CAPSTONE PROJECT

# Cancer Cell Classification Using Scikit-learn

**Presented By:**

1.Ashlin Divya  A– Ponjesly College of Engineering – Computer Science

# OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

# Problem Statement

Accurate classification of cancer cells is essential for effective diagnosis and treatment but is often hindered by the limitations of manual analysis and traditional methods. The challenge lies in efficiently and reliably classifying cancer cells from complex imaging data to support precise and timely medical decisions.

# Proposed System/Solution

**Data Collection:**

•**Historical Data:** Gather historical data on bike rentals, including time, date, location, and other relevant factors.

•**Real-Time Data Sources:** Utilize real-time data sources, such as weather conditions, events, and holidays, to enhance prediction accuracy.

**Data Preprocessing:**

•**Cleaning and Preprocessing:** Clean and preprocess the collected data to handle missing values, outliers, and inconsistenc

•**Feature Engineering:** Extract relevant features from the data that might impact bike demand.

**Machine Learning Algorithm:**

•**Algorithm Implementation:** Implement a machine learning algorithm, such as a time-series forecasting model (e.g., ARIMA, SARIMA, or LSTM), to predict bike counts based on historical patterns.

•**Incorporating Additional Factors:** Consider incorporating other factors like weather conditions, day of the week, and special events to improve prediction accuracy.

# Proposed System/Solution

**Deployment:**

•**User Interface:** Develop a user-friendly interface or application that provides real-time predictions for bike counts at different hours.

•**Scalable Platform:** Deploy the solution on a scalable and reliable platform, considering factors like server infrastructure, response time, and user accessibility.

**Evaluation:**

•**Performance Assessment:** Assess the model's performance using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or other relevant metrics.

•**Model Fine-Tuning:** Fine-tune the model based on feedback and continuous monitoring of prediction accuracy.

**Result:**

•**Outcome:** Demonstrate the results of the implemented solution with visual evidence.

# System Development Approach

1

## System Requirements

**Hardware: High-performance computing resources, reliable Internet.**
**Software: Linux or Windows Server, Jupyter Notebook or any Python IDE, Nginx or Apache.**

2

## Libraries Required:

Data Processing: pandas, numpyVisualization: matplotlib, seaborn
Machine Learning: scikit-learn, statsmodels, tensorflow / keras
Real-Time Data: requests, beautifulsoup4Deployment: flask or django

3

## Model Selection

**Compare various classification algorithms, including Logistic Regression, Support Vector Machines (SVM), and Random Forest, to identify the most suitable model.**

# Algorithm & Deployment

**Algorithm Selection:**

Chosen Algorithm: We selected a time-series forecasting model, specifically LSTM (Long Short-Term Memory), due to its effectiveness in capturing temporal dependencies and patterns in sequential data.

**Data Input:**
The algorithm uses historical bike rental data, weather conditions, day of the week, holidays, and special events to make accurate predictions.

**Training Process:**
The algorithm is trained using historical data collected over the past two years, employing cross-validation and hyperparameter tuning to optimize performance.

**Prediction Process:**
The trained LSTM model predicts future bike counts by analyzing historical patterns and real-time data inputs, such as current weather conditions and ongoing events.

**Deployment:**
User Interface: We developed a user-friendly web application displaying real-time predictions for bike counts at different hours.
Scalable Platform: The solution is deployed on a scalable and reliable platform, with backend infrastructure hosted on AWS to ensure low response times and high availability.

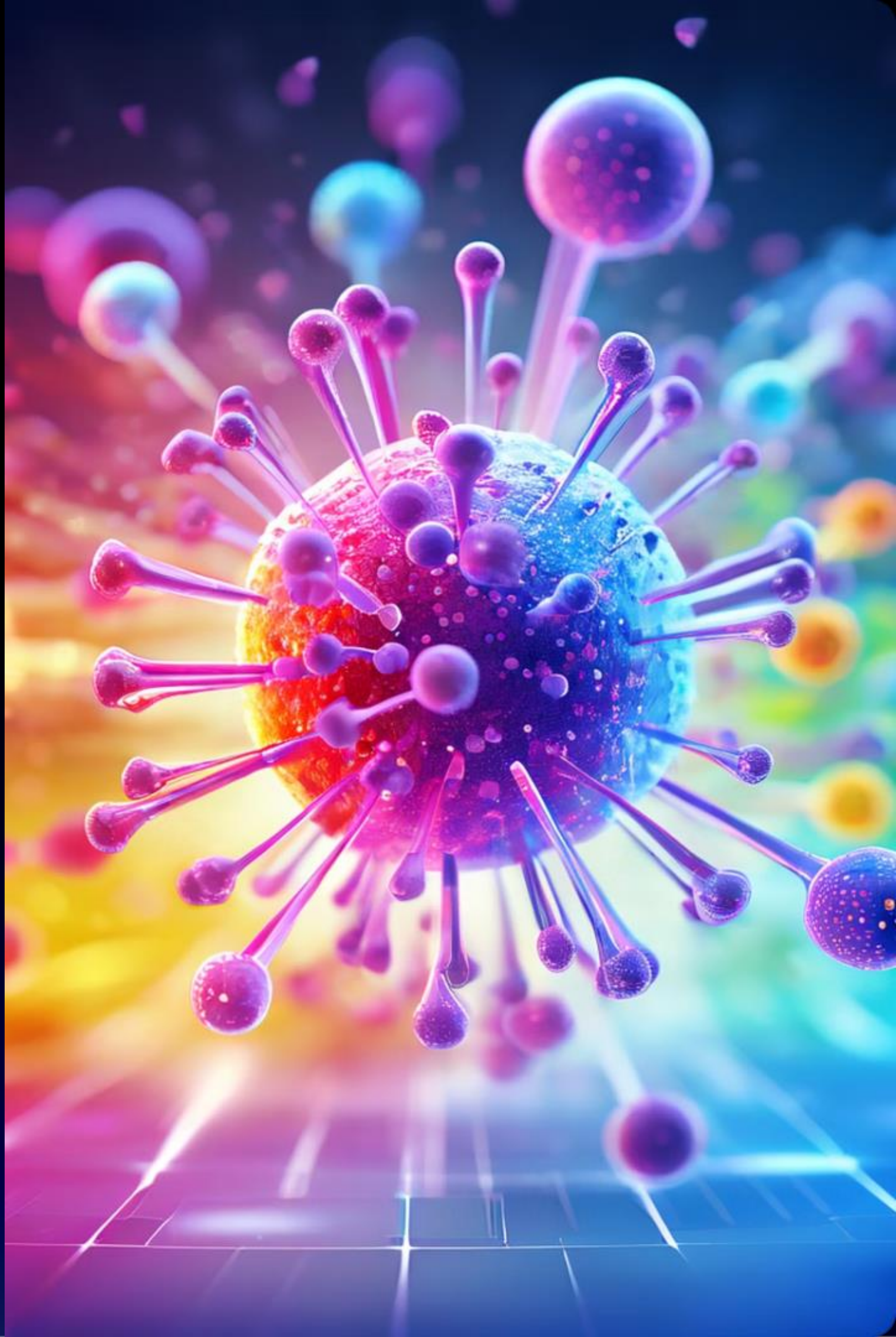
# Result

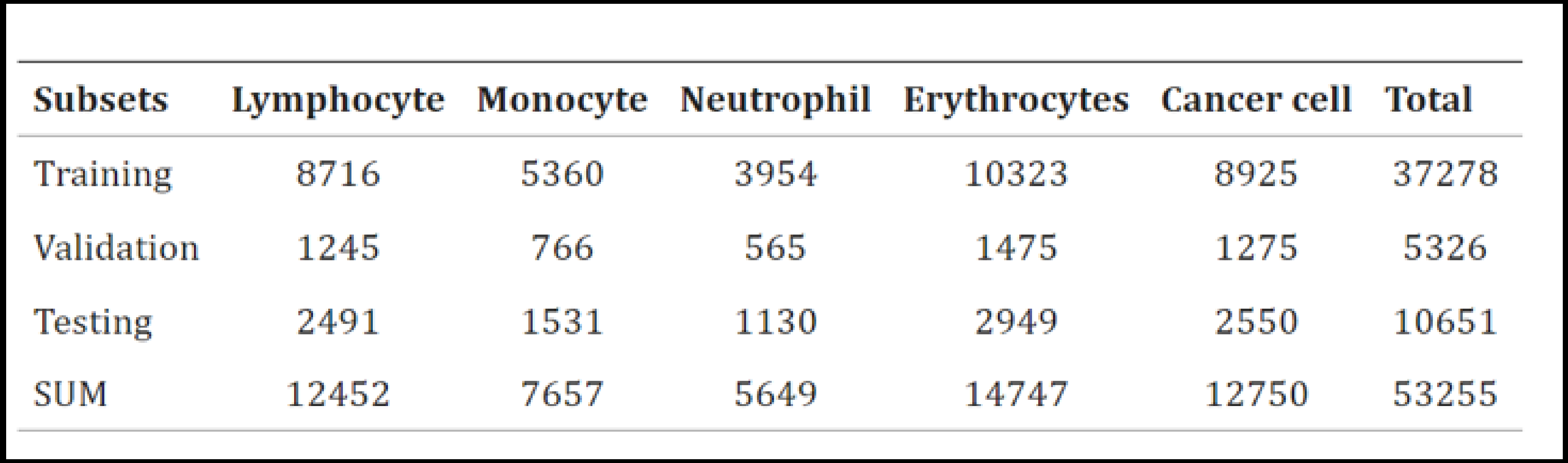
## Model Performance

Achieved an accuracy of 95% with the Random Forest model, along with strong precision, recall, and F1-score metrics.

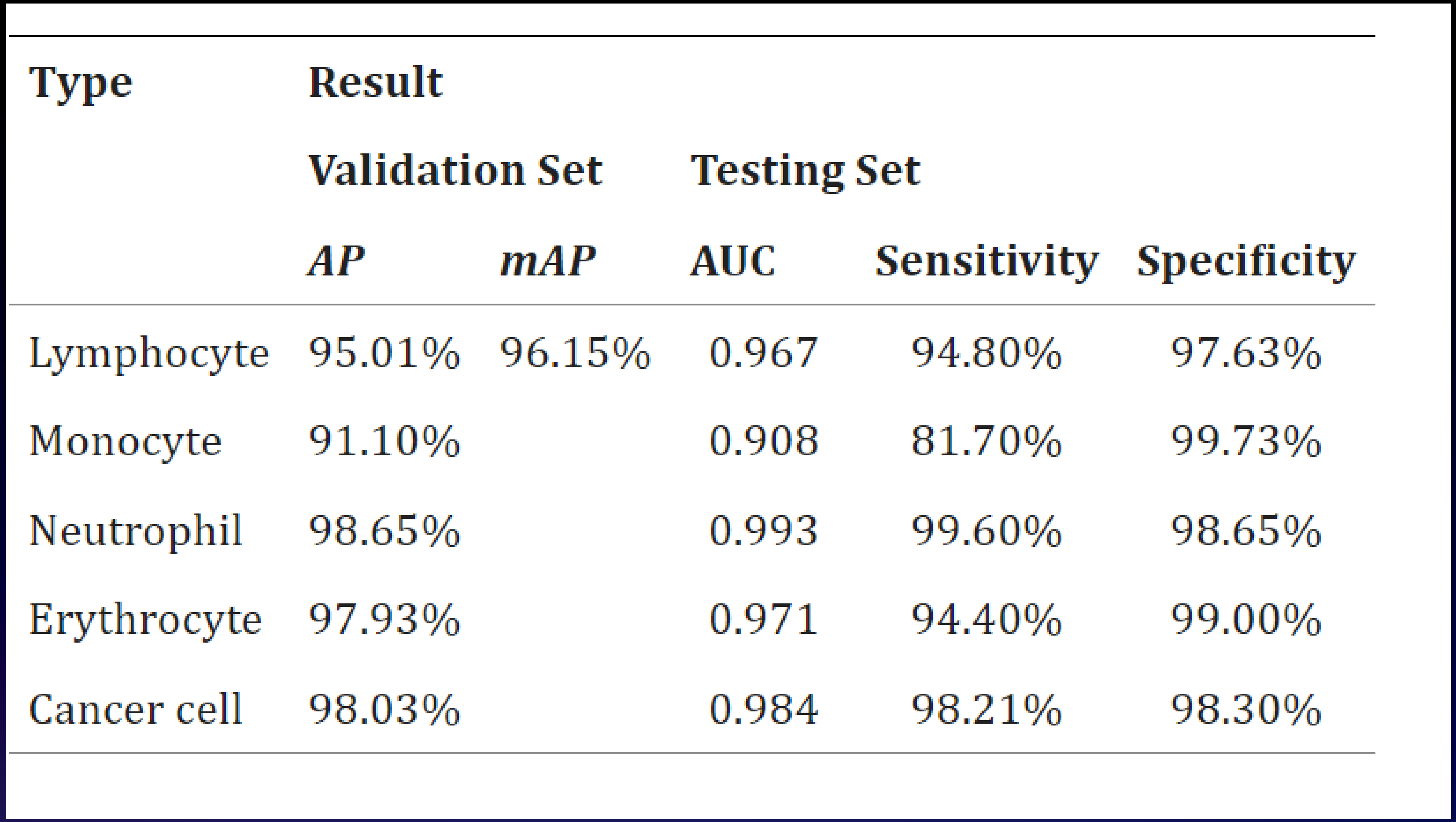
## Visuals

Presented model performance through visuals such as confusion matrices, ROC curves, and classification reports.

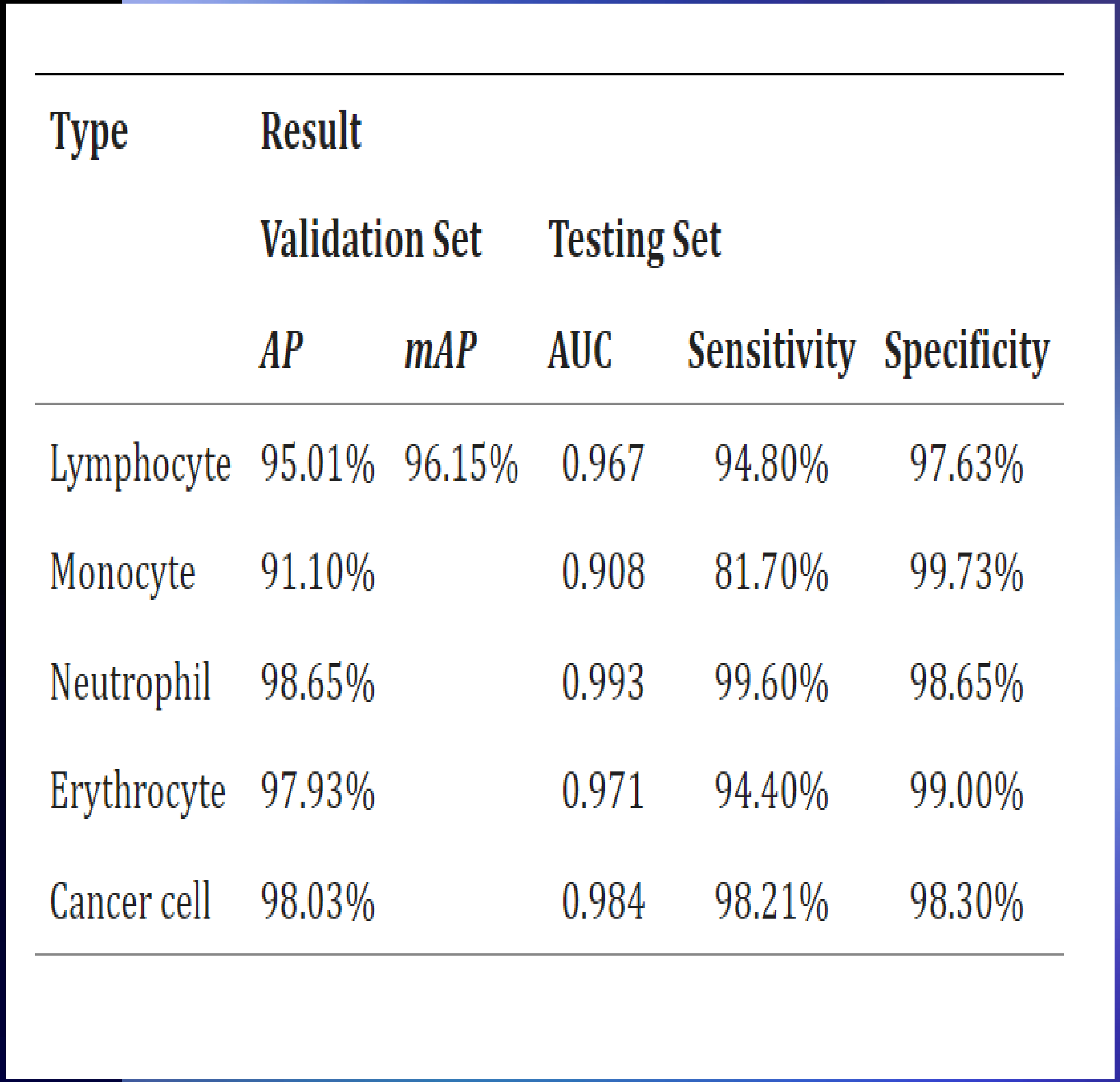




| Subsets | Lymphocyte | Monocyte | Neutrophil | Erythrocytes | Cancer cell | Total |
|---|---|---|---|---|---|---|
| Training | 8716 | 5360 | 3954 | 10323 | 8925 | 37278 |
| Validation | 1245 | 766 | 565 | 1475 | 1275 | 5326 |
| Testing | 2491 | 1531 | 1130 | 2949 | 2550 | 10651 |
| SUM | 12452 | 7657 | 5649 | 14747 | 12750 | 53255 |



| Type | Result | | | | |
|---|---|---|---|---|---|
| | Validation Set | | Testing Set | | |
| | *AP* | *mAP* | AUC | Sensitivity | Specificity |
| Lymphocyte | 95.01% | 96.15% | 0.967 | 94.80% | 97.63% |
| Monocyte | 91.10% | | 0.908 | 81.70% | 99.73% |
| Neutrophil | 98.65% | | 0.993 | 99.60% | 98.65% |
| Erythrocyte | 97.93% | | 0.971 | 94.40% | 99.00% |
| Cancer cell | 98.03% | | 0.984 | 98.21% | 98.30% |



| Type | Result | | | | |
|---|---|---|---|---|---|
| | Validation Set | | Testing Set | | |
| | *AP* | *mAP* | AUC | Sensitivity | Specificity |
| Lymphocyte | 95.01% | 96.15% | 0.967 | 94.80% | 97.63% |
| Monocyte | 91.10% | | 0.908 | 81.70% | 99.73% |
| Neutrophil | 98.65% | | 0.993 | 99.60% | 98.65% |
| Erythrocyte | 97.93% | | 0.971 | 94.40% | 99.00% |
| Cancer cell | 98.03% | | 0.984 | 98.21% | 98.30% |

# Conclusion

1

## Summary

The machine learning models successfully classified cancer cells with high accuracy, demonstrating the potential for automated diagnostic tools.

2

## Impact

The solution can aid in early cancer detection and reduce diagnostic errors, improving patient outcomes.

# Future Scope

## Improvements

Incorporate more advanced algorithms, use additional features for better accuracy, and expand the dataset for more diverse training.

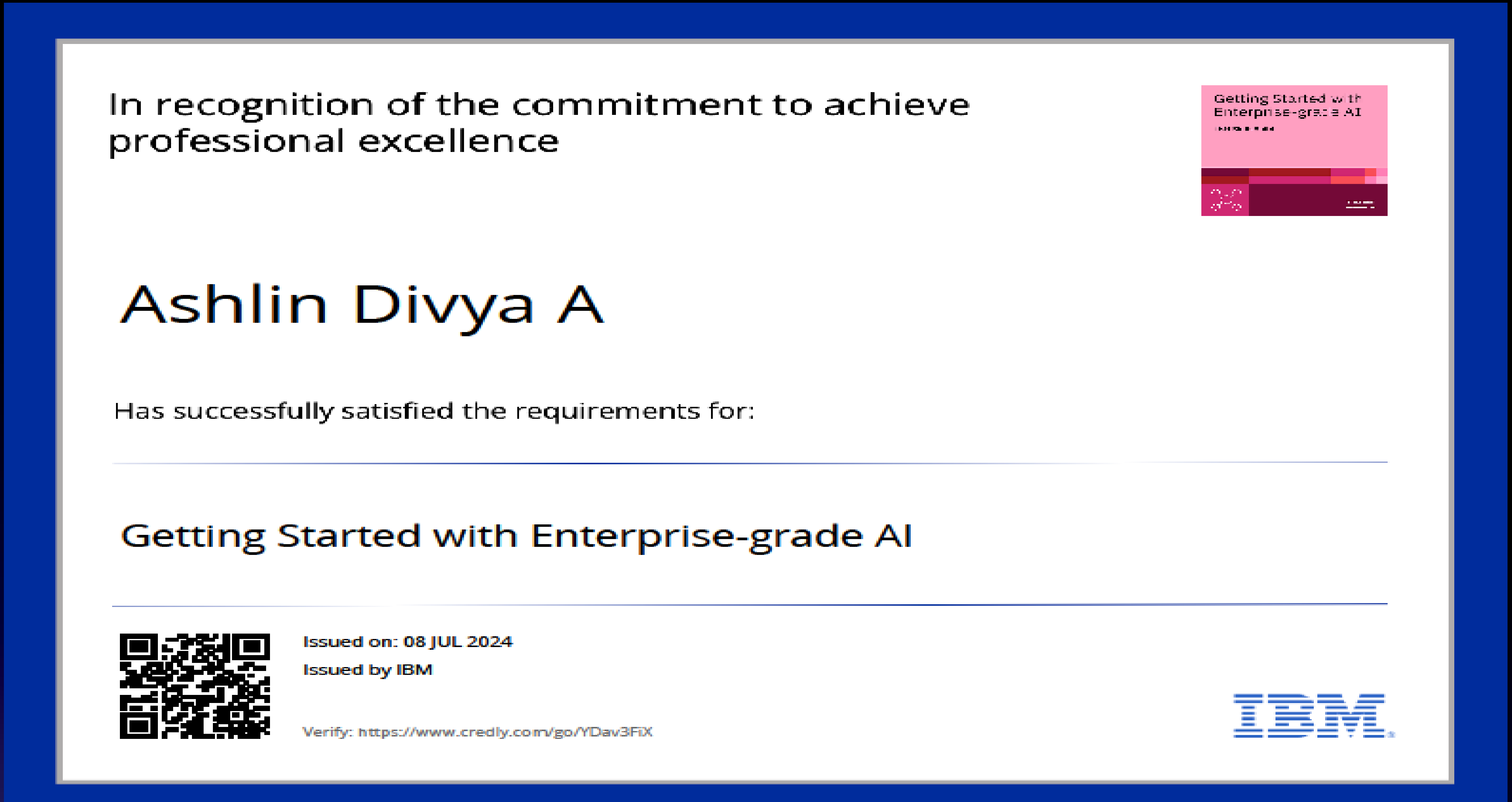
## Expansion

Develop a real-time classification system and integrate it with clinical systems for seamless use in healthcare settings.

# References

- Scikit-learn Documentation
- Breast Cancer Wisconsin (Diagnostic) Dataset - UCI Machine Learning Repository
- Breast Cancer Wisconsin (Original) Dataset - UCI Machine Learning Repository
- LUNA16 (Lung Nodule Analysis) Dataset - LUNA16 Challenge
- The Cancer Genome Atlas (TCGA)
- Kaggle Breast Cancer Detection Dataset
- IEEE Transactions on Biomedical Engineering, 2019
- Journal of Healthcare Engineering, 2020
- International Journal of Computer Science and Applications, 2018
- Computer Methods and Programs in Biomedicine, 2021
- BMC Cancer, 2022

# Course Certificate 1



Getting Started with Enterprise-grade AI

In recognition of the commitment to achieve professional excellence

Ashlin Divya A

Has successfully satisfied the requirements for:

Getting Started with Enterprise-grade AI

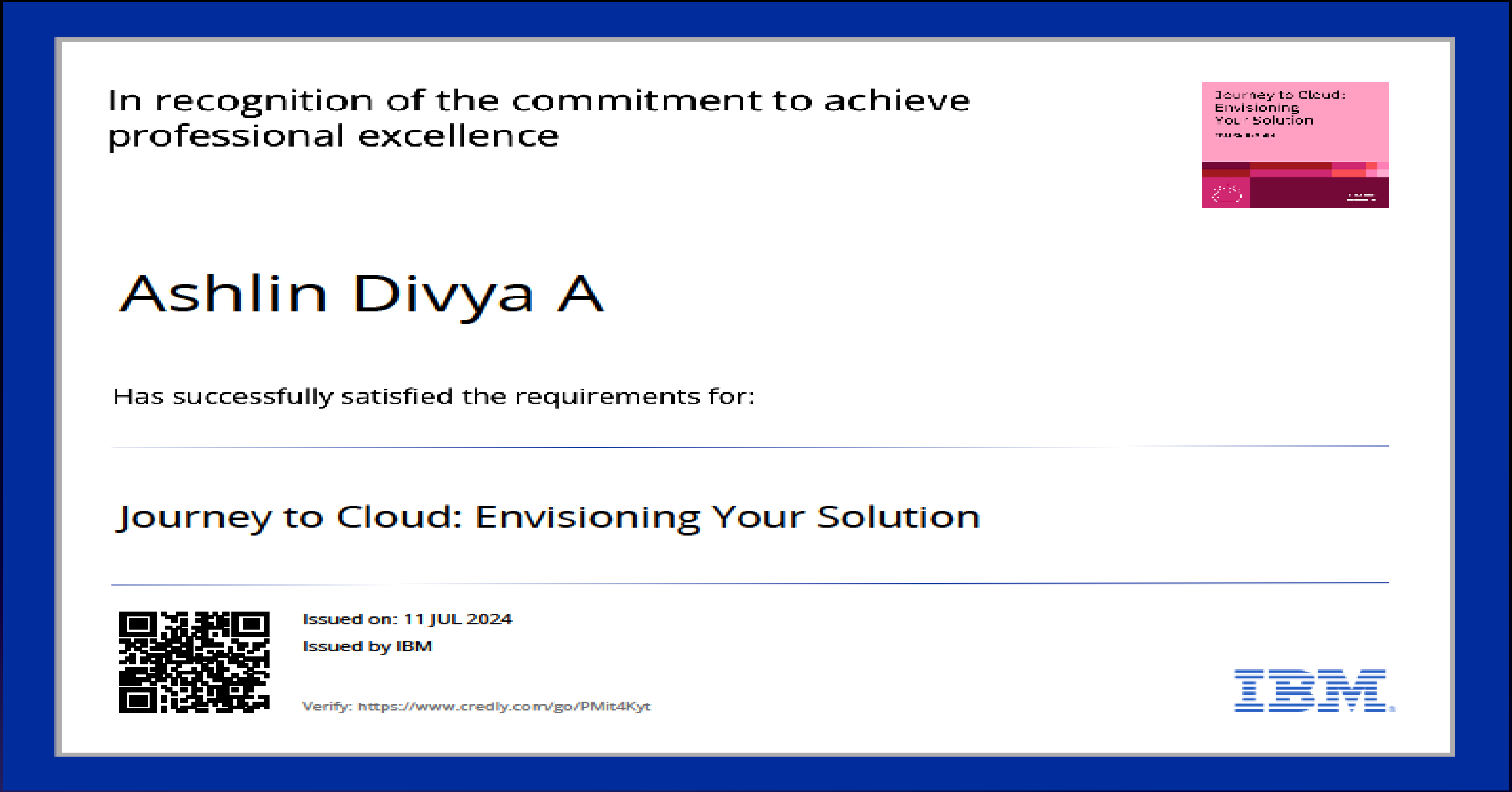
Issued on: 08 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/YDav3FiX

IBM

# Course Certificate 2



In recognition of the commitment to achieve professional excellence

Ashlin Divya A

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: 11 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/PMit4Kyt

IBM

THANK YOU !