

SOURCE CODE:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
import warnings
from sklearn import metrics
from IPython.display import display
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import sys

# To ignore warning messages
warnings.filterwarnings('ignore')

# Check if running in an IPython environment
if 'IPython' in sys.modules:
    from IPython import get_ipython
    get_ipython().run_line_magic('config', 'Completer.use_jedi = False')

# Output settings for tables
np.set_printoptions(threshold=None, precision=2)

dataset = pd.read_csv('predictive_maintenance.csv')
print('Dataset dimensions:', dataset.shape, '\n')
print(dataset.head())

# Statistical summary
print(dataset.describe())

null_df = dataset.isnull().sum() * 100 / len(dataset)
```

```

info_df = pd.DataFrame({
    'missing_count': dataset.isnull().sum(),
    'percent_missing': null_df,
    'dtype': dataset.dtypes
}).reset_index().rename(columns={'index': 'column'})
print(info_df)

```

```

qtd_product_type = dataset[['Product ID', 'Type']].groupby('Type').count().reset_index()
sns.set(rc={'figure.figsize': (5, 6)})
colors = sns.color_palette('pastel')
plt.pie(x=qtd_product_type['Product ID'],
        labels=qtd_product_type['Type'], colors=colors, autopct='%0.1f%%')
plt.title("Vehicle Type Percentage")
plt.show()

```

```

print('Analyzing the vehicle failures:')
print(50 * '-')
print("\nNumber of failed and non-failed vehicle engines:\n0 - No Failure\n1 - Failure")

```

```

target = dataset[['Product ID', 'Target']].groupby('Target',
as_index=False).count().rename(columns={'Product ID': 'Count'})
display(target)
print(50 * '-')
print("\nFrequency of failure type:")

```

```

type_failure = dataset[['Failure Type', 'Target']].groupby('Failure Type',
as_index=False).count().sort_values('Target',
ascending=False).reset_index().rename(columns={'Target': 'Count'})
display(type_failure)
print(50 * '-')

```

```

qtd_failure = dataset[['Target', 'Failure Type']].loc[(dataset['Failure Type'] != 'No
Failure')].groupby('Failure Type').count().reset_index()
sns.set(rc={'figure.figsize': (5, 6)})

```

```

colors = sns.color_palette('pastel')

plt.pie(x=qtd_failure['Target'],
        labels=qtd_failure['Failure Type'], colors=colors, autopct='%1f%%')

plt.title("Vehicle Failure Type Percentage")

plt.show()


print('\nClassification of failed vehicles but no failure type:')

countnofailure = dataset.query("Target == 1 and Failure Type == 'No Failure'")[['Failure Type',
'Target']]

display(countnofailure)

print(30 * '-')

print('\nVehicle classification without failure but with failure type:')

countfailure = dataset.query("Target == 0 and Failure Type != 'No Failure'")[['Failure Type', 'Target']]

display(countfailure)


columns_number = dataset.drop(columns=['Target', 'UDI']).select_dtypes(exclude=['object']).columns

plt.figure(figsize=(25, 50))

sns.set_style("white")

col_count = 1

for col in columns_number[:10]:

    plt.subplot(8, 2, col_count)

    sns.histplot(x=dataset[col], kde=False, bins=10, color='#a1c9f4')

    plt.title(f'Histogram {col}')

    plt.xlabel("")

    plt.ylabel("")

    col_count += 1

    sns.despine()


columns_number = dataset.drop(columns=['Target', 'UDI']).select_dtypes(exclude=['object']).columns

plt.figure(figsize=(25, 50))

sns.set_style("white")

col_count = 1

for col in columns_number[:10]:

```

```
plt.subplot(8, 2, col_count)
sns.boxplot(x=dataset[col], color='#a1c9f4')
plt.title(f'BoxPlot {col}')
plt.xlabel("")
plt.ylabel("")
col_count += 1
sns.despine()
```

```
plt.show()
```