421     100.0
203422     100.0
Length:  203423, dtype: float64
Quality:  99.59345796689657
OEE: 0          -8.226420e+06
1          -8.226420e+06
2          -8.226420e+06
3          -8.226420e+06
4          -8.226420e+06
                 ...
203418   -8.226420e+06
203419   -8.226420e+06
203420   -8.226420e+06
203421   -8.226420e+06
203422   -8.226420e+06
Length: 203423, dtype: float64
```

[103]:

```python
# Data Selection
OEE_data_3 = HSM_data[HSM_data["id_ProgP"] == 3]
print(OEE_data_3)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_3["PosX"].quantile(0.9)
```

```
threshold_Y = OEE_data_3["PosY"].quantile(0.9)
threshold_Z = OEE_data_3["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_3["PosX"] > threshold_X) & (OEE_data_3["PosY"] >
 ↪threshold_Y) & (OEE_data_3["PosZ"] > threshold_Z)

#Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_3[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_3["tps B"] - OEE_data_3["tpsT"]) / 60
idle_time = (OEE_data_3["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *
 ↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /
 ↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_3) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_3)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id  pc  mode  id_outil  n  outil  \
203423  5764202  4292619  191208039         3      33     2         7         0
203424  5764203  4292619  191208040         3      33     2         7         0
203425  5764204  4292619  191208041         3      33     2         7         0
203426  5764205  4292619  191208048         3      33     2         7         0
```

```
203427  5764206  4292619  191208049    3   33   2   7   0
...     ...      ...      ...        ...  ...  ...  ...  ...
837954  6398733  4783338  192367140    3   33   2   0   0
837955  6398734  4783338  192367141    3   33   2   0   0
837956  6398735  4783338  192367142    3   33   2   0   0
837957  6398736  4783338  192367143    3   33   2   0   0
837958  6398737  4783338  192367144    3   33   2   0   0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18 FFT_19  \
203423           20       0  ...     0.0     0.0     0.0     0.0    0.0
203424           20       0  ...     0.0     0.0     0.0     0.0    0.0
203425           20      31  ...     0.0     0.0     0.0     0.0    0.0
203426           20      33  ...     0.0     0.0     0.0     0.0    0.0
203427           20      33  ...     0.0     0.0     0.0     0.0    0.0
...             ...     ...  ...     ...     ...     ...     ...
837954           20       0  ...     0.0     0.0     0.0     0.0    0.0
837955           20       0  ...     0.0     0.0     0.0     0.0    0.0
837956           20       0  ...     0.0     0.0     0.0     0.0    0.0
837957           20       0  ...     0.0     0.0     0.0     0.0    0.0
837958           20       0  ...     0.0     0.0     0.0     0.0    0.0

        FFT_20  FFT_21  FFT_22 FFT_23  FFT_24
203423     0.0     0.0     0.0    0.0     0.0
203424     0.0     0.0     0.0    0.0     0.0
203425     0.0     0.0     0.0    0.0     0.0
203426     0.0     0.0     0.0    0.0     0.0
203427     0.0     0.0     0.0    0.0     0.0
...        ...     ...     ...    ...     ...
837954     0.0     0.0     0.0    0.0     0.0
837955     0.0     0.0     0.0    0.0     0.0
837956     0.0     0.0     0.0    0.0     0.0
837957     0.0     0.0     0.0    0.0     0.0
837958     0.0     0.0     0.0    0.0     0.0

[18108 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1, Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3, Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX, VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5, FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15, FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 203423     1.0
203424    1.0
```

```
203425     1.0
203426     1.0
203427     1.0
              ...
837954     1.0
837955     1.0
837956     1.0
837957     1.0
837958     1.0
Length: 18108, dtype: float64
Performance: 203423     100.0
203424     100.0
203425     100.0
203426     100.0
203427     100.0
              ...
837954     100.0
837955     100.0
837956     100.0
837957     100.0
837958     100.0
Length: 18108, dtype: float64
Quality: 100.0
OEE: 203423     10000.0
203424     10000.0
203425     10000.0
203426     10000.0
203427     10000.0
              ...
837954     10000.0
837955     10000.0
837956     10000.0
837957     10000.0
837958     10000.0
Length: 18108, dtype: float64
```

[104]:
```
# Data Selection
OEE_data_32 = HSM_data[HSM_data["id_ProgP"] == 32]
print(OEE_data_32)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_32["PosX"].quantile(0.9)
threshold_Y = OEE_data_32["PosY"].quantile(0.9)
threshold_Z = OEE_data_32["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
```

```
malfunction_time = (OEE_data_32["PosX"] > threshold_X) & (OEE_data_32["PosY"] >␣
↪threshold_Y) & (OEE_data_32["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_32[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_32["tps B"] - OEE_data_32["tpsT"]) / 60
idle_time = (OEE_data_32["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_32) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_32)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
206071  5766850  4292619  191212597        32     70     2         0        0
206072  5766851  4292619  191212598        32     70     2         0        0
206073  5766852  4292619  191212599        32     70     2         0        0
206074  5766853  4292619  191212600        32     70     2         0        0
206075  5766854  4292619  191212601        32     70     2         0        0
...         ...      ...        ...       ...    ...   ...       ...      ...
481254  6042033  4473434  191717433        32     70     2         7        0
481255  6042034  4473434  191717440        32     70     2         7        0
```

```
481256  6042035  4473434  191717441        32     70      2          7          0
481257  6042036  4473434  191717442        32     70      2          7          0
481258  6042037  4473434  191717443        32     70      2          7          0

        usure outil   nligne   ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
206071           20        0   ...     0.0     0.0     0.0     0.0     0.0
206072           20        0   ...     0.0     0.0     0.0     0.0     0.0
206073           20        0   ...     0.0     0.0     0.0     0.0     0.0
206074           20        0   ...     0.0     0.0     0.0     0.0     0.0
206075           20        0   ...     0.0     0.0     0.0     0.0     0.0
...             ...      ...   ...     ...     ...     ...     ...     ...
481254           20        0   ...     0.0     0.0     0.0     0.0     0.0
481255           20        0   ...     0.0     0.0     0.0     0.0     0.0
481256           20        0   ...     0.0     0.0     0.0     0.0     0.0
481257           20        0   ...     0.0     0.0     0.0     0.0     0.0
481258           20        0   ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22  FFT_23 FFT_24
206071     0.0     0.0     0.0     0.0    0.0
206072     0.0     0.0     0.0     0.0    0.0
206073     0.0     0.0     0.0     0.0    0.0
206074     0.0     0.0     0.0     0.0    0.0
206075     0.0     0.0     0.0     0.0    0.0
...        ...     ...     ...     ...    ...
481254     0.0     0.0     0.0     0.0    0.0
481255     0.0     0.0     0.0     0.0    0.0
481256     0.0     0.0     0.0     0.0    0.0
481257     0.0     0.0     0.0     0.0    0.0
481258     0.0     0.0     0.0     0.0    0.0

[209585 rows x 72 columns]
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
209800  5770579  4292893  191218948        32     39     2         0        1
209801  5770580  4292894  191218949        32     39     2         0        1
209802  5770581  4292895  191218950        32     39     2         0        1
209803  5770582  4292896  191218951        32     39     2         0        1
209804  5770583  4292897  191218952        32     39     2         0        1
...         ...      ...        ...       ...    ...   ...       ...      ...
293743  5854522  4369019  191374503        32     13     2         3        1
293744  5854523  4369019  191374504        32     13     2         3        1
293745  5854524  4369019  191374505        32     13     2         3        1
293746  5854525  4369019  191374512        32     13     2         3        1
293747  5854526  4369019  191374513        32     13     2         3        1

        usure outil  nligne   ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
209800           20      21   ...  2819.485   0.172   0.145   0.094  3734.176
209801           20      21   ...  2857.357   0.180   0.164   0.113  3732.744
209802           20      21   ...  3709.187   0.156   0.156   0.125  3730.466
```

```
209803            20       21  ... 3332.070   0.148   0.137   0.125  1899.800
209804            20       21  ... 3387.659   0.176   0.133   0.109  3328.117
...              ...      ...  ...      ...     ...     ...     ...
293743            20      115  ...    0.000   0.000   0.000   0.000     0.000
293744            20      115  ...    0.000   0.000   0.000   0.000     0.000
293745            20      115  ...    0.000   0.000   0.000   0.000     0.000
293746            20      115  ...    0.000   0.000   0.000   0.000     0.000
293747            20      115  ...    0.000   0.000   0.000   0.000     0.000

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
209800  3802.163  3712.686   0.168   0.129   0.117
209801  2368.797  2837.245   0.160   0.137   0.137
209802  3799.379  1899.299   0.152   0.148   0.133
209803  3732.809  3799.990   0.152   0.137   0.129
209804  1900.470  3733.004   0.145   0.145   0.137
...          ...       ...     ...     ...     ...
293743     0.000     0.000   0.000   0.000   0.000
293744     0.000     0.000   0.000   0.000   0.000
293745     0.000     0.000   0.000   0.000   0.000
293746     0.000     0.000   0.000   0.000   0.000
293747     0.000     0.000   0.000   0.000   0.000

[1278 rows x 72 columns]
Availability: 206071   -1277.0
206072   -1277.0
206073   -1277.0
206074   -1277.0
206075   -1277.0
           ...
481254   -1277.0
481255   -1277.0
481256   -1277.0
481257   -1277.0
481258   -1277.0
Length: 209585, dtype: float64
Performance: 206071    100.0
206072    100.0
206073    100.0
206074    100.0
206075    100.0
          ...
481254    100.0
481255    100.0
481256    100.0
481257    100.0
481258    100.0
Length: 209585, dtype: float64
Quality: 99.39022353698977
```

```
OEE: 206071    -1.269213e+07
206072   -1.269213e+07
206073   -1.269213e+07
206074   -1.269213e+07
206075   -1.269213e+07
                 ...
481254   -1.269213e+07
481255   -1.269213e+07
481256   -1.269213e+07
481257   -1.269213e+07
481258   -1.269213e+07
Length: 209585, dtype: float64
```

[105]:
```python
# Data Selection
OEE_data_28 = HSM_data[HSM_data["id_ProgP"] == 28]
print(OEE_data_28)

# Calculate the 90th percentile for each column
threshold_X  =  OEE_data_28["PosX"].quantile(0.9)
threshold_Y  =  OEE_data_28["PosY"].quantile(0.9)
threshold_Z = OEE_data_28["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_28["PosX"] > threshold_X) & (OEE_data_28["PosY"] >
 ↪threshold_Y) & (OEE_data_28["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_28[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_28["tps B"] - OEE_data_28["tpsT"]) / 60
idle_time = (OEE_data_28["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *
 ↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /
 ↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_28) - len(malfunctioned_data)
```

```
# Calculate the total number of units
total_number_of_units = len(OEE_data_28)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
          tpsT     tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
397588 5958367  4453828  191564020        28     66     0        36        0
397589 5958368  4453828  191564021        28     66     0        36        0
397590 5958369  4453828  191564022        28     66     0        36        0
397591 5958370  4453828  191564023        28     66     0        36        0
397592 5958371  4453828  191564024        28     66     0        36        0
...        ...      ...        ...       ...    ...   ...       ...      ...
455064 6015843  4453828  191668617        28     66     2        40        0
455065 6015844  4453828  191668624        28     66     2        40        0
455066 6015845  4453828  191668625        28     66     2        40        0
455067 6015846  4453828  191668626        28     66     2        40        0
455068 6015847  4453828  191668627        28     66     2        40        0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
397588          200       0  ...     0.0     0.0     0.0     0.0     0.0
397589          200       0  ...     0.0     0.0     0.0     0.0     0.0
397590          200       0  ...     0.0     0.0     0.0     0.0     0.0
397591          200       0  ...     0.0     0.0     0.0     0.0     0.0
397592          200       0  ...     0.0     0.0     0.0     0.0     0.0
...             ...     ...  ...     ...     ...     ...     ...     ...
455064            0       0  ...     0.0     0.0     0.0     0.0     0.0
455065            0       0  ...     0.0     0.0     0.0     0.0     0.0
455066            0       0  ...     0.0     0.0     0.0     0.0     0.0
455067            0       0  ...     0.0     0.0     0.0     0.0     0.0
455068            0       0  ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22  FFT_23  FFT_24
397588     0.0     0.0     0.0     0.0     0.0
397589     0.0     0.0     0.0     0.0     0.0
397590     0.0     0.0     0.0     0.0     0.0
397591     0.0     0.0     0.0     0.0     0.0
397592     0.0     0.0     0.0     0.0     0.0
...        ...     ...     ...     ...     ...
```

```
455064    0.0    0.0    0.0    0.0    0.0
455065    0.0    0.0    0.0    0.0    0.0
455066    0.0    0.0    0.0    0.0    0.0
455067    0.0    0.0    0.0    0.0    0.0
455068    0.0    0.0    0.0    0.0    0.0

[48147 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1, Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3, Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX, VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5, FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15, FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 397588    1.0
397589    1.0
397590    1.0
397591    1.0
397592    1.0
          ...
455064    1.0
455065    1.0
455066    1.0
455067    1.0
455068    1.0
Length: 48147, dtype: float64
Performance: 397588    100.0
397589    100.0
397590    100.0
397591    100.0
397592    100.0
          ...
455064    100.0
455065    100.0
455066    100.0
455067    100.0
455068    100.0
Length: 48147, dtype: float64
Quality: 100.0
OEE: 397588    10000.0
397589    10000.0
397590    10000.0
397591    10000.0
397592    10000.0
```

```
          ...
455064    10000.0
455065    10000.0
455066    10000.0
455067    10000.0
455068    10000.0
Length: 48147, dtype: float64
```

[106]:
```
# Data Selection
OEE_data_33 = HSM_data[HSM_data["id_ProgP"] == 33]
print(OEE_data_33)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_33["PosX"].quantile(0.9)
threshold_Y = OEE_data_33["PosY"].quantile(0.9)
threshold_Z = OEE_data_33["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_33["PosX"] > threshold_X) & (OEE_data_33["PosY"] >␣
↪threshold_Y) & (OEE_data_33["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_33[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_33["tps B"] - OEE_data_33["tpsT"]) / 60
idle_time = (OEE_data_33["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_33) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_33)

# Calculate quality
```

```python
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
         tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
399384  5960163  4453828  191567412        33     71     0        40        0
399385  5960164  4453828  191567413        33     71     0        40        0
409428  5970207  4453828  191584905        33     71     0        40        0
440954  6001733  4453828  191643191        33     71     0        40        0
440955  6001734  4453828  191643192        33     71     0        40        0
447045  6007824  4453828  191654960        33     71     0        40        0

        usure outil  nligne  ... FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
399384            0       0  ...    0.0     0.0     0.0     0.0     0.0
399385            0       0  ...    0.0     0.0     0.0     0.0     0.0
409428            0       0  ...    0.0     0.0     0.0     0.0     0.0
440954            0       0  ...    0.0     0.0     0.0     0.0     0.0
440955            0       0  ...    0.0     0.0     0.0     0.0     0.0
447045            0       0  ...    0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22 FFT_23  FFT_24
399384     0.0     0.0     0.0    0.0     0.0
399385     0.0     0.0     0.0    0.0     0.0
409428     0.0     0.0     0.0    0.0     0.0
440954     0.0     0.0     0.0    0.0     0.0
440955     0.0     0.0     0.0    0.0     0.0
447045     0.0     0.0     0.0    0.0     0.0

[6 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1, Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3, Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX, VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5, FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15, FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 399384    1.0
399385    1.0
```

```
409428    1.0
440954    1.0
440955    1.0
447045    1.0
dtype: float64
Performance: 399384    100.0
399385    100.0
409428    100.0
440954    100.0
440955    100.0
447045    100.0
dtype: float64
Quality: 100.0
OEE: 399384    10000.0
399385    10000.0
409428    10000.0
440954    10000.0
440955    10000.0
447045    10000.0
dtype: float64
```

[107]:

```
# Data Selection
OEE_data_3 = HSM_data[HSM_data['id_ProgP'] == 3]
print(OEE_data_3)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_3["PosX"].quantile(0.9)
threshold_Y = OEE_data_3["PosY"].quantile(0.9)
threshold_Z = OEE_data_3["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_3["PosX"] > threshold_X) & (OEE_data_3["PosY"] >
  threshold_Y) & (OEE_data_3["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_3[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_3["tps B"] - OEE_data_3["tpsT"]) / 60
idle_time = (OEE_data_3["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *
  scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
```

```
availability = (scheduled_production_time - downtime) /
    scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_3) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_3)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
203423  5764202  4292619  191208039         3     33     2         7        0
203424  5764203  4292619  191208040         3     33     2         7        0
203425  5764204  4292619  191208041         3     33     2         7        0
203426  5764205  4292619  191208048         3     33     2         7        0
203427  5764206  4292619  191208049         3     33     2         7        0
...         ...      ...        ...       ...    ...   ...       ...      ...
837954  6398733  4783338  192367140         3     33     2         0        0
837955  6398734  4783338  192367141         3     33     2         0        0
837956  6398735  4783338  192367142         3     33     2         0        0
837957  6398736  4783338  192367143         3     33     2         0        0
837958  6398737  4783338  192367144         3     33     2         0        0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
203423           20       0  ...     0.0     0.0     0.0     0.0     0.0
203424           20       0  ...     0.0     0.0     0.0     0.0     0.0
203425           20      31  ...     0.0     0.0     0.0     0.0     0.0
203426           20      33  ...     0.0     0.0     0.0     0.0     0.0
203427           20      33  ...     0.0     0.0     0.0     0.0     0.0
...             ...     ...  ...     ...     ...     ...     ...     ...
837954           20       0  ...     0.0     0.0     0.0     0.0     0.0
837955           20       0  ...     0.0     0.0     0.0     0.0     0.0
837956           20       0  ...     0.0     0.0     0.0     0.0     0.0
837957           20       0  ...     0.0     0.0     0.0     0.0     0.0
837958           20       0  ...     0.0     0.0     0.0     0.0     0.0
```

```
        FFT_20  FFT_21  FFT_22  FFT_23 FFT_24
203423     0.0     0.0     0.0     0.0    0.0
203424     0.0     0.0     0.0     0.0    0.0
203425     0.0     0.0     0.0     0.0    0.0
203426     0.0     0.0     0.0     0.0    0.0
203427     0.0     0.0     0.0     0.0    0.0
...        ...     ...     ...     ...    ...
837954     0.0     0.0     0.0     0.0    0.0
837955     0.0     0.0     0.0     0.0    0.0
837956     0.0     0.0     0.0     0.0    0.0
837957     0.0     0.0     0.0     0.0    0.0
837958     0.0     0.0     0.0     0.0    0.0

[18108 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure
outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1,
Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3,
Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX,
VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5,
FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15,
FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 203423     1.0
203424    1.0
203425    1.0
203426    1.0
203427    1.0
         ...
837954    1.0
837955    1.0
837956    1.0
837957    1.0
837958    1.0
Length: 18108, dtype: float64
Performance: 203423     100.0
203424    100.0
203425    100.0
203426    100.0
203427    100.0
          ...
837954    100.0
837955    100.0
837956    100.0
837957    100.0
```

```
837958    100.0
Length: 18108, dtype: float64
Quality: 100.0
OEE: 203423     10000.0
203424    10000.0
203425    10000.0
203426    10000.0
203427    10000.0
            ...
837954    10000.0
837955    10000.0
837956    10000.0
837957    10000.0
837958    10000.0
Length: 18108, dtype: float64
```

[108]:

```python
# Data Selection
OEE_data_2 = HSM_data[HSM_data["id_ProgP"] == 2]
print(OEE_data_2)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_2["PosX"].quantile(0.9)
threshold_Y = OEE_data_2["PosY"].quantile(0.9)
threshold_Z = OEE_data_2["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_2["PosX"] > threshold_X) & (OEE_data_2["PosY"] >
 ↪threshold_Y) & (OEE_data_2["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_2[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_2["tps B"] - OEE_data_2["tpsT"]) / 60
idle_time = (OEE_data_2["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *
 ↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /
 ↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100
```

```python
# Calculate the number of good units
number_of_good_units = len(OEE_data_2) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_2)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
400100  5960879  4453828  191568624         2     26     1        40        0
400101  5960880  4453828  191568625         2     26     1        40        0
400102  5960881  4453828  191568626         2     26     1        40        0
400103  5960882  4453828  191568627         2     26     1        40        0
400104  5960883  4453828  191568628         2     26     1        40        0
...         ...      ...        ...       ...    ...   ...       ...      ...
463279  6024058  4460283  191682660         2     26     1         0        0
463280  6024059  4460283  191682661         2     26     1         0        0
463281  6024060  4460283  191682662         2     26     1         0        0
463282  6024061  4460283  191682663         2     26     1         0        0
463283  6024062  4460283  191682664         2     26     1         0        0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
400100            0       0  ...     0.0     0.0     0.0     0.0     0.0
400101            0       0  ...     0.0     0.0     0.0     0.0     0.0
400102            0       0  ...     0.0     0.0     0.0     0.0     0.0
400103            0       0  ...     0.0     0.0     0.0     0.0     0.0
400104            0       0  ...     0.0     0.0     0.0     0.0     0.0
...             ...     ...  ...     ...     ...     ...     ...     ...
463279           20       0  ...     0.0     0.0     0.0     0.0     0.0
463280           20       0  ...     0.0     0.0     0.0     0.0     0.0
463281           20       0  ...     0.0     0.0     0.0     0.0     0.0
463282           20       0  ...     0.0     0.0     0.0     0.0     0.0
463283           20       0  ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22 FFT_23  FFT_24
400100     0.0     0.0     0.0    0.0     0.0
400101     0.0     0.0     0.0    0.0     0.0
400102     0.0     0.0     0.0    0.0     0.0
```

```
400103      0.0     0.0     0.0     0.0     0.0
400104      0.0     0.0     0.0     0.0     0.0
...         ...     ...     ...     ...     ...
463279      0.0     0.0     0.0     0.0     0.0
463280      0.0     0.0     0.0     0.0     0.0
463281      0.0     0.0     0.0     0.0     0.0
463282      0.0     0.0     0.0     0.0     0.0
463283      0.0     0.0     0.0     0.0     0.0

[17450 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure
outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1,
Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3,
Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX,
VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5,
FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15,
FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 400100      1.0
400101    1.0
400102    1.0
400103    1.0
400104    1.0
         ...
463279    1.0
463280    1.0
463281    1.0
463282    1.0
463283    1.0
Length: 17450, dtype: float64
Performance: 400100     100.0
400101     100.0
400102     100.0
400103     100.0
400104     100.0
           ...
463279     100.0
463280     100.0
463281     100.0
463282     100.0
463283     100.0
Length: 17450, dtype: float64
Quality: 100.0
OEE: 400100      10000.0
400101    10000.0
```

```
400102    10000.0
400103    10000.0
400104    10000.0
           ...
463279    10000.0
463280    10000.0
463281    10000.0
463282    10000.0
463283    10000.0
Length: 17450, dtype: float64
```

[109]:
```
# Data Selection
OEE_data_34 = HSM_data[HSM_data["id_ProgP"] == 34]
print(OEE_data_34)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_34["PosX"].quantile(0.9)
threshold_Y = OEE_data_34["PosY"].quantile(0.9)
threshold_Z = OEE_data_34["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_34["PosX"] > threshold_X) & (OEE_data_34["PosY"] >␣
↪threshold_Y) & (OEE_data_34["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_34[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_34["tps B"] - OEE_data_34["tpsT"]) / 60
idle_time = (OEE_data_34["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_34) - len(malfunctioned_data)

# Calculate the total number of units
```

```python
total_number_of_units = len(OEE_data_34)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
          tpsT    tps B      date  id_ProgP  id pc  mode  id_outil  n outil  \
487614 6048393  4473434 191728249        34     72     2         0        0
487615 6048394  4473434 191728256        34     72     2         0        0
487616 6048395  4473434 191728257        34     72     2         0        0
487617 6048396  4473434 191728258        34     72     2         0        0
487618 6048397  4473434 191728259        34     72     2         0        0
...        ...      ...       ...       ...    ...    ...      ...
687734 6248513  4659851 192094404        34     72     2         7        0
687735 6248514  4659851 192094405        34     72     2         7        0
687736 6248515  4659851 192094406        34     72     2         7        0
687737 6248516  4659851 192094407        34     72     2         7        0
687738 6248517  4659851 192094408        34     72     2         7        0

        usure outil  nligne  ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
487614           20       0  ...     0.0     0.0     0.0     0.0     0.0
487615           20       0  ...     0.0     0.0     0.0     0.0     0.0
487616           20       0  ...     0.0     0.0     0.0     0.0     0.0
487617           20       0  ...     0.0     0.0     0.0     0.0     0.0
487618           20       0  ...     0.0     0.0     0.0     0.0     0.0
...             ...     ...  ...     ...     ...     ...     ...     ...
687734           20       0  ...     0.0     0.0     0.0     0.0     0.0
687735           20       0  ...     0.0     0.0     0.0     0.0     0.0
687736           20       0  ...     0.0     0.0     0.0     0.0     0.0
687737           20       0  ...     0.0     0.0     0.0     0.0     0.0
687738           20       0  ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22  FFT_23  FFT_24
487614     0.0     0.0     0.0     0.0     0.0
487615     0.0     0.0     0.0     0.0     0.0
487616     0.0     0.0     0.0     0.0     0.0
487617     0.0     0.0     0.0     0.0     0.0
487618     0.0     0.0     0.0     0.0     0.0
...        ...     ...     ...     ...     ...
687734     0.0     0.0     0.0     0.0     0.0
687735     0.0     0.0     0.0     0.0     0.0
```

```
687736      0.0      0.0      0.0      0.0      0.0
687737      0.0      0.0      0.0      0.0      0.0
687738      0.0      0.0      0.0      0.0      0.0

[200125 rows x 72 columns]
            tpsT    tps B       date  id_ProgP  id pc  mode  id_outil   n outil  \
488073  6048852  4473751  191729048        34     39     2         0         1
488074  6048853  4473752  191729049        34     39     2         0         1
488075  6048854  4473753  191729056        34     39     2         0         1
488076  6048855  4473754  191729057        34     39     2         0         1
488077  6048856  4473755  191729058        34     39     2         0         1
...         ...      ...        ...       ...   ...   ...       ...       ...
686421  6247200  4658617  192092177        34     32     2         7         1
686422  6247201  4658618  192092178        34     32     2         7         1
686423  6247202  4658619  192092179        34     32     2         7         1
686424  6247203  4658620  192092180        34     32     2         7         1
686425  6247204  4658621  192092181        34     32     2         7         1

        usure outil  nligne  ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
488073           20      21  ...  2394.571   0.379   0.176   0.152  3329.547
488074           20      21  ...  2394.377   0.352   0.184   0.129  3330.492
488075           20      21  ...  3800.209   0.270   0.129   0.121  3332.118
488076           20      21  ...  3240.120   0.219   0.145   0.121  3329.445
488077           20      21  ...  2396.426   0.203   0.125   0.125  3330.149
...             ...     ...  ...       ...     ...     ...     ...       ...
686421           20      34  ...  3340.995   0.340   0.301   0.211   117.108
686422           20      34  ...  3341.663   0.496   0.285   0.207  3394.965
686423           20      34  ...  2634.564   0.348   0.305   0.246   116.191
686424           20      34  ...  2634.564   0.555   0.316   0.305  3395.499
686425           20      34  ...  3430.307   0.461   0.297   0.172  3394.765

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
488073  3310.211  1854.718   0.293   0.172   0.137
488074  3305.693  1855.046   0.285   0.152   0.137
488075  3733.200  3240.884   0.234   0.164   0.141
488076  3731.895  2771.801   0.195   0.156   0.148
488077  2771.283  3731.767   0.156   0.156   0.145
...          ...       ...     ...     ...     ...
686421  3341.907  3394.309   0.379   0.258   0.238
686422   117.083  1482.906   0.387   0.371   0.301
686423  2634.109  3394.587   0.375   0.273   0.270
686424  2634.109   116.646   0.457   0.391   0.367
686425   117.564  2635.158   0.367   0.352   0.230

[786 rows x 72 columns]
Availability:  487614    -785.0
487615   -785.0
487616   -785.0
```

```
487617   -785.0
487618   -785.0
          ...
687734   -785.0
687735   -785.0
687736   -785.0
687737   -785.0
687738   -785.0
Length: 200125, dtype: float64
Performance: 487614     100.0
487615     100.0
487616     100.0
487617     100.0
487618     100.0
          ...
687734     100.0
687735     100.0
687736     100.0
687737     100.0
687738     100.0
Length: 200125, dtype: float64
Quality: 99.60724547158026
OEE: 487614    -7.819169e+06
487615   -7.819169e+06
487616   -7.819169e+06
487617   -7.819169e+06
487618   -7.819169e+06
              ...
687734   -7.819169e+06
687735   -7.819169e+06
687736   -7.819169e+06
687737   -7.819169e+06
687738   -7.819169e+06
Length: 200125, dtype: float64
```

[110]:

```
# Data Selection
OEE_data_35 = HSM_data[HSM_data["id_ProgP"] == 35]
print(OEE_data_35)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_35["PosX"].quantile(0.9)
threshold_Y = OEE_data_35["PosY"].quantile(0.9)
threshold_Z = OEE_data_35["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_35["PosX"] > threshold_X) & (OEE_data_35["PosY"] >
 ↪threshold_Y) & (OEE_data_35["PosZ"] > threshold_Z)
```

```
# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_35[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_35["tps B"] - OEE_data_35["tpsT"]) / 60
idle_time = (OEE_data_35["Vf"] == 1).sum() * scheduled_production_time
#

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *
  ↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /
  ↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_35) - len(malfunctioned_data)

# Calculate the total number of units
total_number_of_units = len(OEE_data_35)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
690310 6251089  4659851  192098784        35     73     2         0        0
690311 6251090  4659851  192098785        35     73     2         0        0
690312 6251091  4659851  192098786        35     73     2         0        0
690313 6251092  4659851  192098787        35     73     2         0        0
690314 6251093  4659851  192098788        35     73     2         0        0
...        ...      ...        ...       ...    ...   ...       ...
831420 6392199  4783338  192355984        35     73     2         7        0
831421 6392200  4783338  192355985        35     73     2         7        0
831422 6392201  4783338  192355986        35     73     2         7        0
```

```
831423  6392202  4783338  192355987       35     73     2          7          0
831424  6392203  4783338  192355988       35     73     2          7          0

        usure outil   nligne   ...  FFT_15  FFT_16  FFT_17  FFT_18  FFT_19  \
690310             20        0  ...     0.0     0.0     0.0     0.0     0.0
690311             20        0  ...     0.0     0.0     0.0     0.0     0.0
690312             20        0  ...     0.0     0.0     0.0     0.0     0.0
690313             20        0  ...     0.0     0.0     0.0     0.0     0.0
690314             20        0  ...     0.0     0.0     0.0     0.0     0.0
...               ...      ...  ...     ...     ...     ...     ...     ...
831420             20        0  ...     0.0     0.0     0.0     0.0     0.0
831421             20        0  ...     0.0     0.0     0.0     0.0     0.0
831422             20        0  ...     0.0     0.0     0.0     0.0     0.0
831423             20        0  ...     0.0     0.0     0.0     0.0     0.0
831424             20        0  ...     0.0     0.0     0.0     0.0     0.0

        FFT_20  FFT_21  FFT_22  FFT_23 FFT_24
690310     0.0     0.0     0.0     0.0     0.0
690311     0.0     0.0     0.0     0.0     0.0
690312     0.0     0.0     0.0     0.0     0.0
690313     0.0     0.0     0.0     0.0     0.0
690314     0.0     0.0     0.0     0.0     0.0
...        ...     ...     ...     ...     ...
831420     0.0     0.0     0.0     0.0     0.0
831421     0.0     0.0     0.0     0.0     0.0
831422     0.0     0.0     0.0     0.0     0.0
831423     0.0     0.0     0.0     0.0     0.0
831424     0.0     0.0     0.0     0.0     0.0

[141115 rows x 72 columns]
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
690793  6251572  4660215  192099619        35     39     2         0        1
690794  6251573  4660216  192099620        35     39     2         0        1
690795  6251574  4660217  192099621        35     39     2         0        1
690796  6251575  4660218  192099622        35     39     2         0        1
690797  6251576  4660219  192099623        35     39     2         0        1
...         ...      ...        ...       ...    ...   ...       ...      ...
696458  6257237  4665880  192109256        35     39     2         0        1
696459  6257238  4665881  192109257        35     39     2         0        1
696460  6257239  4665882  192109264        35     39     2         0        1
696461  6257240  4665883  192109265        35     39     2         0        1
696462  6257241  4665884  192109266        35     39     2         0        1

        usure outil  nligne  ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
690793           20      21  ...  3309.961   0.172   0.133   0.121  3331.826
690794           20      21  ...  3710.804   0.160   0.129   0.121  3333.063
690795           20      21  ...  3710.085   0.168   0.113   0.109  2770.892
690796           20      21  ...  2770.449   0.145   0.137   0.125  2770.253
```

```
690797             20      21  ...  2771.125   0.145   0.141   0.125  2771.125
...               ...     ...  ...       ...     ...     ...     ...       ...
696458             20      83  ...  4027.872   1.562   1.113   0.676   395.374
696459             20      83  ...   790.431   1.551   1.160   0.707   395.215
696460             20      83  ...   790.443   1.473   1.090   0.711   395.221
696461             20      83  ...   790.870   1.531   1.105   0.684   395.435
696462             20      83  ...   790.968   1.500   1.023   0.711   395.484

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
690793  3365.990  2826.986   0.148   0.129   0.129
690794  2822.117  2768.991   0.117   0.109   0.109
690795  3327.219  3315.303   0.148   0.125   0.121
690796  3708.974  3308.479   0.156   0.117   0.117
690797  3328.633  2825.840   0.164   0.125   0.113
...          ...       ...     ...     ...     ...
696458  1581.496   790.748   1.937   1.008   0.824
696459  1580.861   790.431   1.926   1.090   0.828
696460  1582.429   790.443   1.867   1.031   0.820
696461  1581.740   790.870   1.926   1.098   0.797
696462  1581.935   790.968   1.863   1.027   0.840

[220 rows x 72 columns]
Availability: 690310    -219.0
690311   -219.0
690312   -219.0
690313   -219.0
690314   -219.0
          ...
831420   -219.0
831421   -219.0
831422   -219.0
831423   -219.0
831424   -219.0
Length: 141115, dtype: float64
Performance: 690310     100.0
690311     100.0
690312     100.0
690313     100.0
690314     100.0
          ...
831420     100.0
831421     100.0
831422     100.0
831423     100.0
831424     100.0
Length: 141115, dtype: float64
Quality: 99.84409878467916
OEE: 690310    -2.186586e+06
```

```
690311   -2.186586e+06
690312   -2.186586e+06
690313   -2.186586e+06
690314   -2.186586e+06
              ...
831420  -2.186586e+06
831421  -2.186586e+06
831422  -2.186586e+06
831423  -2.186586e+06
831424  -2.186586e+06
Length: 141115, dtype: float64
```

[111]:
```
# Data Selection
OEE_data_36 = HSM_data[HSM_data["id_ProgP"] == 36]
print(OEE_data_36)

# Calculate the 90th percentile for each column
threshold_X = OEE_data_36["PosX"].quantile(0.9)
threshold_Y = OEE_data_36["PosY"].quantile(0.9)
threshold_Z = OEE_data_36["PosZ"].quantile(0.9)

# Assuming KI_data2 is your DataFrame
malfunction_time = (OEE_data_36["PosX"] > threshold_X) & (OEE_data_36["PosY"] >␣
↪threshold_Y) & (OEE_data_36["PosZ"] > threshold_Z)

# Use the boolean mask to filter the DataFrame
malfunctioned_data = OEE_data_36[malfunction_time]
print(malfunctioned_data )
scheduled_production_time = (OEE_data_36["tps B"] - OEE_data_36["tpsT"]) / 60
idle_time = (OEE_data_36["Vf"] == 1).sum() * scheduled_production_time

# malfunction_time = (KI_data2["is_malfunctioned"] == 1).sum() *␣
↪scheduled_production_time
malfunction_time = len(malfunctioned_data) * scheduled_production_time
downtime = idle_time + malfunction_time
production_time = scheduled_production_time - downtime

# Calculate availability
availability = (scheduled_production_time - downtime) /␣
↪scheduled_production_time

# Calculate performance
performance = (production_time / (scheduled_production_time - downtime)) * 100

# Calculate the number of good units
number_of_good_units = len(OEE_data_36) - len(malfunctioned_data)
```

```python
# Calculate the total number of units
total_number_of_units = len(OEE_data_36)

# Calculate quality
quality = (number_of_good_units / total_number_of_units) * 100

# Calculate OEE
oee = availability * performance * quality
print("Availability:", availability)
print("Performance:", performance)
print("Quality:", quality)
print("OEE:", oee)
```

```
           tpsT    tps B       date  id_ProgP  id pc  mode  id_outil  n outil  \
837959 6398738  4783338  192367145        36     74     2         0        0
837960 6398739  4783338  192367152        36     74     2         0        0
837961 6398740  4783338  192367153        36     74     2         0        0
837962 6398741  4783338  192367154        36     74     2         0        0
837963 6398742  4783338  192367155        36     74     2         0        0
...        ...      ...        ...       ...    ...    ...       ...      ...
862569 6423348  4802871  192409145        36     74     2        22        6
862570 6423349  4802872  192409152        36     74     2        22        6
862571 6423350  4802873  192409153        36     74     2        22        6
862572 6423351  4802874  192409154        36     74     2        22        6
862573 6423352  4802875  192409155        36     74     2        22        6

        usure outil  nligne  ...    FFT_15  FFT_16  FFT_17  FFT_18    FFT_19  \
837959           20       0  ...     0.000   0.000   0.000   0.000     0.000
837960           20       0  ...     0.000   0.000   0.000   0.000     0.000
837961           20       0  ...     0.000   0.000   0.000   0.000     0.000
837962           20       0  ...     0.000   0.000   0.000   0.000     0.000
837963           20       0  ...     0.000   0.000   0.000   0.000     0.000
...             ...     ...  ...       ...     ...     ...     ...       ...
862569          300     402  ...  2372.390   4.629   3.148   3.145  1186.195
862570          300     402  ...  2371.731   4.594   3.160   2.945  1185.866
862571          300     402  ...  2767.105   4.492   3.418   3.098  1185.902
862572          300     402  ...  2767.960   4.605   3.074   3.027  1186.268
862573          300     402  ...  2372.061   4.328   3.293   3.094  1186.030

          FFT_20    FFT_21  FFT_22  FFT_23  FFT_24
837959     0.000     0.000   0.000   0.000   0.000
837960     0.000     0.000   0.000   0.000   0.000
837961     0.000     0.000   0.000   0.000   0.000
837962     0.000     0.000   0.000   0.000   0.000
837963     0.000     0.000   0.000   0.000   0.000
...          ...       ...     ...     ...
862569  2372.390  1583.138   5.457   4.285   1.906
```

```
862570  2371.731  1582.698   5.453   4.113   1.660
862571  2371.804  1582.747   5.340   4.461   1.746
862572  2372.537  1583.236   5.473   4.250   1.820
862573  2372.061  2767.404   5.145   4.277   1.656

[24615 rows x 72 columns]
Empty DataFrame
Columns: [tpsT, tps B, date, id_ProgP, id pc, mode, id_outil, n outil, usure outil, nligne, nbloc, Abloc, Cbloc, Temp_1, Temp_2, Temp_3, Temp_4, Arms_1, Arms_2, Arms_3, Arms_4, Apic_1, Apic_2, Apic_3, Apic_4, Vrms_1, Vrms_2, Vrms_3, Vrms_4, Vpic_1, Vpic_2, Vpic_3, Vpic_4, PosX, PosY, PosZ, PosA, PosC, VitX, VitY, VitZ, VitA, VitC, Vf, N, P, %Vf, %N, FFT_1, FFT_2, FFT_3, FFT_4, FFT_5, FFT_6, FFT_7, FFT_8, FFT_9, FFT_10, FFT_11, FFT_12, FFT_13, FFT_14, FFT_15, FFT_16, FFT_17, FFT_18, FFT_19, FFT_20, FFT_21, FFT_22, FFT_23, FFT_24]
Index: []

[0 rows x 72 columns]
Availability: 837959      1.0
837960    1.0
837961    1.0
837962    1.0
837963    1.0
         ...
862569    1.0
862570    1.0
862571    1.0
862572    1.0
862573    1.0
Length: 24615, dtype: float64
Performance: 837959     100.0
837960     100.0
837961     100.0
837962     100.0
837963     100.0
          ...
862569     100.0
862570     100.0
862571     100.0
862572     100.0
862573     100.0
Length: 24615, dtype: float64
Quality: 100.0
OEE: 837959      10000.0
837960     10000.0
837961     10000.0
837962     10000.0
837963     10000.0
           ...
```

```
862569    10000.0
862570    10000.0
862571    10000.0
862572    10000.0
862573    10000.0
Length: 24615, dtype: float64
```

5 Objective 4: machining incidents

[114]:
```python
# import pandas as pd
# Define a threshold for cutting force
threshold = 1000 # Change this value to your desired threshold
threshold = float(threshold)
HSM_data = pd.read_csv("data777.csv")

# Load the data into a DataFrame
selected_columns = ["tpsT","tps B","PosX","PosY","PosZ","Vf", "N","P",]
KI_data2 = HSM_data[selected_columns]

# Iterate through the 'Vf' column
malfunction_timestamps = []  # Initialize the list
malfunction_durations = []  # Initialize the list

for i, cutting_force_value in enumerate(KI_data2["Vf"]):
    cutting_force_str = str(cutting_force_value)

    #  Do something with cutting_force_str
    #  print(cutting_force_str)    # Replace this line with your desired operation

    # Convert cutting_force_str to the appropriate numeric type
    cutting_force_numeric  =  float(cutting_force_str)    # Assuming␣
↪cutting_force_str is a numeric value
    # Do something with cutting_force_numeric
    # print(cutting_force_numeric)

    # Check if the cutting force exceeds the threshold
    if cutting_force_numeric > threshold:

        # If so, add the timestamp to the malfunction timestamps list
        malfunction_timestamps.append(i)

#  Calculate malfunction durations
for i in range(len(malfunction_timestamps) - 1):
    start_timestamp = malfunction_timestamps[i]
    end_timestamp = malfunction_timestamps[i + 1]
    malfunction_duration  =  end_timestamp  -  start_timestamp
    malfunction_durations.append(malfunction_duration)
```

```python
# Identify machining incidents
for duration in malfunction_durations:
    if duration > 60:  # Threshold for prolonged malfunction
        # Generate different options for machining incidents
        if cutting_force_numeric < 2000:
            incident_type = 'Minor malfunction'
        elif cutting_force_numeric < 3000:
            incident_type = 'Moderate malfunction'
        else:
            incident_type = 'Major malfunction'
        incident_description = f"Cutting force exceeded the threshold of␣
↪{threshold} for {duration} seconds. This could lead to premature tool wear,␣
↪surface defects, or even workpiece breakage."
        print("Machining incident detected:", incident_type,␣
↪incident_description)
```

```
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 110 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 69 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 93 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 81 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 66 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 72 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 84 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 67 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 69 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the
threshold of 1000.0 for 69 seconds. This could lead to premature tool wear,
surface defects, or even workpiece breakage.
```

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 122 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 122 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 72 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 78 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 98 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 93 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 77 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 98 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 122 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 114 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 122 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 114 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 128 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 78 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 5534 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 103 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 91 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 79 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 83 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 100 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 92 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 96 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 75 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 97 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 118 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 111 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 146 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 88 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 80 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 93 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 145 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 146 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 146 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 146 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 87 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 243 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 147 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 242 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 79 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 2965 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 208 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71136 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 3225 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 384 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 117 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 110 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 120 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 123 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 77 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 5934 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 94 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 92 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 81 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 86 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 100 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 71 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 75 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 100 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 96 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 75 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 97 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 272 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 85 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 271 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 270 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 78 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 118 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 111 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 77 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 101 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 111 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 116 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 78 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 2128 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 94 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 92 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 80 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 85 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 62 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 119 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 119 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 99 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 91 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 76 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 100 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 77 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 99 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 76 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 102 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 8818 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 150 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 91 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 102 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 7456 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 1221 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 120 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 119 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 118 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 114 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 91 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 120 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 113 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 119 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 121 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 124 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 78 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 6087 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 94 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 70 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 91 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 79 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 84 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 69 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 66 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 64 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 63 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 65 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 67 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 61 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 100 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 68 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.
Machining incident detected: Major malfunction Cutting force exceeded the threshold of 1000.0 for 76 seconds. This could lead to premature tool wear, surface defects, or even workpiece breakage.

Created in Deepnote