CAPSTONE PROJECT

CORONARY HEART DISEASE RISK PREDICTION

Presented By:
Student Name- Mayank Mahapatra
College Name- Siksha O Anusandhan
Department – Computer Science Engineering

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

- The dataset is from an ongoing cardiovascular study on residents of a town. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). The dataset provides the patients' information. It includes over 4,000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.

PROPOSED SOLUTION

- The proposed solution involves training machine learning models to predict the 10-year risk of CHD.
- We will use Logistic Regression, Random Forest, Naive Bayes, XGBoost and Support Vector Machine classifiers to build predictive models.
- The models will be evaluated using Accuracy, Precision, Recall, F1 macro, and ROC-AUC.

SYSTEM APPROACH

- **System Requirements:**
  - A computer with a Python environment installed.
  - Access to the house price dataset.
  - Sufficient memory and processing power to handle data preprocessing, model training, and prediction.

- **Libraries Required to Build the Model:**
  - **Pandas**: Used for data manipulation and analysis. It provides data structures and functions needed to manipulate structured data.
  - **Matplotlib and Seaborn**: Used for data visualization. They provide a flexible and powerful declarative framework for creating static, animated, and interactive visualizations in Python.
  - **Scikit-learn**: Used for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction.
  - **NumPy**: Used for numerical computations and working with arrays.

SYSTEM APPROACH

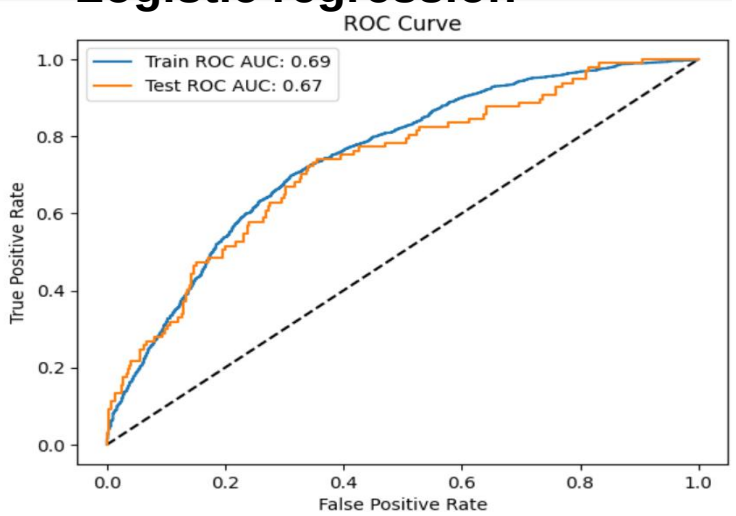
- **Data Preprocessing**
  - **Handling Missing Values**: Imputation using KNN.
  - **Encoding Categorical Variables**: Label encoding.
  - **Feature Scaling**: Standardization of numerical features.
  - **Balancing Classes**: Applying SMOTE (Synthetic Minority Over-sampling Technique).
- **Model Training**: Train Logistic Regression, Random Forest, Naive Bayes, XGBoost, and Support Vector Machine classifiers.
- **Model Evaluation**: Evaluate the models using Accuracy, Precision, Recall, F1 macro, and ROC-AUC.

ALGORITHM & DEPLOYMENT

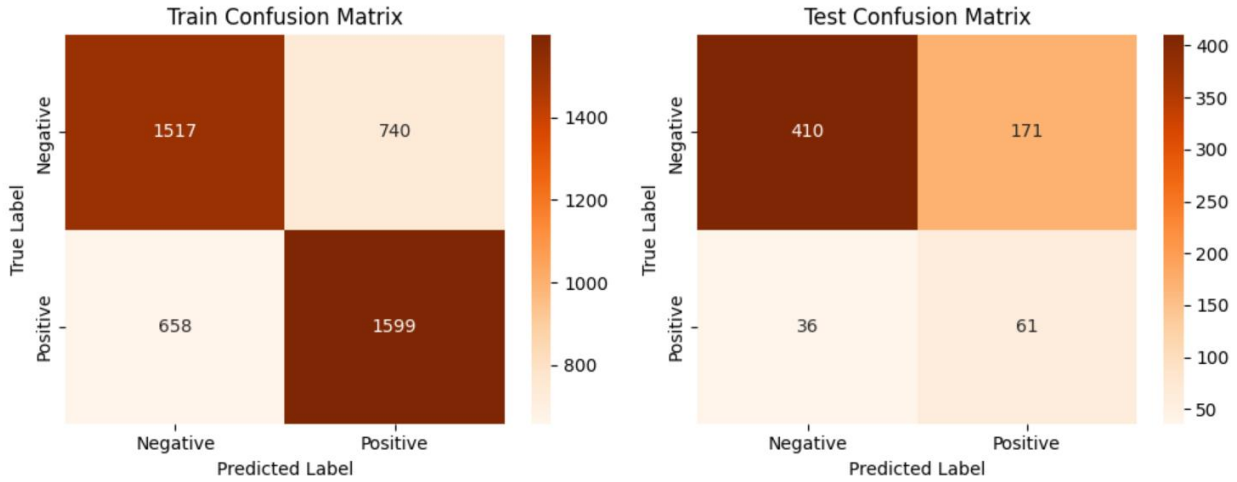
- **Algorithms Used**:
  - Logistic Regression
  - Random Forest Classifier
  - XGBoost Classifier
  - Naive Bayes Classifier
  - Support Vector Machine Classifiers
- **Deployment Steps**:
  - 1. Train the models using the training dataset.
  - 2. Evaluate the models on the testing dataset.
  - 3. Choose the best-performing model based on evaluation metrics.

RESULT

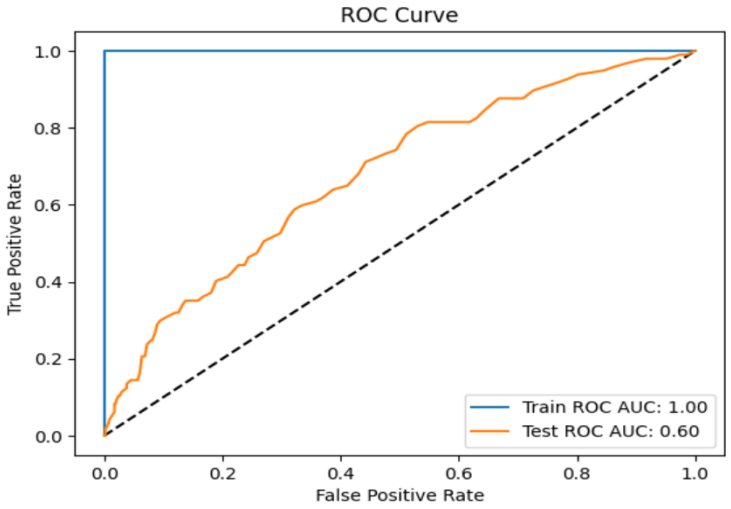
Logistic regression

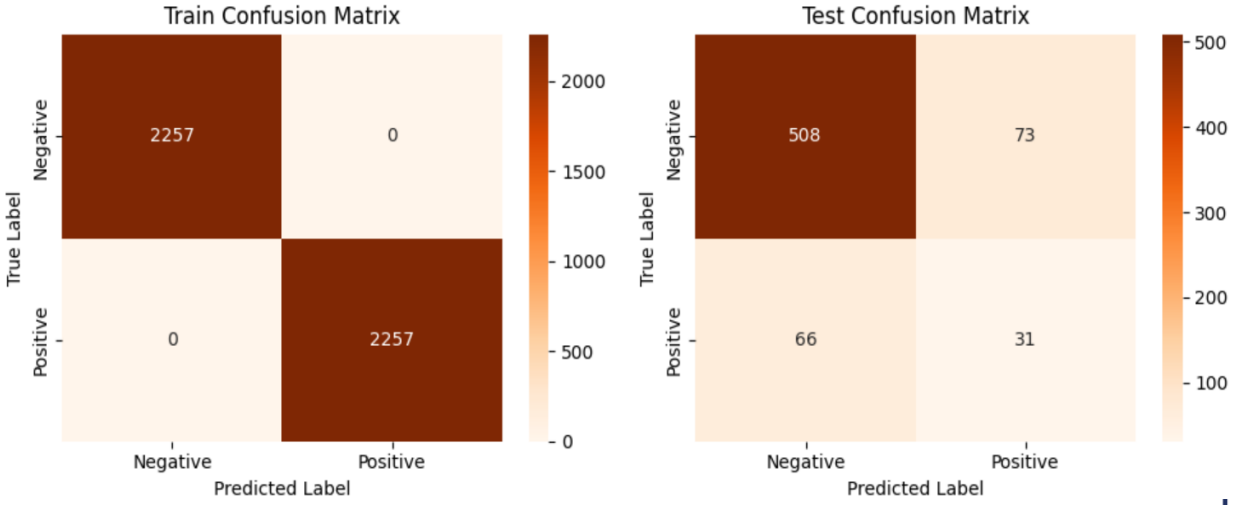


Confusion Matrix:



Random forest

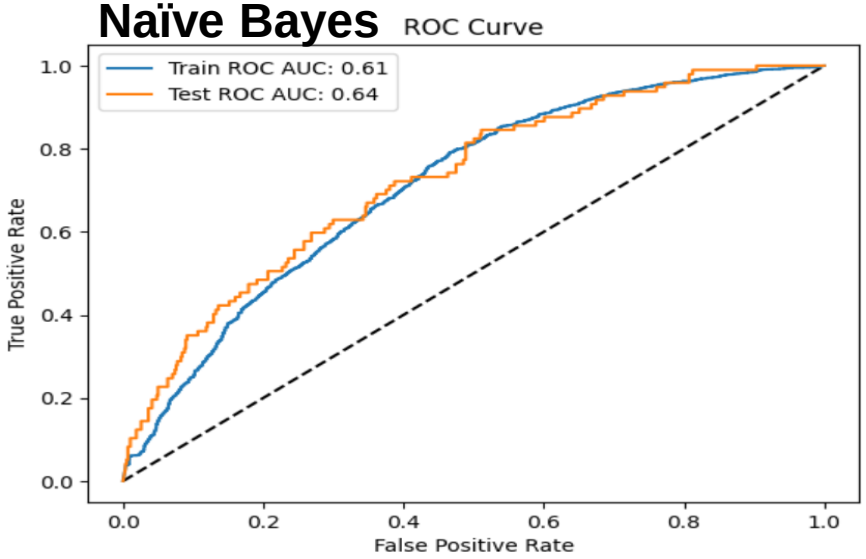


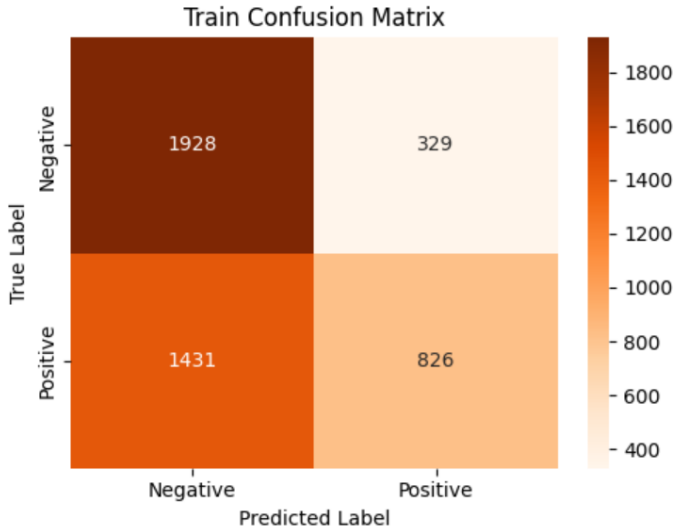


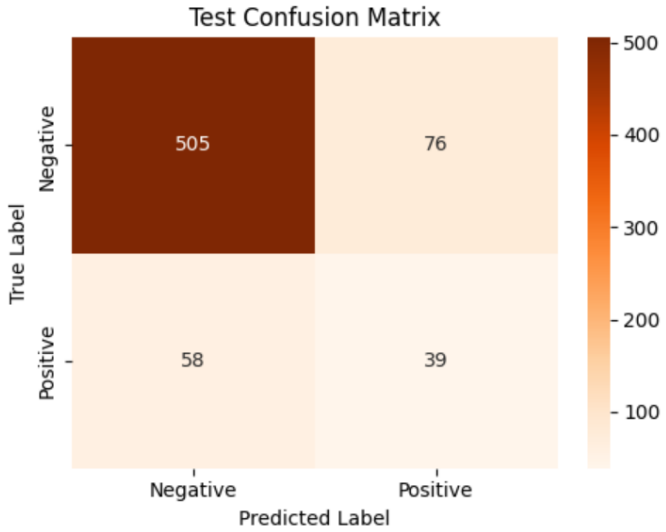
edunet foundation

RESULT

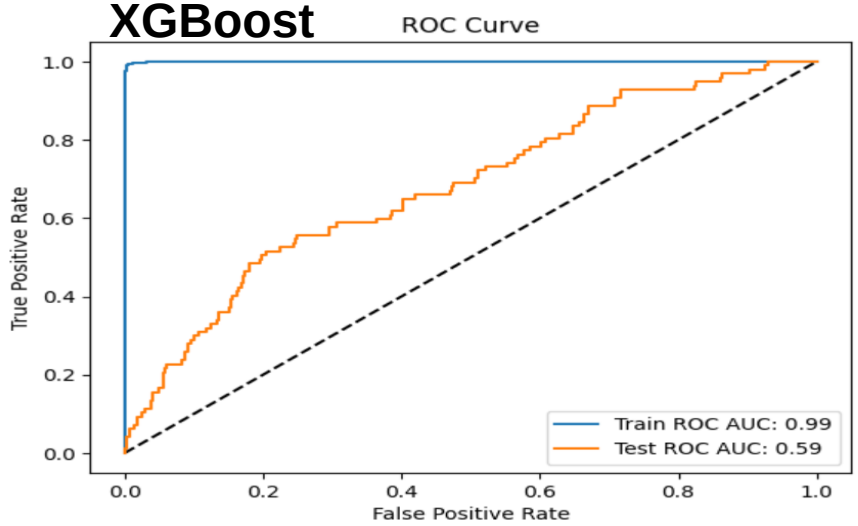
Naïve Bayes

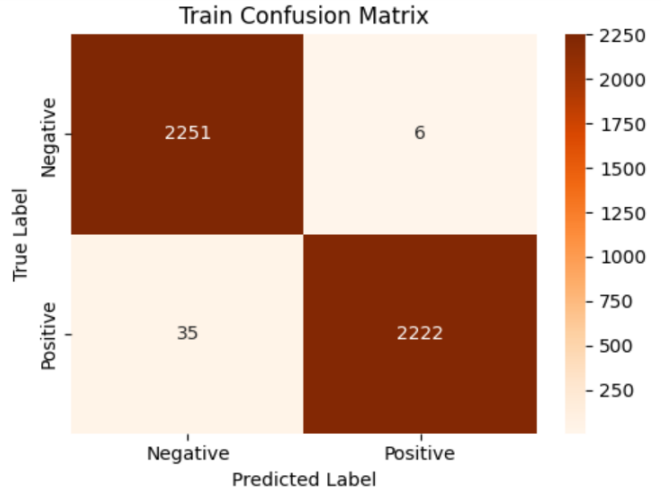


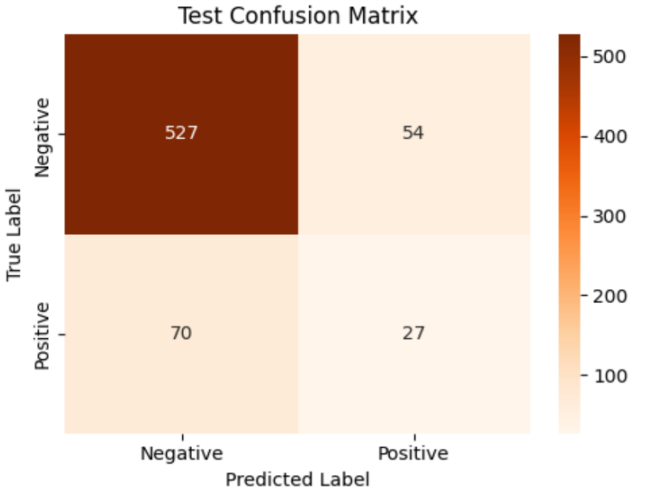




XGBoost



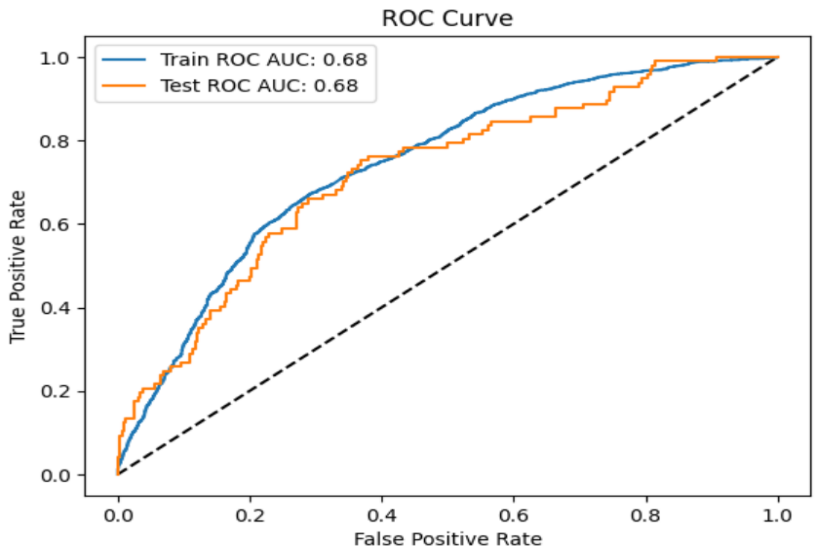


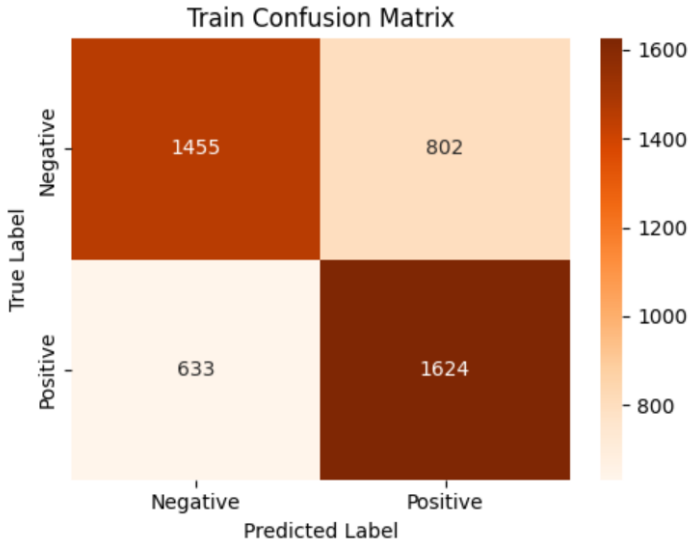


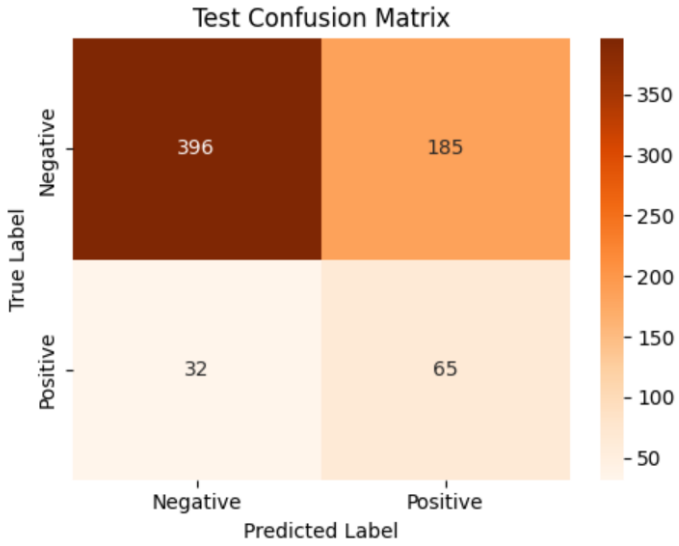
edunet foundation

RESULT

Support Vector Machine







Score

	Precision Train	Precision Test	Recall Train	Recall Test	Accuracy Train	Accuracy Test	ROC-AUC Train	ROC-AUC Test	F1 macro Train	F1 macro Test
Logistic regression	0.690548	0.825380	0.690297	0.694690	0.690297	0.694690	0.690297	0.667273	0.690195	0.737263
Random Forest	1.000000	0.801045	1.000000	0.794985	1.000000	0.794985	1.000000	0.596971	1.000000	0.797934
XGB	0.990998	0.804144	0.990917	0.817109	0.990917	0.817109	0.990917	0.592704	0.990917	0.810131
Naive Bayes	0.644566	0.817170	0.610102	0.802360	0.610102	0.802360	0.610102	0.635626	0.585392	0.809195
SVM	0.683127	0.830060	0.682100	0.679941	0.682100	0.679941	0.682100	0.675843	0.681654	0.726235

CONCLUSION

- In our study to predict the 10-year risk of coronary heart disease (CHD). Each algorithm has its strengths and weaknesses, which make it suitable for different aspects of the problem.
- Ensuring that overfitted algorithms are not considered, so Random Forest and XGB Classifier were removed.

	Precision Train	Precision Test	Recall Train	Recall Test	Accuracy Train	Accuracy Test	ROC-AUC Train	ROC-AUC Test	F1 macro Train	F1 macro Test
Logistic regression	0.690548	0.82538	0.690297	0.694690	0.690297	0.694690	0.690297	0.667273	0.690195	0.737263
Naive Bayes	0.644566	0.81717	0.610102	0.802360	0.610102	0.802360	0.610102	0.635626	0.585392	0.809195
SVM	0.683127	0.83006	0.682100	0.679941	0.682100	0.679941	0.682100	0.675843	0.681654	0.726235

- These results indicate that the Random Forest algorithm is the best in terms of Recall, Accuracy, and F1 Macro, while the SVM algorithm performs best in terms of ROC-AUC and Precision.

```
The best models are:
Precision: SVM - 0.8301
Recall: Naive Bayes - 0.8024
Accuracy: Naive Bayes - 0.8024
ROC-AUC: SVM - 0.6758
F1 macro: Naive Bayes - 0.8092
```

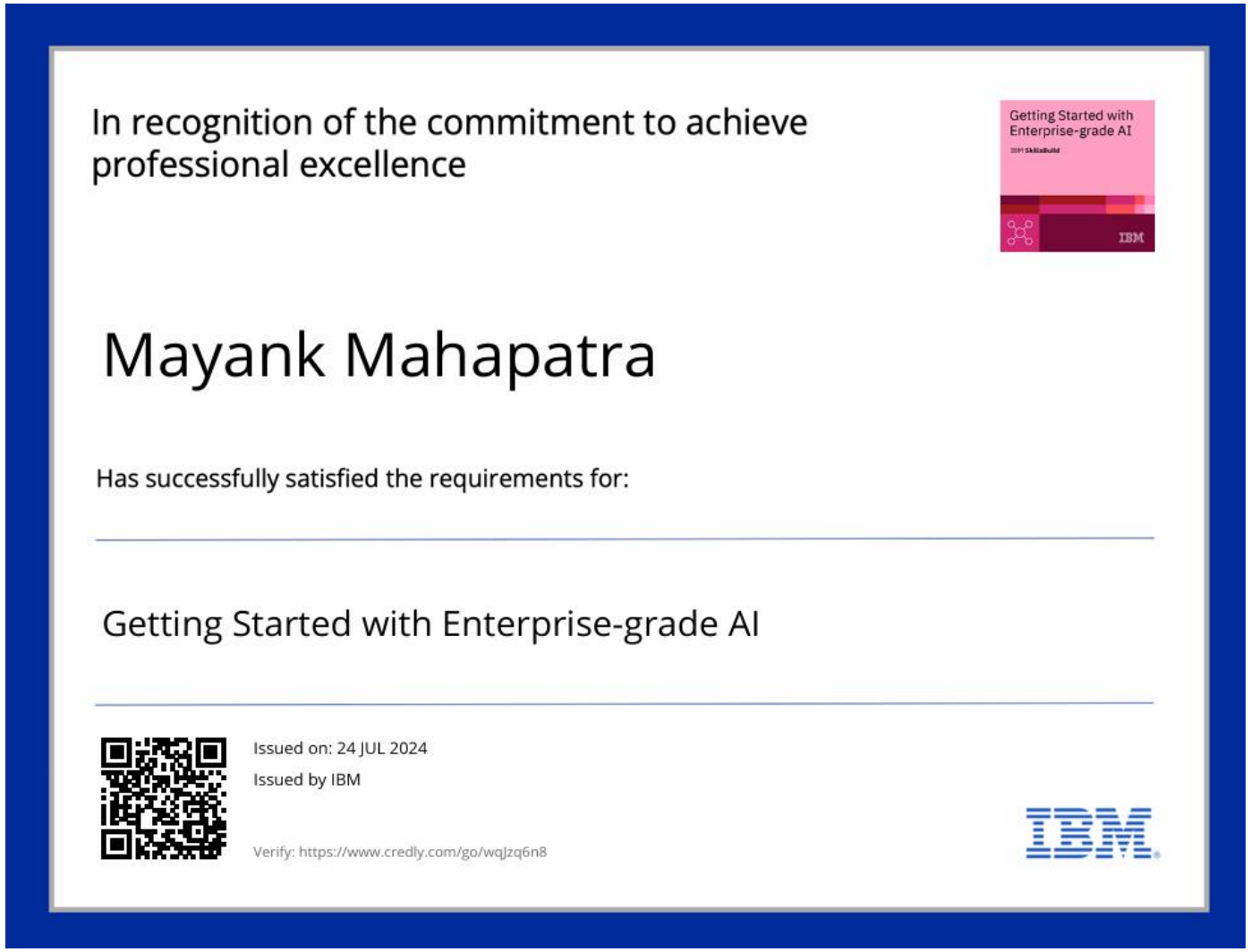
FUTURE SCOPE

- **Feature Engineering and Selection**
  - •Incorporate additional features like genetic data and lifestyle factors.
  - •Use advanced feature selection techniques.
- **Advanced Machine Learning Techniques**
  - •Optimize hyperparameters.
  - •Explore ensemble methods and deep learning models.
- **Handling Imbalanced Data**
  - •Use advanced sampling techniques and anomaly detection methods.
- **Explainability and Interpretability**
  - •Implement SHAP or LIME for model transparency.
  - •Develop user-friendly decision support systems.

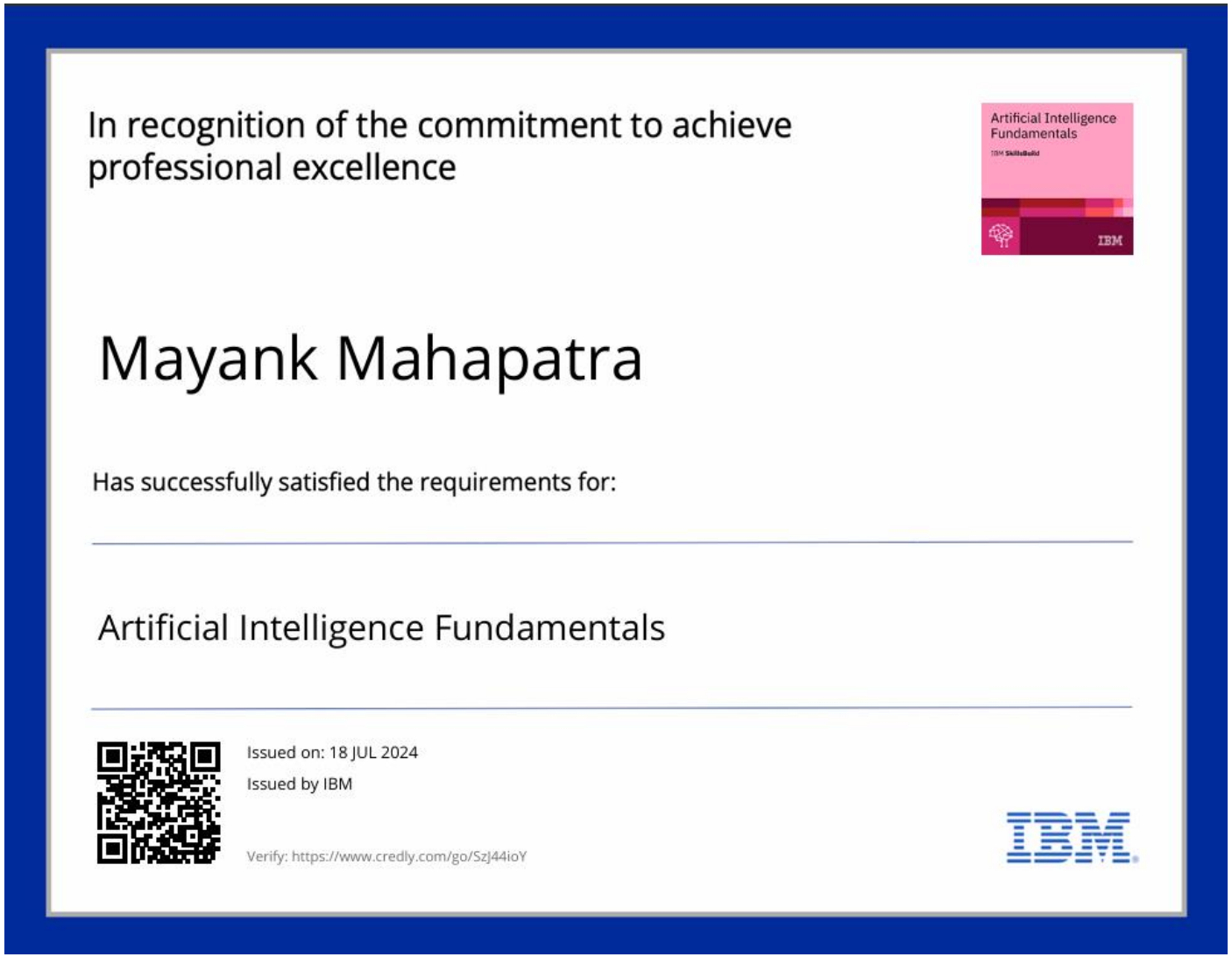
REFERENCES

- https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data
- https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html
- https://seaborn.pydata.org/https://matplotlib.org/stable/contents.html
- https://matplotlib.org/stable/contents.html
- https://scikit-learn.org/stable/user_guide.html

CERTIFICATE 1



CERTIFICATE 2



THANK YOU

edunet foundation