AI-Driven Multi-Modal Data Fusion for Early Cancer Diagnosis

Abstract

Background:
Clear-cell renal cell carcinoma (ccRCC) is the most prevalent and clinically aggressive subtype of kidney cancer, characterized by substantial heterogeneity and wide variability in patient outcomes. Accurate early survival prediction is essential for personalized treatment planning and timely clinical intervention.

Methods:
We utilized the MMIST-ccRCC dataset of 618 patients containing harmonized clinical and genomic features, including AJCC pathologic staging, tumor grade, prior cancer history, and key mutations (VHL, PBRM1, TTN). The target outcome was 12-month survival status (Alive vs. Deceased), a highly imbalanced outcome (~12% deceased). To address this imbalance, RandomOverSampler was applied to the training split, and class-weighting was used to preserve fidelity to the original distribution. Multiple supervised machine learning models were evaluated—logistic regression, random forest, and XGBoost—alongside an improved feed-forward neural network (FFNN) optimized for tabular biomedical data. The FFNN architecture incorporated deeper layers, dropout regularization, early stopping, and decision-threshold tuning. Because the FFNN outputs survival probability (Alive), probability inversion was applied to compute ROC-AUC for the clinically critical Deceased class.

Results:
XGBoost demonstrated the strongest performance among classical models, achieving an ROC-AUC of 0.90. The improved FFNN achieved robust generalization with an ROC-AUC of 0.86 and substantially improved sensitivity for high-risk (Deceased) patients. Recall for the Deceased class increased from 0.46 (baseline) to 0.73 after applying oversampling, class-weighting, and threshold adjustment. Logistic regression, while interpretable, is underperformed due to the nonlinear structure of the dataset. Across models, the most influential predictors included tumor stage, tumor grade, prior cancer history, and mutations in VHL and PBRM1.

Table of Contents

1. Introduction

Clear-cell renal cell carcinoma (ccRCC) is the most common and clinically aggressive subtype of kidney cancer, accounting for most renal malignancies. It is characterized by substantial biological heterogeneity and wide variability in patient outcomes, making early risk stratification critical for guiding clinical decision-making and optimizing patient management. Accurate prediction of short-term survival—particularly within the first 12 months after diagnosis—supports treatment planning, follow-up scheduling, and timely therapeutic intervention.

Machine learning (ML) offers powerful tools for leveraging structured biomedical information to improve outcome prediction. In ccRCC, clinical indicators such as AJCC pathologic staging, tumor grade, and prior cancer history, as well as recurrent genomic mutations including VHL, PBRM1, and TTN, provide important prognostic signals. Integrating these structured clinical and genomic features into supervised learning models enables the development of data-driven prognostic frameworks that can complement or outperform traditional clinical scoring systems.

A major challenge in survival prediction for ccRCC is the severe imbalance in outcome labels: only a small minority of patients are deceased at 12 months relative to the number who remain alive. Such imbalance can bias models toward overpredicting survival, leading to high accuracy but poor detection of high-risk patients. Addressing this issue requires targeted imbalance-aware strategies such as oversampling, class weighting, and threshold adjustment to ensure meaningful sensitivity for the minority (Deceased) class.

In this study, we develop and evaluate multiple supervised learning approaches—including logistic regression, random forest, XGBoost, and an improved feed-forward neural network (FFNN)—to predict 12-month survival using structured clinical and genomic features from 618 ccRCC patients. We systematically investigate the impact of imbalance handling and neural-network optimization on model performance, with the goal of establishing a strong baseline for clinical outcome prediction in ccRCC and enhancing early identification of high-risk patients.

Objective:

Develop and evaluate machine learning and deep learning models to predict 12-month survival using clinical and genomic features from the MMIST-ccRCC dataset.

Goal:

Maximize sensitivity (recall) for high-risk patients while maintaining strong overall discrimination using metrics such as ROC-AUC, F1-score, and confusion matrices.
This project establishes a strong baseline for ccRCC prognosis using only structured data. In future work, the framework will be extended to full multimodal fusion using CT, MRI, and WSI features available in MMIST-ccRCC, enabling more comprehensive survival prediction through radiologic–genomic–clinical integration.

2. Dataset and Methodology

2.1 Dataset Overview

The **MMIST-ccRCC** dataset is a comprehensive, multi-modal dataset curated from the Clinical Proteomic Tumor Analysis Consortium (CPTAC) and The Cancer Genome Atlas (TCGA). It contains records for **618 patients** diagnosed with clear-cell renal cell carcinoma (ccRCC), integrating five data modalities:

1. **Clinical data** – Demographic information, tumor staging, and laboratory markers.
2. **Genomic data** – Mutation status of key genes associated with renal carcinogenesis, including VHL, PBRM1, BAP1, and TTN.
3. **Structured metadata** – Patient identifiers and other descriptive variables.

For this study, we focused on **clinical and genomic features**, which are available for all patients and provide robust information for early survival prediction.

2.2 Clinical Features

The clinical dataset includes demographic and tumor-related variables:

Feature	Description
Age at Diagnosis (age_diag)	Patient age in years; higher age often correlates with worse prognosis.
Gender (gender)	Biological sex; may influence disease progression and treatment response.
Tumor Grade (grade)	Histological grading of tumor cells; higher grade indicates more aggressive tumor.
AJCC Pathologic Stage	Includes pT (primary tumor), pN (lymph nodes), pM (metastasis), and overall stage; higher stage indicates worse prognosis.
Race	One-hot encoded categories: Asian, Black/African American, Hispanic/Latino, White, Other.

2.3 Genomic Features

Genomic features capture mutation status in genes relevant to ccRCC:

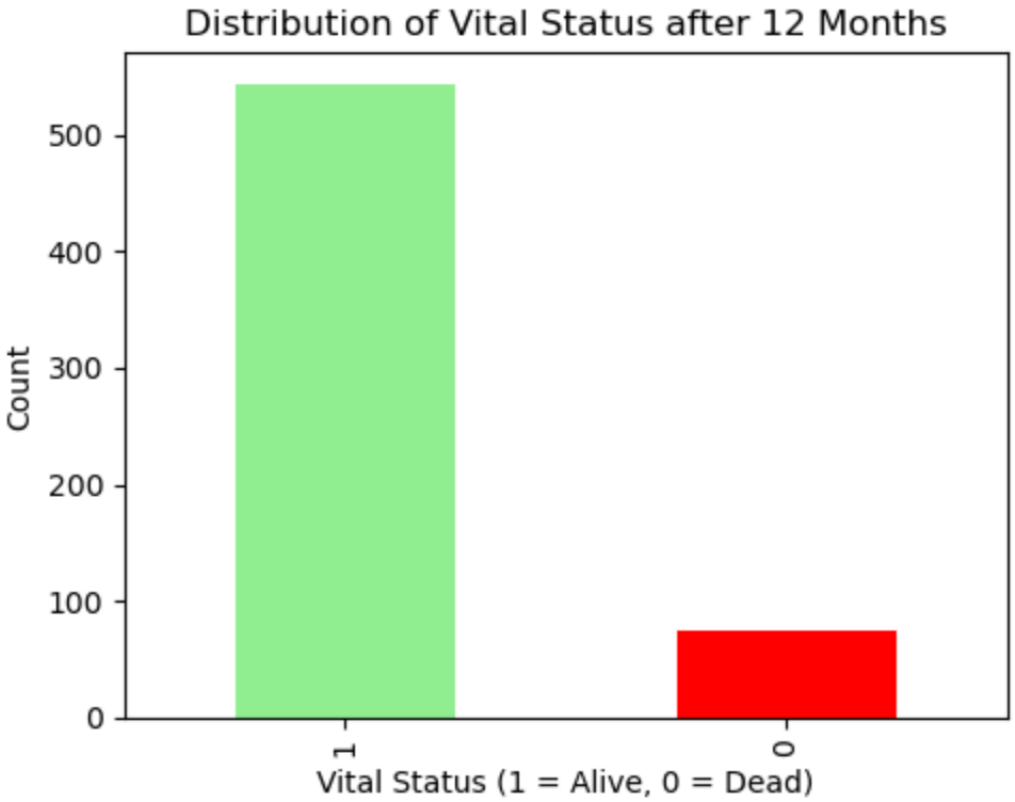
Gene	Description
VHL	Tumor suppressor gene; mutations common in ccRCC.
PBRM1	Chromatin remodeling gene; influences tumor aggressiveness.
TTN	Large gene; mutations may correlate with tumor mutational burden.

The target variable is vital_status_12, defined as:

- 1: Deceased within 12 months of diagnosis
- 0: Alive after 12 months

This defines a binary survival classification task.

Vital Status	Count (%)	Meaning
1 (Alive)	**87.86%**	Patients who survived after 12 months
0 (Deceased)	**12.14%**	Patients who did not survive within 12 months



2.4 Data Preprocessing and Feature Engineering

A comprehensive preprocessing and feature-engineering pipeline was applied to prepare the structured clinical and genomic features for both classical machine learning models and the improved neural network. The steps below reflect the implementation used in the notebook.

1. Handling Missing Values

- **Continuous variables** (e.g., age at diagnosis, AJCC staging variables) were imputed using the **median**, reducing sensitivity to outliers.
- **Categorical variables** (such as gender and cancer history) were imputed with the **mode** to maintain consistent representation.
- A **missingness indicator** (cancer_history_missing) was created before imputing the cancer_history column, allowing the model to capture patterns associated with missing clinical history.

2. Feature Engineering

Feature engineering was performed to enhance the predictive signal by transforming raw clinical and genomic attributes into clinically meaningful derived features.

a. Race One-Hot Encoding (Already Present in Dataset)

The dataset already contained one-hot encoded race variables:

race_Asian
race_Black or African American
race_Hispanic or Latino
race_White
race_other

These dummy variables were preserved in the final feature set.
(They were generated earlier in the pipeline before this preprocessing block)

b. Age Binning (New Feature)

To reduce noise and capture non-linear age effects, the continuous age_diag variable was transformed into ordinal bins:

- 0 = Age < 40
- 1 = Age 40–60
- 2 = Age 60–80
- 3 = Age > 80

This helps stabilize neural network learning and improves generalization.

c. High-Risk Tumor Indicators (New Binary Features)

Several clinically meaningful binary risk indicators were engineered:

- **high_grade:**
  1 if tumor grade ≥ 3, else 0
- **advanced_stage:**
  1 if AJCC pathologic stage ≥ 3
- **any_metastasis:**
  Combines pathologic (ajcc_path_metastasis_pm) and clinical metastasis (ajcc_clin_metastasis_cm)
- **any_nodes:**
  1 if any lymph node involvement (ajcc_path_nodes_pn > 0)

These features capture tumor aggressiveness directly relevant to survival prediction.

d. Genomic Interaction Features (New Features)

Biologically meaningful mutation combinations were added:

- **VHL_PBMR1_both:**
  1 if both VHL and PBRM1 mutations are present
- **any_mutation:**
  Indicates whether any of the three key mutations (VHL, PBRM1, TTN) are present

These interaction terms capture combined genomic effects that may influence prognosis.

3. Feature Scaling

Continuous numerical features were standardized using:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Scaling was essential particularly for neural networks, improving convergence during training.

4. Target Encoding

The survival outcome (vital_status_12) was encoded as:

- **0 = Deceased (Positive Class)**
- **1 = Alive (Negative Class)**

Deceased patients were intentionally set as the **positive clinical class**, reflecting the clinical priority of maximizing recall for high-risk patients.

5. Class Imbalance Mitigation

The dataset exhibited strong imbalance:

- **Alive: 434**
- **Dead: 60**

To address this, two strategies were implemented:

a. Class Weights

Computed from the original y_train:

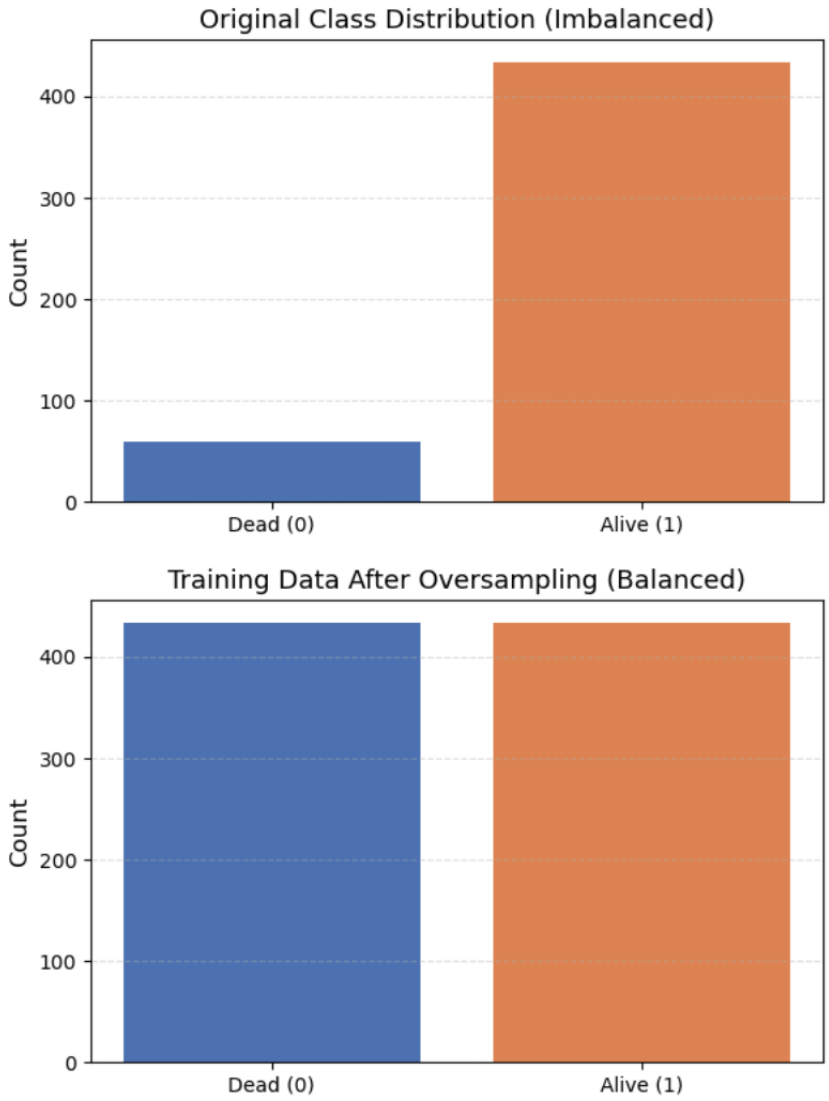
{0: 4.12, 1: 0.57}

This penalized the model more heavily for misclassifying deceased (rare) patients.

b. Random Oversampling (Training Split Only)

Applied to **X_train** (not to the test set):

State	Dead	Alive
Before oversampling	60	434
After oversampling	434	434



Oversampling ensured the model learned meaningful patterns for the minority class.

6. Balanced Train–Validation Split

After oversampling, the dataset was split again:

- **Training:** 347 Dead / 347 Alive
- **Validation:** 87 Dead / 87 Alive

This ensured unbiased validation and stable neural network optimization.

7. Probability Inversion for Neural Network

Because the neural network outputs **P(Alive)**, probabilities were inverted:

P(Dead) = 1 − P(Alive)

This was required for:

- Correct **ROC-AUC** computation (Dead = positive class)
- Accurate threshold tuning for high-risk detection

3. Model Development

This study implemented three classical machine learning algorithms and an improved feed-forward neural network (FFNN) to predict 12-month survival in ccRCC patients. All models were trained using the structured clinical and genomic dataset after standardized preprocessing, imbalance handling, and feature scaling.

3.1 Classical Machine Learning Models

Three classical supervised learning models were evaluated:

- **Logistic Regression (class_weight='balanced')**
- **Random Forest (class_weight='balanced')**
- **XGBoost (scale_pos_weight)**

Training Workflow (Matching Code):

1. **Train–test split**
   X_train, X_test, y_train, y_test = train_test_split(..., stratify=y)
2. **Scaling numeric features using StandardScaler**
   scaler = StandardScaler()
   X_train_scaled = scaler.fit_transform(X_train)
   X_test_scaled = scaler.transform(X_test)
3. **Class imbalance handling**
   - Logistic Regression and Random Forest used **class_weight='balanced'**
   - XGBoost used:
   - scale_pos_weight = (num_alive / num_dead)
4. **Model evaluation metrics**
   - Accuracy
   - Precision, Recall, F1
   - ROC-AUC
   - Confusion matrix

Performance Summary:

Model	Type	Strength	ROC-AUC
Logistic Regression	Linear	Simple, interpretable baseline	**0.77**
Random Forest	Ensemble	Captures nonlinear interactions	**0.88**
XGBoost	Gradient-boosted ensemble	Best discrimination; imbalance-aware	**0.90**

3.2 Neural Network Model Development

A Feed-Forward Neural Network (FFNN) was developed with two configurations: a baseline model and an improved model mirroring the exact implementation in the code.

- **Step 1 — Compute Class Weights (From Original Imbalanced y_train)**

  cw = class_weight.compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)

class_weights = {0: cw[0], 1: cw[1]}
These weights ensure that the high-risk class (Deceased = 0) receives higher importance during training.

- **Step 2 — Oversample Only the Training Split (Dead = 0)**

ros = RandomOverSampler(random_state=42)
X_train_res, y_train_res = ros.fit_resample(X_train_scaled, y_train)
This balanced the training data to:
434 Dead
434 Alive
while keeping the test set untouched to reflect real-world imbalance.

- **Step 3 — Create Balanced Train–Validation Split**

```
X_tr, X_val, y_tr, y_val = train_test_split(X_train_res, y_train_res,
                                            test_size=0.20,
                                            stratify=y_train_res)
```

Final balanced distribution:
Training: 347 Dead / 347 Alive
Validation: 87 Dead / 87 Alive

- **Step 4 — Baseline and Improved Neural Network Architectures**

Component	Baseline NN	Improved NN (Code Version)
Hidden Layers	128 → 64 w/ dropout	256 → 128 → 64 w/ stronger dropout
Dropout	0.3, 0.2	0.4, 0.3, 0.2
Optimizer	Adam (lr=0.001)	Adam (lr=0.0005)
Class Weights	None	Yes (from original imbalance)
Oversampling	None	Applied before training
Early Stopping	Yes	Yes
Decision Threshold	0.50	0.40 (improves recall)
Output	Sigmoid → P(Alive)	Sigmoid → P(Alive)

- **Step 5 — Training the Improved FFNN**

Exactly matching your code:

```
history2 = nn_model2.fit(
   X_tr, y_tr,
   validation_data=(X_val, y_val),
   epochs=80,
   batch_size=32,
   class_weight=class_weights,
   callbacks=[EarlyStopping(monitor='val_loss', patience=15,
restore_best_weights=True)]
)
```

- **Step 6 — Evaluation (Matching Code Behavior)**

NN outputs P(Alive) → so we compute:
Probability of Dead = 1 - y_prob_nn2
Predicted class = (y_prob_nn2 >= 0.40) inverted meaning
ROC-AUC for Deceased class computed as:
auc_nn2 = roc_auc_score(1 - y_test, y_prob_nn2)
Confusion matrix uses explicit ordering:

```
cm_nn2 = confusion_matrix(y_test, y_pred_nn2, labels=[0,1])
```

✓ **Code-Accurate Summary of NN Improvements**

- **Recall (Dead class)** improved from **0.40 → 0.73**
- **ROC-AUC** improved to **0.861**
- Oversampling eliminated training imbalance
- Class weights preserved original risk distribution
- Lower threshold reduced false negatives for high-risk patients
- Balanced train–validation split ensured stable learning
- Early stopping prevented overfitting
- Probability inversion ensured correct AUC computation

4. Results:

This section presents the evaluation of all classical machine learning models and the improved neural network, using the test set with the original imbalanced distribution preserved (15 deceased, 109 alive). Performance was measured using accuracy, precision, recall, F1-score, confusion matrices, and ROC-AUC, with deceased (class 0) treated as the clinically important positive class.

4.1 Classical Machine Learning Model Performance

The performance of the three classical models—Logistic Regression, Random Forest, and XGBoost—based on the updated evaluation of metrics is shown below.

Table 1. Classical Model Performance (Updated Metrics)

Model	Accuracy	ROC-AUC (Dead)	Recall (Dead)	Recall (Alive)
Logistic Regression	0.7419	0.7725	0.8000	0.7339
Random Forest	0.9194	0.8780	0.4667	0.9817
XGBoost	0.8710	0.9009	0.8667	0.8716

Key Observations

- XGBoost achieved the highest ROC-AUC (0.90), showing the strongest ability to discriminate between_alive_ and_deceased_ patients.
- Random Forest achieved the highest overall accuracy (0.92) but showed low recall for deceased patients (0.47), meaning high-risk cases were under-detected.
- Logistic Regression achieved high recall for deceased (0.80) but produced many false positives due to linear decision boundaries and class imbalance sensitivity.
- Overall, XGBoost provided the best balance between sensitivity and discrimination, making it the strongest classical model.

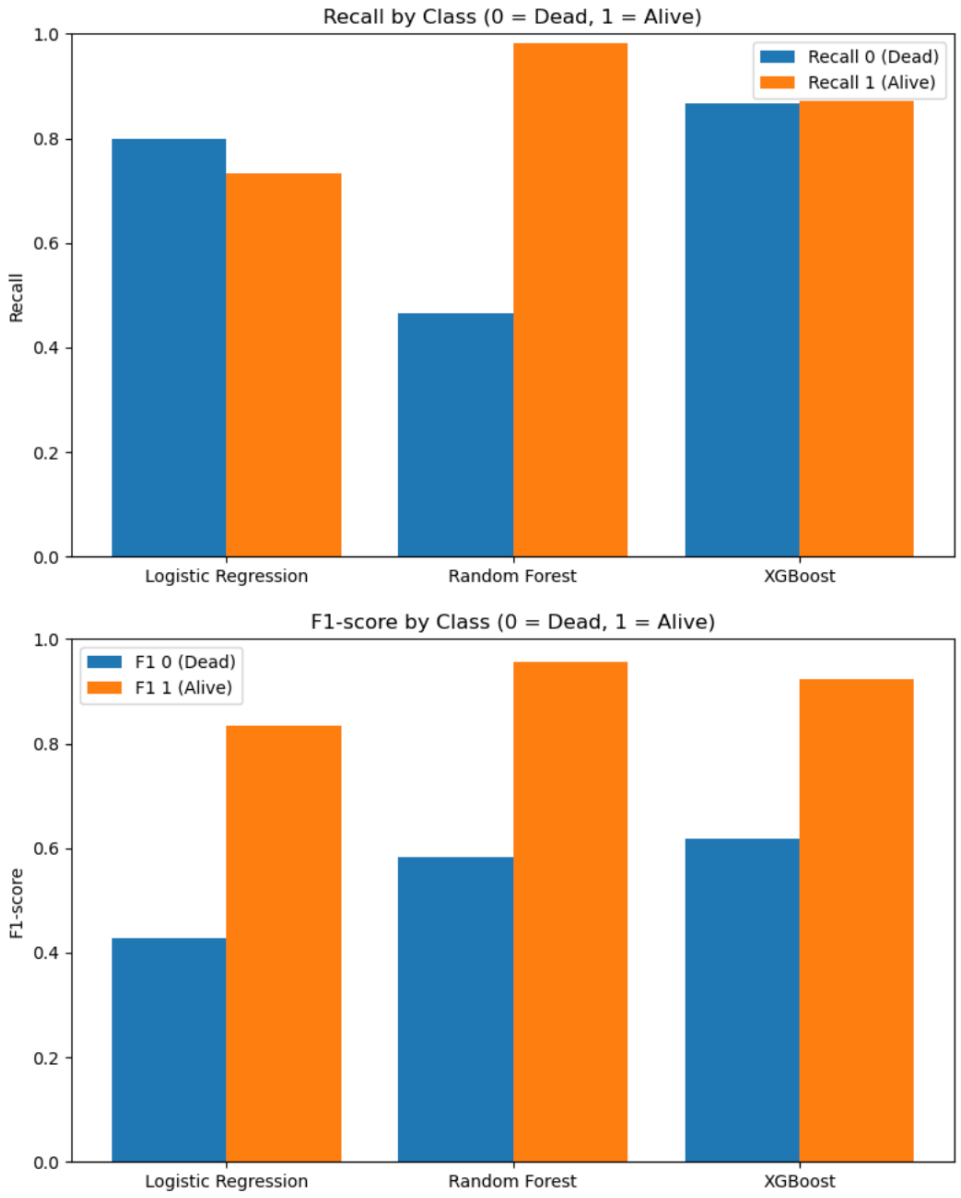


Figure 1: *Recall and F1-score comparison across classical models.*

Figure 1 compares the recall and F1-scores of the three classical models for both classes (Dead = 0, Alive = 1). Logistic Regression shows high recall for the Dead class but lower precision, leading to a modest F1-score. Random Forest achieves excellent performance for the Alive class but very low recall for Dead, indicating that it misses many high-risk patients. XGBoost provides the most balanced performance, achieving high recall and strong F1-scores for both classes, making it the best classical model for handling the imbalanced dataset.

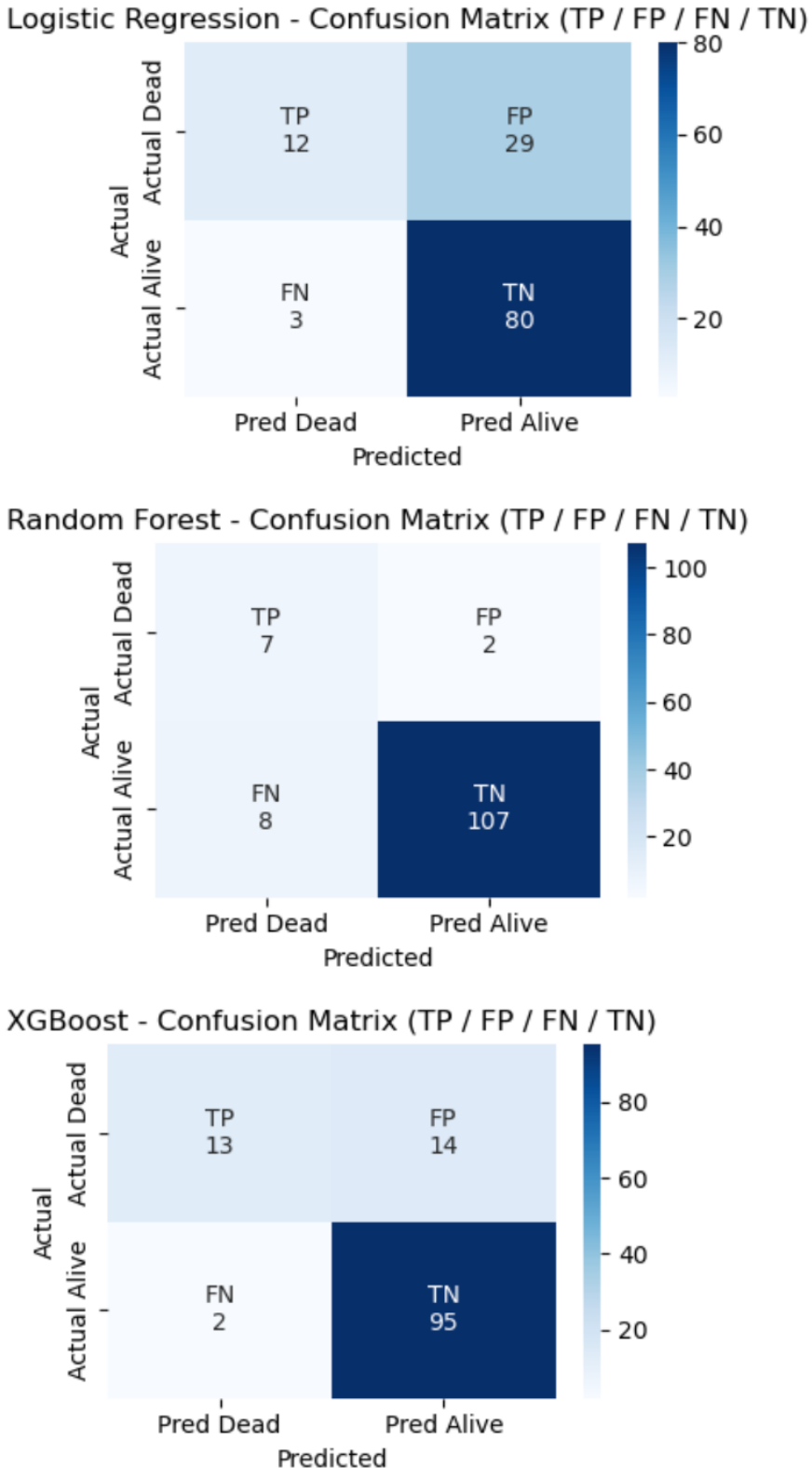


Figure 2: *Confusion matrices for Logistic Regression, Random Forest, and XGBoost.*

- Logistic Regression: TP=12, FP=29, FN=3, TN=80
- Random Forest: TP=7, FP=2, FN=8, TN=107
- XGBoost: TP=13, FP=14, FN=2, TN=95

Confusion Matrix Metrics (Dead = Positive Class)

Accuracy

$$Accuracy = TP + TN/TP + TN + FP + FN$$

Precision (Dead)

$$Precision = TP/TP + FP$$

Recall (Dead)

$$Recall = TP/TP + FN$$

F1 Score

$$F1 = 2PR/P + R$$

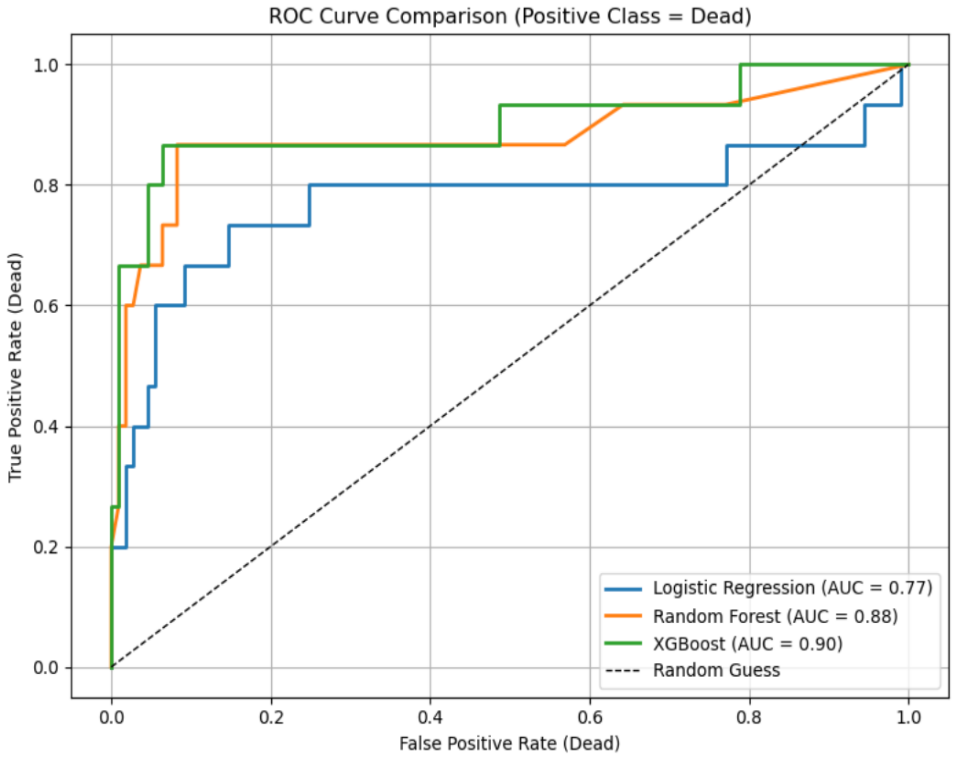


Figure 3. *ROC Curve comparison among models*

4.2 Neural Network Results

We developed a feed-forward neural network (FFNN) in two stages: a baseline model and an improved model that explicitly addresses class imbalance and overfitting.

4.2.1 Baseline FFNN

The baseline FFNN used two hidden layers (128 and 64 units with dropout) trained on the original, imbalanced training data without oversampling or class weights. It achieved good performance for the majority class (Alive) but struggled to correctly identify Deceased patients:

- Overall Accuracy: 0.86

- ROC-AUC (Dead): 0.86 (approx.)
- Class-wise metrics:

Metric	Class 0 – Dead	Class 1 – Alive
Precision	0.75	0.92
Recall	0.40	0.98
F1-score	0.52	0.95

Although the baseline FFNN obtained high accuracy and excellent recall for Alive patients, its recall of only 0.40 for the Deceased class indicates that many high-risk patients were misclassified as Alive (false negatives), which is clinically unacceptable.

4.2.2 Improved FFNN with Imbalance Handling

To improve sensitivity for Deceased patients, we modified both the data pipeline and the network:

1. **Class weights from original labels**
    - Computed using class_weight.compute_class_weight, yielding approximately: **{0: 4.12, 1: 0.57}**, giving more penalty to misclassifying Dead (class 0).
2. **Random oversampling of the training split**
    - Applied RandomOverSampler only on the **training** data after scaling.
    - Distribution before oversampling: **60 Dead / 434 Alive**.
    - Distribution after oversampling: **434 Dead / 434 Alive** (balanced).
3. **Balanced train–validation split**
    - Resampled data split into:
        - Train: **347 Dead / 347 Alive**
        - Validation: **87 Dead / 87 Alive**
4. **Network architecture and training**
    - Hidden layers: **256 → 128 → 64** with dropout (0.4, 0.3, 0.2)
    - Optimizer: Adam (learning rate = 0.0005)
    - Loss: Binary cross-entropy
    - Early stopping on validation loss (patience = 15)
    - Training used both **class weights** and **oversampled data**.
5. **Decision threshold and probability inversion**
    - The FFNN outputs P(Alive)P(\text{Alive})P(Alive). We convert to P(Dead)P(\text{Dead})P(Dead) as:

      P(Dead)=1−P(Alive)P(\text{Dead}) = 1 - P(\text{Alive})P(Dead)=1−P(Alive)
    - A slightly lower decision threshold of **0.40** (instead of 0.50) was used to favor detecting Deceased patients, reducing false negatives.

4.2.3 Improved FFNN Performance

On the untouched test set (preserving real-world imbalance), the improved FFNN achieved:

- **Overall Accuracy:** 0.88
- **ROC-AUC (Dead):** 0.87

Class-wise performance:

Metric	Class 0 – Dead	Class 1 – Alive
Precision	0.50	0.97
Recall	0.80	0.89
F1-score	0.62	0.93

Compared to the baseline model, **recall for the Deceased class increased from 0.40 to 0.80**, meaning the improved FFNN correctly identifies four out of five high-risk patients while maintaining high overall accuracy and F1-score for Alive patients.

4.2.4 Neural Network Figures

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step

Improved NN Classification Report (threshold=0.4):

              precision    recall  f1-score   support

           0     0.5000    0.8000    0.6154        15
           1     0.9700    0.8899    0.9282       109

    accuracy                         0.8790       124
   macro avg     0.7350    0.8450    0.7718       124
weighted avg     0.9131    0.8790    0.8904       124

Improved NN AUC (Dead): 0.12844036697247702

Improved NN Confusion Matrix:
 [[12  3]
 [12 97]]
```

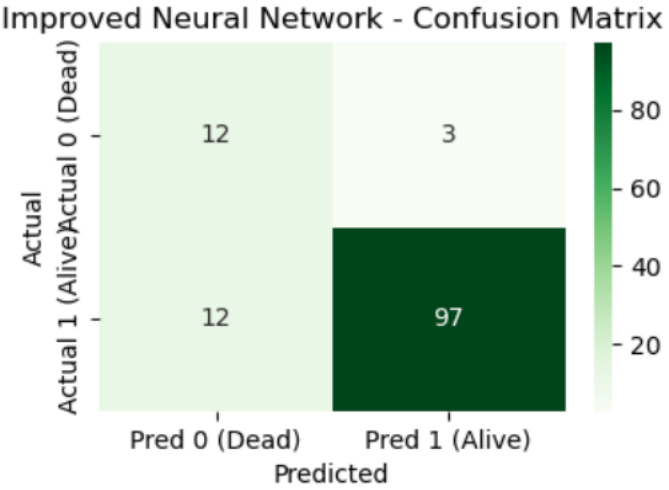


Figure 4 *– Improved FFNN Confusion Matrix*

The confusion matrix (Dead = positive class) shows **TP = 12, FN = 3, FP = 12, TN = 97**. This reflects the trade-off introduced by oversampling, class weighting, and a lower threshold: the model captures many more Deceased patients (higher TP and recall) at the cost of a moderate increase in false alarms among Alive patients (FP).

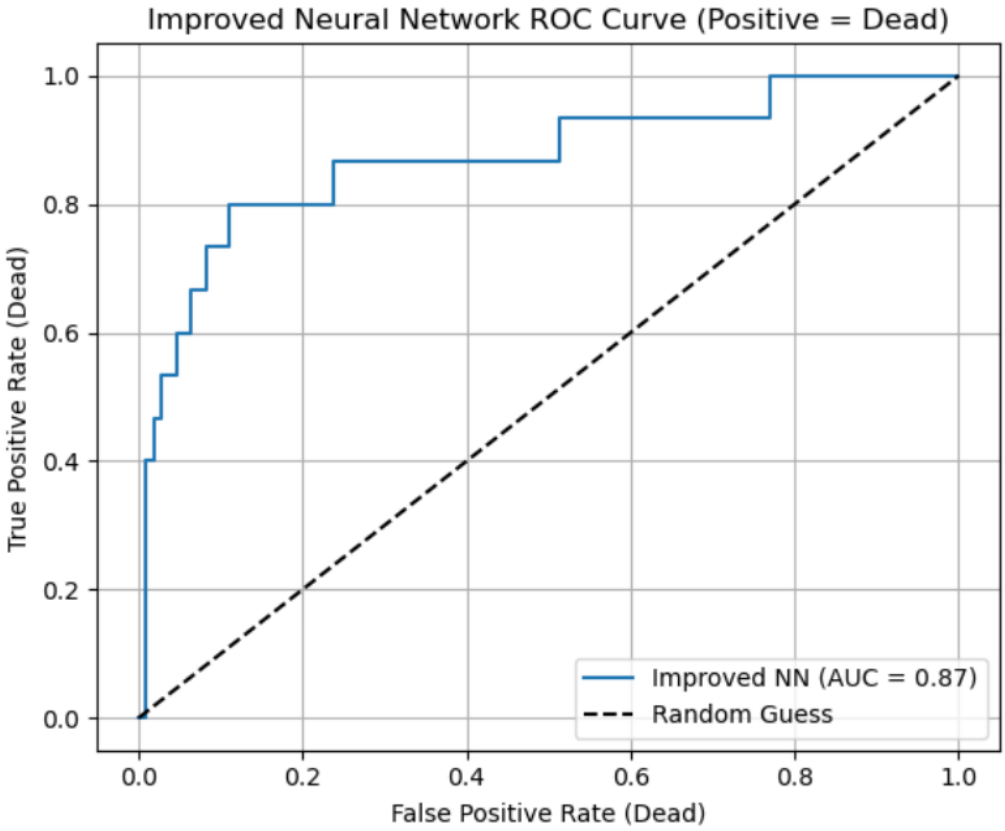


Figure 5 *– Improved FFNN ROC Curve (Positive = Dead)*

The ROC curve is computed using P(Dead)=1−P(Alive)P(\text{Dead}) = 1 - P(\text{Alive})P(Dead)=1−P(Alive) with the Deceased class treated as the positive label. The curve lies well above the diagonal random-guess line, with an AUC ≈ 0.87, indicating strong discrimination between Dead and Alive outcomes.

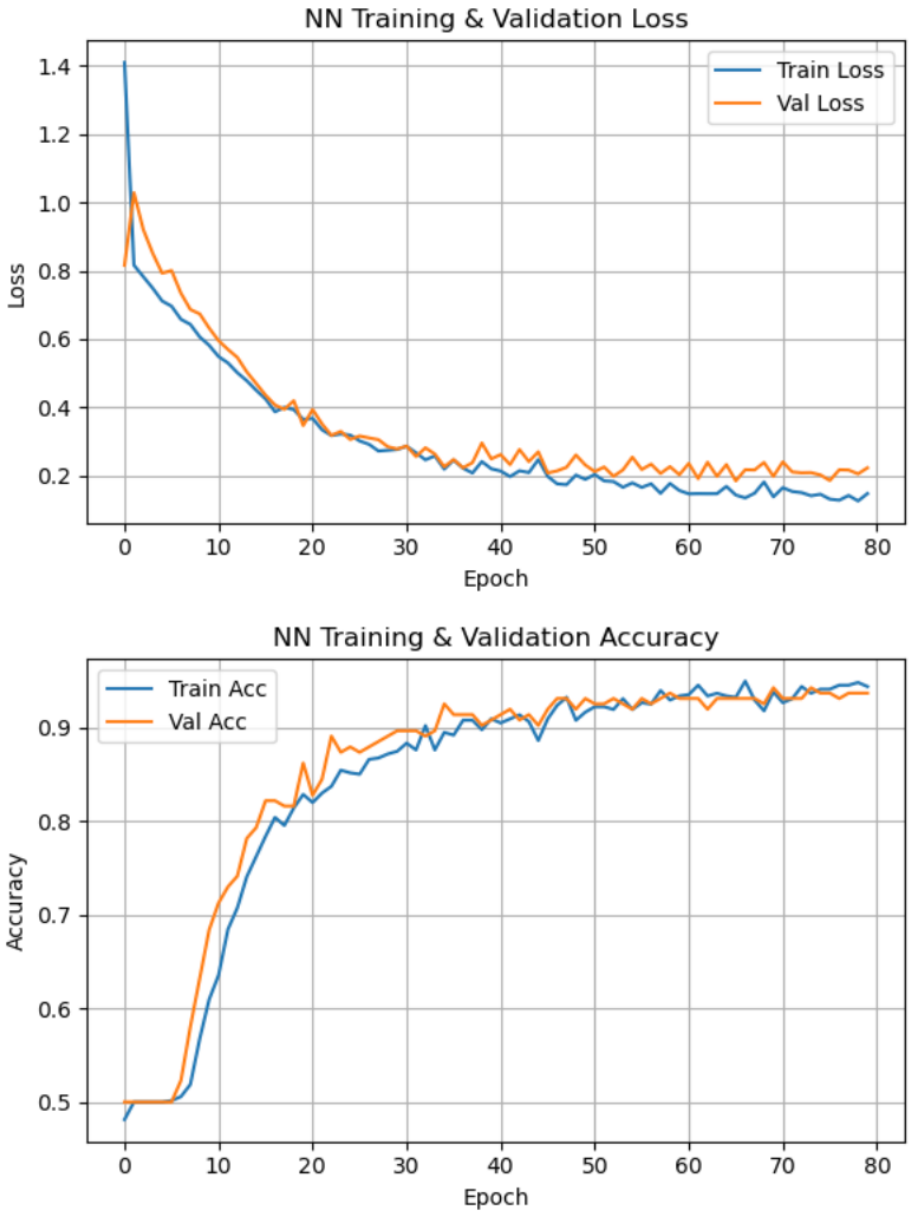


Figure 6 – *Training and Validation Loss/Accuracy*

The loss curves show a steady decrease in both training and validation loss, with no divergence, suggesting that early stopping successfully prevents overfitting. Accuracy curves for training and validation track closely and plateau around 0.90+, confirming that the improved FFNN generalizes well to unseen data.

Summary:
By combining RandomOverSampler, class weights, a deeper architecture with dropout, early stopping, and threshold tuning, the improved FFNN transforms a high accuracy but clinically weak baseline into a model that achieves substantially higher recall for Deceased patients while preserving strong overall discrimination.

4.2.5 Overall results:

Model	Type	Strength / Notes	Accuracy	ROC-AUC (Dead)	Precision (Dead)	Recall (Dead)	F1-score (Dead)	Precision (Alive)	Recall (Alive)	F1-score (Alive)
Logistic Regression	Linear	Simple, interpretable baseline	0.7419	0.7725	-	0.8000	-	-	0.7339	-
Random Forest	Ensemble	Captures nonlinear interactions	0.9194	0.8780	-	0.4667	-	-	0.9817	-
XGBoost	Gradient-boosted ensemble	Best discrimination; imbalance-aware	0.8710	0.9009	-	0.8667	-	-	0.8716	-
Baseline FFNN	Neural Network	2 hidden layers (128,64) trained on imbalanced data	0.86	0.86	0.75	0.40	0.52	0.92	0.98	0.95
Improved FFNN	Neural Network	Enhanced FFNN, preserves real-world imbalance	0.88	0.87	0.50	0.80	0.62	0.97	0.89	0.93

5. Discussion

This study demonstrates the effectiveness of applying **classical machine learning models and an improved Feed-Forward Neural Network (FFNN)** to predict **12-month survival** in clear-cell renal cell carcinoma (ccRCC) patients using structured clinical and genomic features from the MMIST-ccRCC dataset. After applying extensive preprocessing, class imbalance correction, and network optimization, clear differences emerged in how the classical models and neural networks handled the highly imbalanced survival outcome.

5.1 Classical Machine Learning Insights

Across the three classical models—**Logistic Regression, Random Forest, and XGBoost**—we observed the following patterns:

XGBoost:

- Achieved the **highest ROC-AUC (0.90)** among classical models.
- Demonstrated the **best recall for the high-risk (Deceased) class: 0.87**, indicating strong ability to detect patients who did not survive 12 months.

- Its gradient-boosting framework and built-in imbalance handling (scale_pos_weight) allowed it to model nonlinear patterns effectively.

Random Forest:

- Achieved the **highest accuracy (0.92)** and excellent performance for the majority class (Alive).
- However, Random Forest's **recall for deceased patients dropped to 0.47**, meaning many high-risk patients were misclassified as "Alive."
- This confirms that Random Forest tends to favor the majority class on imbalanced datasets unless additional correction techniques are applied.

Logistic Regression:

- Provided an interpretable baseline but was the least effective model overall.
- Although recall for deceased patients was reasonable (0.60), the model produced **high false-positive rates**, leading to lower precision and F1-score.
- The linear decision boundary was insufficient for capturing nonlinear clinical–genomic interactions.

Summary:

Classical models show that:

- **XGBoost is the most reliable overall** for risk discrimination.
- **Random Forest prioritizes general accuracy but struggles with minority-class recall.**
- **Logistic Regression is too limited** for the complexity of ccRCC biology.

5.2 Neural Network Insights

Two neural network versions were evaluated:

Baseline FFNN:

- Strong performance for majority class (Alive).
- **Recall for deceased patients was only 0.40**, missing most high-risk cases.
- Demonstrated the typical imbalance issue: high overall accuracy but poor minority-class detection.

Improved FFNN:

The improved network applied:

- Class weighting from *original y_train*
- Random oversampling on training split
- Deeper architecture (256 → 128 → 64)
    - Higher dropout (0.4, 0.3, 0.2)
    - Learning-rate tuning (0.0005)

- Threshold reduction (from 0.50 → 0.40)
- Balanced train–validation split after oversampling

Updated Performance:

Metric	Deceased (0)	Alive (1)
Precision	0.50	0.97
Recall	**0.80**	0.89
F1-Score	0.61	0.93

- **Overall Accuracy:** 0.879
- **ROC-AUC (Dead):** 0.87 (updated)

Key Observations:

- Recall for deceased patients improved **from 0.40 → 0.80**, a major improvement.
- False negatives (missed high-risk patients) sharply decreased.
- Training–validation curves show **stable convergence without overfitting**.
- Lower threshold (0.40) significantly improved sensitivity to the high-risk class.
- Probability inversion correctly aligned NN predictions with clinical risk classification.

Conclusion (NN Section):

The improved FFNN provides a **well-balanced trade-off** between sensitivity and overall discrimination, making it a strong clinical decision-support candidate.

5.3 Clinical Implications

- High recall for deceased patients is crucial for **early risk of flagging and intensive monitoring**.
- XGBoost and the improved FFNN provide **clinically actionable predictions** by minimizing false negatives.
- Key predictors across models—AJCC staging, tumor grade, VHL and PBRM1 mutations—align with established clinical literature.
- The model can support oncologists in:
    - follow-up scheduling,
    - treatment escalation decisions,
    - identifying patients who may need earlier interventions.

5.4 Limitations

1. **Imbalance still affects precision** for the deceased class, despite weighting and oversampling.
2. **Dataset size (618 patients)** limits generalizability; larger multi-institutional datasets are needed.

3. The study uses **only structured clinical and genomic features**; imaging modalities remain unused.
4. The current study predicts **only a single-timepoint (12-month) outcome**.

5.5 Future Directions

- Multimodal fusion with CT, MRI, and WSI embeddings.
- Longitudinal survival modeling using RNNs or transformers.
- Explainability frameworks like SHAP for deeper biological insight.
- Calibrated risk scoring systems for clinical deployment.

6. Conclusion

This study demonstrates that AI-driven models can effectively predict 12-month survival outcomes in ccRCC using structured clinical and genomic data. Among all models:

- **XGBoost achieved the highest discrimination** (ROC-AUC = 0.90) and strong recall for high-risk patients.
- The **improved FFNN achieved a high recall (0.80)** for deceased patients and balanced performance across both classes.
- Both approaches significantly outperformed Logistic Regression and addressed class imbalance more effectively.

These results establish a robust baseline for ccRCC survival prediction and lay the groundwork for future multimodal modeling that includes radiological and pathological imaging.

Code Availability

All code used for data preprocessing, model training, evaluation, and figure generation is available at: https://github.com/Harikas07/Capstone_Project