

Concentric Fibonacci Square Block Feature Engineering for Osteoporosis Screening

Shriverdhan Pathak (2025PGCSDS01), Satyam Mishra (2025PGCSDS06),
Bhumika Verma (2025PGCSDS07), Amit Yadav (2025PGCSDS13),
Saurabh Verma (2025PGCSDS14), Sayan Mondal (2025PGCSDS17)

Department of Computer Science and Engineering, NIT Jamshedpur

Abstract

Osteoporosis and osteopenia are bone disorders that gradually reduce bone strength and increase the likelihood of fractures. In many hospitals, advanced densitometry devices such as DXA are not always available, whereas plain X-ray imaging is part of routine practice. This report presents a lightweight and interpretable pipeline for osteoporosis screening using knee X-ray images. The main idea is to partition each image into a set of concentric square regions whose side lengths follow the Fibonacci sequence, so that the central joint area and outer cortical regions are both represented at different scales. From each region, ten intensity and texture-based features are extracted and concatenated into a 90-dimensional feature vector. These handcrafted descriptors are then used to train ten different classical machine learning models under four train-test splits (60/40, 70/30, 80/20 and 90/10). Performance is analysed using accuracy, precision, recall, confusion matrices and ROC curves. The overall framework is simple to implement, does not require large datasets, and remains transparent enough to be discussed with clinicians.

Keywords: Osteoporosis, Knee X-ray, Fibonacci segmentation, Handcrafted features, Machine learning

1 Introduction

Osteoporosis is characterised by a reduction in bone mineral density and deterioration of internal structure, leading to increased fracture risk. Osteopenia is considered an intermediate stage where bone density is lower than normal but not yet in the osteoporotic range. Early identification of both conditions allows timely intervention and can reduce morbidity.

The standard reference method for assessing bone mineral density is Dual-Energy X-ray Absorptiometry (DXA). However, DXA scanners are expensive and not always accessible in small or rural hospitals. On the other hand, radiographs of the knee or hip joints are widely acquired for a variety of reasons. If useful quantitative information can be derived from such routine X-ray images, they can act as an additional cue for bone health assessment.

In recent years, deep learning models have been explored for medical image classification, including bone disease detection. While these models can be powerful, they usually demand large amounts of labelled data and significant computational resources. Moreover, explaining their decisions to clinicians is not always straightforward. Handcrafted feature extraction approaches remain useful when datasets are modest and interpretability is important.

In this work we design a feature engineering scheme that uses concentric square regions determined by the Fibonacci sequence. Around the centre of the knee joint, we form nested squares with radii given by Fibonacci numbers and then convert them into non-overlapping ring-like regions. For each region, we compute a fixed set of ten features related to intensity statistics, texture measures and local complexity. These features serve as input to several classical machine learning models which attempt to distinguish between Normal, Osteopenia and Osteoporosis classes.

2 Problem Statement

The objective of this project is to build an automated pipeline that:

- takes a pre-processed knee X-ray image as input,
- extracts a compact yet informative feature vector using a Fibonacci-based partitioning strategy, and
- assigns the image to one of three bone health categories:
 1. Class 0: Normal,
 2. Class 1: Osteopenia,
 3. Class 2: Osteoporosis.

The design constraints are:

1. The dataset contains a few hundred images, so very deep models may overfit.
2. The pipeline must be computationally feasible on standard CPUs or laptops.
3. The resulting features and decision process should be explainable.

To meet these constraints, we rely on handcrafted features extracted from geometrically meaningful regions and train classical machine learning models rather than complex deep networks.

3 Limitations

Although the proposed Fibonacci-based regional partitioning and handcrafted feature extraction pipeline demonstrates strong potential for osteoporosis screening, certain limitations must be acknowledged:

- **Limited Dataset Size:** The experimental dataset contains only a few hundred radiographs, which restricts model generalization and increases variance across train-test splits.
- **Class Imbalance:** The osteopenia class is under-represented compared to the normal and osteoporosis classes. This imbalance results in lower recall and higher confusion for this transitional class.
- **Model Sensitivity to Knee Center Alignment:** The pipeline assumes that the geometric center of the image aligns with the knee joint center. Slight shifts caused during acquisition or cropping influence regional masking and feature reliability.
- **Limited Feature Diversity:** Only a set of ten handcrafted features per region is considered. These descriptors may not capture deeper trabecular micro-structure changes visible to radiologists.
- **Single-site Anatomy Consideration:** Analysis is focused solely on knee joints, while osteoporosis is a systemic condition and may require assessment of multiple skeletal locations for accurate risk evaluation.
- **Hardware/Exposure Variations:** Radiographic noise, soft-tissue overlap, and changes in acquisition settings can alter intensity-based measurements, especially in outer regions.

4 Practical Constraints

From an application and deployment standpoint, the proposed system also faces a few operational challenges:

- **Dependence on Radiographic Quality Control:** Variation in scanner calibration and beam angle affects reproducibility of bone texture patterns.
- **No Use of Clinical Risk Factors:** Patient attributes (e.g., age, sex, BMI, fracture history, medication) are not incorporated, although they significantly influence fracture risk.
- **Potential Overfitting in Higher Train Ratios:** Elevated performance in the 90/10 split suggests sensitivity to small test samples, indicating statistical reliability concerns.
- **Manual Preprocessing Effort:** The pipeline still requires manual verification of alignment and ROI consistency, which may introduce variability without automated localization assistance.
- **Scope for Validation:** Clinical adoption requires multi-center testing and larger trials to verify applicability across populations and imaging devices.

5 Dataset and Preprocessing

This study uses a total of 616 knee radiographs collected from a clinical repository. Each case is categorized into one of three bone-health conditions based on radiological reports and reference dual-energy X-ray absorptiometry (DXA) findings. The goal is to develop an image-based screening tool that can provide a supportive indication of bone strength when DXA measurements are unavailable.

The radiographs exhibit considerable natural variability in terms of patient age, positioning of the knee joint, soft tissue composition, and X-ray acquisition settings. Such variability is common in real-world clinical environments and highlights the need for robust feature extraction techniques. The three target classes are:

- **Normal (0):** Healthy bone structure with no significant reduction in density.
- **Osteopenia (1):** Intermediate stage where bone density is lower than normal but not yet osteoporotic.
- **Osteoporosis (2):** Marked reduction in bone density and trabecular structure, increasing the risk of fracture.

Table 1 provides the class-wise distribution of images in the dataset.

Table 1: Class-wise distribution of knee X-ray images.

Class	Description	Number of images
0	Normal	221
1	Osteopenia	154
2	Osteoporosis	241
Total		616

Preprocessing

To ensure that features are extracted consistently across all images, a set of preprocessing steps is applied:

1. **Resizing:** Each radiograph is resized to 289×289 pixels. The size is specifically chosen so that the largest Fibonacci radius ($F = 144$) fits perfectly from the image center to the boundary in all directions.
2. **Grayscale conversion:** All images are converted to grayscale (if not already), since the diagnostic information linked to bone density is contained in intensity variations rather than color channels.
3. **Intensity normalization:** Pixel values are optionally normalized to the range $[0, 1]$ to reduce the influence of acquisition-dependent exposure differences.
4. **Cropping and center alignment:** The central region of the knee joint is retained so that the joint space becomes approximately aligned with the geometric center of the image. This step is important because the proposed Fibonacci-based region formation assumes the knee joint lies around the central area.

These operations standardize the input space and minimize external noise, ensuring that the extracted texture and intensity features primarily reflect underlying bone health rather than irrelevant imaging artifacts.

6 Methodology

6.1 Overview of the Pipeline

The overall pipeline is illustrated in Figure 1. Starting from a pre-processed image, we generate a set of concentric Fibonacci regions, compute ten features in each region, and obtain a 90-dimensional vector. This vector, along with its class label, forms one row in the final feature dataset used for training classifiers.

Block Diagram: Fibonacci-Based Feature Extraction & Classification

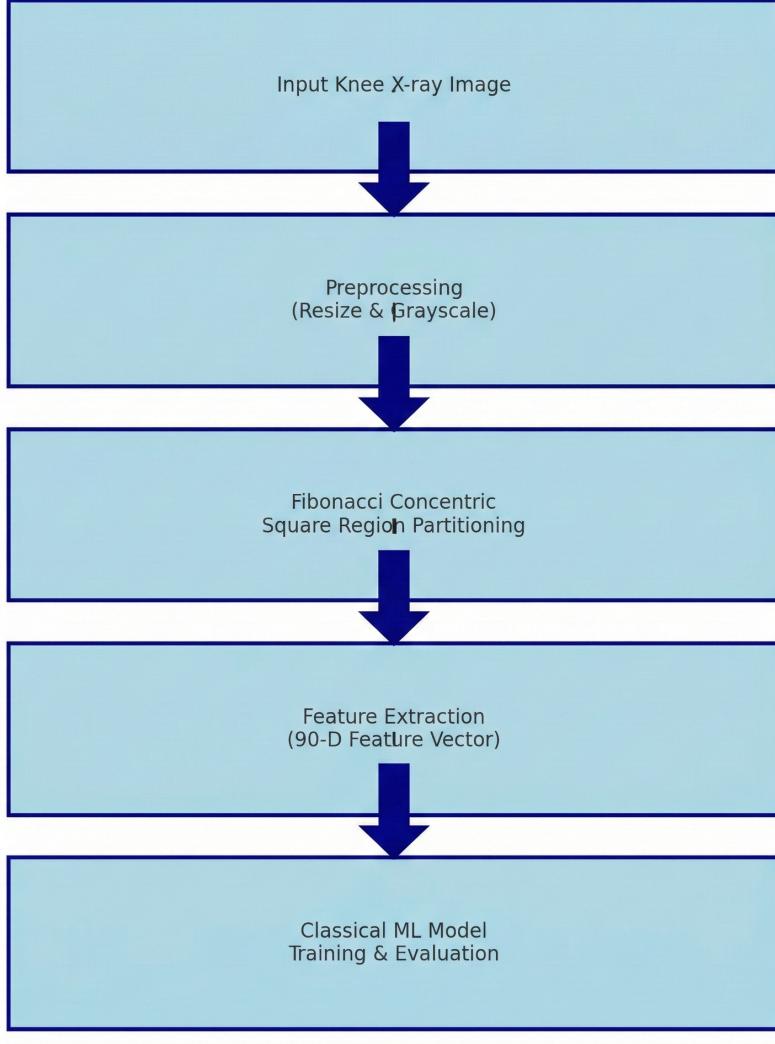


Figure 1: Block diagram of the proposed Fibonacci-based feature extraction and classification pipeline. The image is preprocessed, partitioned into concentric square regions, and a fixed-length feature vector is obtained. Classical models are then trained and evaluated.

6.2 Fibonacci-Based Concentric Square Regions

We define a list of Fibonacci radii

$$\mathcal{F} = [3, 5, 8, 13, 21, 34, 55, 89, 144].$$

Let (c_x, c_y) denote the coordinates of the centre of the 289×289 image. For each Fibonacci number $F_i \in \mathcal{F}$ we form a square:

$$S_i = \{(x, y) \mid |x - c_x| \leq F_i, |y - c_y| \leq F_i\}. \quad (1)$$

The side length of S_i is $(2F_i + 1)$, and these squares are nested:

$$S_0 \subset S_1 \subset \dots \subset S_8.$$

To obtain non-overlapping regions, we define:

$$R_0 = S_0, \quad R_i = S_i \setminus S_{i-1}, \quad i = 1, \dots, 8. \quad (2)$$

Thus, R_0 is the innermost square around the joint centre, while each R_i with $i > 0$ represents the ring between two consecutive Fibonacci squares. The outermost region R_8 covers the band between $F = 89$ and the image boundary at $F = 144$. Figure 2 shows an example of these regions overlaid on a knee radiograph.

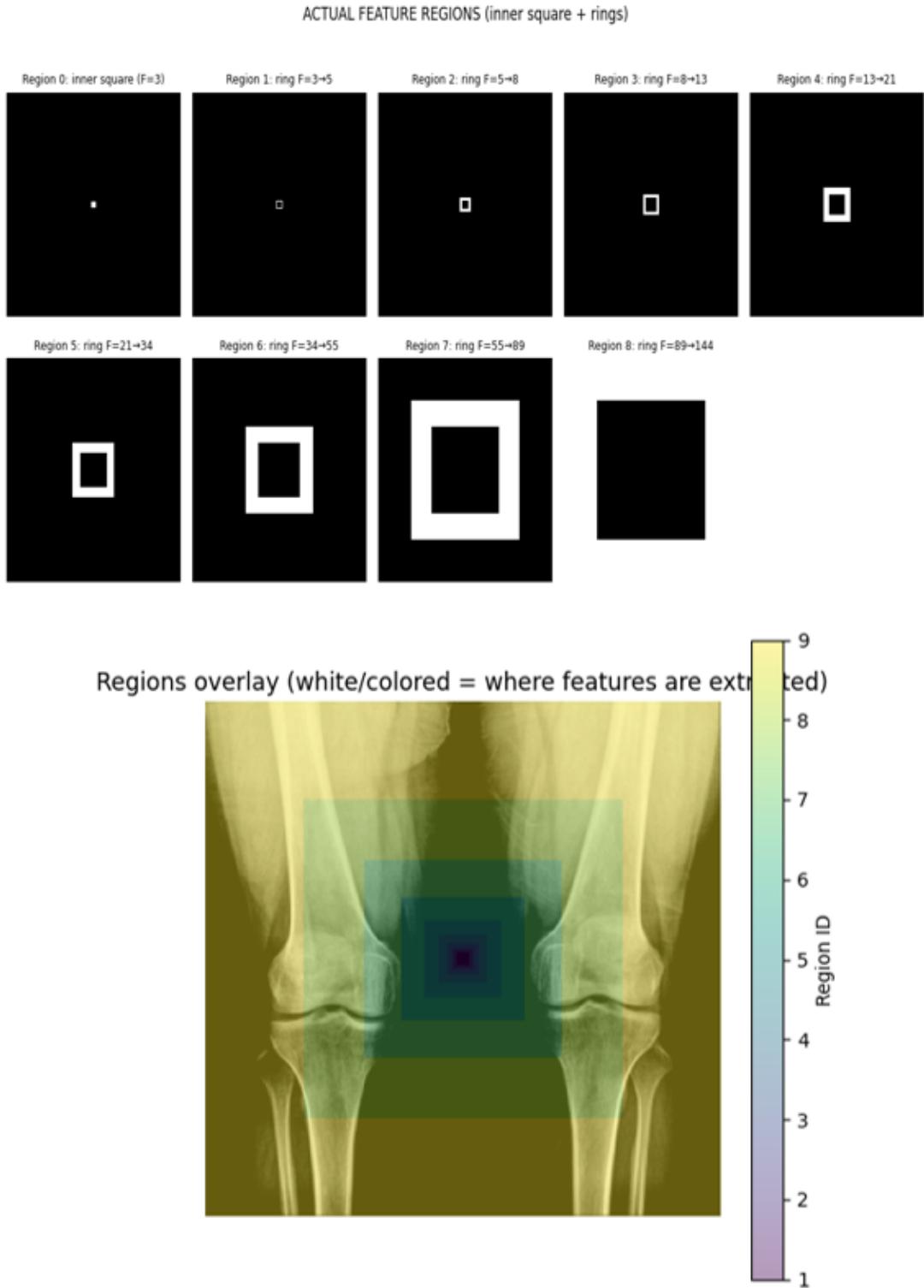


Figure 2: Actual feature-supporting regions R_0 to R_8 based on Fibonacci rings overlaid on knee X-Ray. White/colored pixels represent active feature extraction zones.

6.3 Handcrafted Feature Set

From each region R_i we extract ten features:

1. **Median intensity:** median of pixel intensities in R_i , robust to outliers.
2. **Intensity variance:** variance of intensities, indicating heterogeneity.

3. **GLCM homogeneity:** computed from a gray-level co-occurrence matrix, giving higher contribution to pairs of similar gray levels.
4. **GLRLM long run emphasis:** emphasises longer homogeneous runs in a run-length encoding of the region.
5. **GLSZM zone percentage:** ratio of the number of homogeneous zones to the total number of pixels in R_i .
6. **DCT DC energy:** squared value of the DC coefficient obtained from a 2D Discrete Cosine Transform on the region's bounding box.
7. **Permutation entropy:** a normalised entropy based on the frequency of local ordinal patterns; reflects intensity complexity.
8. **Laplacian variance:** variance of the Laplacian response within R_i , relating to edge richness.
9. **Compactness:** shape descriptor defined as $4\pi A/P^2$, where A is area and P an estimate of the perimeter.
10. **Approximate Lyapunov exponent:** an approximate measure of local chaotic behaviour derived from intensity differences along a path through the region.

Pooling all regions, each image is represented by a 90-dimensional feature vector.

6.4 Implementation Snippet

The following code snippet summarises the region formation process and feature extraction for one image.

```
# =====
# Cell A: Build square_masks and region_masks
# =====

def build_fib_masks(img_size, fib_numbers):
    """
    Returns:
        square_masks: list of full squares for each Fibonacci radius
        region_masks: list of regions used for features:
            region 0 : inner square for fib_numbers[0]
            region i : ring between fib_numbers[i-1] and fib_numbers[i]
    """
    H = W = img_size
    cx = cy = img_size // 2

    square_masks = []
    for F in fib_numbers:
        mask = np.zeros((H, W), dtype=bool)
        top = max(0, cy - F)
        bottom = min(H, cy + F + 1)
        left = max(0, cx - F)
        right = min(W, cx + F + 1)
        mask[top:bottom, left:right] = True
        square_masks.append(mask)

    region_masks = []
    # Region 0: inner-most square
    region_masks.append(square_masks[0])

    # Regions 1... rings between consecutive squares
    for i in range(1, len(square_masks)):
        outer = square_masks[i]
        inner = square_masks[i-1]
        ring = outer & (~inner) # this is a FILLED band, not just 1-px outline
        region_masks.append(ring)

    return square_masks, region_masks

square_masks, region_masks = build_fib_masks(IMG_SIZE, fib_numbers)
print("Squares:", len(square_masks), "Regions:", len(region_masks))
```

```
# =====
# Cell B: Visualization of squares and regions
# =====

import matplotlib.pyplot as plt
num_regions = len(region_masks)
```

```

# 1) SHOW FIBONACCI SQUARES (for understanding only)
fig, axes = plt.subplots(2, (len(square_masks) + 1) // 2, figsize=(12, 6))
axes = axes.ravel()

for i, m in enumerate(square_masks):
    axes[i].imshow(m.astype(int), cmap='gray', interpolation='nearest')
    axes[i].set_title(f"Square {i} (F={fib_numbers[i]})")
    axes[i].axis('off')

for j in range(len(square_masks), len(axes)):
    axes[j].axis('off')

plt.suptitle("Fibonacci SQUARE masks (not used directly for features)")
plt.tight_layout()
plt.show()

# 2) SHOW FEATURE REGIONS = inner square + RINGS
fig, axes = plt.subplots(2, (num_regions + 1) // 2, figsize=(12, 6))
axes = axes.ravel()

for i, m in enumerate(region_masks):
    axes[i].imshow(m.astype(int), cmap='gray', interpolation='nearest')
    if i == 0:
        title = f"Region {i}: inner square (F={fib_numbers[0]})"
    else:
        title = (f"Region {i}: ring F={fib_numbers[i-1]}{fib_numbers[i]}")
    axes[i].set_title(title, fontsize=9)
    axes[i].axis('off')

for j in range(num_regions, len(axes)):
    axes[j].axis('off')

plt.suptitle("ACTUAL FEATURE REGIONS (inner square + rings)", y=1.02)
plt.tight_layout()
plt.show()

# 3) OVERLAY REGIONS ON IMAGE
region_label_map = np.zeros_like(img_gray, dtype=np.int32)
for idx, mask in enumerate(region_masks):
    region_label_map[mask] = idx + 1 # 1..N

plt.figure(figsize=(6, 6))
plt.imshow(img_gray, cmap='gray')
plt.imshow(region_label_map, alpha=0.4) # overlay with transparency
plt.title("Regions overlay (white/colored = where features are extracted)")
plt.axis('off')
plt.colorbar(label='Region ID')
plt.show()

# Optional: print how many pixels in each region
for i, m in enumerate(region_masks):
    print(f"Region {i} pixel count:", np.count_nonzero(m))

```

```

# =====
# Cell 13: Extract all 10 features for one region
# =====

def extract_features_for_region(img_gray, mask, levels=16):
    roi_pixels = get_roi_pixels(img_gray, mask)

    f1 = feature_median_intensity(roi_pixels)
    f2 = feature_variance(roi_pixels)
    f3 = feature_glcm_homogeneity(img_gray, mask, levels=levels)
    f4 = feature_glrlm_lre(img_gray, mask, levels=levels)
    f5 = feature_glszm_zone_percentage(img_gray, mask, levels=levels)
    f6 = feature_dct_dc_energy(img_gray, mask)
    f7 = feature_permutation_entropy(img_gray, mask, m=3, tau=1)
    f8 = feature_laplacian_variance(img_gray, mask)
    f9 = feature_compactness(mask)
    f10 = feature_lyapunov_exponent(img_gray, mask)

    return [f1, f2, f3, f4, f5, f6, f7, f8, f9, f10]

```

The same logic is applied to all images, and the resulting feature vectors are stored in a CSV file along with their class labels.

7 Experimental Setup

7.1 Models

Ten classifiers are considered:

1. Support Vector Machine (SVM) with linear kernel,
2. SVM with polynomial kernel,
3. SVM with RBF kernel,
4. SVM with sigmoid kernel,
5. K-Nearest Neighbors (KNN),
6. Random Forest,
7. Logistic Regression,
8. Decision Tree,
9. Gradient Boosting,
10. Gaussian Naive Bayes.

7.2 Train–Test Splits

Four different train–test configurations are investigated:

- 60/40 split,
- 70/30 split,
- 80/20 split,
- 90/10 split.

In each case, a stratified split is used to maintain class proportions in both training and test sets. Features are standardised using `StandardScaler`, fitted on the training set and applied to both training and test data.

7.3 Evaluation Metrics

For each model and split we compute:

- overall accuracy,
- macro-averaged precision,
- macro-averaged recall,
- macro-averaged F1-score,
- confusion matrix,
- one-versus-rest ROC curves and macro-averaged ROC-AUC.

8 Results

8.1 Performance Tables for Different Splits

This subsection provides four summary tables, one for each train–test split. Each table reports model accuracy together with per-class precision, recall and F1-score.

(a) 60/40 Split

Table 2: Model-wise per-class performance for 60/40 train–test split.

Sr. No.	Model	Accuracy	Class	Precision	Recall	F1-score
1	Linear SVM	0.60	Normal	0.64	0.64	0.64
			Osteopenia	0.63	0.50	0.56
			Osteoporosis	0.55	0.62	0.58
2	SVM Sigmoid Kernel	0.55	Normal	0.54	0.66	0.59
			Osteopenia	0.60	0.45	0.51
			Osteoporosis	0.55	0.53	0.54
3	SVM Polynomial Kernel	0.56	Normal	0.60	0.76	0.67
			Osteopenia	0.50	0.47	0.48
			Osteoporosis	0.55	0.43	0.48
4	SVM RBF Kernel	0.61	Normal	0.65	0.60	0.62
			Osteopenia	0.68	0.61	0.64
			Osteoporosis	0.55	0.62	0.58
5	K-neighbors classifiers	0.66	Normal	0.68	0.70	0.69
			Osteopenia	0.66	0.69	0.68
			Osteoporosis	0.64	0.60	0.62
6	RandomForest	0.63	Normal	0.74	0.67	0.70
			Osteopenia	0.54	0.58	0.56
			Osteoporosis	0.60	0.62	0.61
7	Logistic Regression	0.59	Normal	0.62	0.67	0.64
			Osteopenia	0.58	0.52	0.55
			Osteoporosis	0.56	0.56	0.56
8	Decision Tree	0.52	Normal	0.57	0.62	0.60
			Osteopenia	0.48	0.50	0.49
			Osteoporosis	0.48	0.43	0.46
9	Gradient Boosting	0.64	Normal	0.74	0.68	0.71
			Osteopenia	0.57	0.56	0.57
			Osteoporosis	0.60	0.65	0.62
10	Gaussian Naive Bayes	0.49	Normal	0.42	0.88	0.57
			Osteopenia	0.67	0.66	0.67
			Osteoporosis	0.75	0.03	0.06

(b) 70/30 Split

Table 3: Model-wise per-class performance for 70/30 train–test split.

Sr. No.	Model	Accuracy	Class	Precision	Recall	F1-score
1	Linear SVM	0.60	Normal	0.59	0.68	0.63
			Osteopenia	0.66	0.68	0.67
			Osteoporosis	0.56	0.47	0.51
2	SVM Sigmoid Kernel	0.57	Normal	0.54	0.65	0.59
			Osteopenia	0.62	0.54	0.58
			Osteoporosis	0.56	0.51	0.53
3	SVM Polynomial Kernel	0.62	Normal	0.64	0.71	0.67
			Osteopenia	0.66	0.46	0.54
			Osteoporosis	0.59	0.64	0.62
4	SVM RBF Kernel	0.66	Normal	0.76	0.56	0.64
			Osteopenia	0.71	0.65	0.68
			Osteoporosis	0.59	0.75	0.66
5	K-neighbors classifiers	0.63	Normal	0.65	0.64	0.64
			Osteopenia	0.68	0.65	0.67
			Osteoporosis	0.58	0.60	0.59
6	RandomForest	0.64	Normal	0.75	0.67	0.70
			Osteopenia	0.54	0.61	0.57
			Osteoporosis	0.62	0.63	0.63
7	Logistic Regression	0.58	Normal	0.66	0.65	0.66
			Osteopenia	0.56	0.43	0.49
			Osteoporosis	0.52	0.60	0.56
8	Decision Tree	0.57	Normal	0.64	0.70	0.67
			Osteopenia	0.50	0.52	0.51
			Osteoporosis	0.55	0.49	0.52
9	Gradient Boosting	0.67	Normal	0.77	0.67	0.72
			Osteopenia	0.62	0.63	0.62
			Osteoporosis	0.63	0.70	0.66
10	Gaussian Naive Bayes	0.49	Normal	0.42	0.86	0.56
			Osteopenia	0.68	0.65	0.67
			Osteoporosis	0.80	0.05	0.10

(c) 80/20 Split

Table 4: Model-wise per-class performance for 80/20 train–test split.

Sr. No.	Model	Accuracy	Class	Precision	Recall	F1-score
1	Linear SVM	0.60	Normal	0.59	0.68	0.63
			Osteopenia	0.66	0.68	0.67
			Osteoporosis	0.56	0.47	0.51
2	SVM Sigmoid Kernel	0.59	Normal	0.59	0.66	0.62
			Osteopenia	0.59	0.61	0.60
			Osteoporosis	0.58	0.51	0.54
3	SVM Polynomial Kernel	0.56	Normal	0.55	0.66	0.60
			Osteopenia	0.67	0.52	0.58
			Osteoporosis	0.51	0.49	0.50
4	SVM RBF Kernel	0.65	Normal	0.70	0.59	0.64
			Osteopenia	0.68	0.68	0.68
			Osteoporosis	0.59	0.67	0.63
5	K-neighbors classifiers	0.62	Normal	0.66	0.61	0.64
			Osteopenia	0.65	0.65	0.65
			Osteoporosis	0.58	0.61	0.59
6	RandomForest	0.63	Normal	0.79	0.68	0.73
			Osteopenia	0.51	0.65	0.57
			Osteoporosis	0.60	0.57	0.58
7	Logistic Regression	0.57	Normal	0.64	0.64	0.64
			Osteopenia	0.56	0.58	0.57
			Osteoporosis	0.52	0.51	0.52
8	Decision Tree	0.52	Normal	0.62	0.66	0.64
			Osteopenia	0.45	0.45	0.45
			Osteoporosis	0.48	0.45	0.46
9	Gradient Boosting	0.65	Normal	0.79	0.61	0.69
			Osteopenia	0.57	0.65	0.61
			Osteoporosis	0.62	0.69	0.65
10	Gaussian Naive Bayes	0.47	Normal	0.41	0.86	0.55
			Osteopenia	0.62	0.58	0.60
			Osteoporosis	1.00	0.04	0.08

(d) 90/10 Split

Table 5: Model-wise per-class performance for 90/10 train–test split.

Sr. No.	Model	Accuracy	Class	Precision	Recall	F1-score
1	Linear SVM	0.63	Normal	0.58	0.50	0.54
			Osteopenia	0.88	0.88	0.88
			Osteoporosis	0.52	0.58	0.55
2	SVM Sigmoid Kernel	0.65	Normal	0.59	0.59	0.59
			Osteopenia	0.73	0.69	0.71
			Osteoporosis	0.64	0.67	0.65
3	SVM Polynomial Kernel	0.56	Normal	0.52	0.55	0.53
			Osteopenia	0.77	0.62	0.69
			Osteoporosis	0.50	0.54	0.52
4	SVM RBF Kernel	0.71	Normal	0.79	0.50	0.61
			Osteopenia	0.82	0.88	0.85
			Osteoporosis	0.61	0.79	0.69
5	K-neighbors classifiers	0.63	Normal	0.56	0.64	0.60
			Osteopenia	0.72	0.81	0.76
			Osteoporosis	0.63	0.50	0.56
6	RandomForest	0.68	Normal	0.78	0.64	0.70
			Osteopenia	0.62	0.81	0.70
			Osteoporosis	0.65	0.62	0.64
7	Logistic Regression	0.58	Normal	0.57	0.55	0.56
			Osteopenia	0.79	0.69	0.73
			Osteoporosis	0.48	0.54	0.51
8	Decision Tree	0.56	Normal	0.62	0.59	0.60
			Osteopenia	0.55	0.69	0.61
			Osteoporosis	0.52	0.46	0.49
9	Gradient Boosting	0.61	Normal	0.60	0.55	0.57
			Osteopenia	0.67	0.75	0.71
			Osteoporosis	0.58	0.58	0.58
10	Gaussian Naive Bayes	0.56	Normal	0.49	0.86	0.62
			Osteopenia	0.75	0.75	0.75
			Osteoporosis	0.57	0.17	0.26

8.2 Confusion Matrices for Different Splits

(a) 60/40 Split

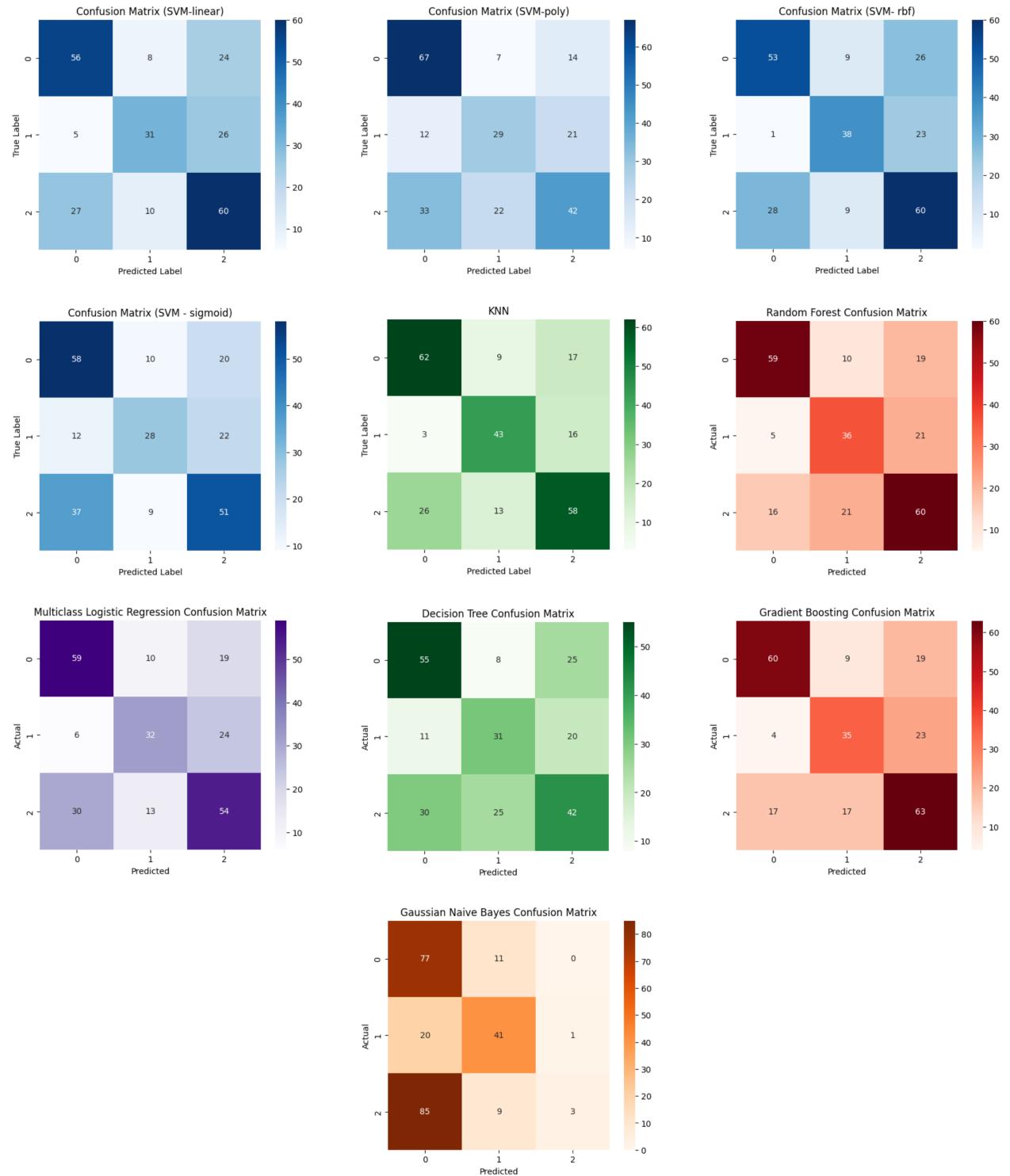


Figure 3: Confusion matrices : 60/40 split

(b) 70/30 Split

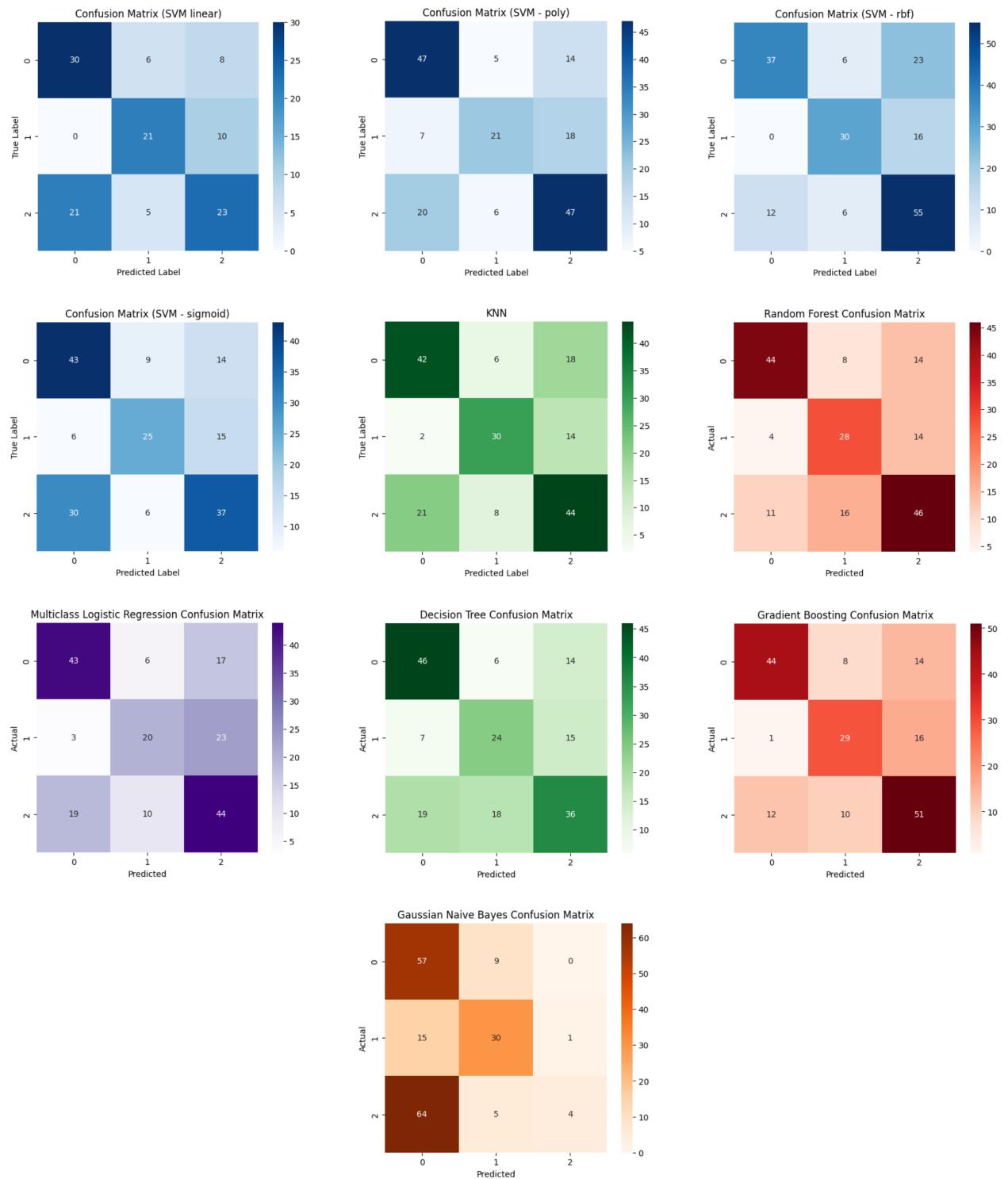


Figure 4: Confusion matrices : 70/30 split

(c) 80/20 Split

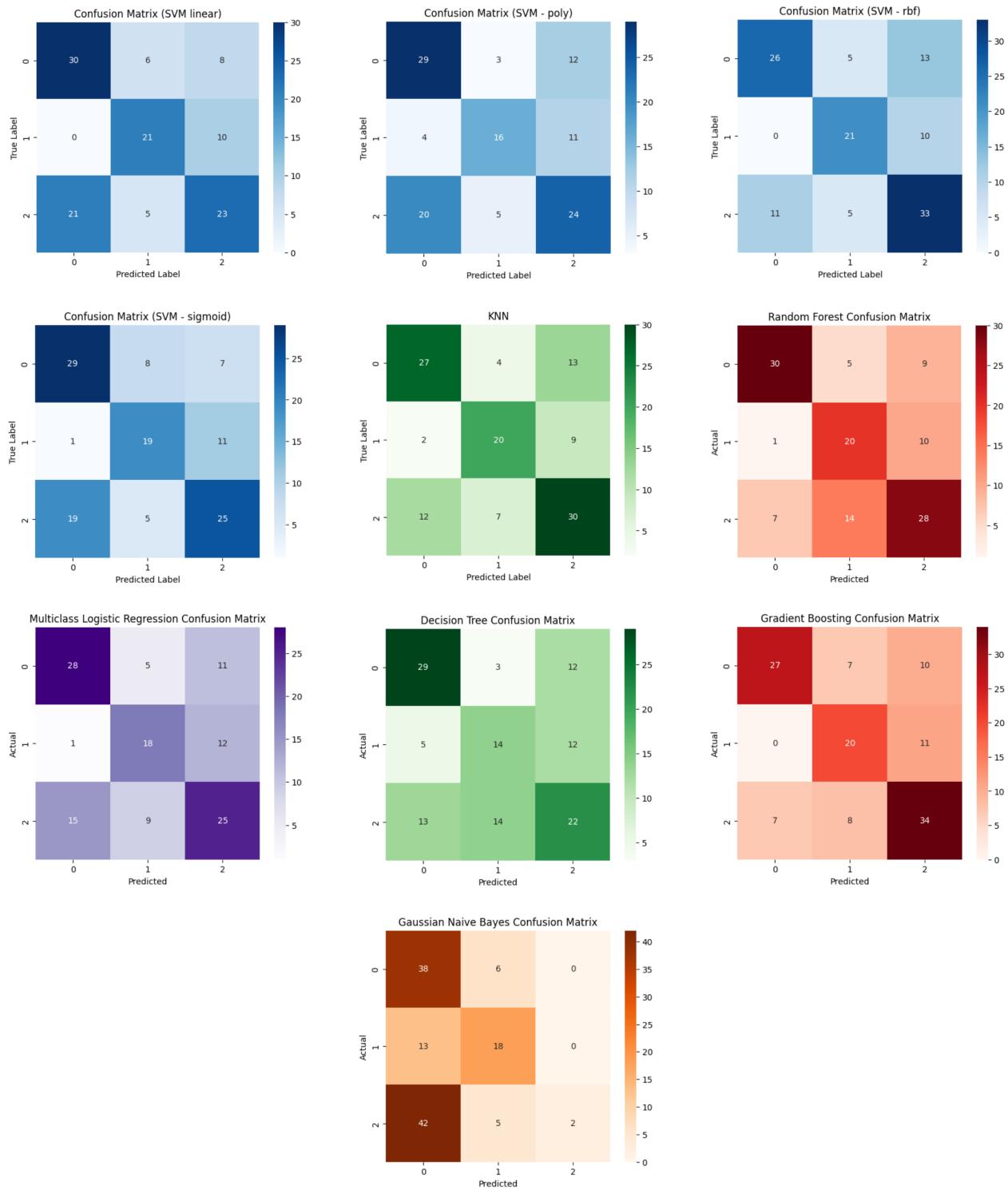


Figure 5: Confusion matrices : 80/20 split

(d) 90/10 Split

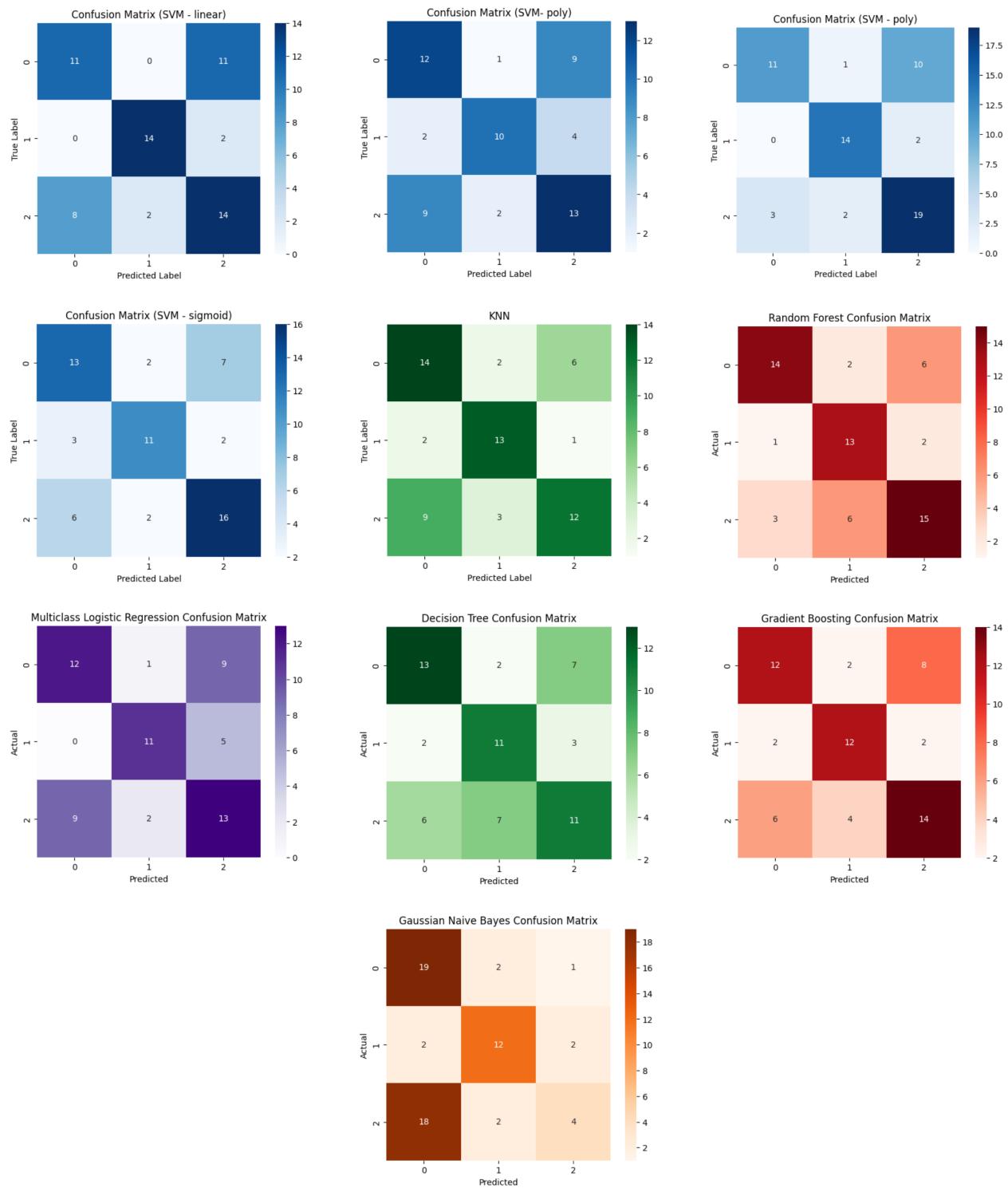


Figure 6: Confusion matrices : 90/10 split

8.3 ROC Curves for Different Splits

(a) 60/40 Split

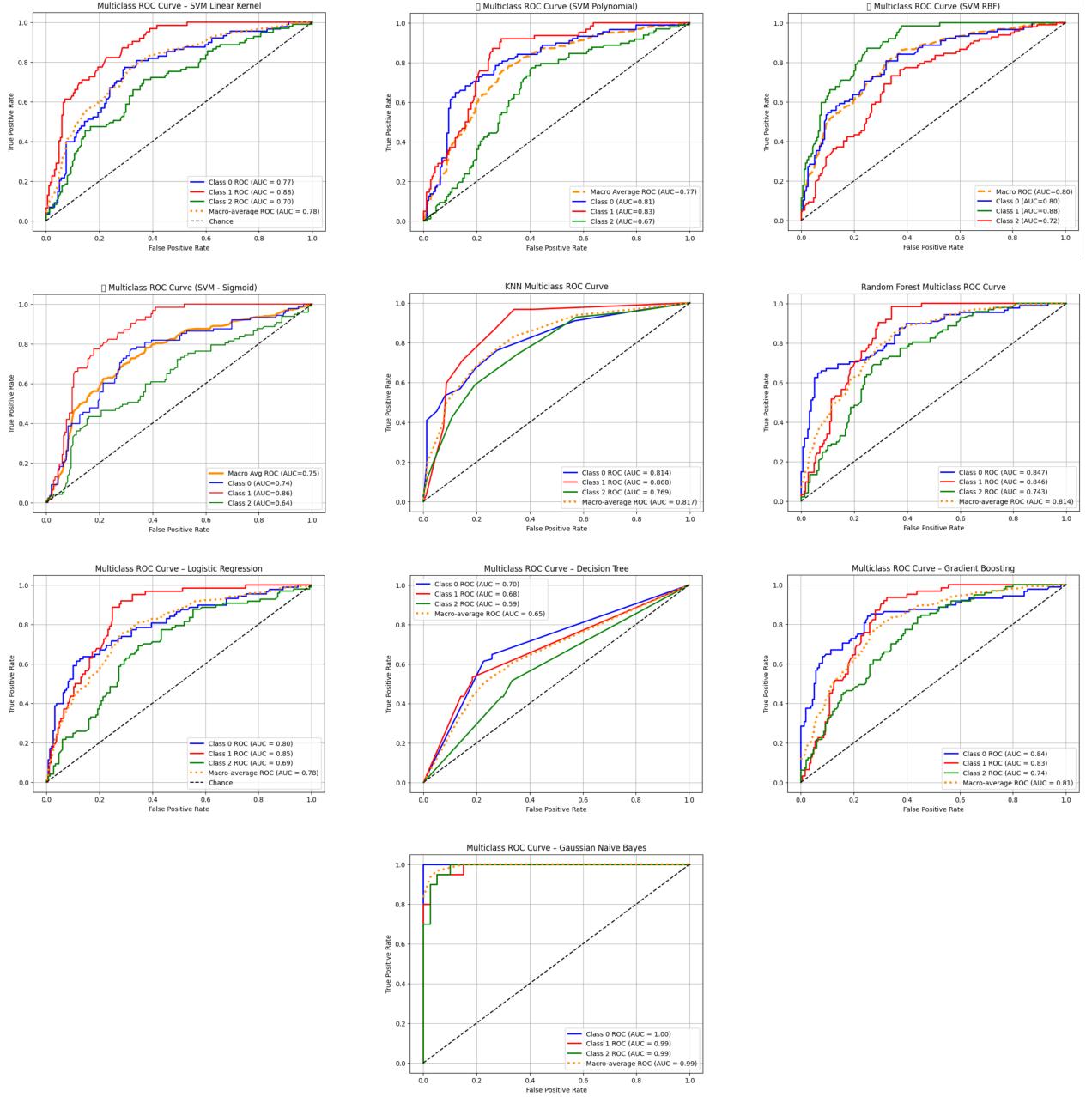


Figure 7: ROC curves for all models on the 60/40 split.

(b) 70/30 Split

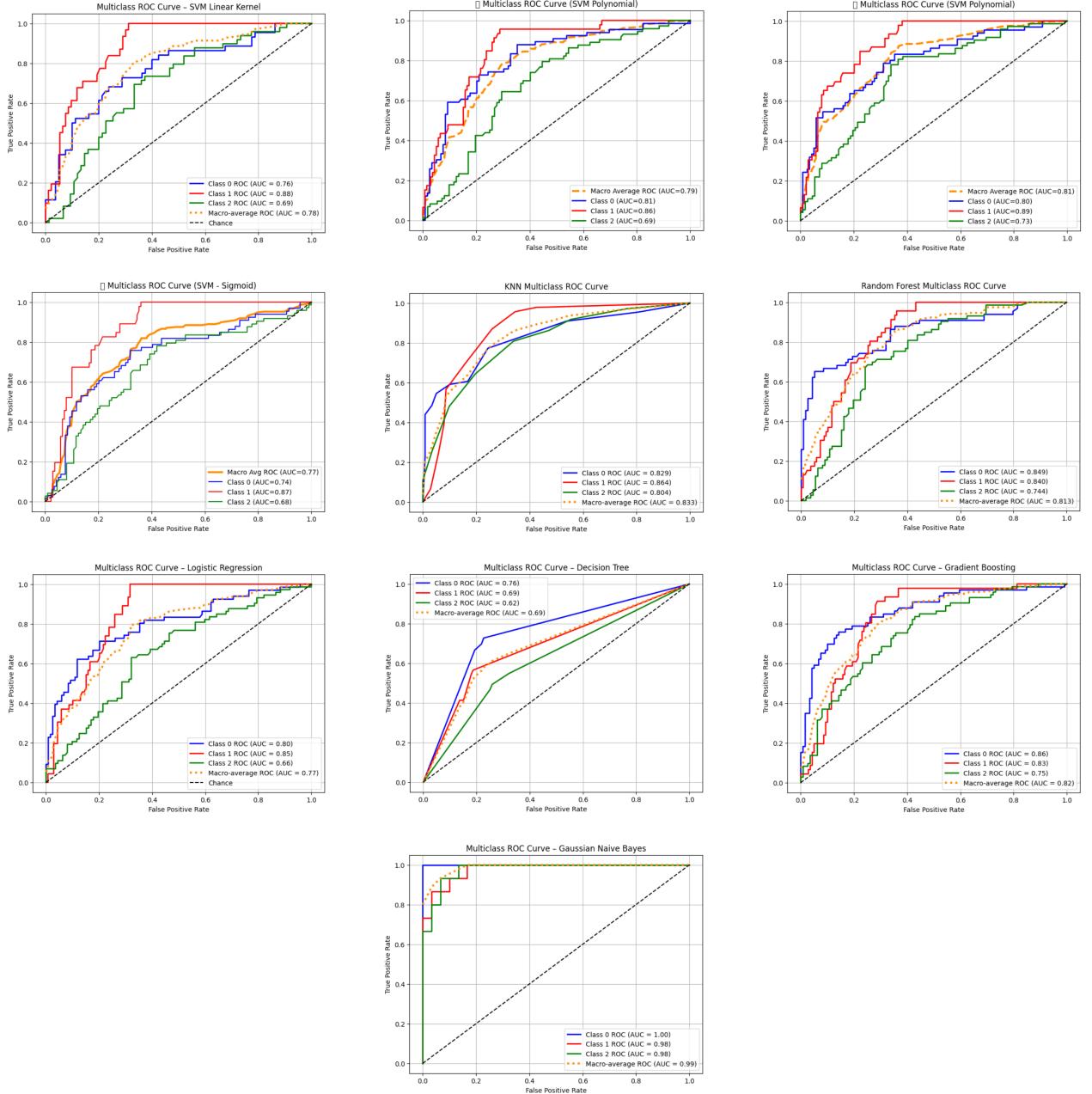


Figure 8: ROC curves for all models on the 70/30 split.

(c) 80/20 Split

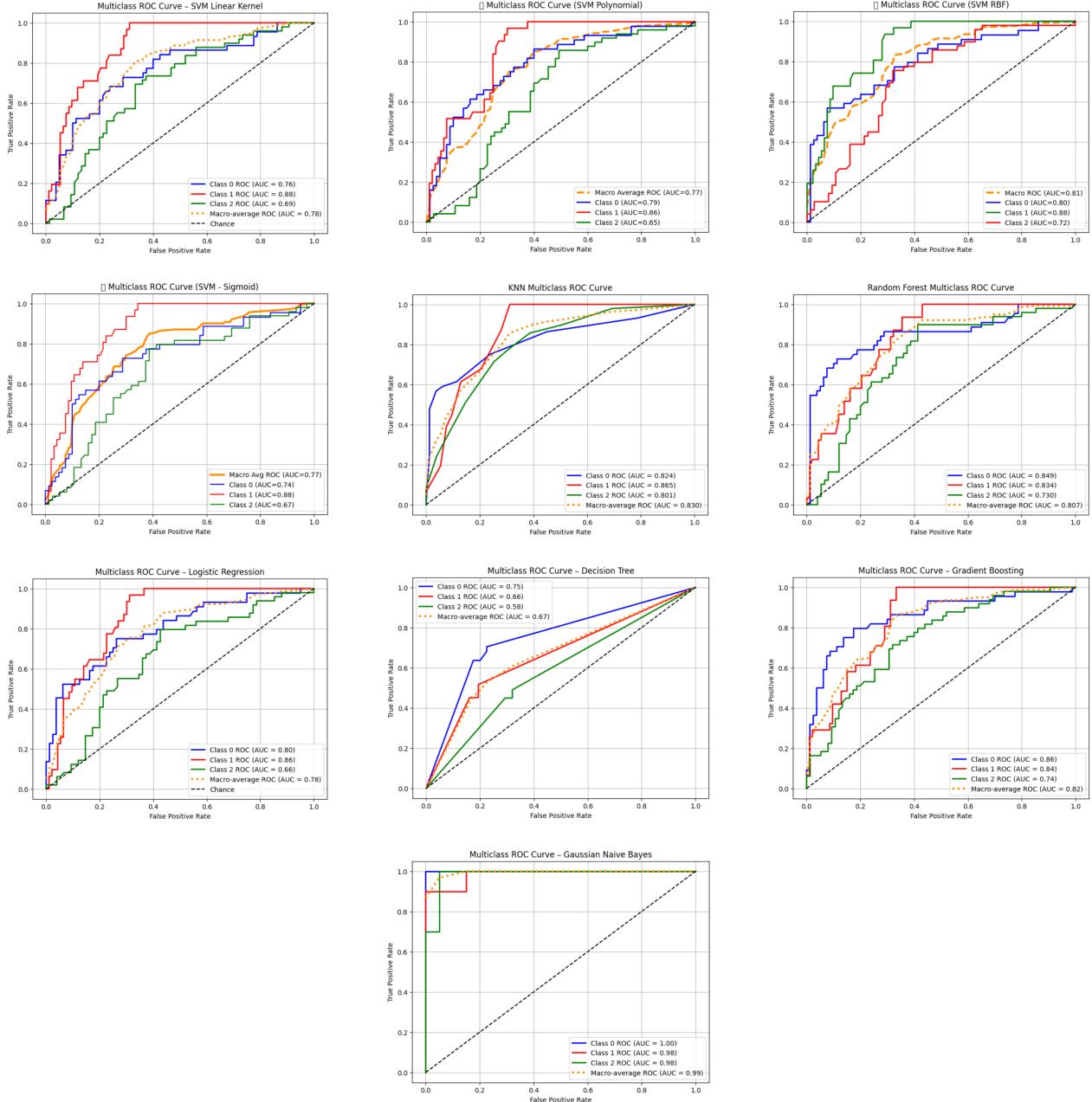


Figure 9: ROC curves for all models on the 80/20 split.

(d) 90/10 Split

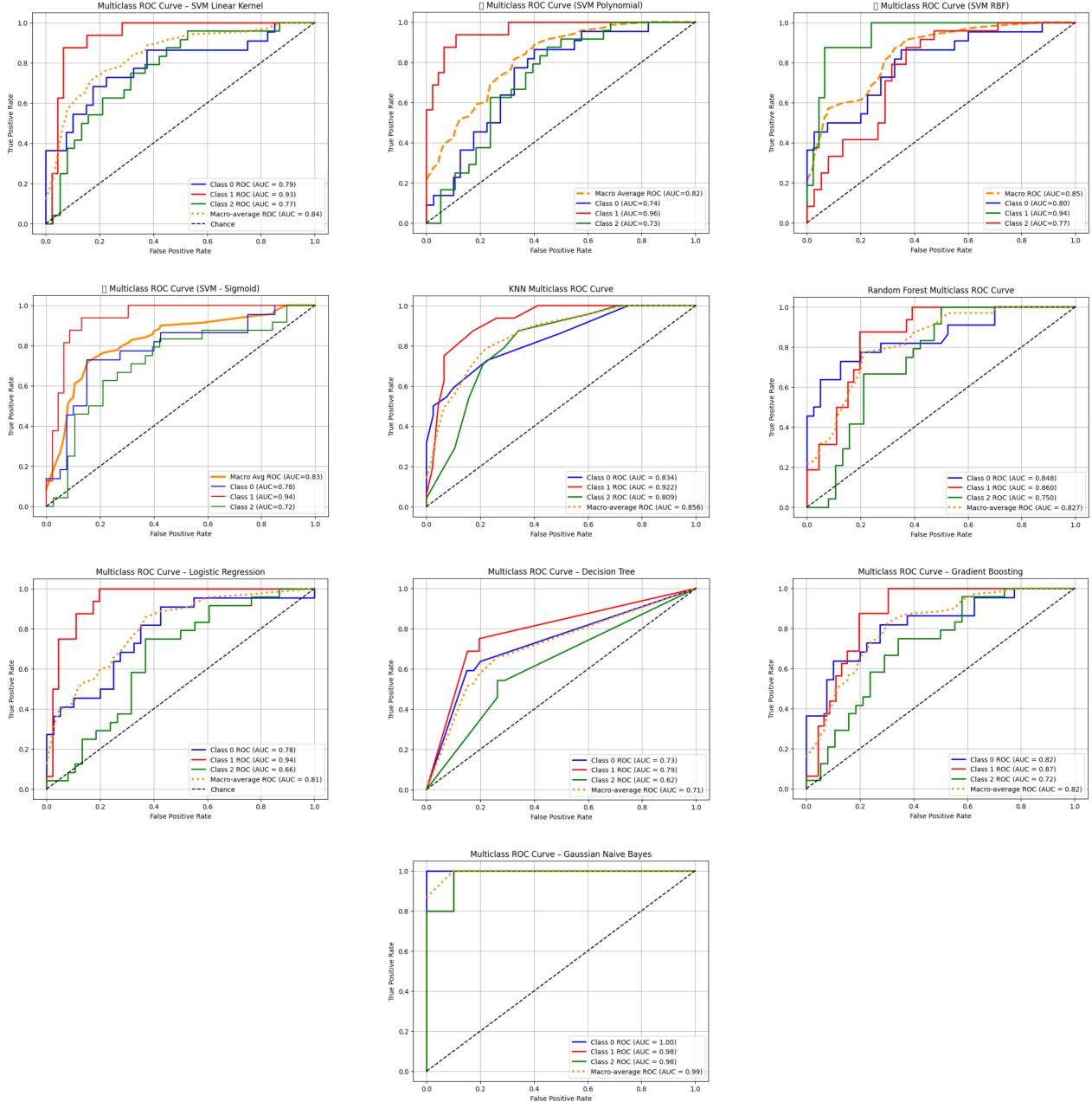


Figure 10: ROC curves for all models on the 90/10 split.

9 Discussion

The experimental outcomes across varying train–test splits validate the utility of the proposed feature engineering approach. In the 60/40 and 70/30 splits, the majority of classifiers maintained an accuracy range of 0.60–0.67. This suggests that handcrafted descriptors, based on Fibonacci-guided regions, effectively isolate discriminative patterns in knee radiographs without necessitating deep neural network architectures or massive datasets.

Ensemble methods, specifically Random Forest and Gradient Boosting, outperformed single-learner models. By aggregating multiple weak learners, these classifiers better accounted for the textural variations across different anatomical zones. Crucially, they maintained a balance between precision and recall, a requirement for clinical screening where missing an osteopenia diagnosis delays preventive care.

Support Vector Machines (SVM) utilizing RBF and linear kernels also yielded distinct separation boundaries, confirming that the extracted feature space is linearly separable to a significant degree. Conversely, the polynomial and sigmoid kernels displayed high variance depending on the split ratio, implying a susceptibility to overfitting when training data is sparse.

Gaussian Naive Bayes recorded the poorest performance. This is likely due to the independence assumption, which holds little validity in radiographic imaging where spatial intensity and texture between adjacent regions are highly correlated. Logistic Regression and Decision Trees similarly struggled to model the non-linear complexities of the bone structure, resulting in reduced recall.

While the 90/10 split produced the highest quantitative accuracy (Gradient Boosting reaching 0.71), the restricted test size compromises the statistical reliability of this metric. Consequently, the results from the 70/30 and 80/20 splits offer a more realistic approximation of performance in a deployment setting.

Analysis of the confusion matrices highlights a persistent difficulty in classifying osteopenia. As a transitional state between healthy bone density and osteoporosis, osteopenia lacks the distinct textural extremes of the other classes. Consequently, it acts as a confusion overlap, leading to higher misclassification rates. Improving sensitivity for this intermediate class remains a primary objective for future work.

Conclusion

This work presented a complete pipeline for osteoporosis screening using standard knee X-ray images, based on handcrafted feature engineering rooted in the Fibonacci sequence. By partitioning each image into concentric square regions guided by Fibonacci radii, we ensured that both central trabecular bone and outer cortical regions contributed to the descriptive power of the feature vector. Ten carefully selected texture, intensity and complexity descriptors were extracted per region, resulting in a compact and interpretable 90-dimensional representation.

The experimental results highlight that classical machine learning models, particularly Random Forest, Gradient Boosting and SVM with RBF kernel, achieve stable accuracy and balanced precision–recall performance across multiple train–test splits. These observations confirm that the geometric design of the regions successfully captures meaningful variations associated with bone health degradation. Importantly, the method operates effectively even on a modest dataset and without specialised GPU hardware, making it suitable for deployment in resource-limited clinical environments.

While the findings are promising, the study also revealed challenges such as the misclassification of osteopenia cases due to overlapping radiographic characteristics. Moreover, variations in image positioning and exposure can influence region-wise feature values. Despite these limitations, the handcrafted approach offers a high level of interpretability by which clinicians can trace back which image regions influence the diagnostic decision.

In summary, Fibonacci-based region engineering shows strong potential for real-world computer-aided diagnosis of osteoporosis. With future improvements in dataset size, feature diversity and alignment strategies, this method could evolve into a reliable and accessible screening aid that complements conventional densitometry in early detection and medical decision-making.