Global Scientific JOURNALS

GSJ: Volume 13, Issue 11, November 2025, Online: ISSN 2320-9186

www.globalscientificjournal.com

Performance Evaluation of Machine Learning Algorithms in Prostate Cancer Detection

A paper presented by:

NWENE, OGBONNAYA VINCENT

COMPUTER ENGINEERING DEPARTMENT, FEDERAL POLYTECHNIC OKO, ANAMBRA STATE

nwene.ogbonnaya@federalpolyoko.edu.ng (08066867988)

and

ENGR. DR. ANANTI JOHN EGBUNIKE

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT, FEDERAL POLYTECHNIC OKO,ANAMBRA STATE.

john.ananti@federalpolyoko.edu.ng (08030997272)

Abstract

Prostate cancer remains a major health concern among men, necessitating efficient and accurate diagnostic approaches. This study presents a comparative evaluation of various machine learning algorithms for prostate cancer prediction using clinical diagnostic features such as radius, texture, area, smoothness, and symmetry. Data preprocessing involved normalization and correlation-based feature selection to enhance model performance. Several algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), XGBoost, LightGBM, CatBoost, and a Deep Learning model were trained and assessed using accuracy, precision, recall, F1-score, and ROC-AUC metrics. Results revealed that ensemble-based methods, notably XGBoost and Random Forest, outperformed other models in predictive accuracy, while Logistic Regression provided higher interpretability. The study highlights the potential of machine learning models in facilitating early and reliable prostate cancer detection to support clinical decision-making.

Keywords: Prostate cancer, Machine learning, Classification, Model evaluation, Early detection

Introduction

The prostate is a small, walnut-shaped gland that forms part of the male reproductive system. It is located below the bladder and in front of the rectum, surrounding the urethra—the tube responsible for carrying urine and semen out of the body. The prostate gland plays a vital role in producing seminal fluid, which nourishes and protects sperm, and also contributes to the regulation of urinary flow.

Prostate cancer is a malignant condition that originates in the prostate gland when abnormal cells begin to grow uncontrollably, forming a tumor. While some prostate cancers progress slowly and may pose minimal health risks, others are aggressive and can metastasize to other parts of the body. Early detection and accurate diagnosis are essential for effective treatment and improved patient survival rates.

In recent years, the emergence of machine learning (ML) techniques has provided innovative approaches for analyzing complex medical datasets, enabling improved accuracy in disease prediction and diagnosis. ML algorithms apply artificial intelligence to identify hidden patterns in clinical data and make reliable diagnostic predictions.

Several studies have demonstrated the potential of machine learning in prostate cancer detection and prognosis. For instance, (Wang et al. 2018) utilized Logistic Regression and Decision Tree models to analyze clinical data, achieving high predictive accuracy in assessing prostate cancer risk. Similarly, (Zhang et al. 2020) employed Random Forest algorithms on genomic datasets to predict cancer recurrence, highlighting the effectiveness of ML in personalized medicine. Furthermore, the integration of ML with medical imaging has significantly enhanced diagnostic precision. (Zhu et al. 2019) developed a deep learning model that combined magnetic resonance imaging (MRI) with clinical data, achieving superior classification performance compared to traditional diagnostic techniques. In addition, (Radovic et al. 2021) implemented ensemble learning methods, including Random Forest, to analyze multiparametric MRI data, demonstrating improved sensitivity and specificity in prostate cancer diagnosis.

These findings collectively underscore the transformative potential of machine learning in enhancing early detection, accurate diagnosis, and treatment planning for prostate cancer.

Problem Statement

Despite significant medical advancements, the detection and diagnosis of prostate cancer remain challenging due to issues of under diagnosis and misclassification. Traditional diagnostic methods, including clinical assessments and imaging evaluations, are often time-consuming, expensive, and dependent on expert interpretation, which may lead to inconsistent outcomes. These limitations underscore the urgent need for intelligent and automated diagnostic systems capable of supporting clinicians in decision-making. Machine learning presents a promising approach for developing predictive models that can efficiently analyze clinical data, enhance diagnostic accuracy, and provide scalable solutions, particularly in low-resource healthcare environments.

Objective (s) of the study:

The basic objectives of this research are as listed below:

1. To source and preprocess a prostate cancer dataset.
2. To analyze feature relationships within the dataset.
3. To implement and train different machine learning models.
4. To compare model performance based on standard evaluation metrics.
5. To recommend the most effective model for clinical application.

Literature Review

Historical Review of Machine Learning

The foundation of artificial intelligence (AI) and machine learning (ML) can be traced back to the 1940s, during and after World War II, when researchers began exploring the potential of computers to replicate aspects of human intelligence. In 1950, **Alan Turing** introduced the concept of "learning machines" in his seminal work *Computing Machinery and Intelligence*, proposing that computers could be designed to emulate human reasoning and intelligent behavior (Luchini et al. 2021). This idea laid the groundwork for the development of artificial intelligence as a scientific discipline.

The formal term **"**Artificial Intelligence**"** was coined in 1956 by John McCarthy, who defined it as *the science and engineering of making intelligent machines*. This marked the official beginning of AI research and the vision of computers capable of performing tasks traditionally requiring human cognition (McCarthy et al. 1956).

The concept of **"**Machine Learning" emerged shortly thereafter. In 1959, Arthur Samuel introduced the term to describe the ability of computers to learn from experience without being explicitly programmed (Samuel, 1959). His pioneering work on a self-learning checkers-playing program demonstrated that computers could improve their performance through iterative learning — a foundational principle of modern ML.

Over the decades, artificial intelligence has evolved from a theoretical concept into a transformative technology embedded in nearly every aspect of modern life. Today, AI applications span diverse fields such as robotics, search engines, law enforcement, autonomous systems, and medical diagnostics. The definition of AI has also expanded to encompass the performance of cognitive tasks—such as perception, reasoning, and decision-making—by machines that mimic human intelligence (Bi, Q et al. 2019).

Within the broader domain of AI, machine learning has become one of the most influential subfields, enabling systems to automatically identify patterns, adapt to new data, and make predictions with minimal human intervention. ML goes beyond traditional statistical approaches by leveraging vast, multi-dimensional datasets to discover hidden relationships and optimize outcomes. In the field of healthcare, ML has gained significant attention for its potential to improve disease prediction, diagnosis, and patient management. It excels in utilizing large-scale electronic health record (EHR) data, selecting relevant variables, and uncovering complex interactions among clinical parameters to enhance individualized care (Levin et al. 2018).

Application of Machine Learning in Prostate Cancer Detection

Machine learning (ML) has shown significant potential in improving the detection, classification, and prognosis of prostate cancer. Traditional diagnostic methods—such as prostate-specific antigen (PSA) testing, digital rectal examination (DRE), and magnetic resonance imaging (MRI)—though widely used, often suffer from limitations such as false positives, over-diagnosis, and inconsistent interpretation among clinicians. ML techniques address these challenges by leveraging large datasets to identify complex, non-linear patterns that enhance diagnostic accuracy and clinical decision-making (Ahmed et al. 2017).

ML algorithms have been successfully applied to predict prostate cancer risk using clinical parameters such as PSA levels, patient age, prostate volume, and biopsy results. For instance, **(**Lawrence et al. 2019**)** demonstrated that Logistic Regression and Support Vector Machine (SVM) models could effectively classify patients at high risk of prostate cancer using clinical biomarkers, outperforming traditional threshold-based PSA analysis. Similarly, **(**Gann et al. 2018) developed ML-based risk stratification models that incorporated demographic and biochemical data, resulting in improved patient screening efficiency.

The integration of ML with medical imaging**,** particularly multiparametric MRI (mpMRI), has revolutionized prostate cancer diagnosis. Algorithms such as Convolutional Neural Networks (CNNs) have been trained to automatically detect and localize prostate lesions, significantly reducing the dependence on manual image interpretation. (Zhu et al. 2019) proposed a deep learning model that combined MRI and clinical data, achieving higher sensitivity and specificity in cancer detection compared to conventional radiological assessments. Likewise, **(**Ishioka et al. 2020) utilized ensemble learning methods to classify prostate lesions based on MRI features, achieving enhanced diagnostic reliability.

In addition to diagnosis, ML has been employed in prognostic modeling to predict tumor aggressiveness, treatment response, and recurrence probability. For example, **(**Nitta et al. 2021) developed an ML-based framework integrating genomic and pathological features to predict biochemical recurrence after prostatectomy. These models provide clinicians with valuable insights for personalized treatment planning and risk management.

Research Gap

Several studies have explored the application of machine learning in prostate cancer prediction; however, key limitations persist that warrant further investigation.

(Chen et al. 2022)**,** in their study *"Machine Learning-Based Models Enhance Prostate Cancer Prediction,"* employed four supervised learning algorithms—Logistic Regression, Decision Trees, Random Forest, and Support Vector Machine—to develop predictive models for prostate cancer. Although the multivariate Logistic Regression model achieved the highest performance with an AUC of 0.918, the study revealed challenges in accurately detecting prostate cancer cases that do not present as typical nodular formations. This limitation suggests that model generalization remains a significant concern in clinical application.

Similarly, **(**Lee et al. 2019**)**, in *"Machine Learning Approaches for Predicting Prostate Cancer Based on Age and Prostate Specific Antigen Level: Experience from the Field,"* utilized SVM, RF, LR, LGBM, and XGBoost algorithms. Their reported accuracy ranged from 64.4% to 74%, indicating relatively low predictive power. This outcome highlights the need to explore additional and more advanced algorithms that can improve classification performance and diagnostic accuracy.

Furthermore, **(**Saqib Iqbal et al. 2016)**,** in *"Prostate Cancer Detection Using Deep Learning and Traditional Techniques,"* examined the synergy between deep learning and conventional machine learning models through a structured approach involving data preprocessing, training, and evaluation. While the integration of traditional and deep learning methods provided complementary strengths—combining interpretability with pattern recognition—issues related to data quality, model interpretability, and ethical implications remained unresolved.

Methodology

The methodology of this project involves the evaluation of different machine learning and Deep Learning models in order to adjudge their effectiveness in early detection of prostate cancer.

The following tools were used to carry out this process

i. Kaggle platform
ii. Python 3
iii. Pytorch, Pandas, NumPy, Scikit-learn, Scipy

Kaggle platform is the experimental environment for this project work. Kaggle is actually a cloud-based ecosystem fully designed for data science and machine learning projects. Kaggle offers robust infrastructure with access to advanced computing resources, such as GPUs and TPUs, making it suitable for handling of datasets like prostate and implementing it in machine learning models and deep learning models.

The entire experiment was conducted within a single Jupyter notebook, a versatile tool for interactive computing that integrates code, visuals, and text in one environment. Python 3 was the programming language employed for the implementation. Python's extensive ecosystem of libraries such as Pytorch, Pandas, NumPy, and scikit-learn were inclusive to facilitate preprocessing, model training, and evaluation.

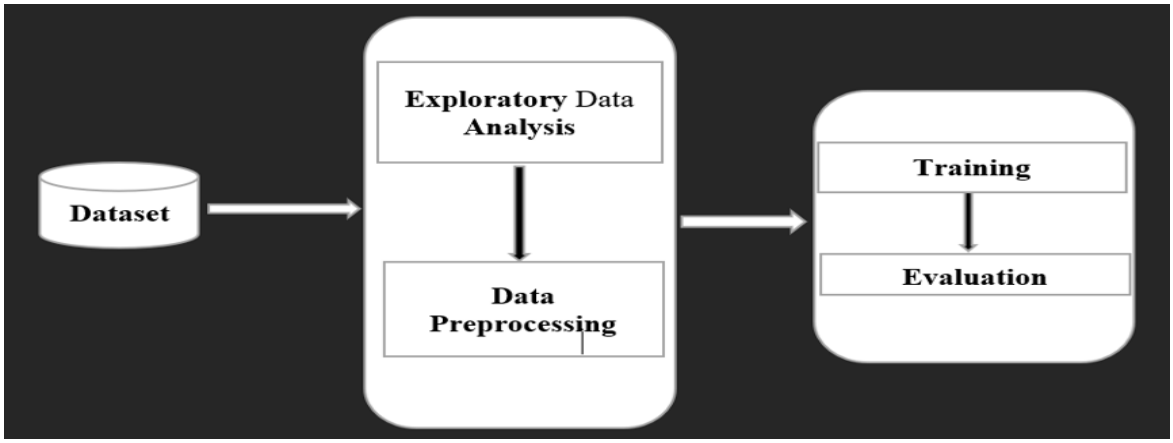

Figure 1: Methodology

Importation of Dataset

The first step involves obtaining the Prostate Cancer dataset from Kaggle, the dataset was imported into the python platform for further exploration data analysis and preprocessing.

```
#data_pth = "/kaggle/input/prostate-cancer/Prostate_Cancer.csv"

data_pth = r"c:/Users/LENOVO/Desktop/DATASET/Practices/main_Prostate_Cancer.csv"

toolkit = ProjectToolkit(data_path=data_pth)
```

Figure 2: Importation of Dataset

Exploratory Data Analysis (EDA)

This section describes the steps taken to load the dataset and preprocess it for analysis.

Data Loading

The prostate cancer dataset was loaded into Python using pandas. The prostate cancer dataset was uploaded for further preprocessing.

```
df = pd.read_csv(toolkit.data_path)

df.head()
```

Python

	id	diagnosis_result	radius	texture	perimeter	area	smoothness	compactness	symmetry	fractal_dimension
0	1	M	23	12	151	954	0.143	0.278	0.242	0.079
1	2	B	9	13	133	1326	0.143	0.079	0.181	0.057
2	3	M	21	27	130	1203	0.125	0.160	0.207	0.060
3	4	M	14	16	78	386	0.070	0.284	0.260	0.097
4	5	M	9	19	135	1297	0.141	0.133	0.181	0.059

Figure 3: Data Loading

Dataset Information and Statistics

After uploading the dataset, the dataset's structure was reviewed using the info() function in pandas. This offered a summary of each feature's data type, the count of non-null values, and the dataset's memory usage.

```
df.info()
```
Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 100 non-null    int64
 1   diagnosis_result   100 non-null    object
 2   radius             100 non-null    int64
 3   texture            100 non-null    int64
 4   perimeter          100 non-null    int64
 5   area               100 non-null    int64
 6   smoothness         100 non-null    float64
 7   compactness        100 non-null    float64
 8   symmetry           100 non-null    float64
 9   fractal_dimension  100 non-null    float64
dtypes: float64(4), int64(5), object(1)
memory usage: 7.9+ KB
```

Figure 4: Summary of Data information

The describe() function was also utilized to produce summary statistics for the numerical columns. This provided insights into key characteristics of the dataset, including the mean, standard deviation, and range. These metrics were essential for understanding the data's distribution and identifying potential outliers or errors.

:

```
df.describe().T
```
Python

	count	mean	std	min	25%	50%	75%	max
id	100.0	50.50000	29.011492	1.000	25.7500	50.5000	75.250	100.000
radius	100.0	16.85000	4.879094	9.000	12.0000	17.0000	21.000	25.000
texture	100.0	18.23000	5.192954	11.000	14.0000	17.5000	22.250	27.000
perimeter	100.0	96.78000	23.676089	52.000	82.5000	94.0000	114.250	172.000
area	100.0	702.88000	319.710895	202.000	476.7500	644.0000	917.000	1878.000
smoothness	100.0	0.10273	0.014642	0.070	0.0935	0.1020	0.112	0.143
compactness	100.0	0.12670	0.061144	0.038	0.0805	0.1185	0.157	0.345
symmetry	100.0	0.19317	0.030785	0.135	0.1720	0.1900	0.209	0.304
fractal_dimension	100.0	0.06469	0.008151	0.053	0.0590	0.0630	0.069	0.097

Figure 5: Summary statistics for the numerical columns

Encoding of the Diagnostic_Result

In datasets like the one used for this research work where the **diagnosis_result**is labeled as **"M" or "B"**, these labels typically represent target variable:

M= Malignant indicating the presence of cancerous tumor which tends to grow uncontrollably and invade surrounding tissues.

B= Benign indicating that tumor is not cancerous which usually localize and does not spread.

M is coded **1** while **B** is coded **0** as shown in the figure below.

```
encoded_df = toolkit.EncodeCategoricalColumn(df, df.select_dtypes(include="O").columns)

encoded_df.head()
```
Python

```
Encoded Features: {'diagnosis_result': {0: 'B', 1: 'M'}}
```

	diagnosis_result	radius	texture	perimeter	area	smoothness	compactness	symmetry	fractal_dimension
0	1	23	12	151	954	0.143	0.278	0.242	0.079
1	0	9	13	133	1326	0.143	0.079	0.181	0.057
2	1	21	27	130	1203	0.125	0.160	0.207	0.060
3	1	14	16	78	386	0.070	0.284	0.260	0.097
4	1	9	19	135	1297	0.141	0.133	0.181	0.059

Figure 6: Encoding of Diagnosis Result.

Data Visualization

Data visualization is the graphical representation of data and information. It transforms complex datasets into easily interpretable visual formats, enabling a better understanding of patterns, trends, and relationships within the data. In the context of machine learning, data visualization plays a crucial role in exploratory data analysis (EDA), model evaluation, and result presentation.

Data Preprocessing

Data preprocessing is a crucial step in preparing datasets for machine learning models. It involves cleaning, transforming, and scaling the data to ensure it is suitable for training.

Scaling the Data

The features were standardized using z-score normalization to ensure they had a mean of 0 and a standard deviation of 1. Scaling was necessary to prevent features with larger magnitudes from dominating and causing biased predictions. Moreover, models that rely on optimization techniques, such as gradient descent, perform better with scaled data as it enhances convergence speed.

The standardization was performed using the following formula:

$$z = \frac{x - \mu}{\sigma}$$

Where:

- x is the original feature value,
- μ is the mean of the feature,
- σ is the standard deviation of the feature,
- z is the scaled feature.

This transformation ensures that each feature has:

- A **mean** of 0: $\mu = 0$,
- A **standard deviation** of 1: $\sigma = 1$.

The scaling was done as follows:

```
# Scale data (using zscore), split data into X and y
scaled_X, y = toolkit.ScaleData(encoded_df.drop("diagnosis_result", axis=1), SS=True), encoded_df["diagnosis_result"]
```
29] Python

```
scaled_X
```
30] Python

	radius	texture	perimeter	area	smoothness	compactness	symmetry	fractal_dimension
0	1.266830	-1.205746	2.301611	0.789417	2.764210	2.486970	1.594151	1.764464
1	-1.617011	-1.012208	1.537520	1.958830	2.764210	-0.784061	-0.397314	-0.948199
2	0.854853	1.697335	1.410172	1.572169	1.528655	0.547364	0.451507	-0.578290
3	-0.587068	-0.431591	-0.797201	-0.996139	-2.246650	2.585594	2.181796	3.983916
4	-1.617011	0.149025	1.622419	1.867666	2.626926	0.103555	-0.397314	-0.701593
...	...	...	...	...	...	...	...	...
95	1.266830	-0.431591	1.495071	1.763927	-0.805170	0.070681	0.549448	-1.071502
96	1.060841	-0.818669	-0.797201	-0.791806	0.155817	-0.915560	-0.103491	0.161527
97	0.442876	1.697335	-1.476393	-1.282205	-0.050109	-1.211432	-1.899074	0.531435

Figure 8: Scaling the Data Using Z-score

Data Splitting

The dataset was divided into training and testing sets while preserving the original proportions. This ensures the model is trained and tested on representative data. Subsequently, the training set was further partitioned into training and validation sets to facilitate model evaluation during training. The splitting was performed using the train_test_split method.

```
Xtrain, Xtest, ytrain, ytest = train_test_split(scaled_X, y, test_size=.2, stratify=y, random_state=toolkit.
random_state)

print(Xtrain.shape, Xtest.shape)
```

Python

```
(80, 8) (20, 8)
```

Figure 9: Data Splitting

Table 1. Distribution of Train Data Split

X_train		Y_train
No. of Samples	**No. of Features**	**No. of Samples**
80	8	80

Table 2. Distribution of Test Data split

X_test		Y_test
No. of Samples	**No. of Features**	**No. of Samples**
20	8	20

The dataset was split into training and testing sets in an 80-20 ratio, with stratification to maintain class balance.

Modeling

The modeling process involved applying eight machine learning algorithms alongside a deep learning algorithm to the prostate dataset to predict outcomes. The machine learning models served as benchmarks to validate the performance of the deep learning model on the training data. Each machine learning algorithm was tested to establish baseline performance, leveraging its unique strengths and mathematical principles. The machine learning models used are as follows:

SVC, LogisticRegression, DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier, XGBClassifier, LGBMClassifier, CatBoostClassifier

```
model_list = [SVC, DecisionTreeClassifier, KNeighborsClassifier,
              RandomForestClassifier, LogisticRegression, XGBClassifier,
              LGBMClassifier, CatBoostClassifier]
```

Python

```
pred_pre_model_f, model_score_f = toolkit.TrainMLModelInFold(Xtrain, ytrain, Xtest, ytest, model_list, 5)
```

Python

```
SVC: 100%|██████████| 5/5 [00:01<00:00,  3.52it/s]
DecisionTreeClassifier: 100%|██████████| 5/5 [00:01<00:00,  4.68it/s]
KNeighborsClassifier: 100%|██████████| 5/5 [00:01<00:00,  3.55it/s]
RandomForestClassifier: 100%|██████████| 5/5 [00:08<00:00,  1.74s/it]
LogisticRegression: 100%|██████████| 5/5 [00:01<00:00,  2.74it/s]
XGBClassifier: 100%|██████████| 5/5 [00:04<00:00,  1.21it/s]
LGBMClassifier: 100%|██████████| 5/5 [00:04<00:00,  1.10it/s]
CatBoostClassifier: 100%|██████████| 5/5 [00:48<00:00,  9.75s/it]
```

Figure 10: Machine Learning Development

Deep Learning Model Implementation

The deep learning model, built using PyTorch for binary classification, implements a fully connected neural network architecture adaptable for multiclass classification tasks. The architecture features dense layers utilizing the Rectified Linear Unit (ReLU) activation function. Both the input and hidden layers employ ReLU activation (as described in Equation 1) to introduce non-linearity, enabling the model to effectively capture complex data patterns.

$$ReLU(x) = \max(0, x) \qquad (1)$$

A softmax activation function is applied in the output layer to convert raw outputs into probabilities, ensuring they sum to 1 and represent the predicted class probabilities for each

category. To address overfitting, dropout layers with a 30% dropout rate were incorporated. During training,

$$\text{Softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}}$$

Where:

- z_j: Logit (raw score) for the j-th class.
- e^{z_j}: Exponential of the logit z_j.
- $\sum_{k=1}^{C} e^{z_k}$: Normalizing factor that ensures the sum of probabilities equals 1.

these layers randomly deactivate 30% of neurons, enhancing the model's generalization ability and improving its capacity to learn robust patterns.

The model is structured as follows: The Input Layer accepts the input features, with the shape determined by the dataset. These hidden layers are activated using the ReLU (Rectified Linear Unit) function, with a Dropout rate of 0.3 to prevent over fitting. The Output Layer has a number of neurons equal to the number of classes in the dataset. It uses the Softmax activation function, which converts the raw output values into probability scores for each class.

3.5.1 Deep Learning Model Parameters

The model was compiled and trained using the following parameters:

Optimizer: The Adam (Adaptive Moment Estimation) optimizer was utilized. Adam demonstrates superior performance compared to other optimizers, although its effectiveness may vary depending on the architecture. Adam is a widely used optimization algorithm in deep learning, combining the strengths of AdaGrad and RMSProp. It excels in handling sparse gradients and non-stationary objectives, making it a robust and reliable choice for training neural networks.

Loss Function: The model utilized sparse categorical cross-entropy as the loss function, which is well-suited for multiclass classification tasks. This loss function is particularly effective when target labels are encoded as integers (e.g., 0, 1, 2, ..., n−1n-1 for nn classes) rather than one-hot encoded vectors.

The sparse categorical cross-entropy loss can be defined as:

$$L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i)$$

$$L(y, \hat{y}) = -\sum_{i=1} y_i \cdot \log(\hat{y}_i)$$

Where:

yi is the true label for class i (one-hot encoded).

yi^ is the predicted probability for class iii, obtained using the softmax activation function.

For **sparse categorical Cross entropy**, the target labels y is not one-hot encoded. Instead, they are just integer values representing the class index for each sample.

Evaluation Metric: Accuracy, which measures the proportion of correctly classified instances.

The model was trained for 20 epochs with a batch size of 32, and early stopping was employed to prevent over fitting. Additionally, learning rate reduction was applied when the validation performance plateaued, allowing the model to refine its learning.

```
    result, tst_logit, test_prob = toolkit.Train_DL_MODEL(Xtrain, ytrain, Xval, yval, Xtest, ytest)
9]                                                                                                  Python

  Epoch 1/20, Train Loss: 1.8349, Val Loss: 0.6498, Val Accuracy: 0.6111
  Epoch 2/20, Train Loss: 1.8252, Val Loss: 0.6462, Val Accuracy: 0.6667
  Epoch 3/20, Train Loss: 1.7631, Val Loss: 0.6432, Val Accuracy: 0.6667
  Epoch 4/20, Train Loss: 1.8148, Val Loss: 0.6409, Val Accuracy: 0.6667
  Epoch 5/20, Train Loss: 1.8129, Val Loss: 0.6387, Val Accuracy: 0.6667
  Epoch 6/20, Train Loss: 1.7594, Val Loss: 0.6361, Val Accuracy: 0.6667
  Epoch 7/20, Train Loss: 1.8348, Val Loss: 0.6336, Val Accuracy: 0.6667
  Epoch 8/20, Train Loss: 1.6746, Val Loss: 0.6313, Val Accuracy: 0.6111
  Epoch 9/20, Train Loss: 1.7095, Val Loss: 0.6294, Val Accuracy: 0.6111
  Epoch 10/20, Train Loss: 1.6274, Val Loss: 0.6278, Val Accuracy: 0.6111
  Epoch 11/20, Train Loss: 1.7763, Val Loss: 0.6261, Val Accuracy: 0.6667
  Epoch 12/20, Train Loss: 1.6879, Val Loss: 0.6240, Val Accuracy: 0.6111
  Epoch 13/20, Train Loss: 1.7135, Val Loss: 0.6220, Val Accuracy: 0.6111
  Epoch 14/20, Train Loss: 1.5574, Val Loss: 0.6200, Val Accuracy: 0.6111
  Epoch 15/20, Train Loss: 1.5286, Val Loss: 0.6181, Val Accuracy: 0.6111
  Epoch 16/20, Train Loss: 1.5693, Val Loss: 0.6166, Val Accuracy: 0.6111
  Epoch 17/20, Train Loss: 1.5036, Val Loss: 0.6152, Val Accuracy: 0.6667
  Epoch 18/20, Train Loss: 1.6609, Val Loss: 0.6138, Val Accuracy: 0.6667
  Epoch 19/20, Train Loss: 1.6476, Val Loss: 0.6121, Val Accuracy: 0.6667
  Epoch 20/20, Train Loss: 1.4740, Val Loss: 0.6106, Val Accuracy: 0.6667
```


Figure 11: Deep Learning Training Model

After training the models, their performance was evaluated using several metrics, which provide insights into how well the models classify instances into the correct categories. These evaluation metrics help to understand the strengths and weaknesses of each model.

Machine Learning Model Performance

The performance of the selected machine learning models in predicting prostate cancer was evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Table 3: Model Performance Chart

Models	Test Accuracy	Test Precision	Test Recall	Test F1
SVC	0.68	0.70	0.80	0.75
Decision Tree Classifier	0.71	0.73	0.82	0.77
K-Neighbors Classifier	0.62	0.65	0.78	0.71
Random Forest Classifier	0.75	0.73	0.91	0.81
Logistic Regression	0.71	0.71	0.85	0.78
XGBClassifier	0.75	0.73	0.91	0.81
LGBMClassifier	0.74	0.75	0.85	0.80
CatBoostClassifier	0.75	0.73	0.91	0.81
Deep Learning	0.8333	0.8667	0.9250	0.8148

Comparing the performance of the various models trained you can possibly see how each of the models performed with respect to the testing results seen. Deep learning model seems to show more level of accuracy than the other model.

Conclusion

The integration of machine learning models into prostate cancer prediction has significantly enhanced diagnostic accuracy and patient management. This research has demonstrated that machine ensemble algorithms such as Random Forests, XGB classifier and catBoost classifier and Deep learning outperformed traditional classifiers in terms of accuracy, precision, recall, and ROC-AUC. Their superior performance is attributed to their ability to handle complex, non-linear data relationships and minimize overfitting through feature aggregation and boosting techniques.

Recommendation

In the future, a deep learning image-based analysis for MRI/biopsy image-based prostate cancer detection should be considered, a hybrid approach which combines ensemble learning with deep learning with strong

explainability technique can provide the most accurate and clinically reliable prostate cancer prediction and in subsequent research work the dataset used will be localized in place of online dataset.

Reference

Wang, J., Zhang, Q., & Li, S. (2018). *Application of machine learning models for prostate cancer risk prediction using clinical parameters*. BMC Medical Informatics and Decision Making, 18(1), 150.

Zhang, Y., Li, H., & Chen, X. (2020). *Predicting prostate cancer recurrence with Random Forest algorithm based on genomic features*. Frontiers in Oncology, 10, 745.

Zhu, Y., Tang, L., & Li, R. (2019). *Deep learning-based analysis of multiparametric MRI for prostate cancer classification*. European Radiology, 29(2), 1186–1194.

Radovic, M., Ghalwash, M., Filipovic, N., & Obradovic, Z. (2021). *Ensemble learning for improved prostate cancer diagnosis using MRI data*. Artificial Intelligence in Medicine**,** 114, 102038.

Luchini, C., Stubbs, B., Solmi, M., & Veronese, N. (2021). *The history and evolution of artificial intelligence in medicine*. European Journal of Clinical Investigation, 51(4), e13420.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1956). *A proposal for the Dartmouth summer research project on artificial intelligence*. AI Magazine, 27(4), 12–14.

Samuel, A. L. (1959). *Some studies in machine learning using the game of checkers*. IBM Journal of Research and Development**,** 3(3), 210–229.

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). *What is machine learning? A primer for the epidemiologist*. American Journal of Epidemiology, 188(12), 2222– 2239.

Levin, S., Toerper, M., Hamrock, E., Hinson, J. S., Barnes, S., Gardner, H., Dugas, A., & Linton, B. (2018). *Machine-learning-based electronic triage more accurately differentiates patients with respect to clinical outcomes compared with the Emergency Severity Index*. Annals of Emergency Medicine, 71(5), 565–574.e2.

Ahmed, H. U., Bosaily, A. E., Brown, L. C., et al. (2017). *Diagnostic accuracy of multi- parametric MRI and TRUS biopsy in prostate cancer (PROMIS): a paired validating confirmatory study*. The Lancet**,** 389(10071), 815–822.

Lawrence, E. M., Gnanapragasam, V. J., & Douiri, A. (2019). *Predicting prostate cancer risk using machine learning algorithms on clinical data*. BMC Medical Informatics and Decision Making**,** 19(1), 198.

Gann, P. H., Ma, J., Catalona, W. J., & Stampfer, M. J. (2018). *Machine learning–based models for prostate cancer risk prediction using clinical and biochemical data*. Cancer Epidemiology, Biomarkers & Prevention**,** 27(7), 829–837.

Zhu, Y., Wang, L., & Li, R. (2019). *Deep learning in prostate cancer diagnosis using multiparametric MRI: a comparative study with radiologists*. European Radiology, 29(3), 1186–1194.

Ishioka, J., Matsuoka, Y., Uehara, S., et al. (2020). *Ensemble learning for prostate cancer classification using MRI-derived radiomic features.* Scientific Reports, 10(1), 3385.

Nitta, S., Otsuki, H., Takayama, R., et al. (2021). *Machine learning-based prediction of prostate cancer recurrence using genomic and pathological data*. Frontiers in Oncology**,** 11, 647876.