

Name: Asraa Al-Wattar

Course: Practical Machine Learning

Assignment: Handle missing values, outliers, and data visualization

```
from google.colab import files
```

```
import pandas as pd
```

```
uploaded = files.upload()
```

```
filename = list(uploaded.keys())[0]
```

```
print("تم رفع الملف:", filename)
```

```
df = pd.read_csv(filename)
```

```
print("Shape:", df.shape)
```

```
print(df.head())
```

 **kidney_disease.csv.csv**(text/csv) - 41503 bytes, last modified: 11/13/2025 - 100% done

Saving kidney_disease.csv.csv to kidney_disease.csv (1).csv

تم رفع الملف: kidney_disease.csv (1).csv

Shape: (398, 25)

	age	bp	sg	al	su	rbc	pc	pcc	ba \
0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent
1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent
2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent

```
3 48.0 70.0 1.005 4.0 0.0 normal abnormal present notpresent
4 51.0 80.0 1.010 2.0 0.0 normal normal notpresent notpresent
```

```
    bgr ... pcv  wc  rc htn  dm cad appet  pe  ane classification
0 121.0 ... 44 7800 5.2 yes yes no good no no      ckd
1  NaN ... 38 6000 NaN no no no good no no      ckd
2 423.0 ... 31 7500 NaN no yes no poor no yes     ckd
3 117.0 ... 32 6700 3.9 yes no no poor yes yes     ckd
4 106.0 ... 35 7300 4.6 no no no good no no      ckd
```

```
[5 rows x 25 columns]
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.style.use("seaborn-v0_8")
```

```
print("Info before cleaning:")
print(df.info())
```

```
print("\nMissing values per column before handling:")
print(df.isna().sum())
```

```

numeric_str_cols = ["pcv", "wc", "rc"]

for col in numeric_str_cols:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors="coerce")

cat_cols_manual = ['rbc', 'pc', 'pcc', 'ba', 'htn', 'dm',
                    'cad', 'appet', 'pe', 'ane', 'classification']
cat_cols = [c for c in cat_cols_manual if c in df.columns]

for col in cat_cols:
    df[col] = df[col].astype(str).str.strip()
    df[col] = df[col].replace("nan", np.nan)

print("\nInfo after basic cleaning:")
print(df.info())

print("\nMissing values per column after basic cleaning step:")
print(df.isna().sum())

numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
categorical_cols = df.select_dtypes(include=['object']).columns

df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].median())

```

```
if len(categorical_cols) > 0:
    df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])

print("\nMissing values per column AFTER imputation:")
print(df.isna().sum())
```

```
def iqr_clip(series):
    """
    Clip outliers using the IQR rule:
    values < Q1 - 1.5*IQR → set to lower bound
    values > Q3 + 1.5*IQR → set to upper bound
    """
    Q1 = series.quantile(0.25)
    Q3 = series.quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    return series.clip(lower, upper)
```

```
df[numeric_cols] = df[numeric_cols].apply(iqr_clip)
```

```
plt.figure(figsize=(15, 10))
df[numeric_cols].hist(figsize=(15, 10))
plt.suptitle("Histograms of Numeric Features", fontsize=16)
plt.tight_layout()
```

```
plt.show()
```

```
plt.figure(figsize=(15, 8))
```

```
df[numeric_cols].boxplot(rot=90)
```

```
plt.title("Boxplots of Numeric Features", fontsize=16)
```

```
plt.tight_layout()
```

```
plt.show()
```

```
if 'classification' in df.columns:
```

```
    plt.figure(figsize=(6, 4))
```

```
    sns.countplot(x='classification', data=df)
```

```
    plt.title("Target Variable Distribution (classification)", fontsize=14)
```

```
    plt.xlabel("Class")
```

```
    plt.ylabel("Count")
```

```
    plt.tight_layout()
```

```
    plt.show()
```

```
plt.figure(figsize=(12, 8))
```

```
corr = df[numeric_cols].corr()
```

```
sns.heatmap(corr, annot=False, cmap="coolwarm")
```

```
plt.title("Correlation Heatmap of Numeric Features", fontsize=16)
```

```
plt.tight_layout()
```

```
plt.show()
```

Info before cleaning:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 398 entries, 0 to 397

Data columns (total 25 columns):

#	Column	Non-Null Count	Dtype
0	age	389 non-null	float64
1	bp	386 non-null	float64
2	sg	352 non-null	float64
3	al	353 non-null	float64
4	su	350 non-null	float64
5	rbc	247 non-null	object
6	pc	334 non-null	object
7	pcc	394 non-null	object
8	ba	394 non-null	object
9	bgr	354 non-null	float64
10	bu	379 non-null	float64
11	sc	381 non-null	float64
12	sod	311 non-null	float64
13	pot	310 non-null	float64
14	hemo	347 non-null	float64
15	pcv	329 non-null	object
16	wc	293 non-null	object
17	rc	269 non-null	object
18	htn	396 non-null	object

19 dm 396 non-null object
20 cad 396 non-null object
21 appet 397 non-null object
22 pe 397 non-null object
23 ane 397 non-null object
24 classification 398 non-null object

dtypes: float64(11), object(14)

memory usage: 77.9+ KB

None

Missing values per column before handling:

age	9
bp	12
sg	46
al	45
su	48
rbc	151
pc	64
pcc	4
ba	4
bgr	44
bu	19
sc	17
sod	87
pot	88
hemo	51

```

pcv      69
wc       105
rc       129
htn      2
dm       2
cad      2
appet    1
pe       1
ane      1
classification  0
dtype: int64

```

Info after basic cleaning:

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 398 entries, 0 to 397

Data columns (total 25 columns):

#	Column	Non-Null Count	Dtype
0	age	389 non-null	float64
1	bp	386 non-null	float64
2	sg	352 non-null	float64
3	al	353 non-null	float64
4	su	350 non-null	float64
5	rbc	247 non-null	object
6	pc	334 non-null	object
7	pcc	394 non-null	object

8	ba	394 non-null	object
9	bgr	354 non-null	float64
10	bu	379 non-null	float64
11	sc	381 non-null	float64
12	sod	311 non-null	float64
13	pot	310 non-null	float64
14	hemo	347 non-null	float64
15	pcv	328 non-null	float64
16	wc	292 non-null	float64
17	rc	268 non-null	float64
18	htn	396 non-null	object
19	dm	396 non-null	object
20	cad	396 non-null	object
21	appet	397 non-null	object
22	pe	397 non-null	object
23	ane	397 non-null	object
24	classification	398 non-null	object

dtypes: float64(14), object(11)

memory usage: 77.9+ KB

None

Missing values per column after basic cleaning step:

age	9
bp	12
sg	46
al	45

su	48
rbc	151
pc	64
pcc	4
ba	4
bgr	44
bu	19
sc	17
sod	87
pot	88
hemo	51
pcv	70
wc	106
rc	130
htn	2
dm	2
cad	2
appet	1
pe	1
ane	1
classification	0

dtype: int64

Missing values per column AFTER imputation:

age	0
bp	0

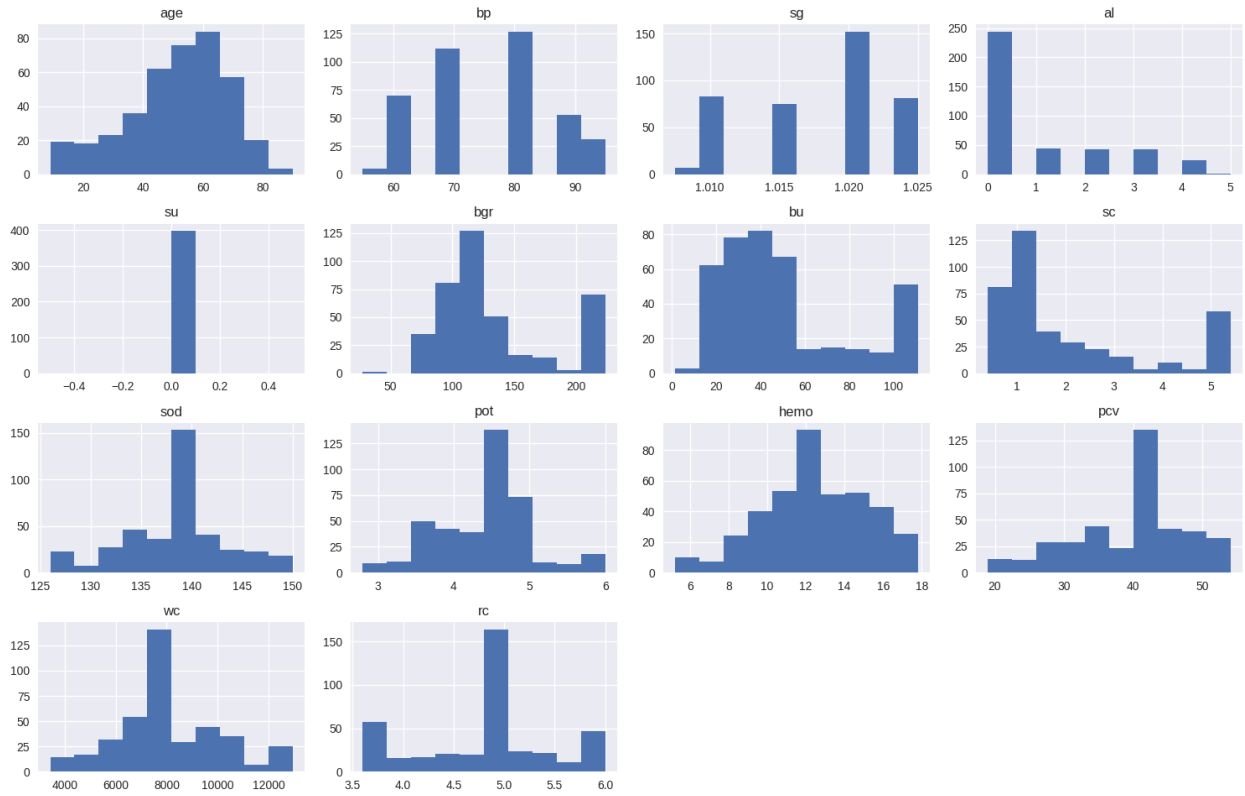
sg	0
al	0
su	0
rbc	0
pc	0
pcc	0
ba	0
bgr	0
bu	0
sc	0
sod	0
pot	0
hemo	0
pcv	0
wc	0
rc	0
htn	0
dm	0
cad	0
appet	0
pe	0
ane	0

classification 0

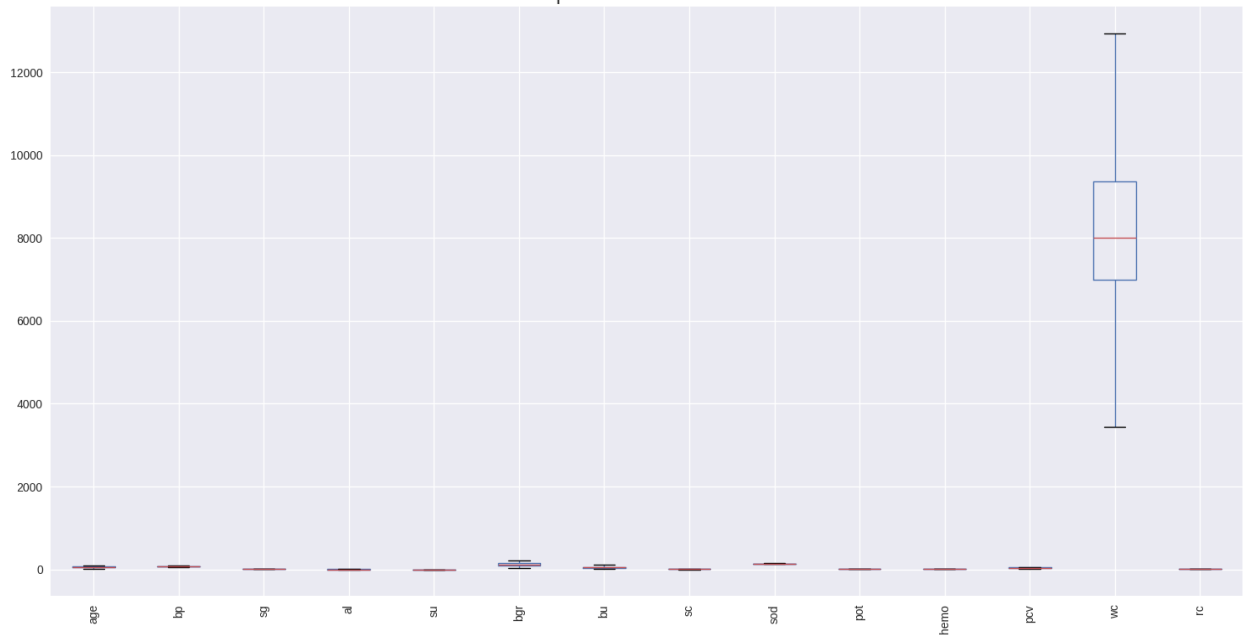
dtype: int64

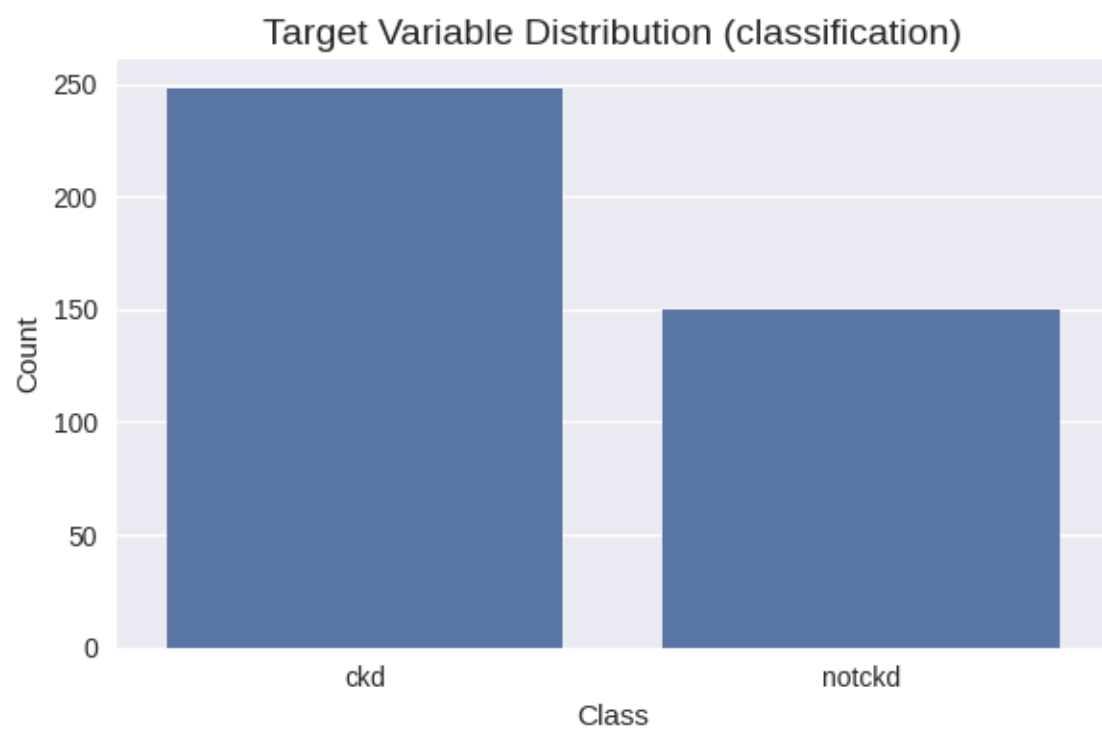
<Figure size 1500x1000 with 0 Axes>

Histograms of Numeric Features



Boxplots of Numeric Features





Correlation Heatmap of Numeric Features

