

## تحلیل داده‌های نگهداری و تعمیرات

ایمیل: ariandkh@gmail.com

لينكدين: www.linkedin.com/in/arian-dadkhah

# تحلیل داده‌های نگهداری و تعمیرات دستگاه کمپرسور

در این پژوهه، هدف ما تحلیل داده‌های مربوط به نگهداری و تعمیرات یک دستگاه کمپرسور است. این داده‌ها شامل اطلاعات مربوط به خرابی‌ها، تعمیرات، زمان‌بندی‌های نگهداری، و ویژگی‌های عملکرد دستگاه می‌باشند. هدف اصلی این است که از این داده‌ها برای شناسایی الگوهای خرابی و روندهای خرابی دستگاه استفاده کنیم تا بتوانیم به پیش‌بینی خرابی‌ها و بهبود برنامه‌ریزی نگهداری دست یابیم.

با تحلیل این داده‌ها، می‌توانیم تشخیص دهیم که چه پارامترهایی بیشترین تأثیر را در خرابی دستگاه دارند. این اطلاعات به ما کمک می‌کند تا برنامه‌ریزی دقیق‌تری برای تعمیرات و نگهداری دستگاه‌ها داشته باشیم و هزینه‌های مربوط به خرابی‌های غیرمنتظره را کاهش دهیم.

در این پژوهه، ما داده‌های مختلفی مانند گشتاور ماشین، سرعت چرخش اجزای ماشین، دما، و مدت زمان سایش ابزار را برای شناسایی الگوهای خرابی مورد بررسی قرار می‌دهیم. همچنین، با تحلیل روند تغییرات این پارامترها در طول زمان، می‌توانیم مشکلات احتمالی را قبل از وقوع خرابی‌های جدی شناسایی کنیم.

هدف نهایی این است که بهوسیله شبیه‌سازی و تحلیل داده‌ها، پیش‌بینی‌های دقیقی از خرابی‌ها و نیاز به نگهداری انجام دهیم و در نتیجه طول عمر دستگاه را افزایش داده و هزینه‌های عملیاتی را کاهش دهیم.

## الگوهای خرابی و روندهای دستگاه

در این بخش، ما بر روی شناسایی الگوهای خرابی دستگاه تمرکز داریم. این الگوها می‌توانند شامل تغییرات مشخصی در پارامترهای دستگاه در طول زمان باشند. برای مثال، افزایش تدریجی دما یا کاهش گشتاور می‌تواند نشانه‌ای از خرابی در آینده باشد. با استفاده از این الگوها، می‌توانیم زمان‌های بهینه برای تعمیرات پیشگیرانه و نگهداری دستگاهها را شناسایی کنیم.

همچنین، با بررسی این الگوهای خرابی‌های مختلف، قادر خواهیم بود تشخیص دهیم که کدام پارامترها برای هر نوع خرابی بیشتر تأثیر دارند و این اطلاعات به ما کمک می‌کند که اقدامات نگهداری مناسب‌تری انجام دهیم.

## اهداف پروژه

هدف اصلی این پروژه تحلیل داده‌های نگهداری و تعمیرات دستگاه کمپرسور به‌منظور شناسایی الگوها و روندهای خرابی است. از این طریق می‌توانیم:

- الگوهای خرابی را شناسایی و روند تغییرات پارامترها را بررسی کنیم.
- پارامترهایی که بیشترین تأثیر را در خرابی‌ها دارند شناسایی کنیم.
- برای نگهداری پیشگیرانه و به‌موقع، برنامه‌ریزی دقیقترا انجام دهیم.
- پیش‌بینی خرابی‌ها را بهبود دهیم و به‌طور مؤثری هزینه‌های نگهداری دستگاهها را کاهش دهیم.

## نتایج مورد انتظار

با تحلیل داده‌ها و شناسایی الگوهای خرابی دستگاه، ما انتظار داریم:

- شناسایی دقیق زمان‌های خرابی و نیاز به نگهداری.
- کاهش خرابی‌های غیرمنتظره و افزایش طول عمر دستگاه.
- بهبود عملکرد دستگاه و کاهش هزینه‌های تعمیرات.

این تحلیل‌ها به شرکت کمک می‌کند تا در هزینه‌های نگهداری صرفه‌جویی کند و با کاهش خرابی‌های غیرمنتظره، بهره‌وری دستگاهها را افزایش دهد.

## ساختار داده‌ها

داده‌ها در این پروژه در قالب یک فایل CSV ذخیره شده‌اند که شامل اطلاعات مختلفی در مورد دستگاه کمپرسور و ویژگی‌های عملکردی آن است. این فایل به صورت جدول داده‌ها (DataFrame) بازگذاری می‌شود که هر ستون نشان‌دهنده یک ویژگی خاص است. به طور کلی، ساختار داده‌ها شامل ستون‌های زیر است:

کد بررسی : شناسه یا کد منحصر به فرد برای هر بررسی انجام شده روی دستگاه.

- دمای هوای اطراف ماشین (کلوین) : دمای محیط یا هوای اطراف دستگاه کمپرسور که می‌تواند تأثیر زیادی بر روی عملکرد آن داشته باشد.
- دمای فرآیند (کلوین) : دمای فرآیند داخل دستگاه که نشان‌دهنده دمای عملیاتی دستگاه است.
- سرعت چرخش اجزای ماشین (دور در دقیقه) : سرعت چرخش قطعات مختلف دستگاه که می‌تواند به عنوان معیاری برای شناسایی بار اضافی یا خرابی در قطعات باشد.
- گشتاور ماشین (نیوتون متر) : میزان گشتاوری که دستگاه تولید می‌کند که اطلاعاتی در مورد عملکرد دستگاه و بار آن فراهم می‌آورد.
- مدت زمان سایش ابزار (دقیقه) : مدت زمانی که ابزارهای دستگاه در حال استفاده بوده‌اند و می‌تواند ارتباطی با خرابی‌ها یا نیاز به تعمیرات داشته باشد.

- وضعیت خرابی : وضعیت کلی دستگاه (0 برای بدون خرابی و 1 برای خرابی ماشین).
- نوع خرابی : نوع خرابی دستگاه، مانند خرابی در اتلاف گرما، خرابی نیرو، خرابی ناشی از فشار بیش از حد، یا خرابی ناشی از ساییدگی ابزار.

با استفاده از این داده‌ها، می‌توانیم تحلیل‌هایی مانند شناسایی الگوهای خرابی در زمان‌های مختلف، بررسی تاثیر شرایط محیطی و عملیاتی دستگاه بر خرابی‌ها، و پیش‌بینی خرابی‌های احتمالی آینده انجام دهیم. به علاوه، تحلیل این داده‌ها می‌تواند به بهینه‌سازی فرآیندهای نگهداری و تعمیرات کمک کند.

```
:In [14]
import pandas as pd
import requests
from io import StringIO

# لینک به فایل Google Drive #
file_url = "https://drive.google.com/file/d/1oPWP0P52vH3cUG_922hvLElpHy3lIqv0/view?usp=drive_link"

# استخراج ID فایل از لینک
file_id = file_url.split('/')[-1].split('.')[0]

# ساخت لینک مستقیم دانلود
direct_url = f"https://drive.google.com/uc?id={file_id}&export=download

# دانلود فایل و بارگذاری به pandas
response = requests.get(direct_url)

# خواندن محتوا به عنوان CSV
data = pd.read_csv(StringIO(response.text))

# نمایش داده‌ها به صورت جدول
()data.head
```

:Out[14]

نوع خرابی	وضعیت خرابی	مدت زمان سایش ابزار (دقیقه)	گشتاور ماشین (نیوتون متر)	سرعت چرخش اجزای ماشین (دور در دقیقه)	دما فرآیند (کلوین)	دماهی هوای اطراف ماشین (کلوین)	کد بررسی
No Failure	0	0	42.8	1551	308.6	298.1	1 0
No Failure	0	3	46.3	1408	308.7	298.2	2 1
No Failure	0	5	49.4	1498	308.5	298.1	3 2
No Failure	0	7	39.5	1433	308.6	298.2	4 3
No Failure	0	9	40.0	1408	308.7	298.2	5 4

## پاکسازی داده‌ها

پاکسازی داده‌ها یکی از مراحل کلیدی برای اطمینان از کیفیت داده‌ها و دقت تحلیل‌ها است. در این پژوهه، مراحل زیر برای پاکسازی داده‌ها انجام شده است:

- استانداردسازی نام ستون‌ها: حذف فاصله‌های اضافی از نام ستون‌ها و اطمینان از یکنواختی نام‌گذاری برای جلوگیری از خطاهای احتمالی.
- جایگزینی مقادیر عددی با توضیحات متى: مقادیر عددی ستون "وضعیت خرابی" به عبارات متى "بدون خرابی" و "خرابی ماشین" تبدیل شدند.
- ترجمه مقادیر نوع خرابی: مقادیر ستون "نوع خرابی" از انگلیسی به فارسی ترجمه شدند تا خوانایی بیشتری داشته باشند. به عنوان مثال:
  - "بدون خرابی" به "No Failure" تبدیل شد.
  - "خرابی دفع گرما" به "Heat Dissipation Failure" تبدیل شد.
  - "Power Failure" به "خرابی توان" تبدیل شد.
- تبدیل داده‌ها به نوع عددی: ستون‌هایی که مقادیر عددی داشتند، بررسی و به نوع عددی تبدیل شدند تا برای محاسبات آماری آماده

باشند.

با انجام این فرآیندها، داده‌ها برای تحلیل‌های دقیق‌تر آماده شدند و امکان تمرکز بر روی نتایج اصلی فراهم شد.

```
:In [15]
import numpy as np
import pandas as pd

# استانداردسازی نام ستون‌ها (حذف فاصله‌های اضافی از نام ستون‌ها)
()data.columns = data.columns.str.strip

# تغییر مقادیر ستون "وضعیت خرابی" به عبارات توصیفی
("وضعیت خرابی" == 0, "بدون خرابی", "خرابی ماشین")
)data["وضعیت خرابی"] = np.where(data["وضعیت خرابی"] == 0, "بدون خرابی", "خرابی ماشین")

# ترجمه مقادیر ستون "نوع خرابی" به فارسی
("نوع خرابی" = ["بدون خرابی", "No Failure",
                   "Heat Dissipation Failure", "خرابی دفع گرما",
                   "Power Failure", "خرابی توان", "Overstrain Failure",
                   "Tool Wear Failure", "خرابی ابزار",
                   "Random Failures"])
)replace("نوع خرابی", {"بدون خرابی": "No Failure",
                           "Heat Dissipation Failure": "خرابی دفع گرما",
                           "Power Failure": "خرابی توان",
                           "Overstrain Failure": "خرابی ناشی از فشار بیش از حد",
                           "Tool Wear Failure": "خرابی ابزار",
                           "Random Failures": "خرابی‌های تصادفی"})

# تبدیل ستون‌های عددی به نوع عددی (در صورت نیاز به اجتناب از خطاهای تبدیل، استفاده از
# numeric_columns = numeric_columns[["دماهی هوای اطراف ماشین (کلوین)", "دماهی فرآیند (کلوین)", "سرعت چرخش اجزای ماشین (دور در دقیقه)", "گشتاور ماشین (نیوتن متر)", "مدت زمان سایش ابزار (دقیقه)"]]
)numeric_columns = data[numeric_columns].apply(pd.to_numeric, errors='coerce')

# نمایش پیش‌نمایش از داده‌های تمیز شده
)data.head
```

:Out[15]

نوع خرابی	وضعیت خرابی	مدت زمان سایش ابزار (دقیقه)	گشتاور ماسین (نیوتن متر)	سرعت چرخش اجزای ماشین در دقیقه (دور کلوین)	دما فرآیند (کلوین)	دماهای هوای اطراف ماشین (کلوین)	کد بررسی
بدون خرابی	بدون خرابی	0	42.8	1551	308.6	298.1	1 0
بدون خرابی	بدون خرابی	3	46.3	1408	308.7	298.2	2 1
بدون خرابی	بدون خرابی	5	49.4	1498	308.5	298.1	3 2
بدون خرابی	بدون خرابی	7	39.5	1433	308.6	298.2	4 3
بدون خرابی	بدون خرابی	9	40.0	1408	308.7	298.2	5 4

## تعداد تکرار خرابی ها

در این بخش، تحلیل داده‌ها را با بررسی تعداد تکرار هر نوع خرابی آغاز می‌کنیم. این تحلیل به ما کمک می‌کند تا متوجه شویم کدام نوع خرابی بیشتر رخ داده و اولویت‌بندی در بررسی‌ها و تحلیل‌های عمیق‌تر را مشخص کنیم.

برای این منظور، تعداد تکرار هر خرابی به صورت نمودار نمایش داده می‌شود تا بتوانیم مقایسه‌ای بصری میان انواع خرابی‌ها انجام دهیم.

:In [16]

```

import warnings
import arabic_reshaper
from bidi.algorithm import get_display
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import font_manager

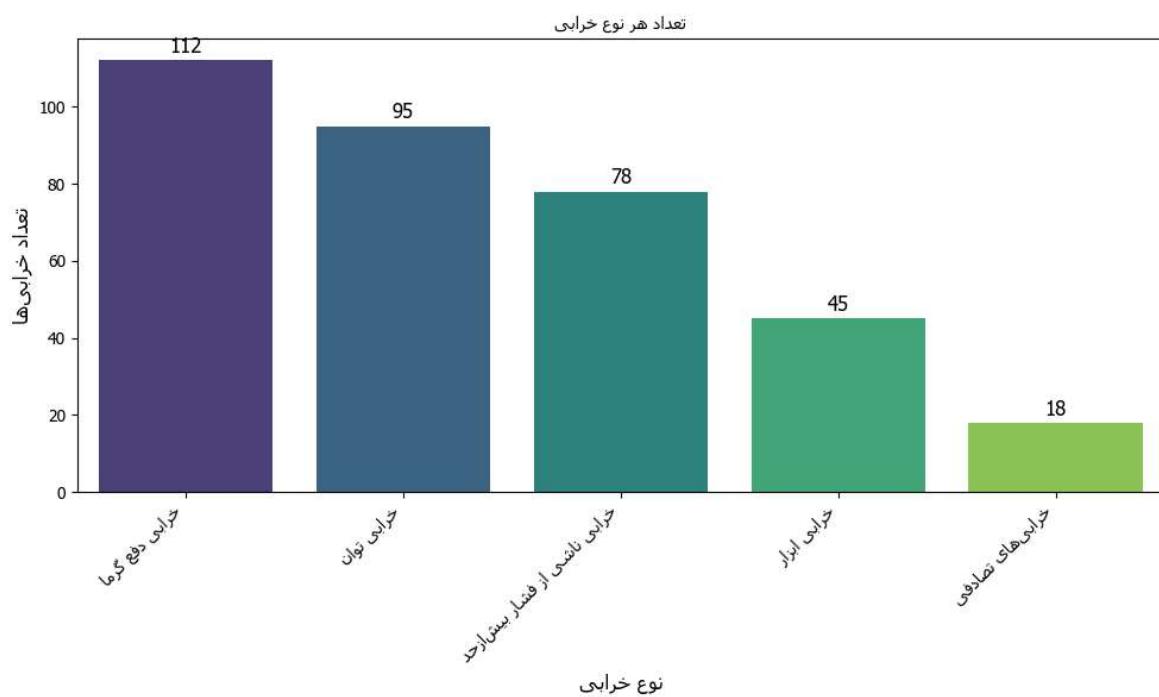
# غیرفعال کردن نمایش پیغام‌های هشدار
warnings.filterwarnings('ignore', category=FutureWarning)

# تنظیم فونت
# font_path = 'C:/Windows/Fonts/Tahoma.ttf'
prop = font_manager.FontProperties(fname=font_path)

# اصلاح متن فارسی برای نمایش صحیح
def reshaped_text(text)
    reshaped = arabic_reshaper.reshape(text)
    return get_display(reshaped)

```

```
# فیلتر کردن داده‌ها تا فقط خرابی‌های ماشین نمایش داده شوند  
[ 'نوع خرابی' ] ! = 'بدون خرابی'  
[ data ] filtered_data = data  
  
# گروهبندی داده‌ها بر اساس نوع خرابی و شمارش تعداد هر نوع خرابی  
failure_counts = failure_counts[ 'نوع خرابی' ].value_counts()  
  
# مرتب‌سازی داده‌ها طبق تعداد خرابی‌ها به ترتیب نزولی  
failure_counts = failure_counts.sort_values(ascending=False)  
  
# رسم نمودار میله‌ای  
plt.figure(figsize=(10,6))  
sns.barplot(x=failure_counts.index, y=failure_counts.values, palette='viridis')  
  
# اعمال فونت فارسی و اصلاح متن برای عنوان و برچسب‌ها  
(fontsize=16, fontproperties=prop)'تعداد هر نوع خرابی') , plt.title  
(fontsize=12, fontproperties=prop)'نوع خرابی' , plt.xlabel  
(fontsize=12, fontproperties=prop)'تعداد خرابی‌ها' , plt.ylabel  
  
# اصلاح ترتیب برچسب‌های محور X و نمایش حروف فارسی  
ticks = range(len(failure_counts))  
for label in failure_counts.index], rotation=45, ha='right', fontproperties=prop)  
  
# اضافه کردن دیتالیلیل به هر میله  
:for p in ax.patches  
    , 'ax.annotate(f'{p.get_height():,.0f}'  
, ((p.get_x() + p.get_width() / 2., p.get_height())  
        , 'ha='center', va='center'  
        , 'fontsize=12, color='black'  
        ('xytext=(0, 8), textcoords='offset points  
  
# اصلاح چیدمان و نمایش نمودار  
(plt.tight_layout()  
(plt.show()
```



در این نمودار تعداد تکرار انواع خرابی‌های دستگاه نمایش داده شده است. این اطلاعات به ما کمک می‌کند تا الگوهای خرابی دستگاه را شناسایی کنیم و درک بهتری از نوع خرابی‌هایی که بیشتر رخ می‌دهند پیدا کنیم.

همانطور که در نمودار مشاهده می‌شود، خرابی‌های دفع گرما بیشترین تعداد تکرار را دارند. این خرابی‌ها ممکن است به دلیل مشکلاتی مانند عدم عملکرد مناسب سیستم‌های خنک‌کننده یا طراحی نامناسب دستگاه باشد. بنابراین، بررسی و تحلیل دقیق‌تر این خرابی‌ها اهمیت بالایی دارد.

در گام بعدی، می‌خواهیم خرابی‌های دفع گرما را به‌طور خاص بررسی کنیم تا علت وقوع آن‌ها و نحوه پیشگیری از آن‌ها را شناسایی کنیم. این مرحله می‌تواند به بهبود نگهداری و عملکرد دستگاه کمک کند.

## تحلیل خرابی دفع گرما

در این بخش از تحلیل، به بررسی رفتار دستگاه در شرایطی پرداخته‌ایم که خرابی دفع گرما رخ داده است. ابتدا سطرهای مربوط به خرابی دفع گرما از داده‌ها استخراج شدند. سپس، با انتخاب ستون‌های مرتبط با عملکرد دستگاه مانند دمای هوای اطراف ماشین، دمای فرآیند، سرعت چرخش اجزای ماشین، گشتاور و مدت زمان سایش ابزار، میانگین مقادیر این ویژگی‌ها محاسبه شد.

این میانگین‌ها به‌منظور بررسی عملکرد دستگاه در شرایط خرابی دفع گرما استفاده شده و می‌توانند برای مقایسه با سایر خرابی‌ها یا شرایط بهینه دستگاه مورد استفاده قرار گیرند.

```
# فیلتر کردن سطرهای مربوط به خرابی دفع گرما
[{"نوع خرابی": "خرابی دفع گرما"}]heat_failure_data = data[["نوع خرابی"]]

# انتخاب ستون‌های خاص برای محاسبه میانگین (مثال: دمای هوای اطراف ماشین، دمای فرآیند و
# "سرعت چرخش اجزای ماشین (دور در دقیقه)", "گشتاور ماشین (نیوتون متر)", "مدت زمان سایش ابزار (دقیقه)"]
selected_columns = ["دمای هوای اطراف ماشین (کلوین)", "دمای فرآیند (کلوین)", "سرعت چرخش اجزای ماشین (دور در دقیقه)", "گشتاور ماشین (نیوتون متر)", "مدت زمان سایش ابزار (دقیقه)"]
```

:In [17]

```
# فیلتر کردن داده‌ها برای ستون‌های خاص
selected_data = heat_failure_data[selected_columns]

# محاسبه میانگین برای ستون‌های انتخاب شده
mean_values = selected_data.mean()

# تبدیل به DataFrame برای نمایش به صورت جدول
mean_values_df = mean_values.to_frame().T

# نمایش نتایج
mean_values_df
```

مدت زمان سایش ابزار (دقیقه)	گشتاور ماشین (نیوتون متر)	سرعت چرخش اجزای ماشین (دور در دقیقه)	دماهای اطراف ماشین (کلوین)	دما فرآیند (کلوین)	:Out[17]
107.339286	52.778571	1337.964286	310.799107	302.567857	0

## مقایسه میانگین خرابی دفع گرما و میانگین کل

در این بخش، به مقایسه میانگین مقادیر مختلف ویژگی‌ها در وضعیت خرابی دفع گرما و میانگین کلی داده‌ها پرداخته‌ایم. ابتدا سطرهای مربوط به خرابی دفع گرما از داده‌ها استخراج شدند. سپس، میانگین مقادیر ویژگی‌هایی مانند دمای هوای اطراف ماشین، دمای فرآیند، سرعت چرخش اجزای ماشین، گشتاور و مدت زمان سایش ابزار، در دو وضعیت خرابی دفع گرما و کل داده‌ها محاسبه گردید.

تفاوت میان میانگین خرابی دفع گرما و میانگین کل محاسبه شده است و این مقادیر در یک نمودار میله‌ای به‌طور مقایسه‌ای نمایش داده شده‌اند. این نمودار نشان می‌دهد که خرابی دفع گرما چگونه می‌تواند بر عملکرد دستگاه تأثیر بگذارد و مقایسه آن با وضعیت کل می‌تواند الگوهای جالبی را برای تحلیل و پیش‌بینی خرابی‌ها در آینده فراهم کند.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper
```

```

from matplotlib import font_manager

# تنظیم فونت فارسی
font_path = 'C:/Windows/Fonts/Tahoma.ttf' # مسیر فونت فارسی
prop = font_manager.FontProperties(fname=font_path)

# اصلاح متن فارسی
def reshaped_text(text):
    reshaped = arabic_reshaper.reshape(text)
    return get_display(reshaped)

# فیلتر کردن سطرهای مربوط به خرابی دفع گرما
heat_failure_data = data[[data['نوع خرابی'] == "خرابی دفع گرما"]]

# انتخاب ستونهای خاص برای محاسبه میانگین (مثالاً: دمای هوای اطراف ماشین، دمای فرآیند و
selected_columns = ["دمای هوای اطراف ماشین (کلوین)", "دمای فرآیند (کلوین)", "سرعت چرخش اجزای ماشین (دور در دقیقه)", "گشتاور ماشین (نیوتون متر)،
                     "مدت زمان سایش ابزار (دقیقه)"]

# فیلتر کردن داده‌ها برای ستونهای خاص
selected_data = heat_failure_data[selected_columns]

# محاسبه میانگین برای ستونهای انتخاب شده در وضعیت خرابی دفع گرما
heat_failure_mean = selected_data.mean()

# محاسبه میانگین برای ستونهای انتخاب شده در تمام داده‌ها
overall_mean = data[selected_columns].mean()

# محاسبه تفاوت میانگین خرابی دفع گرما نسبت به میانگین کل
difference = heat_failure_mean - overall_mean

# تبدیل به DataFrame برای رسم نمودار
df_diff = pd.DataFrame(
    {'میانگین خرابی دفع گرما': heat_failure_mean,
     'میانگین کل': overall_mean,
     'تفاوت': difference
}, {}

# تنظیمات نمودار
plt.figure(figsize=(12, 6))

# رسم نمودار میله‌ای برای مقایسه
df_diff[['میانگین خرابی دفع گرما', 'میانگین کل']].plot(kind='bar', color=['lightblue', 'lightgreen'], edgecolor='black')

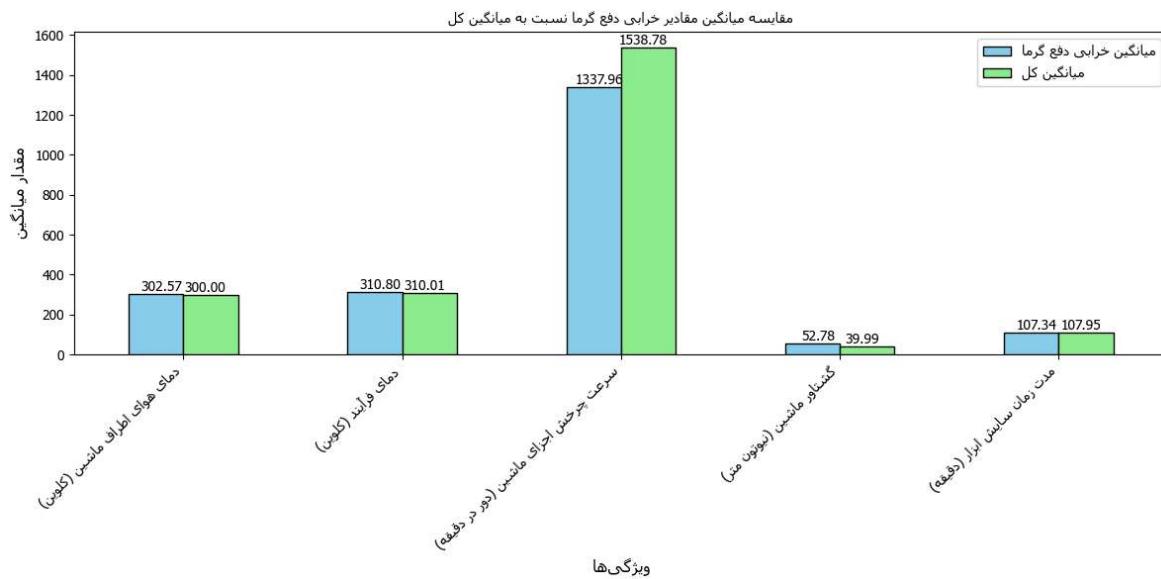
# اضافه کردن دیتالیل
for i in range(len(df_diff)):
    # فاصله بینتری برای دیتالیل‌ها در نظر می‌گیریم
    plt.text(i - 0.1, df_diff["0.1", f + [i]] - 0.1, df_diff["میانگین خرابی دفع گرما"], df_diff["میانگین خرابی دفع گرما"])
    plt.text(i + 0.1, df_diff["0.1", f + [i]] + 0.1, df_diff["میانگین کل"], df_diff["میانگین کل"])

# اصلاح ترتیب برچسب‌های محور افقی
ticks = range(len(df_diff))
label = df_diff.index, rotation=45, ha='right', fontproperties=prop)

# عنوان و برچسب‌ها
plt.title("مقایسه میانگین مقادیر خرابی دفع گرما نسبت به میانگین کل"), fontproperties=prop, reshaped_text=plt.title

```

```
(fontsize=12, fontproperties=prop) "ویژگی ها"), reshaped_text=plt.xlabel(fontsize=12, fontproperties=prop, "مقادیر میانگین"), reshaped_text=plt.ylabel(prop=c="میانگین خرابی دفع گرما", reshaped_text=plt.legend(c="میانگین کل", [prop]), c=prop)
# اصلاح ترتیب حروف در لجند
# چیدمان و نمایش نمودار
()plt.tight_layout()
()plt.show
```



یکی از تحلیل‌های مهم در بررسی‌های انجام شده، مقایسه میانگین سرعت چرخش اجزای ماشین در موقع خرابی دفع گرما با میانگین کل داده‌ها است. همان‌طور که از نمودار بالا پیداست، میانگین سرعت چرخش اجزای ماشین در موقع خرابی دفع گرما تفاوت فاحشی با میانگین کل دارد. در این تحلیل: در موقع خرابی دفع گرما، سرعت چرخش اجزای ماشین به‌طور مشخص پایین‌تر از میانگین کل است این تفاوت می‌تواند نشان‌دهنده یک الگوی خاص در رفتار ماشین در زمان خرابی دفع گرما باشد که نیازمند بررسی بیشتر است نمودار نشان می‌دهد که خرابی دفع گرما با سرعت پایین چرخش اجزای ماشین همراه است. این اطلاعات می‌تواند به ما کمک کند تا بهتر پیش‌بینی کنیم که در چه شرایطی خرابی دفع گرما بیشتر رخ می‌دهد و اقدامات پیشگیرانه مناسب را انجام دهیم. حالا به تحلیل نوسان سرعت چرخش اجزای ماشین در طول خرابی در اثر دفع گرما می‌پردازیم:

: In [19]

```

[1] "سرعت چرخش اجزای ماشین (دور در دقیقه)" rotation_speed = data
[2] "سرعت چرخش اجزای ماشین (دور در دقیقه)" heat_failure_speed = heat_failure_data

# رسم نمودار سرعت چرخش اجزای ماشین
plt.figure(figsize=(15, 6)) # افزایش عرض نمودار
("سرعت چرخش اجزای ماشین در طول بررسی‌ها") rotation_speed, label=reshaped_text) plt.plot

# کمترین و بیشترین مقدار سرعت چرخش در خرابی دفع گرما
()min_failure_speed = heat_failure_speed.min()
()max_failure_speed = heat_failure_speed.max

# رسم خطوط افقی برای کمترین و بیشترین مقدار سرعت چرخش در خرابی دفع گرما
# کد رنگ نارنجی line_color = "#FFA500
(rotation_speed) - 1, color=line_color, linestyle="--", linewidth=2, label=None)
(rotation_speed) - 1, color=line_color, linestyle="--", linewidth=2, label=None)

# اضافه کردن مقدار به خطوط
+, color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)
+, color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)

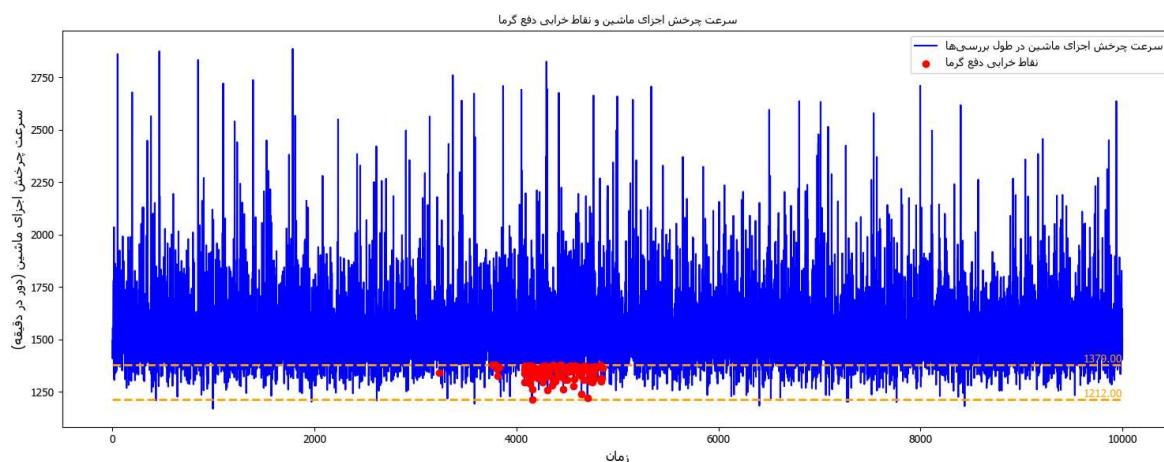
# مارک کردن نقاط خرابی دفع گرما
ure_data.index, heat_failure_speed, color='red', label=reshaped_text) plt.scatter

# اضافه کردن عنوان و برچسب‌ها
properties=prop) reshaped_text) plt.title("سرعت چرخش اجزای ماشین و نقاط خرابی دفع گرما", fontsize=12, fontproperties=prop)
(reshaped_text) plt.xlabel("زمان", fontproperties=prop)
("سرعت چرخش اجزای ماشین (دور در دقیقه)") reshaped_text) plt.ylabel("سرعت چرخش اجزای ماشین (دور در دقیقه)")

# اصلاح ترتیب حروف در لجند
plt.legend(prop=prop)

# نمایش نمودار
()plt.tight_layout
()plt.show

```



همانطور که مشاهده میکنید، نمودار نوسانات سرعت چرخش اجزای ماشین ظاهر زیبایی ندارد و الگوی خاصی را دنبال نمی کند ولی با مخصوص شدن نقاط مارک شده خرابی دفع گرما، میبینیم که خرابی های دفع گرما عدمتا مربوط به

## بازه 1212 تا 1379 برای سرعت چرخش اجزای ماشین میباشد

ولی برای اطمینان حاصل کردن از درستی دیتا و بررسی مجدد سرعت چرخش  
اجزای ماشین رو بصورت بازه های عددی در می آوریم و با یک نمودار خطی  
خرابی های دفع گرما را بررسی می کنیم

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper

# ایجاد بازه ها و بر جسبها
speed_bins = np.arange(1160, 2890, 100) # بازه های 100 واحدی
speed_labels = [f"{b}-{b + 100}" for b in speed_bins[:-1]] # ایجاد ستون بازه های سرعت چرخش

# سرعت بازه [ "سرعت بازه"]data=pd.cut(["سرعت بازه"], data, right=False)
# شمارش تعداد کل بررسی ها در هر بازه
value_counts(sort=False).speed_counts = data

# شمارش تعداد خرابی های دفع گرما در هر بازه
heat_failure_speed_counts = data["نوع خرابی"] == "خرابی دفع گرما"["سرعت بازه"]

# اعمال تنظیمات برای پشتیبانی از متن فارسی
plt.rcParams['font.family'] = 'Tahoma'

# رسم نمودار
fig, ax1 = plt.subplots(figsize=(16, 6))

# نمودار میله ای برای تعداد کل بررسی ها
counts, color="lightgreen", edgecolor="black", label=get_display)bars = ax1.bar(tedad_kl_bresiha_arabic_reshaper.reshape)get_display)ax1.set_ylabel("بازه های سرعت چرخش اجزای ماشین (دور در دقیقه)"[ "سرعت بازه"])
arabic_reshaper.reshape)get_display)ax1.set_xlabel("تعداد بررسی ها و خرابی های دفع گرما در بازه")
arabic_reshaper.reshape)get_display)ax1.set_title("نمودار میله ای برای تعداد کل بررسی ها")
ax1.tick_params(axis="y", labelcolor="green")

# تنظیم چرخش نوشتہ های محور افقی
plt.xticks(rotation=45, ha='right', fontsize=10)

# ایجاد محور دوم برای نمایش تعداد خرابی دفع گرما
()ax2 = ax1.twinx

counts_reindexed = heat_failure_speed_counts.reindex(speed_labels, fill_value=0)
counts_reindexed, color="orange", marker="o", label=get_display)line = ax2.plot("orange")arabic_reshaper.reshape)get_display)ax2.set_ylabel("تعداد خرابی دفع گرما")
ax2.tick_params(axis="y", labelcolor="orange")
ax2.set_ylim(0, max(heat_failure_speed_counts_reindexed) * 1.2)

# افزودن دیتالجند
lines_labels = [bars[0], line[0]]
```

```

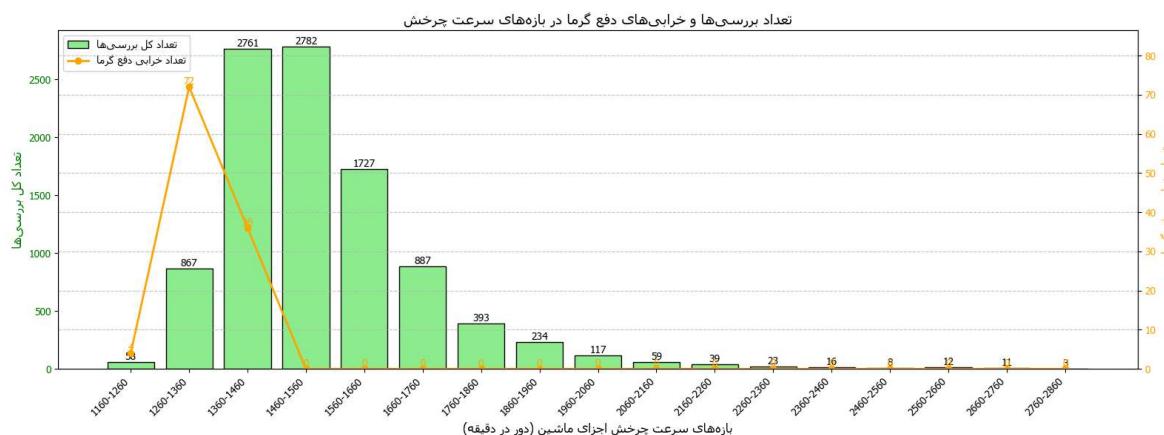
shape).get_display , (( "تعداد کل بررسی‌ها" ) arabic_reshaper.reshape ).get_display ] = labels
ax1.legend( lines_labels , labels , loc="upper left" , fontsize=10 )

# افزودن دیتالیل‌ها
:for bar in bars
    get_height() , f'{bar.get_height()}' , ha='center' , va='bottom' , fontsize=10)

        :for x , y in zip(speed_labels , heat_failure_speed_counts_reindexed)
            t(x , y , f'{int(y)}' , color="orange" , fontsize=10 , ha='center' , va='bottom')

# نمایش نمودار
plt.grid(axis="y" , linestyle="--" , alpha=0.7)
()plt.tight_layout()
()plt.show

```



با توجه به این نمودار متوجه میشویم که بازه عددی نمودار قبل درست بوده.  
پس نتیجه گیری از بررسی سرعت اجزای ماشین در خرابی های دفع گرما به این صورت است که اگر سرعت اجزای ماشین در بازه 1212 تا 1379 (دور در دقیقه) باشد، خطر خرابی دفع گرما مارا تهدید میکند

# بررسی گشتاور ماشین در خرابی های دفع گرما

اکنون میخواهیم ارتباط میان گشتاور ماشین و خرابی های دفع گرما را مورد بررسی قرار دهیم:

```

# استخراج داده‌های مربوط به گشتاور ماشین
] torque = data
[ "گشتاور ماشین (نیوتن متر)" ]
[ "گشتاور ماشین (نیوتون متر)" ]
heat_failure_torque = heat_failure_data

```

In [21]

```

# تنظیم نقاطی که خرابی دفع گرما دارند
plt.figure(figsize=(14, 6))

# رسم نمودار گشتاور ماشین
('color='lightgreen', torque, label=reshaped_text)plt.plot

# کمترین و بیشترین مقدار گشتاور در خرابی دفع گرما
()min_failure_torque = heat_failure_torque.min
()max_failure_torque = heat_failure_torque.max

# رسم خطوط افقی برای کمترین و بیشترین مقدار گشتاور در خرابی دفع گرما
# کد رنگ نارنجی "#FFA500
(max=len(torque) - 1, color=line_color, linestyle="--", linewidth=2, label=None)
(max=len(torque) - 1, color=line_color, linestyle="--", linewidth=2, label=None)

# اضافه کردن مقدار به خطوط
", color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)
", color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)

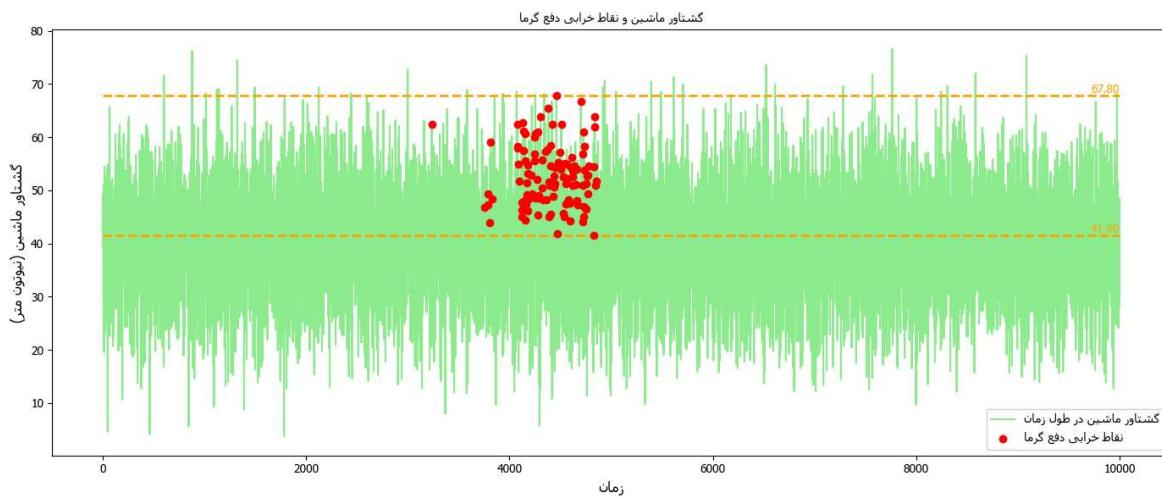
# مارک کردن نقاط خرابی دفع گرما
'e_data.index, heat_failure_torque, color='red', label=reshaped_text)plt.scatter

# اضافه کردن عنوان و برجسب‌ها
e=16, fontproperties=prop, reshaped_text)plt.title
(fontsize=12, fontproperties=prop, reshaped_text)plt.xlabel
ontsize=12, fontproperties=prop, reshaped_text)plt.ylabel

# اصلاح ترتیب حروف در لجند
plt.legend(prop=prop)

# تنظیم نمایش و نمایش نمودار
()plt.tight_layout
()plt.show

```



همانطور که میبینید نمودار گشتاور ماشین در طی بررسی ها مانند نمودار سرعت چرخش اجزای ماشین دارای ظاهر جالبی نیست و الگویابی را کمی سخت میکند ولی همانطور که میبینید خرابی های دفع ماشین در یک بازه

# مشخص نوسان کرده اند ولی مانند بررسی قبلی دوباره برای بررسی دقیق تر به سراج نمودار میله‌ای میرویم

```
:In [22]
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper

# ایجاد بازه‌ها و برجسب‌ها
bins = np.arange(0, 81, 10)
labels = [f"{b}-{b + 10}" for b in bins[:-1]]

# ایجاد ستون گشتاور بازه‌ها
data["گشتاور بازه"] = pd.cut(data["نیوتون متر"], bins=bins, labels=labels, right=False)

# شمارش تعداد کد بررسی در هر بازه
torque_counts = data["گشتاور بازه"].value_counts(sort=False)

# شمارش تعداد خرابی‌های دفع گرما در هر بازه
heat_failure_counts = data["نوع خرابی"].value_counts(sort=False)

# اعمال تنظیمات برای پیش‌بینی از متن فارسی
plt.rcParams['font.family'] = 'Tahoma'

# رسم نمودار
fig, ax1 = plt.subplots(figsize=(12, 6))

# نمودار میله‌ای برای تعداد کل بررسی‌ها
bars = ax1.bar(arabic_reshaper.reshape(get_display('تعداد کل بررسی‌ها')), color="blue", edgecolor="black", label="تعداد کل بررسی‌ها")
ax1.set_ylabel("بازه‌های گشتاور ماشین (نیوتون متر)")
ax1.set_xlabel("تعداد بررسی‌ها و خرابی‌های دفع گرما در بازه")
ax1.set_title("نمودار میله‌ای برای تعداد کل بررسی‌ها")
ax1.tick_params(axis="y", labelcolor="blue")

# افزودن دیتالیل به میله‌ها
for bar in bars:
    height = bar.get_height()
    ax1.annotate(f'{int(height)}', xy=(bar.get_x() + bar.get_width() / 2, height), xytext=(0, 5), textcoords="offset points", ha='center', va='bottom', fontsize=10, color="black")

# ایجاد محور دوم برای نمایش تعداد خرابی دفع گرما
ax2 = ax1.twinx()

heat_failure_counts_reindexed = heat_failure_counts.reindex(labels, fill_value=0)
line = ax2.plot(arabic_reshaper.reshape(get_display('تعداد خرابی دفع گرما')), color="orange", marker="o", label="تعداد خرابی دفع گرما")
ax2.set_ylabel("تعداد خرابی دفع گرما")
ax2.set_xlabel("نیوتون متر")
ax2.set_title("نمودار میله‌ای برای تعداد کل بررسی‌ها")
ax2.set_yticks(arabic_reshaper.reshape(get_display('تعداد خرابی دفع گرما')))

# تنظیم بازه مناسب محو
ax2.set_ylim(0, max(heat_failure_counts_reindexed) * 1.2)

# افزودن دیتالیل به نقاط خط
for i, count in enumerate(heat_failure_counts_reindexed):
    ax2.plot(arabic_reshaper.reshape(get_display('تعداد خرابی دفع گرما'))[i], count, color="orange", marker="o")
```

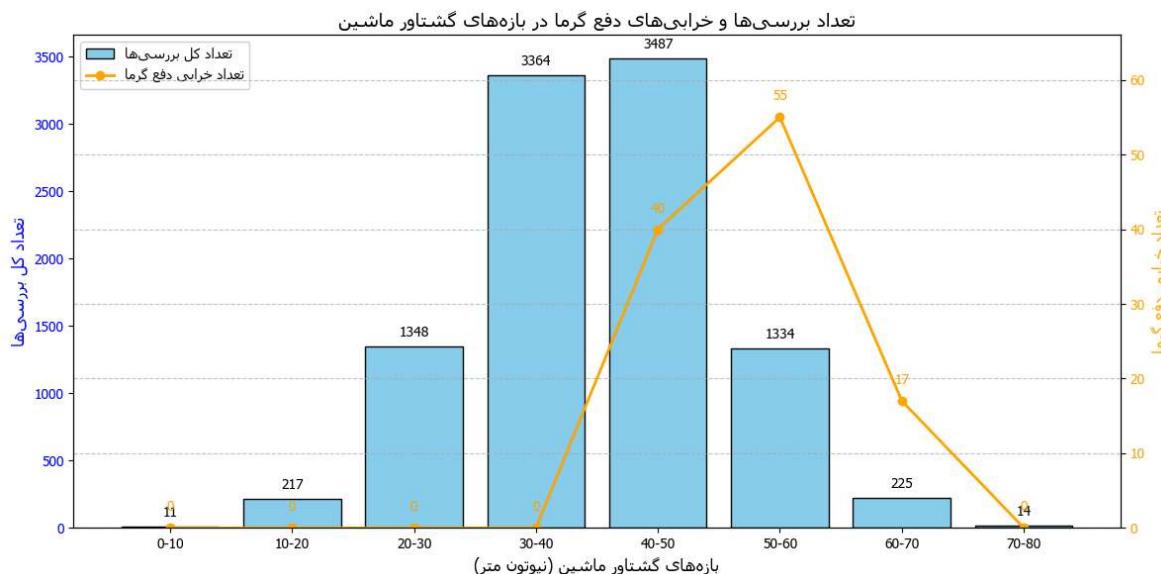
```

        , 'ax2.annotate(f'{int(count)}'
        ,xy=(i, count)
        ,xytext=(0, 10)
        ,textcoords="offset points"
        ("ha='center', va='bottom', fontsize=10, color='orange'

# افزودن دیتالجند
lines_labels = [bars[0], line[0]]
arabic_reshaper.get_display(arabic_reshaper.reshape(arabic_reshaper.get_display) = labels
ax1.legend(lines_labels, labels, loc="upper left", fontsize=10)

# نمایش نمودار
plt.grid(axis="y", linestyle="--", alpha=0.7)
()plt.tight_layout()
()plt.show

```



در این نمودار متوجه میشویم که بیشترین میزان خرابی دفع گرما مربوط به بازه 40 تا 70 نیوتون متر برای گشتاور ماشین است. پس میتوانیم از سیستم مانیتورینگ دستگاه استفاده کنیم و با قرار گرفتن گشتاور ماشین در این بازه اقدامات لازم برای پیشگیری از خرابی دفع گرمای دستگاه را انجام دهیم.

## تحلیل دمای فرآیند و نقاط خرابی دفع گرما

در این تحلیل، هدف بررسی دمای فرآیند دستگاه در طول تمامی بررسی‌ها است. برای این منظور، ابتدا داده‌های مربوط به دمای فرآیند از کل داده‌ها استخراج شده و نقاطی که مربوط به خرابی دفع گرما هستند، مشخص شده‌اند.

سپس، نموداری از دمای فرآیند رسم شده است که تمام مقادیر دمای فرآیند را در طول زمان نمایش می‌دهد. همچنین، نقاطی که در آن‌ها خرابی دفع گرما رخداده است، به صورت خاص با رنگ قرمز مشخص شده‌اند. این نمایش بصری می‌تواند به شناسایی روندها و الگوهای خاص در دمای فرآیند و ارتباط آن با خرابی‌ها کمک کند.

این نوع تحلیل می‌تواند در پیش‌بینی خرابی‌ها و برنامه‌ریزی برای جلوگیری از آن‌ها مفید باشد، به‌ویژه در شرایطی که دمای فرآیند نقش مهمی در عملکرد دستگاه ایفا می‌کند.

:In [23]

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper
from matplotlib import font_manager

# تنظیم فونت فارسی
'font_path = 'C:/Windows/Fonts/Tahoma.ttf' # مسیر فونت فارسی
prop = font_manager.FontProperties(fname=font_path)

# اصلاح متن فارسی
:(def reshaped_text(text)
reshaped = arabic_reshaper.reshape(text)
return get_display(reshaped)

# فیلتر کردن سطرهای مربوط به خرابی دفع گرما
[("نوع خرابی") == "خرابی دفع گرما"]data]heat_failure_data = data

# استخراج دمای فرآیند از کل داده‌ها و خرابی دفع گرما
["دما فرآیند (کلوین)"]process_temp = data
["دما فرآیند (کلوین)"]heat_failure_temp = heat_failure_data

# تنظیم نقاطی که خرابی دفع گرما دارد (در نمودار مارک می‌شود)
heat_failure_points = heat_failure_data.index

# رسم نمودار دمای فرآیند
plt.figure(figsize=(12, 6))
process_temp, label=reshaped_text)plt.plot
color='red', label=reshaped_text)plt.scatter
heat_failure_temp, color='red', label=reshaped_text)plt.scatter
failure_points, heat_failure_temp, color='red', label=reshaped_text)plt.scatter

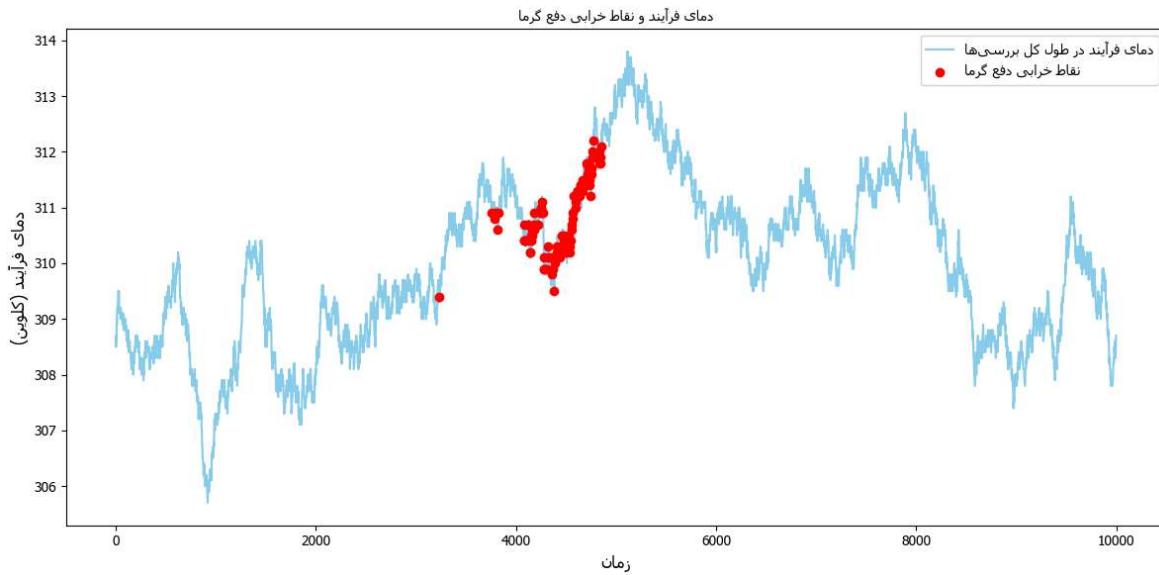
# اضافه کردن عنوان و برچسب‌ها
size=16, fontproperties=prop)("دما فرآیند و نقاط خرابی دفع گرما", , reshaped_text)plt.title
```

```

    (fontsize=12, fontproperties=prop , ("زمان")reshaped_text)plt.xlabel
    (fontsize=12, fontproperties=prop , ("دما فرآیند (کلین)") , reshaped_text)plt.ylabel
# اصلاح ترتیب حروف در لجند
plt.legend(prop=prop)

# نمایش نمودار
()plt.tight_layout
()plt.show

```



یکی از پارامترهای مهم در تحلیل داده‌های دستگاه، «اختلاف دما» است که از تفاوت بین دمای هوا اطراف ماشین (دمای محیط) و دمای فرآیند محاسبه می‌شود. این پارامتر می‌تواند نشان‌دهنده میزان اتلاف انرژی و یا توانایی دستگاه در دفع گرما باشد. اختلاف دما در شرایط مختلف می‌تواند به شناسایی مشکلات دستگاه کمک کند، به‌ویژه زمانی که این اختلاف در نقاط خرابی دفع گرما برجسته‌تر باشد.

در این تحلیل، ابتدا اختلاف دما برای تمامی بررسی‌ها محاسبه شده و به داده‌ها اضافه شده است. سپس، اختلاف دما در نقاطی که خرابی دفع گرما رخ داده است، به‌طور جداگانه استخراج و برای بررسی دقیق‌تر آماده شده است.

نمودار رسم شده در این بخش، اختلاف دما را در طول تمامی بررسی‌ها نمایش می‌دهد. همچنین نقاطی که در آن‌ها خرابی دفع گرما رخ داده است، به‌طور خاص با رنگ قرمز مشخص شده‌اند. این نمایش به ما کمک می‌کند تا الگوهای و

رفتارهای خاص در اختلاف دما را در زمان خرابی‌های دفع گرما شناسایی کنیم.

تحلیل این پارامتر می‌تواند در درک بهتر عملکرد دستگاه و پیش‌بینی مشکلات احتمالی در آینده مفید باشد:

```
:In [24]
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper
from matplotlib import font_manager
import warnings

# خاموش کردن هشدارها
warnings.filterwarnings('ignore')

# تنظیم فونت فارسی
'font_path = 'C:/Windows/Fonts/Tahoma.ttf'
prop = font_manager.FontProperties(fname=font_path)

# اصلاح متن فارسی
def reshaped_text(text):
    reshaped = arabic_reshaper.reshape(text)
    return get_display(reshaped)

# فیلتر کردن سطرهای مربوط به خرابی دفع گرما
[("نوع خرابی"] == "خرابی دفع گرما")
data]heat_failure_data = data

# محاسبه اختلاف دما بین دمای محیط و دمای فرآیند در کل داده‌ها
["اختلاف دما"]data = ["دما"]data - ["دما فرآیند (کلوین)"]
data

# حالا این ستون را به داده‌های فیلتر شده (خرابی دفع گرما) اضافه می‌کنیم
heat_failure_data = ["اختلاف دما"]heat_failure_data - ["دما هواي اطراف ماشين (کلوين)"]
heat_failure_data

# استخراج اختلاف دما و نقاط خرابی دفع گرما
temperature_difference = data["اختلاف دما"]
heat_failure_diff = heat_failure_data["اختلاف دما"]

# پایین‌ترین و بالاترین مقدار خرابی دفع گرما
()min_temp_diff = heat_failure_diff.min()
()max_temp_diff = heat_failure_diff.max()

# رسم نمودار اختلاف دما
plt.figure(figsize=(12, 6))
temperature_difference, label=reshaped_text)plt.plot

# مارک کردن نقاطی که خرابی دفع گرما دارد
lure_data.index, heat_failure_diff, color='red', label=reshaped_text)plt.scatter

# رسم خطوط افقی به رنگ نارنجی
# کد رنگ نارنجی "#FFA500
index.min(), xmax=data.index.max(), color=line_color, linestyle="--", linewidth=2)
```

```

index.min(), xmax=data.index.max(), color=line_color, linestyle="--", linewidth=2)

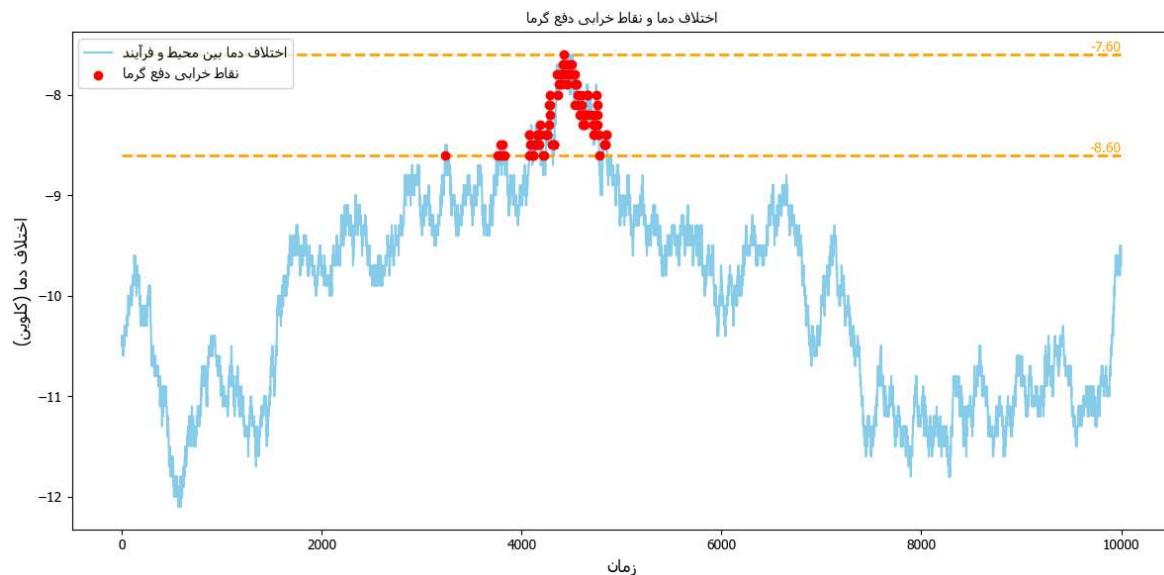
# اضافه کردن مقدار به خطوط
+, color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)
+, color=line_color, fontsize=10, ha="right", va="bottom", fontproperties=prop)

# اضافه کردن عنوان و برجسبها
size=16, fontproperties=prop, "اختلاف دما و نقاط خرابی دفع گرما"), reshaped_text)plt.title
(fontsize=12, fontproperties=prop, "زمان"), reshaped_text)plt.xlabel
(fontsize=12, fontproperties=prop, "اختلاف دما (کلوین)'), reshaped_text)plt.ylabel

# اصلاح ترتیب حروف در لجند
plt.legend(prop=prop)

# نمایش نمودار
()plt.tight_layout
()plt.show

```



در این نمودار متوجه میشویم که تمامی خرابی های دفع گرما در بازه عددی 7.6 تا 8.6 برای اختلاف دمای محیط و فرآیند به وجود می آید. پس میتوانیم برای سیستم کنترل وضعیت دستگاه کمپرسور یک پارامتر تحت عنوان «اختلاف دما» تعریف کنیم و در این بازه کنترل های مورد نیاز برای جلوگیری از خرابی دفع گرما انجام شود.

## مدت زمان سایش ابزار

تنها پارامتر باقیمانده در این نوع خرابی «مدت زمان سایش ابزار» می باشد که در نمودار زیر به بررسی آن میپردازیم:

: In [25]

```

import matplotlib.pyplot as plt
from bidi.algorithm import get_display
import arabic_reshaper

# اصلاح متن فارسی
def reshaped_text(text):
    reshaped = arabic_reshaper.reshape(text)
    return get_display(reshaped)

# استخراج داده‌های مربوط به مدت زمان سایش ابزار
[ "مدت زمان سایش ابزار (دقیقه)" ]wear_duration = data

# داده‌های مربوط به خرابی دفع گرما
[ "نوع خرابی" == "خرابی دفع گرما" ]heat_failure_data = data
[ "مدت زمان سایش ابزار (دقیقه)" ]heat_failure_wear_duration = heat_failure_data

# تنظیم نقاطی که خرابی دفع گرما دارند
plt.figure(figsize=(15, 6))

# رسم نمودار مدت زمان سایش ابزار
plt.plot(wear_duration, label=reshaped_text("مدت زمان سایش ابزار در طول بررسی‌ها"))

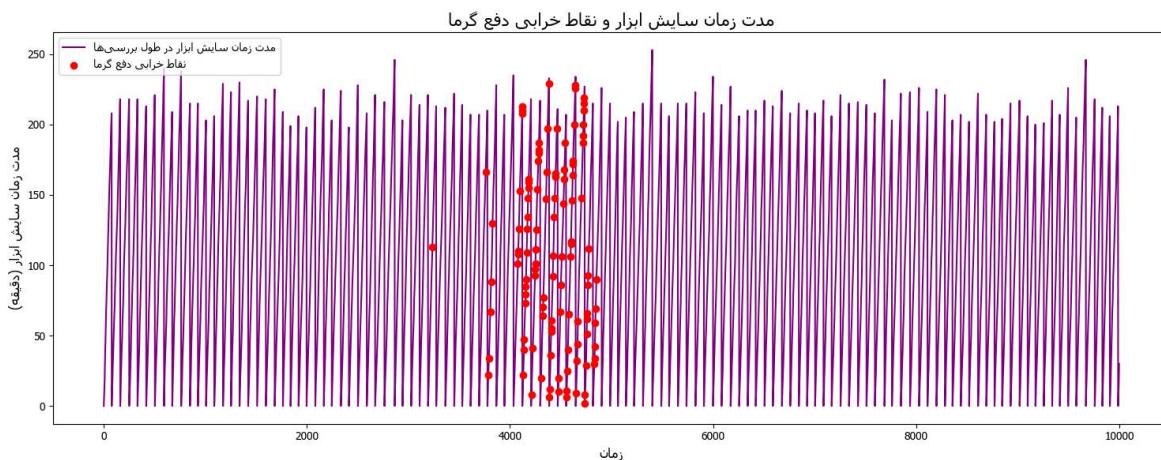
# مارک کردن نقاطی که خرابی دفع گرما دارند
plt.scatter(index, heat_failure_wear_duration, color='red', label=reshaped_text("مارک کردن نقاطی که خرابی دفع گرما دارند"))

# اضافه کردن عنوان و برچسب‌ها
plt.title(reshaped_text("مدت زمان سایش ابزار و نقاط خرابی دفع گرما"), fontsize=16)
plt.xlabel(reshaped_text("زمان"), fontsize=12)
plt.ylabel(reshaped_text("مدت زمان سایش ابزار (دقیقه)"), fontsize=12)

# نمایش لگند
plt.legend()

# تنظیم نمایش و نمایش نمودار
plt.tight_layout()
plt.show()

```



نمودار بررسی این پارامتر نشان میدهد که خرابی دفع گرما ارتباط خاصی با مدت زمان سایش ابزار ندارد و در تمامی بازه های عددی مدت زمان سایش ابزار شاهد

خرابی دفع گرما بوده ایم

# بررسی کلی خرابی دفع گرما

اکنون که تمای پارامتر های دستگاه را بررسی کردیم و متوجه بازه های عددی مختلف برای پارامتر ها شدیم یک نگاه اجمالی به الگوهای یافته شده می اندازیم:

```

import pandas as pd
:In [26]

# فیلتر کردن داده های مربوط به خرابی دفع گرما
data[heat_failure_data == "خرابی دفع گرما"]

# پارامترهای مورد بررسی
parameters = [
    "دماهی هوای اطراف ماشین (کلوین)",
    "دماهی فرآیند (کلوین)",
    "سرعت چرخش اجزای ماشین (دور در دقیقه)",
    "گشتاور ماشین (نیوتون متر)",
    "مدت زمان سایش ابزار (دقیقه)"
]

# محاسبه بازه ها برای هر پارامتر
ranges = {
    parameter: {
        "min": heat_failure_data[parameter].min(),
        "max": heat_failure_data[parameter].max()
    }
    for parameter in parameters
}

# تبدیل بازه ها DataFrame
ranges_df = pd.DataFrame(
    list(ranges.items()),
    columns=["پارامتر", "بازه عددی"]
)

# نمایش جدول
ranges_df

```

پارامتر	بازه عددی	:Out[26]
0	دماهی هوای اطراف ماشین (کلوین)	303.7 - 300.8
1	دماهی فرآیند (کلوین)	312.2 - 309.4
2	سرعت چرخش اجزای ماشین (دور در دقیقه)	1379 - 1212
3	گشتاور ماشین (نیوتون متر)	67.8 - 41.6
4	مدت زمان سایش ابزار (دقیقه)	229 - 2

# بررسی سایر خرابی ها

برای بررسی سایر خرابی ها تک تک پارامتر هارا بررسی نمیکنیم بلکه مانند آخرین جدول خرابی دفع گرما بازه های عددی هر پارامتر را در یک جدول قرار میدهیم و در انتهای تمامی بازه ها را بررسی میکنیم.

```
import pandas as pd :In [27]

# محاسبه بازه ها برای هر پارامتر
def calculate_range(data, column_name)
    "return f'{data[column_name].min()} - {data[column_name].max()}'"

# گروه بندی داده ها بر اساس نوع خرابی
grouped_data = data.groupby('نوع خرابی')

# ایجاد یک لیست برای ذخیره بازه ها
[] = rows

# محاسبه بازه ها برای هر خرابی
for name, group in grouped_data
    } = row
    "نوع خرابی": name,
    "مدت زمان سایش ابزار": group.calculate_range("مدت زمان سایش ابزار (دقیقه)", ),
    "گشتاور ماشین": group.calculate_range("گشتاور ماشین (نیوتن متر)", ),
    "سرعت چرخش اجزای ماشین": group.calculate_range("سرعت چرخش اجزای ماشین (دور دقیقه)", ),
    "دما های هوای اطراف ماشین": group.calculate_range("دما های هوای اطراف ماشین (کلوین)", ),
    "دما فرآیند": group.calculate_range("دما فرآیند (کلوین)", )

    {
        rows.append(row)

# تبدیل لیست به DataFrame
range_table = pd.DataFrame(rows)

# نمایش جدول بازه ها
range_table
```

نوع خرابی	سایش ابزار	مدت زمان	گشتوار ماشین	سرعت چرخش اجزای ماشین	دماهی اطراف ماشین	دماهی فرآیند
بدون خرابی 0	246 - 0	- 12.6 70.0	2695 - 1168	304.5 - 295.3	- 305.7 313.8	
خرابی ابزار 1	253 - 198	- 16.2 62.4	2271 - 1323	304.4 - 296.9	- 307.4 313.7	
خرابی توان 2	234 - 0	76.6 - 3.8	2886 - 1200	304.0 - 295.7	- 306.2 313.2	
خرابی دفع گرما 3	229 - 2	- 41.6 67.8	1379 - 1212	303.7 - 300.8	- 309.4 312.2	
خرابی ناشی از فشار بیش از حد 4	251 - 177	- 46.3 68.2	1515 - 1181	304.0 - 295.6	- 306.1 313.1	
خرابی های تصادفی 5	215 - 2	- 27.7 61.2	1687 - 1306	302.9 - 297.0	- 307.7 312.5	

برای نمایش بازه های عددی هر پارامتر برای هر نوع خرابی به گونه ای که مقایسه آنها ساده و کارآمد باشد، بهترین گزینه نمودار Boxplot (نمودار جعبه ای) است. این نمودار به طور خلاصه اطلاعات آماری مانند کمینه، بیشینه، میانه، و چارک ها را برای هر دسته داده (در اینجا انواع خرابی) نمایش می دهد و بسیار مناسب برای مقایسه بازه های داده ای بین گروه ها است.

دلیل انتخاب Boxplot:

مقایسه های بصری ساده:

می توانید تمام انواع خرابی را در کنار هم نمایش دهید و بازه های عددی هر پارامتر را مقایسه کنید.

اطلاعات بیشتر در مقایسه با جدول:

علاوه بر بازه (کمینه و بیشینه)، این نمودار میانه و چارک ها را نیز نشان می دهد.

نمایش داده های پرت:

نقاط پرت که ممکن است روی تحلیل تأثیر بگذارند بهوضوح در این نمودار

مشخص می‌شوند. این نقاط به عنوان "outliers" در نمودار جعبه‌ای به صورت نقاط جداگانه (به طور معمول با علامت ستاره یا دایره) نمایش داده می‌شوند.

در نتیجه، نمودار جعبه‌ای یک ابزار بصری مفید برای مقایسه و تحلیل داده‌ها است که علاوه بر نمایش بازه‌ها، میانه‌ها، و چارک‌ها، امکان شناسایی داده‌های پرت را نیز فراهم می‌کند.

## آشنایی با اجزای نمودار جعبه‌ای

نمودار جعبه‌ای (Boxplot) یک ابزار گرافیکی است که برای نمایش توزیع داده‌ها و ویژگی‌های آماری آن‌ها استفاده می‌شود. در این نمودار، اطلاعات مهمی مانند میانه، چارک‌ها، و مقادیر پرت برای هر دسته داده به نمایش درمی‌آید.

اجزای اصلی نمودار جعبه‌ای عبارتند از:

### 1. جعبه (Box):

جعبه اصلی نمودار که از دو خط افقی به نام "چارک اول" (Q1) و "چارک سوم" (Q3) تشکیل شده است. جعبه نمایانگر 50% مرکزی داده‌ها است که بین این دو چارک قرار دارند.

### 2. خط میانه (Median):

خطی که به داخل جعبه رسم می‌شود و میانه داده‌ها (Q2) را نشان می‌دهد. این خط نمایانگر نقطه‌ای است که داده‌ها را به دو بخش برابر تقسیم می‌کند.

### 3. بازه بین چارک‌ها (Interquartile Range - IQR):

فاصله بین چارک اول (Q1) و چارک سوم (Q3) که نشان‌دهنده پخش شدن داده‌ها است. این بازه 50 درصد از داده‌ها را در بر می‌گیرد.

#### : (Whiskers) 4. شاخها

خطوط افقی که از دو طرف جعبه خارج می‌شوند و نشان‌دهنده پراکندگی داده‌ها خارج از چارک‌ها هستند. طول این شاخها معمولاً تا 1.5 برابر فاصله IQR از Q1 و Q3 است.

#### : (Outliers) 5. داده‌های پرت

نقاطی که به‌طور واضح از سایر داده‌ها فاصله دارند و خارج از محدوده شاخها قرار می‌گیرند. این نقاط معمولاً به صورت دایره یا ستاره در نمودار نشان داده می‌شوند.

### چرا استفاده از نمودار جعبه‌ای مفید است؟

- **تحلیل توزیع داده‌ها:** این نمودار به‌طور سریع و واضح نشان می‌دهد که داده‌ها چگونه توزیع شده‌اند و آیا تقارن دارند یا خیر.

- **تشخیص داده‌های پرت:** می‌توانید نقاط غیرعادی یا پرت را شناسایی کنید که ممکن است نیاز به بررسی بیشتر داشته باشند.

- **مقایسه گروه‌ها:** اگر چندین مجموعه داده داشته باشد، نمودار جعبه‌ای به شما این امکان را می‌دهد که توزیع و ویژگی‌های مختلف هر گروه را با هم مقایسه کنید.

```
import seaborn as sns
import matplotlib.pyplot as plt
from arabic_reshaper import reshape
from bidi.algorithm import get_display

# تابع برای اصلاح ترتیب حروف
def reshaped_text(text)
    reshaped = reshape(text)
    return get_display(reshaped)

# پارامترهای مورد بررسی
] = parameters
```

: In [28]

"دماي هواي اطراف ماشين (کلوين)" ,  
 "دماي فرآيند (کلوين)" ,  
 "سرعت چرخش اجزاي ماشين (دور در دقيقه)" ,  
 "گشتاور ماشين (نيتون متر)" ,  
 "مدت زمان سايش ابزار (دقيقه)"

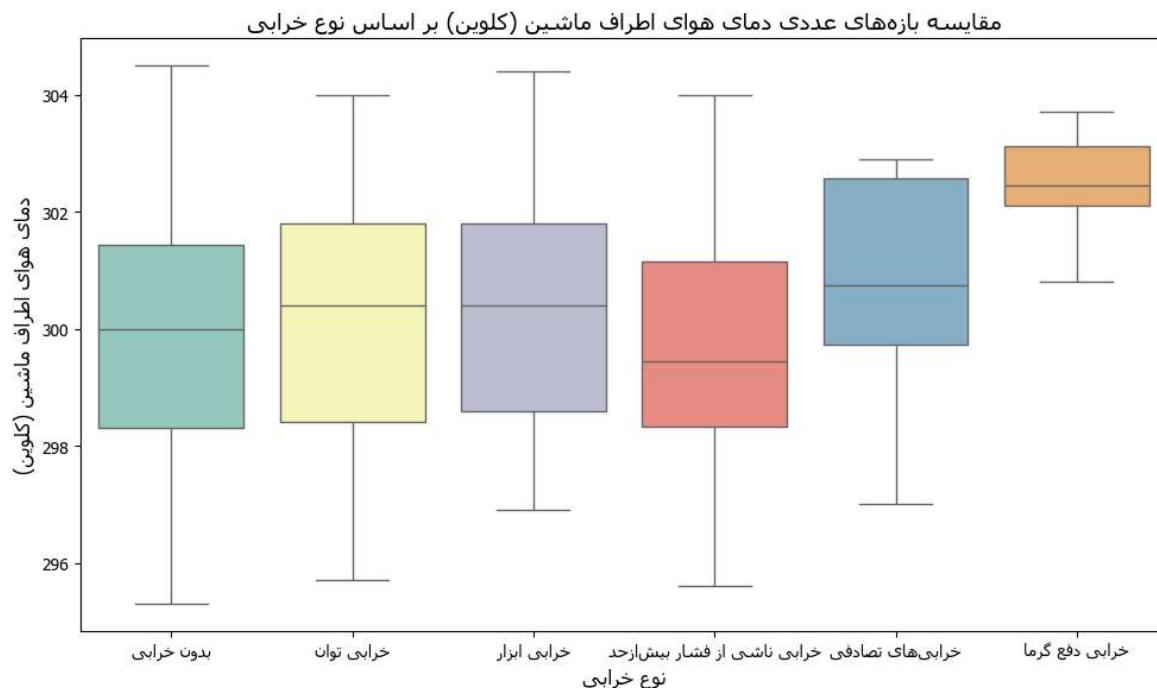
[

```
# رسم نمودار جعبه‌ای برای هر پارامتر
for parameter in parameters
    plt.figure(figsize=(10, 6))
    ("y=data[parameter], palette="Set3", ["نوع خرابی"]x=data)sns.boxplot

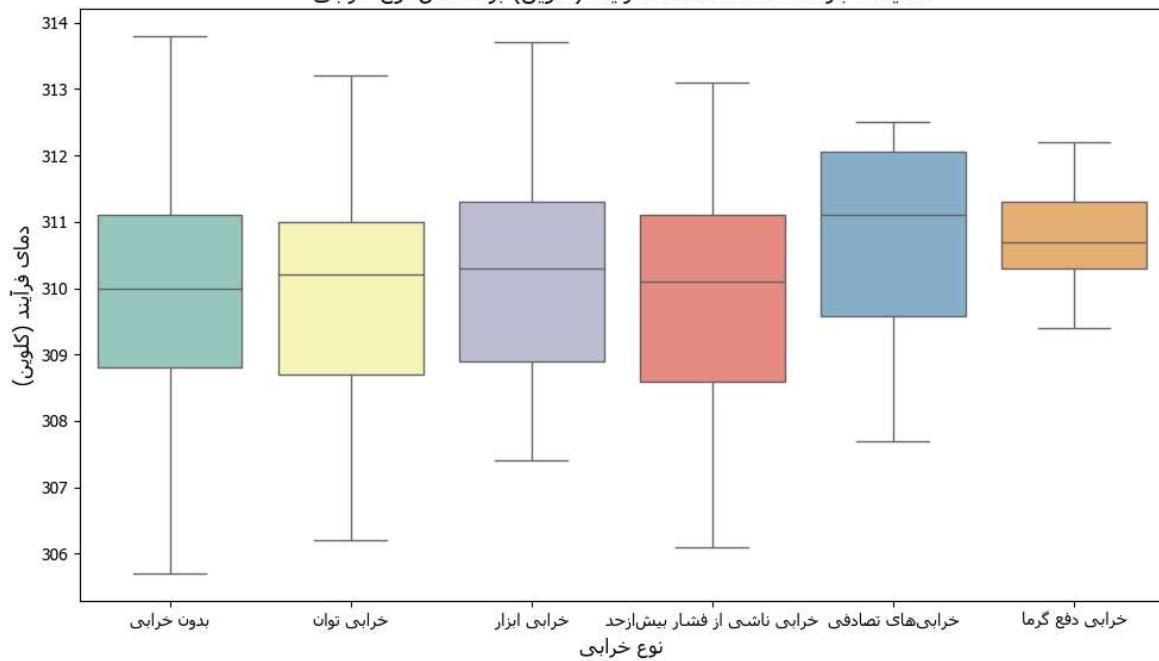
# اصلاح عنوان و برجسبها
ze=14 مقایسه بازه‌های عددی {parameter} بر اساس نوع خرابی() ،
f(reshaped_text)plt.title
(fontsize=12)("نوع خرابی")reshaped_textplt.xlabel
plt.ylabel(reshaped_text(parameter), fontsize=12)

# اصلاح برجسبهای محور افقی
()xticks = plt.gca().get_xticks
[()unique.["نوع خرابی"]reshaped_text(label) for label in data] = xticklabels
plt.gca().set_xticks(xticks)
plt.gca().set_xticklabels(xticklabels, fontproperties=prop)

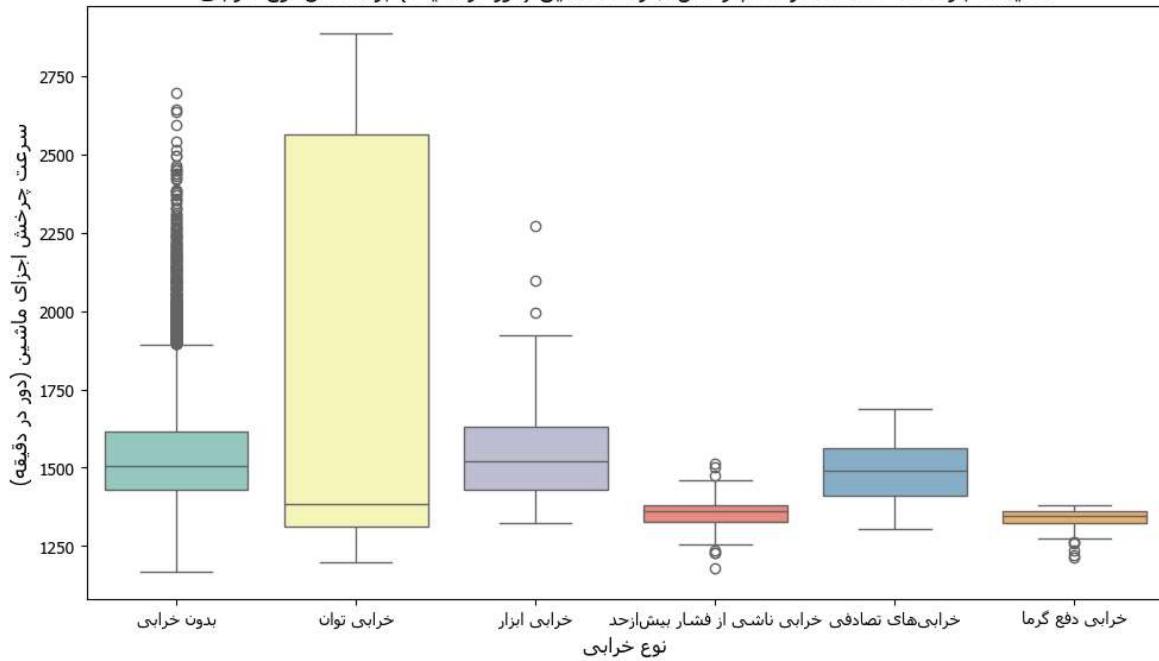
()plt.tight_layout
()plt.show
```



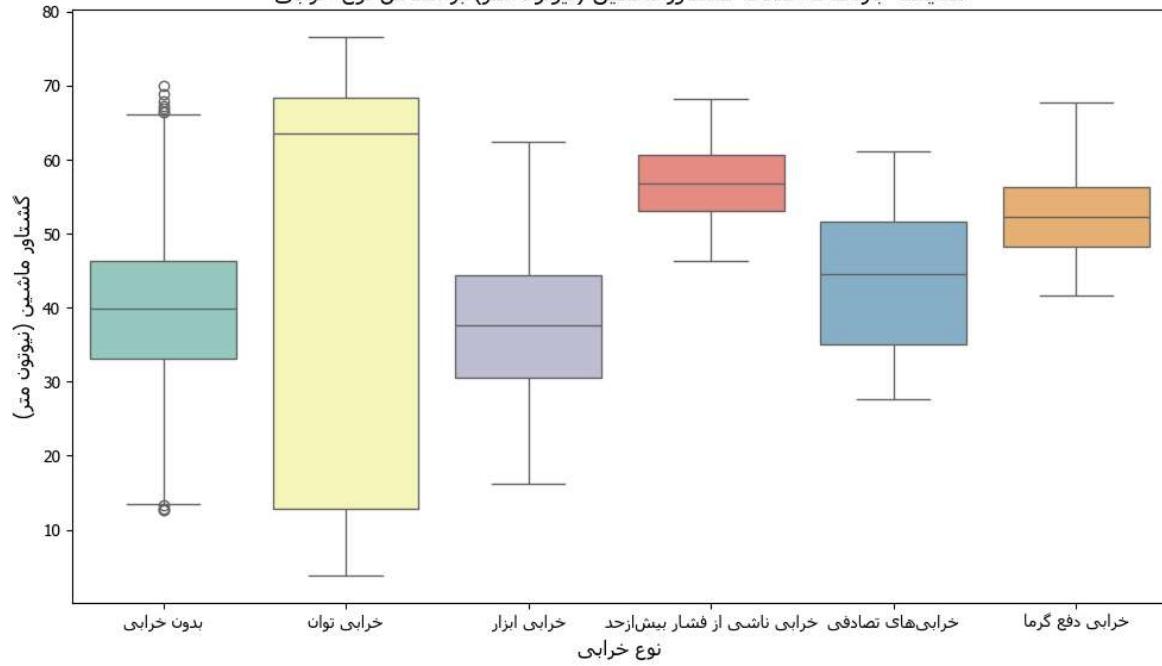
مقایسه بازه‌های عددی دمای فرآیند (کلوین) بر اساس نوع خرابی



مقایسه بازه‌های عددی سرعت چرخش اجزای ماشین (دور در دقیقه) بر اساس نوع خرابی



مقایسه بازه‌های عددی گشتاور ماشین (نیوتن متر) بر اساس نوع خرابی



مقایسه بازه‌های عددی مدت زمان سایش ابزار (دقیقه) بر اساس نوع خرابی

