Final Presentation Notebook

Group 11 — An Exploratory Data Analysis on Malignant Breast Cancer Cells

This notebook consolidates **all datasets, all EDA, and all visualizations** using the updated visualizer structure.

Datasets:

- **Dataset 1:** Wisconsin Breast Cancer (Original)
- **Dataset 2:** Coimbra Breast Cancer (Metabolic)
- **Dataset 3:** Wisconsin Breast Cancer (Diagnostic)

All plots use:

- Cleaned and unified target variables
- Updated palette (0 = benign/healthy, 1 = malignant/patient)
- Dataset-specific functions

In [1]:
```
# ========== IMPORTS ==========

import sys
sys.path.append('./src')

import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

import src.data_loader as data_loader
import src.data_preprocessor as data_preprocessor
import src.visualizer as viz

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

print("All modules loaded successfully!")
```

```
All modules loaded successfully!
```

In [ ]:
```
from IPython.display import Markdown

Markdown("""
# Final Presentation Notebook
### Group 11 — ECE143 Data Science in Practice
This notebook consolidates **all datasets, all EDA, and all visualizations** usi

### Datasets:
- **Dataset 1:** Wisconsin Breast Cancer (Original)
- **Dataset 2:** Coimbra Breast Cancer (Metabolic)
- **Dataset 3:** Wisconsin Breast Cancer (Diagnostic Imaging)
```

```
All plots use:
- Cleaned and unified target variables
- Updated palette (0 = benign/healthy, 1 = malignant/patient)
- Dataset-specific functions
""")
```

In [3]:
```
raw_df1 = data_loader.load_dataset1_raw("dataset/breast_cancer_bd.csv")
df1 = data_preprocessor.clean_dataset1(raw_df1)
df1.head()
```

Out[3]:

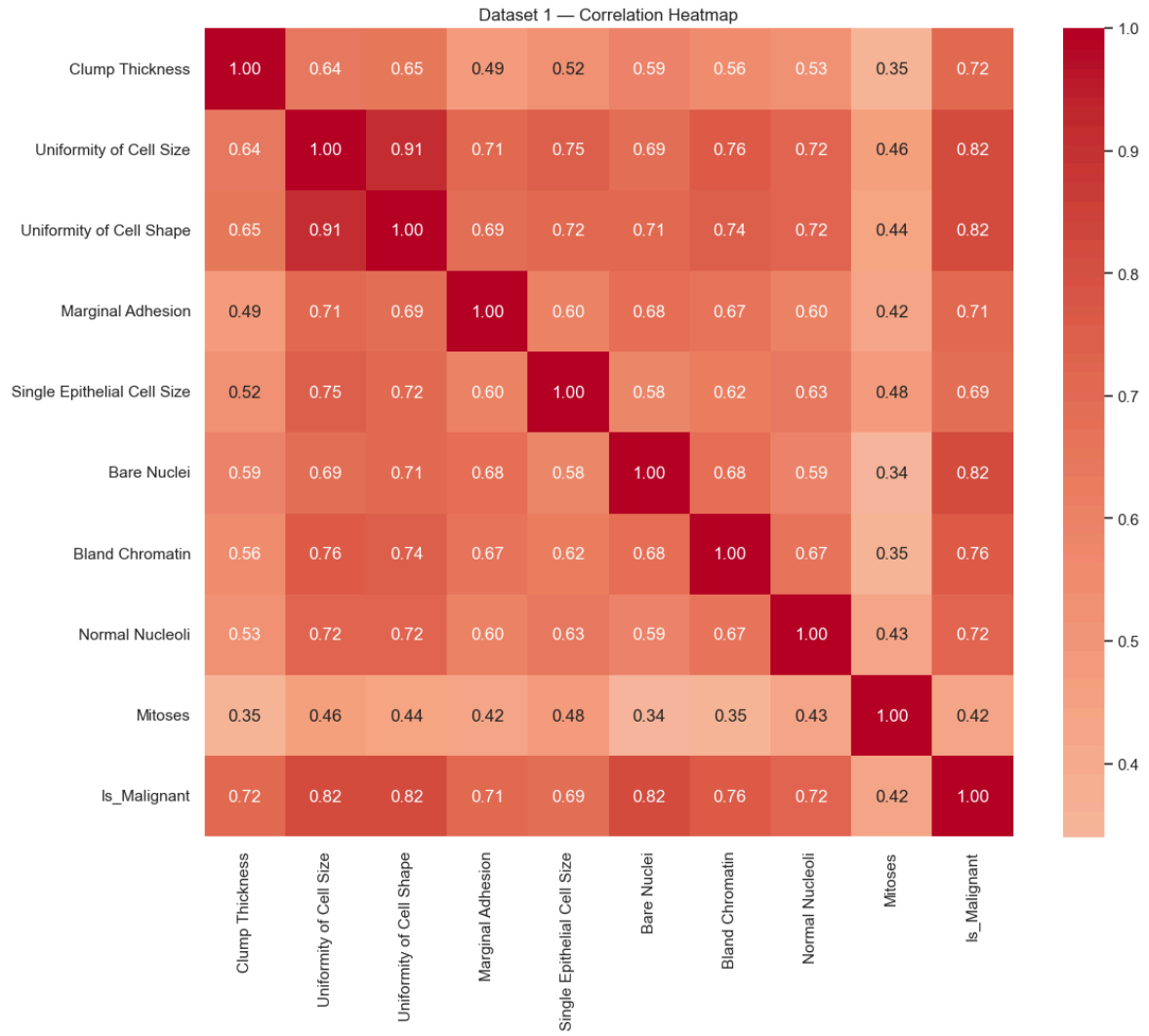
	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Nc Nu
0	5	1	1	1	2	1.0	3	
1	5	4	4	5	7	10.0	3	
2	3	1	1	1	2	2.0	3	
3	6	8	8	1	3	4.0	3	
4	4	1	1	3	2	1.0	3	

In [4]:
```
display(df1.describe())
df1['Is_Malignant'].value_counts()
```

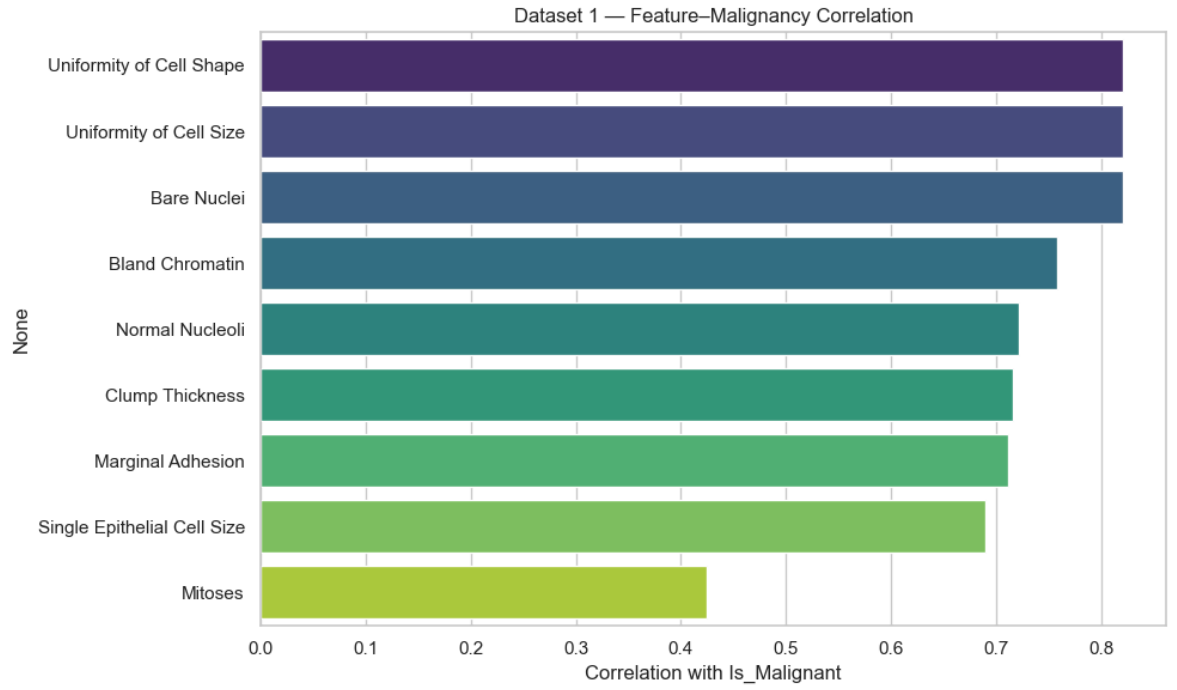
	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Chı
count	675.000000	675.000000	675.000000	675.000000	675.000000	675.000000	675
mean	4.451852	3.146667	3.208889	2.848889	3.229630	3.537778	3
std	2.820859	3.055005	2.976552	2.875917	2.208497	3.637871	2
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
25%	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2
50%	4.000000	1.000000	1.000000	1.000000	2.000000	1.000000	3
75%	6.000000	5.000000	5.000000	4.000000	4.000000	6.000000	5
max	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10

Out[4]:
```
Is_Malignant
0    439
1    236
Name: count, dtype: int64
```

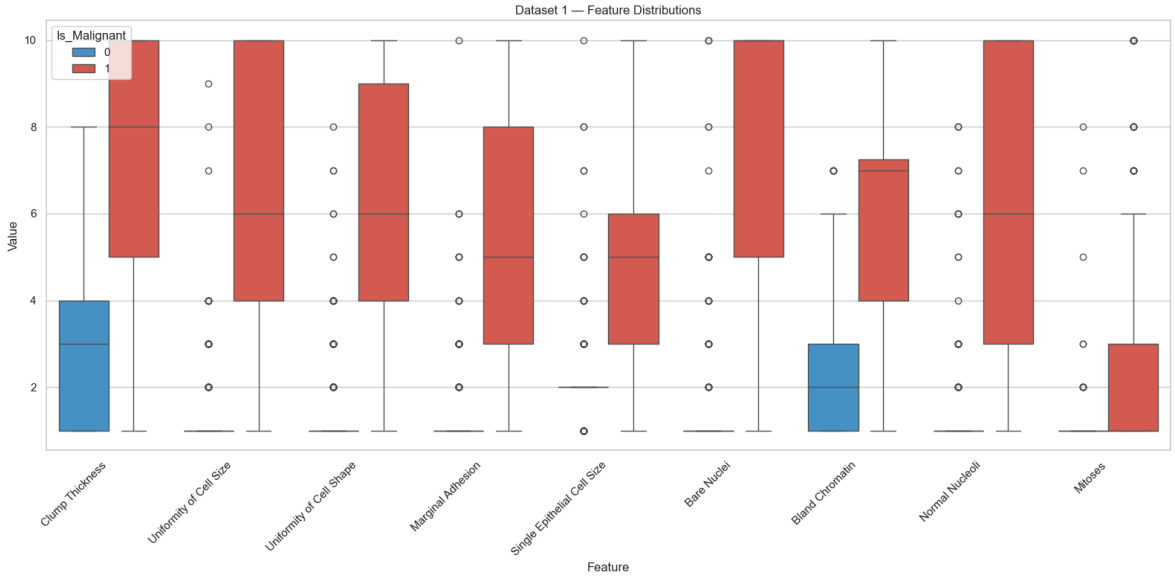
In [5]:
```
viz.plot_d1_heatmap(df1)
```



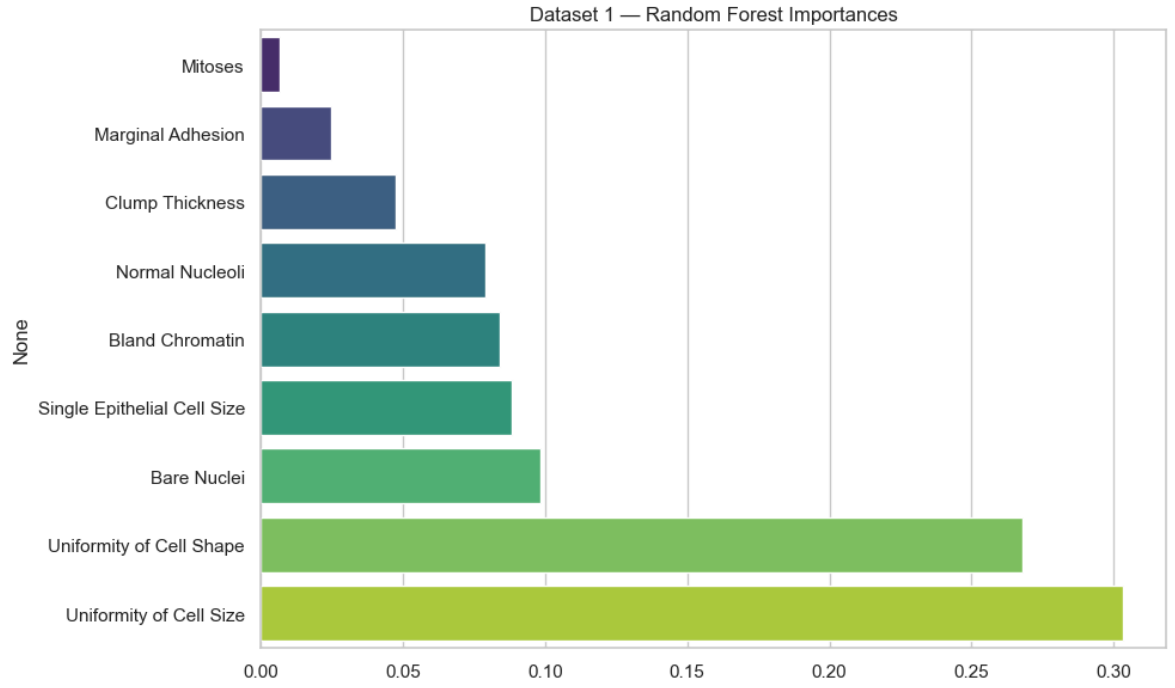
In [6]:
```
viz.plot_d1_feature_ranking(df1)
```



In [7]:
```
viz.plot_d1_boxplots(df1)
```



In [8]:
```
X1 = df1.drop(columns=["Is_Malignant"])
y1 = df1["Is_Malignant"]

rf1 = RandomForestClassifier(random_state=42)
rf1.fit(X1, y1)

viz.plot_d1_rf_importance(df1)
```



In [9]:
```
raw_df2 = data_loader.load_dataset2_raw("dataset/Coimbra_breast_cancer_dataset.c
df2 = data_preprocessor.clean_dataset2(raw_df2)
df2.head()
```

Out[9]:

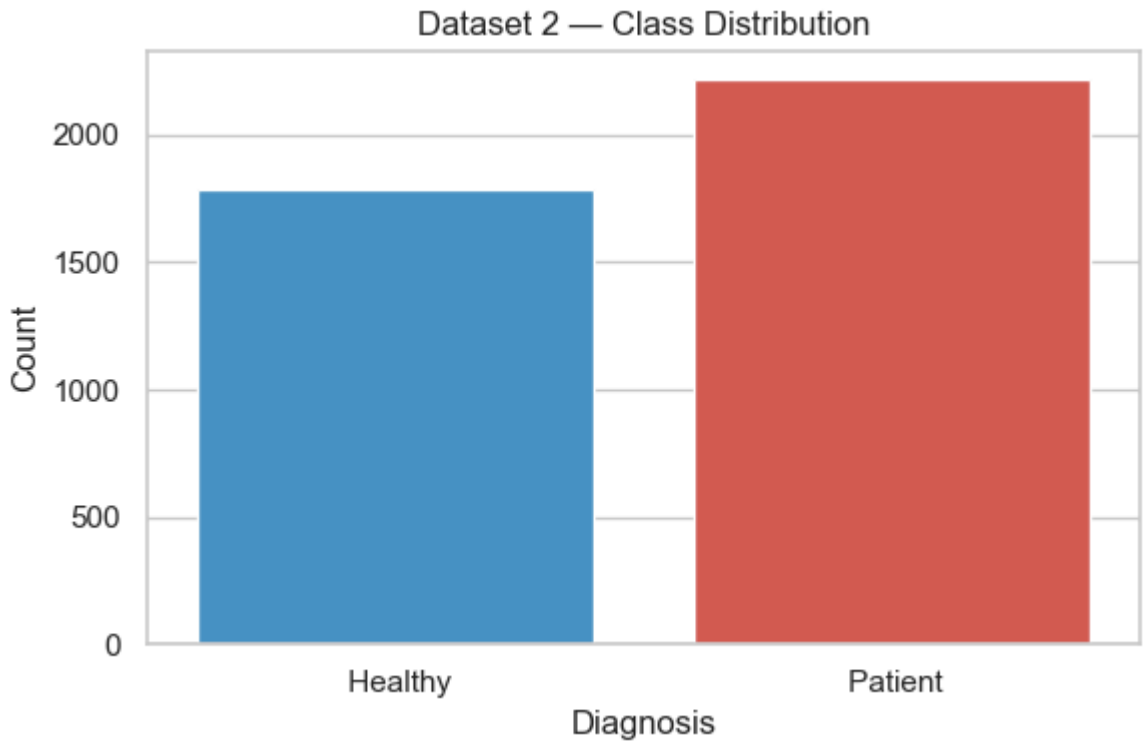
	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	M
0	54	35.207389	103	5.642	1.378660	65.6699	9.738408	31.17499	19
1	52	22.978520	132	6.054	1.145435	47.5445	3.627241	23.03327	42
2	32	21.101341	87	5.668	1.008595	50.5074	5.067841	9.51156	89
3	42	26.761205	132	2.875	1.003837	16.8972	10.096475	9.75652	26
4	55	34.232520	76	3.120	1.597721	17.6852	11.845054	17.21541	78

In [10]:
```
display(df2.describe())
df2['diagnosis'].value_counts()
```

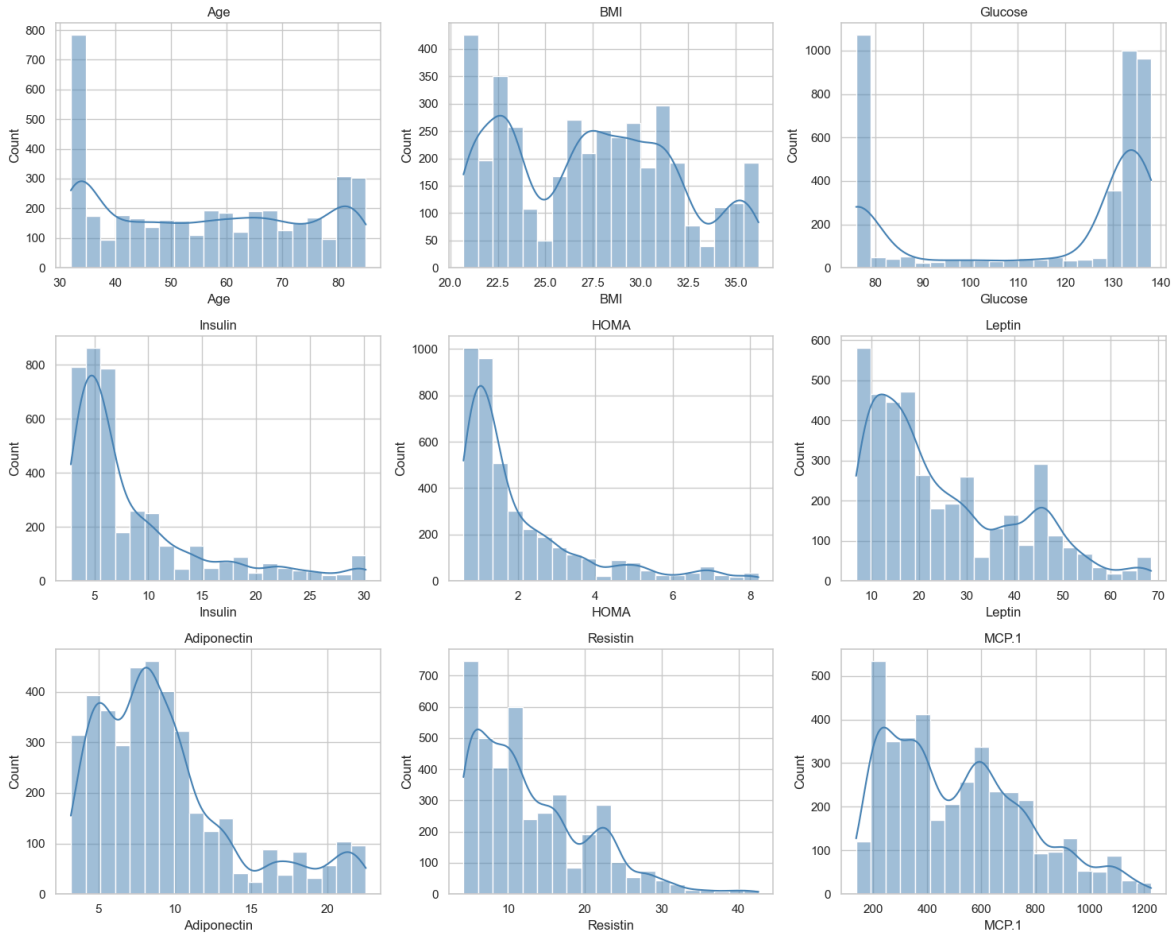
	Age	BMI	Glucose	Insulin	HOMA	Leptin
count	4000.00000	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000
mean	56.21075	27.422280	113.876500	8.654001	2.024332	25.137737
std	17.80965	4.413884	25.837795	6.435160	1.625638	15.096446
min	32.00000	20.690751	76.000000	2.821000	0.590033	6.831900
25%	39.00000	23.079053	76.000000	4.421750	0.970090	12.712750
50%	56.00000	27.558485	131.000000	5.818000	1.373842	19.805050
75%	72.00000	30.814916	134.000000	10.466250	2.502776	36.670250
max	85.00000	36.209606	138.000000	30.211000	8.218456	68.506600

Out[10]:
```
diagnosis
1    2216
0    1784
Name: count, dtype: int64
```

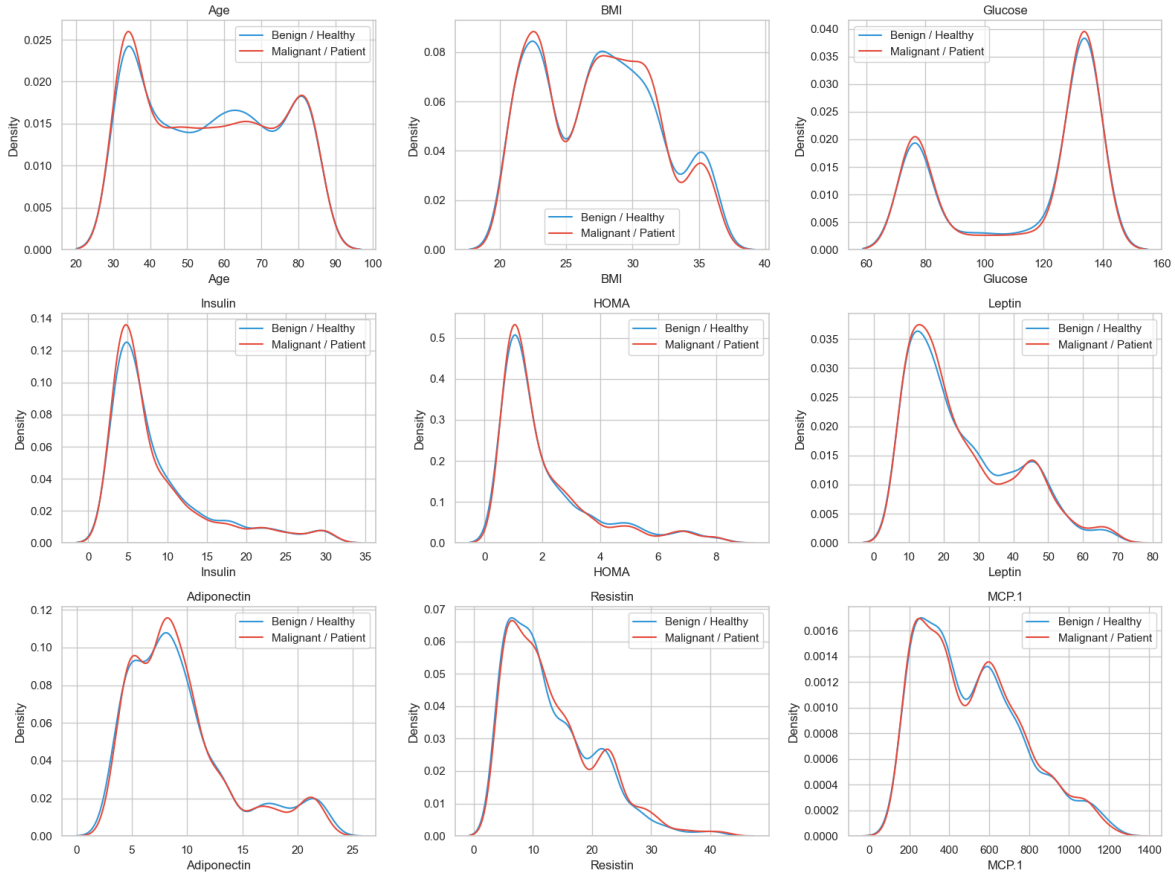
In [11]:
```
viz.plot_d2_class_distribution(df2)
```



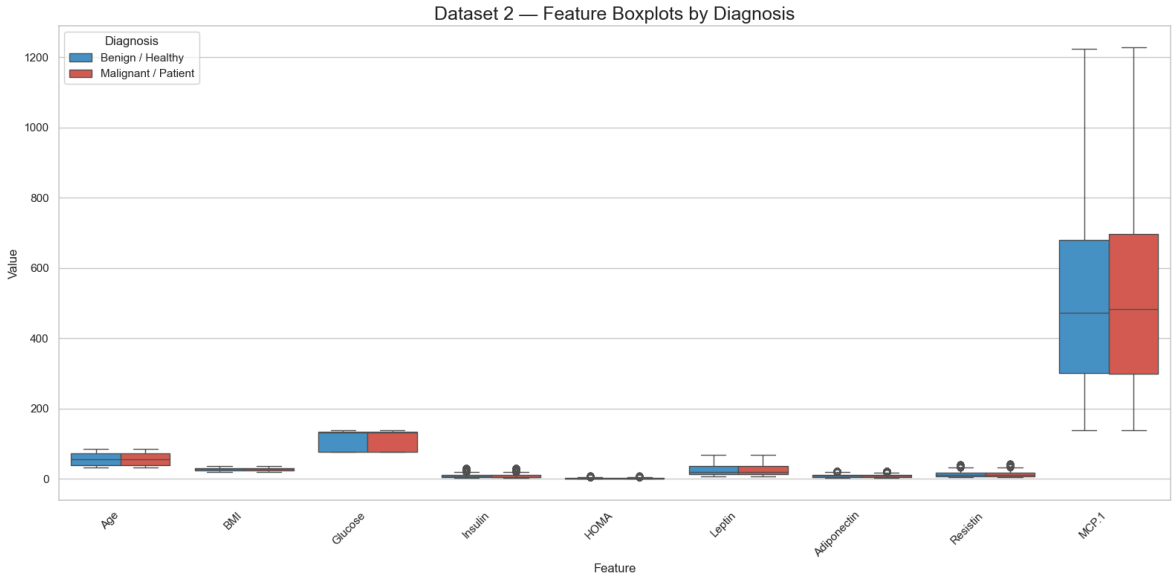
In [12]:
```
viz.plot_d2_hist(df2)
```



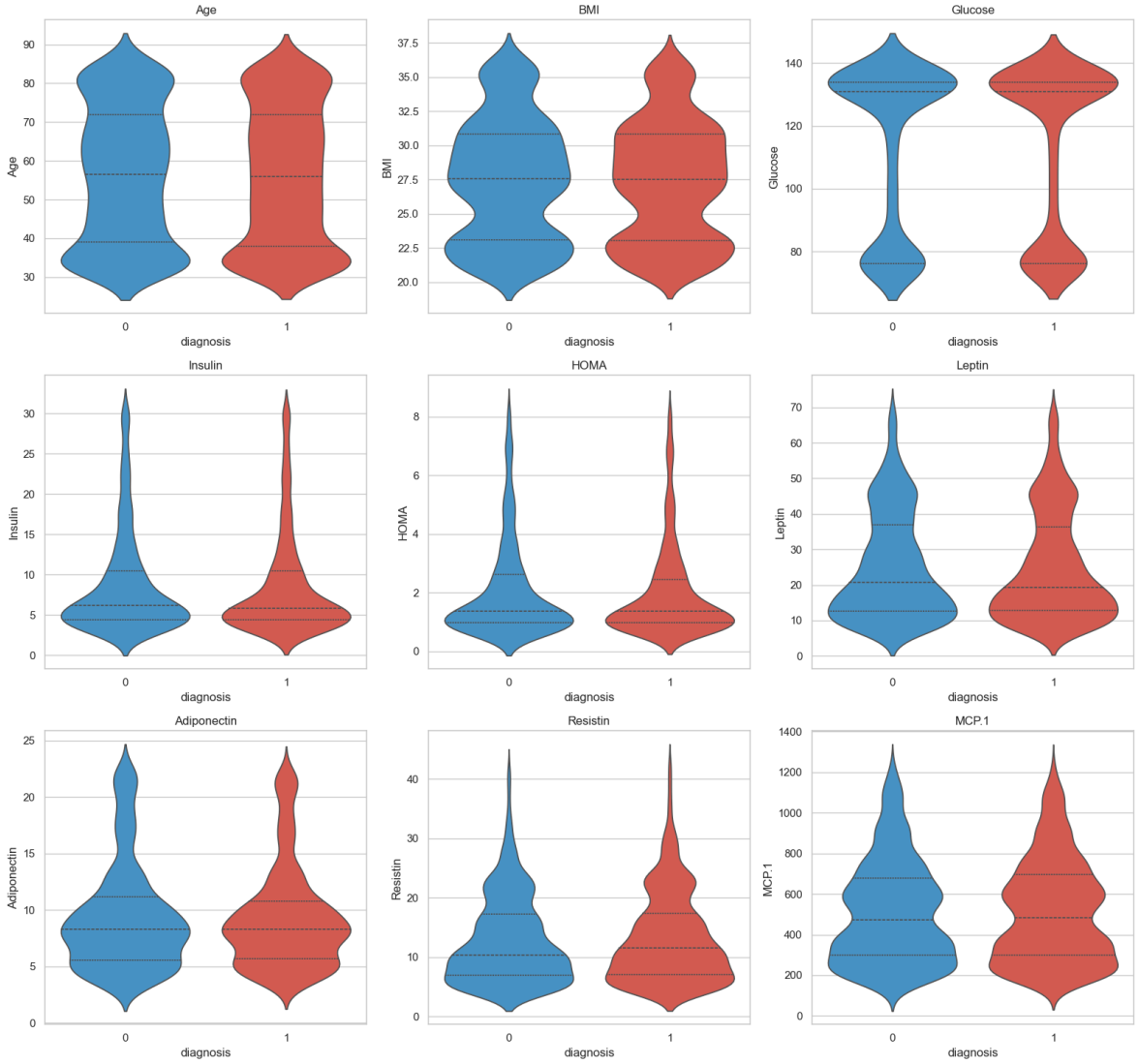
In [13]:
```
viz.plot_d2_kde(df2)
```



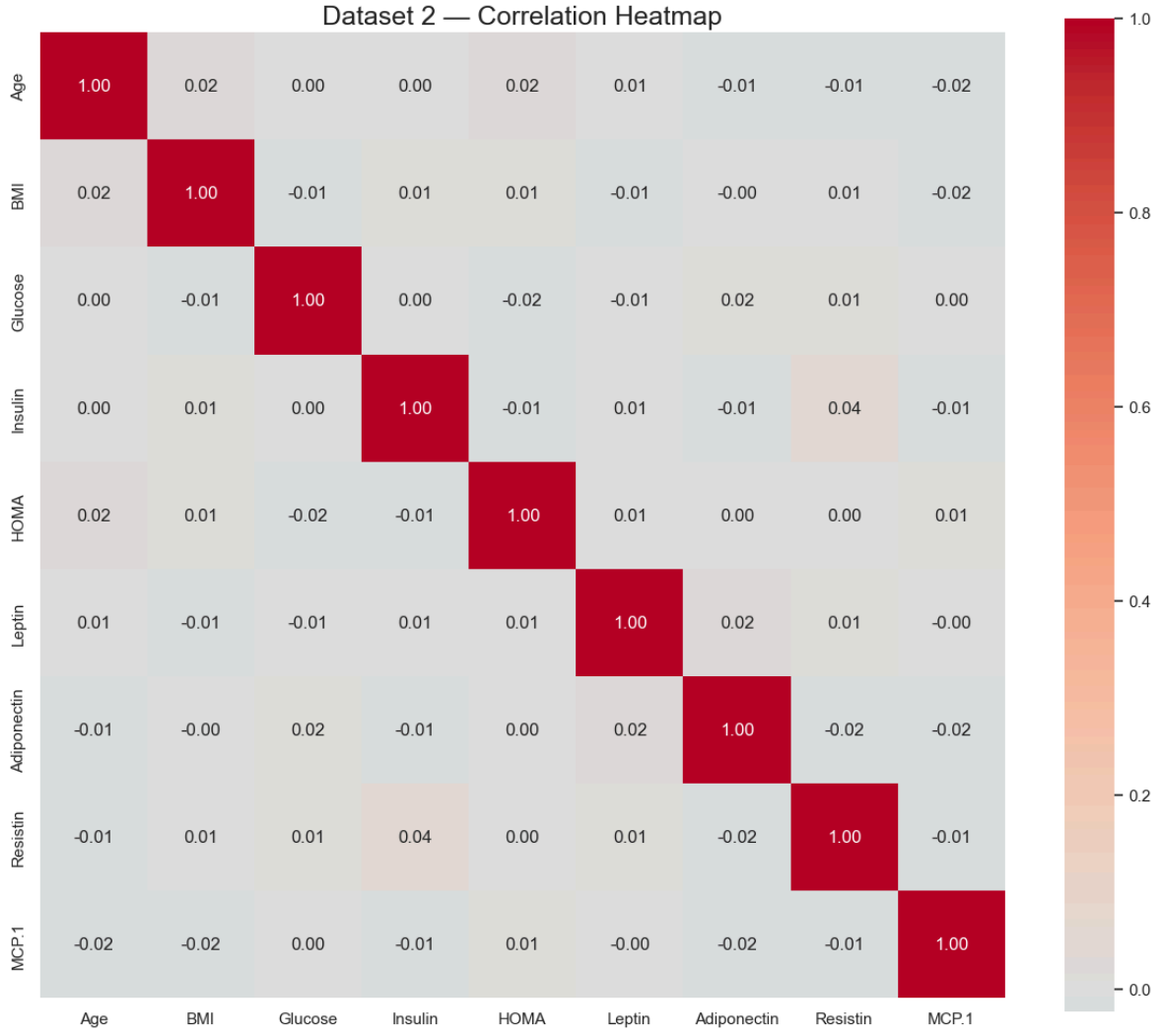
In [14]:
```
viz.plot_d2_boxplots(df2)
```



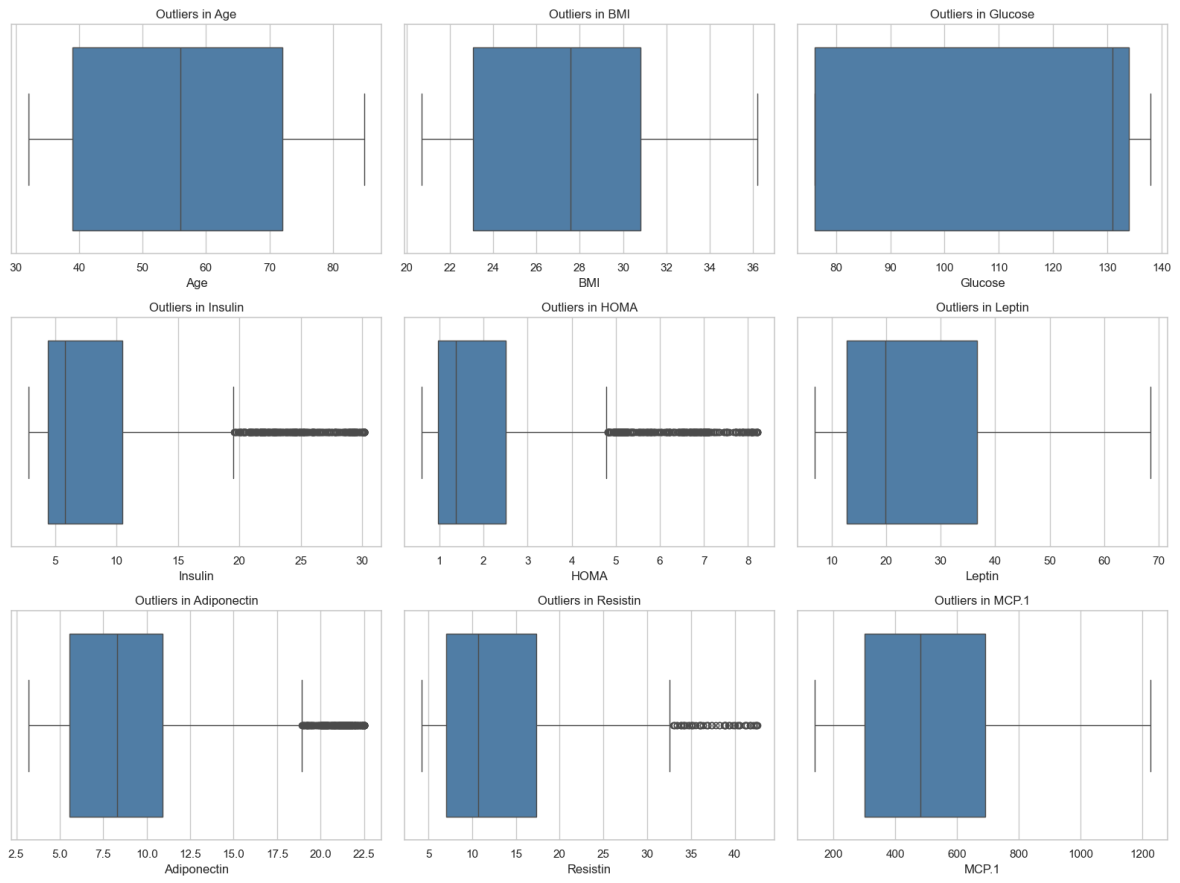
In [15]:
```
viz.plot_d2_violin(df2)
```



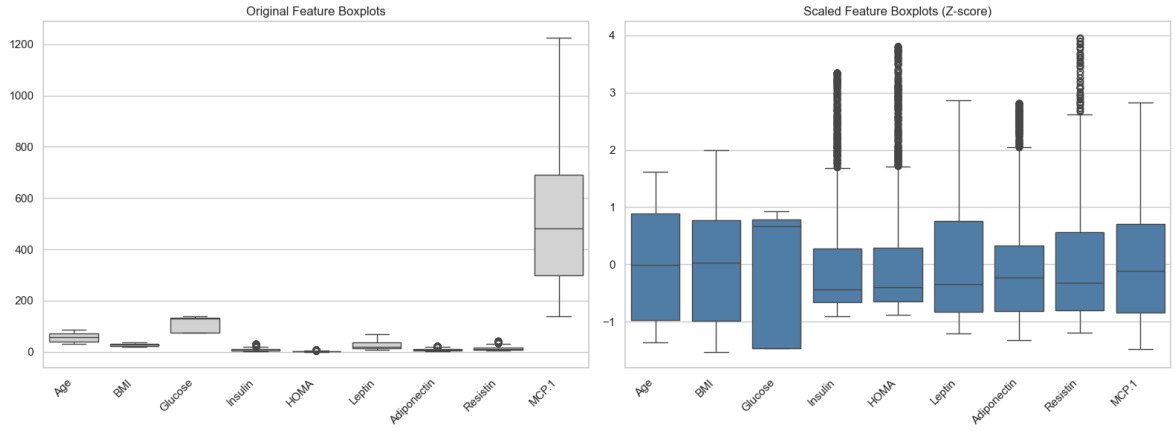
In [16]:
```
viz.plot_d2_heatmap(df2)
```



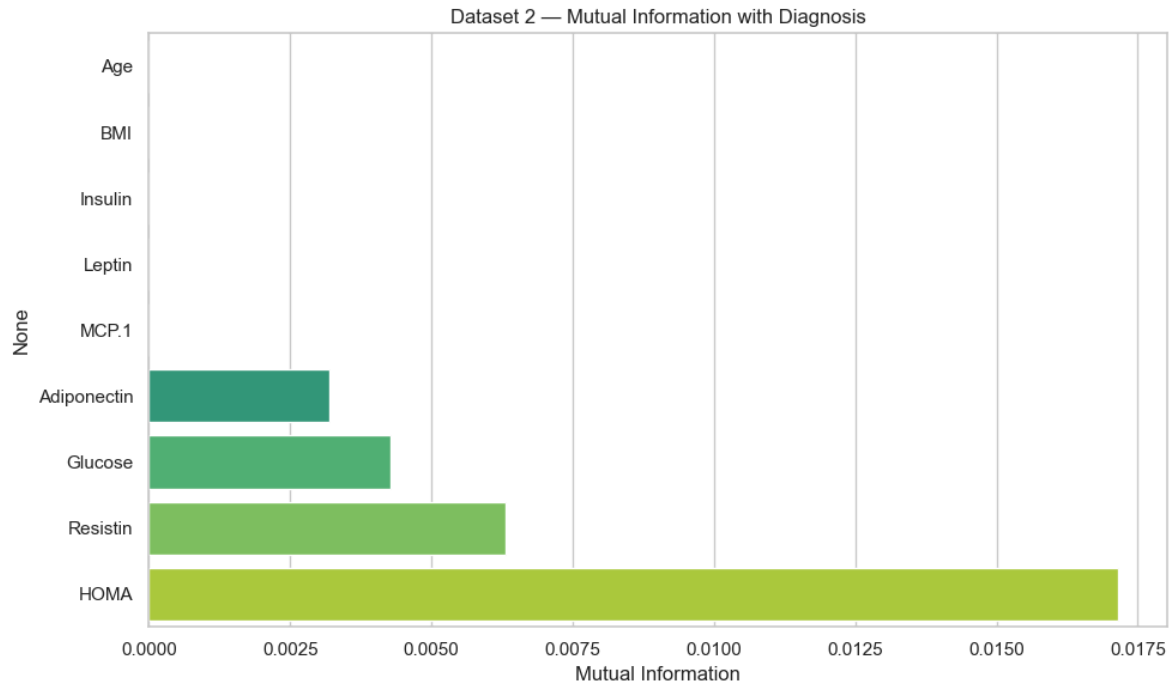
In [17]:
```
viz.plot_d2_outliers(df2)
```



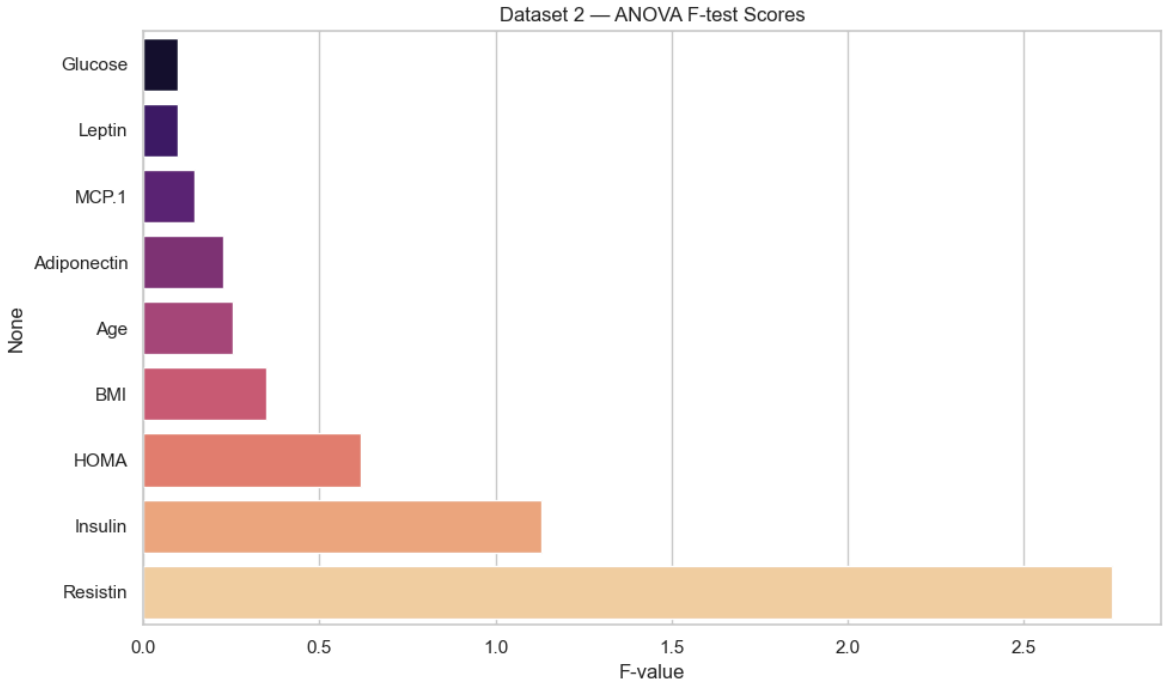
In [18]:
```
viz.plot_d2_scaled_boxplots(df2)
```



In [19]:
```
viz.plot_d2_mutual_info(df2)
```



In [20]:
```
viz.plot_d2_anova(df2)
```



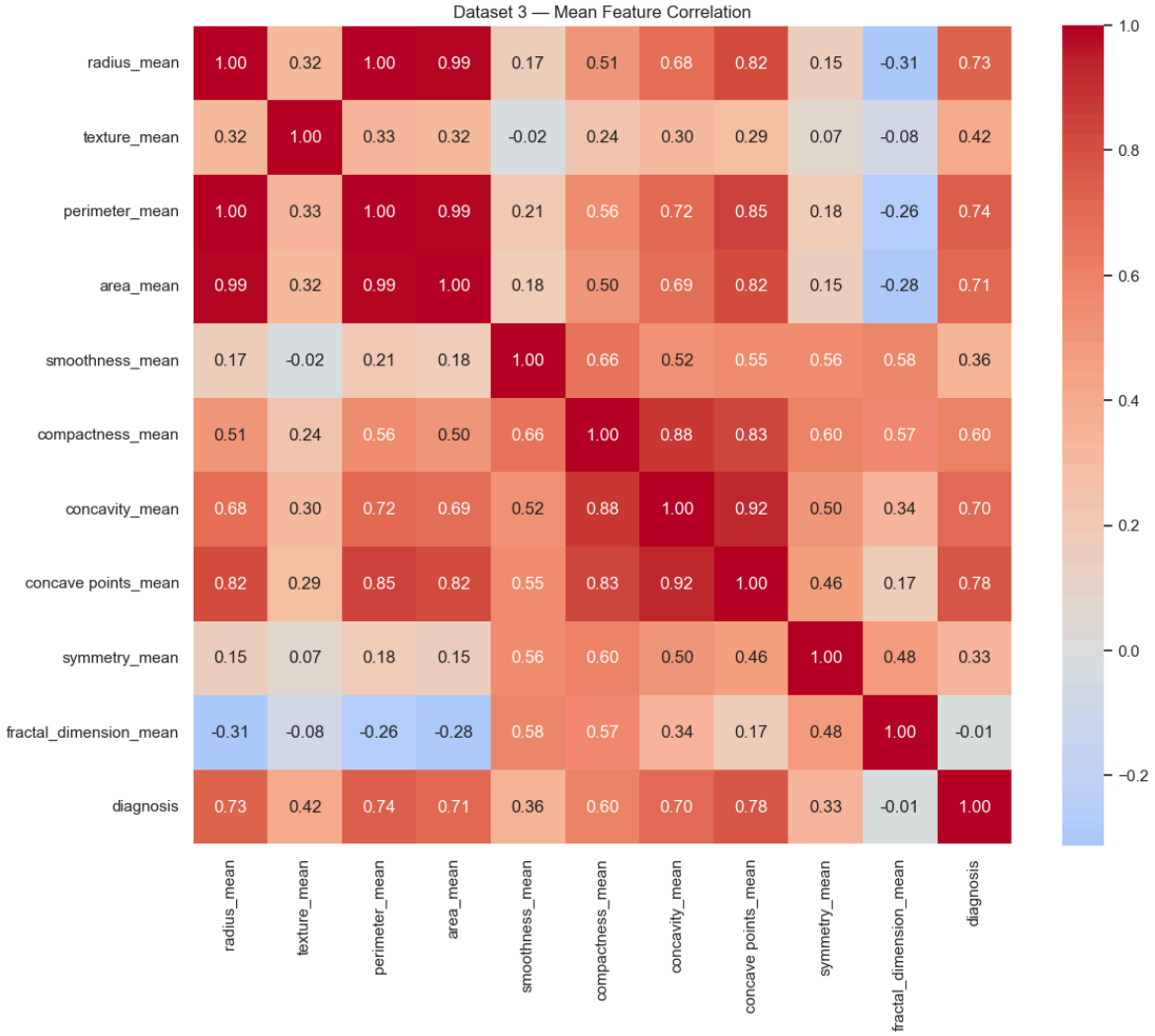
In [21]:
```
# Dataset 3 - Prepare standardized features (matching teammate's code)
raw_df3 = data_loader.load_dataset3_raw("dataset/breast_cancer_dia.csv")
df3 = data_preprocessor.clean_dataset3(raw_df3)

# Split features & target
features = df3.drop(columns=["diagnosis"])
target = df3["diagnosis"].astype(int)

# Standardize (z-score) - this is the KEY missing step
means = features.mean()
stds = features.std()
features_normalized = (features - means) / stds

# Final X and y for modeling
X3 = features_normalized
y3 = target

df3.head()
```

Out[21]:

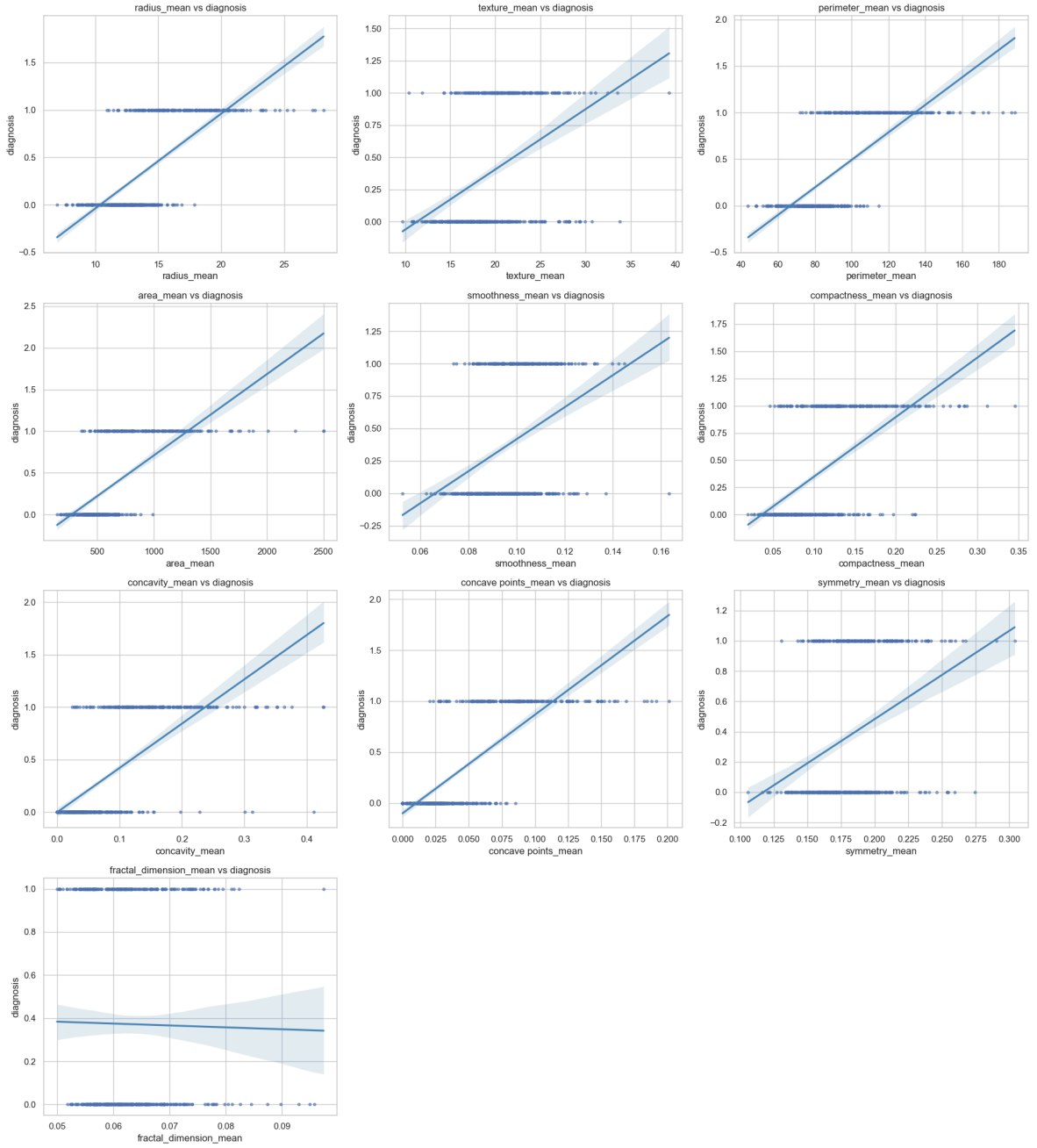
	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	1	17.99	10.38	122.80	1001.0	
1	1	20.57	17.77	132.90	1326.0	
2	1	19.69	21.25	130.00	1203.0	
3	1	11.42	20.38	77.58	386.1	
4	1	20.29	14.34	135.10	1297.0	

5 rows × 31 columns

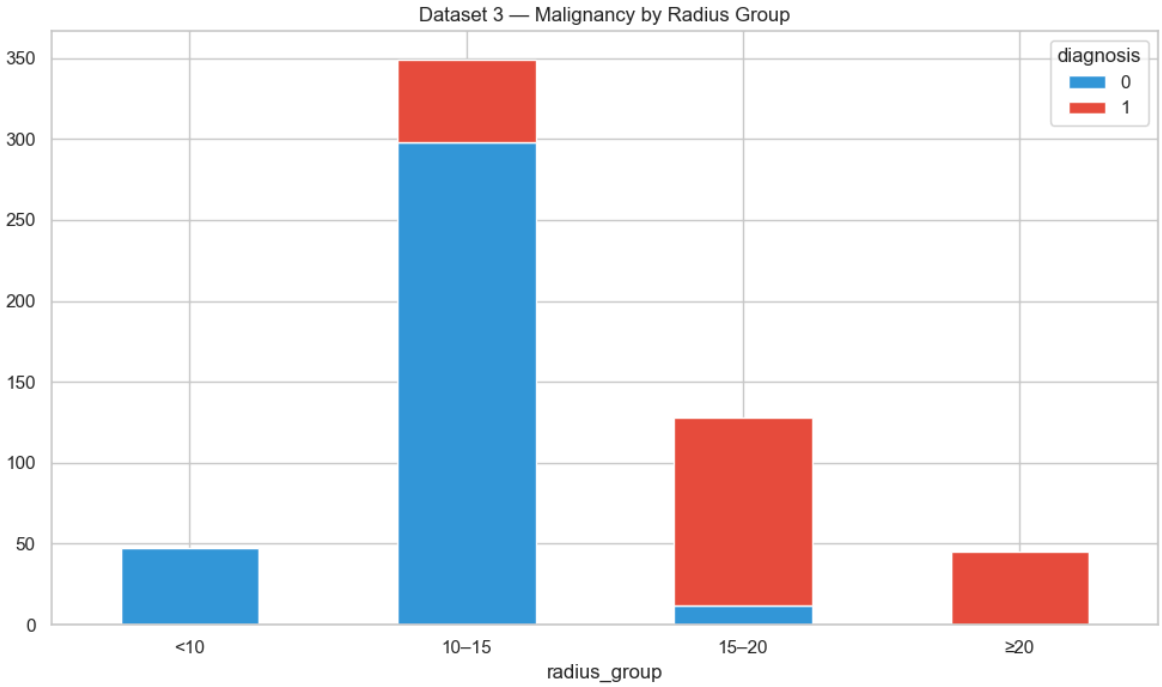
In [22]:
```
viz.plot_d3_mean_heatmap(df3)
```



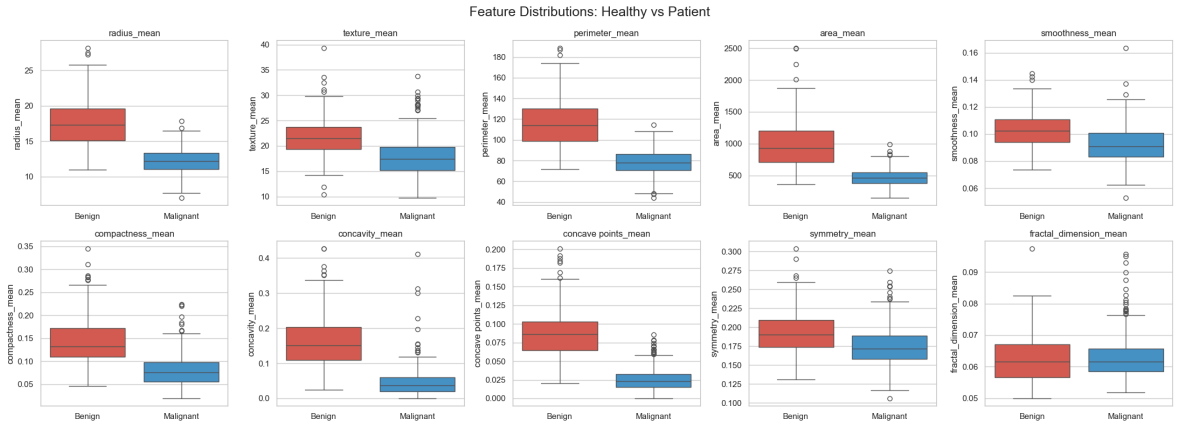
In [23]:
```
viz.plot_d3_regression_mean(df3)
```



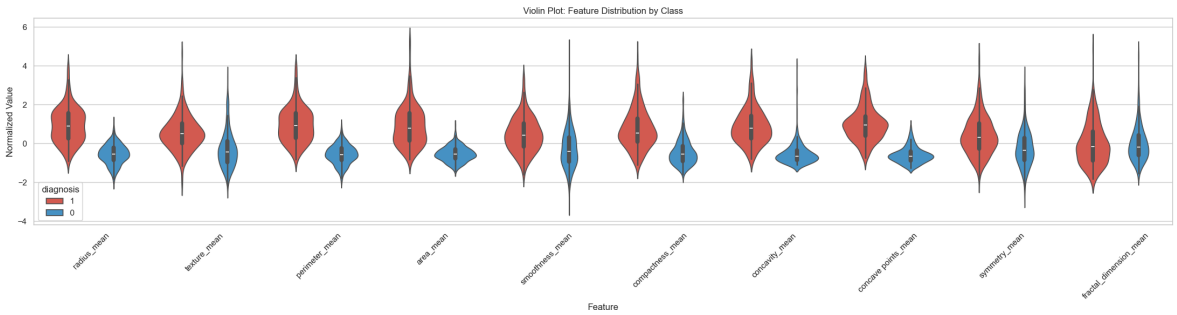
In [24]:
```
viz.plot_d3_radius_group(df3)
```



In [25]:
```
viz.plot_d3_mean_boxplots(df3)
```

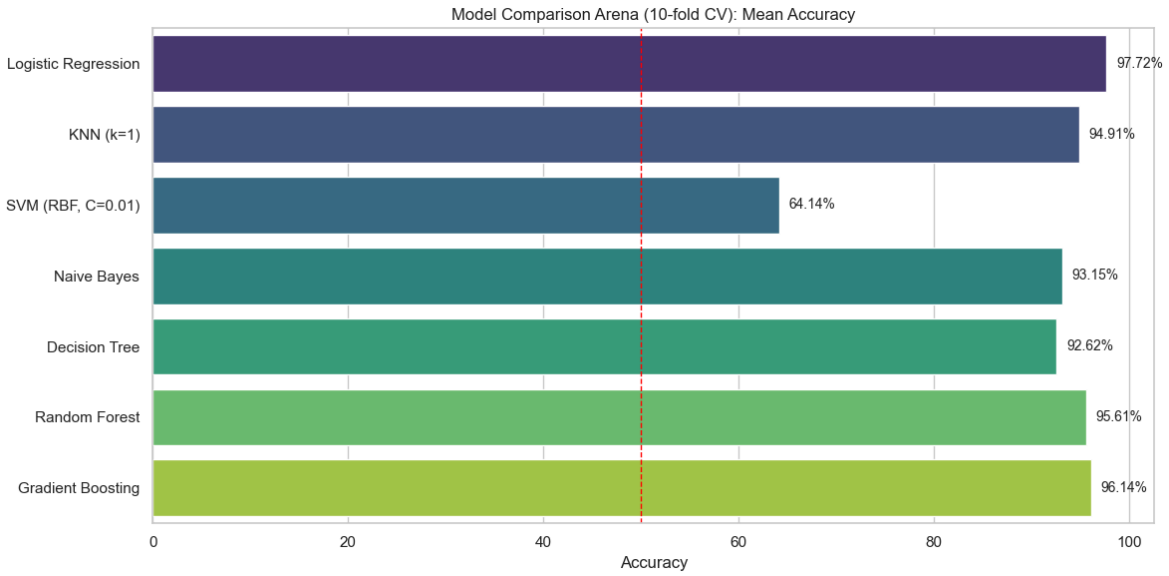


In [26]:
```
viz.plot_d3_violin(df3)
```

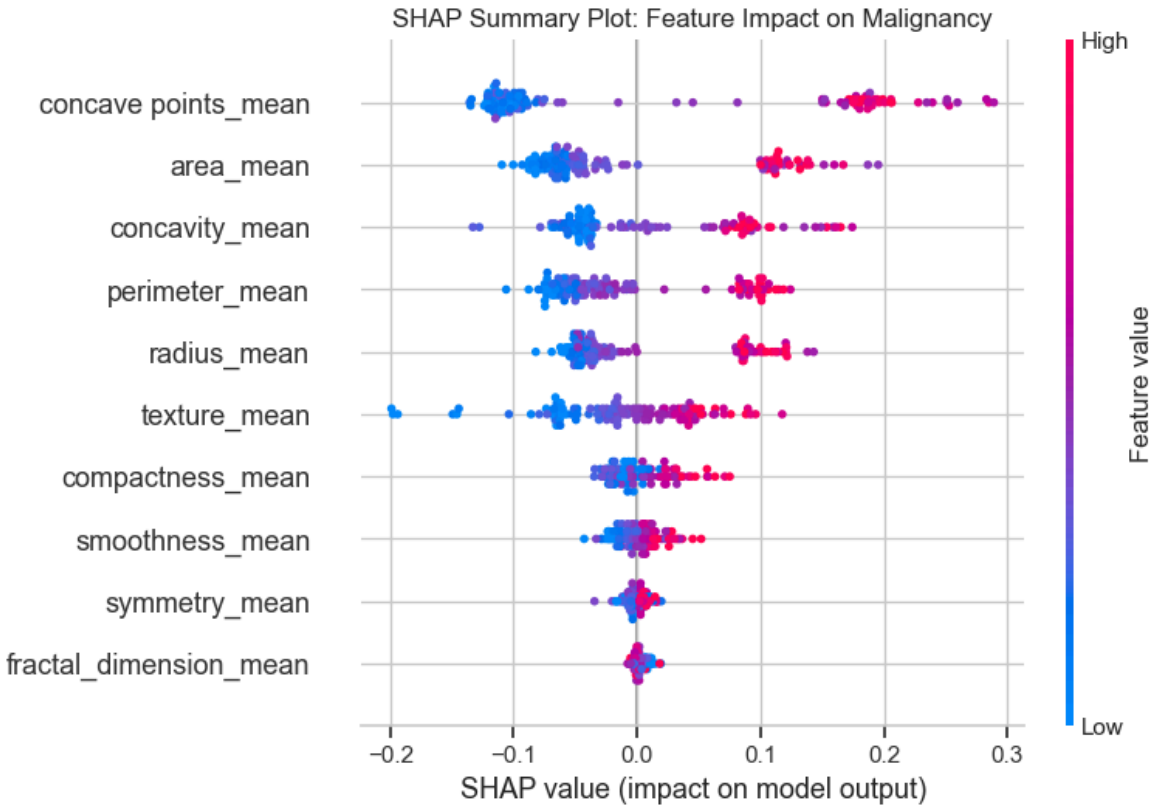


In [27]:
```
viz.run_d3_model_comparison(df3)
```

```
=== Dataset 3 Model Comparison (10-fold CV) ===
Logistic Regression: 97.72%
KNN (k=1): 94.91%
SVM (RBF, C=0.01): 64.14%
Naive Bayes: 93.15%
Decision Tree: 92.62%
Random Forest: 95.61%
Gradient Boosting: 96.14%
```



In [28]:
```
viz.plot_d3_shap(df3)
```




```
Top Influential Feature: concave points_mean
```