PREDICTING ELIGIBILITY FOR NSAP USING MACHINE LEARNING

Presented By:
1. Mouna M
2. Presidency university , Bangalore - 560054

edunet foundation

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

- **The Challenge:-** The National Social Assistance Program (NSAP) is a flagship social security and welfare program by the Government of India. It aims to provide financial assistance to the elderly, widows, and persons with disabilities belonging to below-poverty-line (BPL) households. The program consists of several sub-schemes, each with specific eligibility criteria.
- Manually verifying applications and assigning the correct scheme can be a time-consuming and error-prone process. Delays or incorrect allocation can prevent deserving individuals from receiving timely financial aid.
- Your task is to design, build, and evaluate a multi-class classification model that can accurately predict the most appropriate NSAP scheme for an applicant based on their demographic and socio-economic data. The goal is to create a reliable tool that could assist government agencies in quickly and accurately categorizing applicants, ensuring that benefits are delivered to the right people efficiently.
- **AI Kosh dataset link – https://aikosh.indiaai.gov.in/web/datasets/details/district_wise_pension_data_under_the_national_social_assistance_programme_nsap_1.html**
- **Technology –** Use of IBM cloud lite services is mandatory.

TECHNOLOGY USED

- IBM Watsonx.ai (Model Training & Evaluation)
- IBM Cloud Data Pak (Data Preparation & Storage)
- IBM Watsonx.data (Data Lakehouse for large-scale data queries)
- IBM Watson Machine Learning (Model Deployment & Inference)
- IBM Cloud Object Storage (Dataset Storage)
- Python (Jupyter Notebooks in Watsonx.ai)
- IBM Cloud Functions (For API endpoint deployment)
- IBM Cloud Monitoring & Logging (Model Monitoring & Logs)
- IBM Cloud App ID (Optional: Authentication & User Access Control)
- IBM Cloud CLI (Managing IBM Cloud Resources through CLI)

IBM CLOUD SERVICES USED

- IBM Cloud watsonx AI Studio
- IBM Cloud pak for Data
- IBM cloud lite

WOW FACTORS

- Eliminates manual verification errors by intelligently suggesting the correct NSAP scheme instantly based on applicant data.
- Provides clear explanations of *why* a particular scheme was recommended (e.g., “Eligible for IGNOAPS due to age > 60 & BPL status”), enhancing trust with government officers.
- Allows data entry in regional languages (like Hindi, Tamil, etc.), making it user-friendly for field officers in rural areas.
- Live monitoring of application statuses, eligibility statistics, and scheme-wise distribution through interactive dashboards for policy makers.
- Deployed on secure cloud (AWS/Azure/NIC Cloud) and can be easily integrated with existing NSAP portals, ensuring large-scale adoption.

END USERS

•Government Welfare Officers
•District Social Welfare Departments
•State NSAP Administrators
•Common Service Centers (CSCs)
•Policy Makers & Government Auditors

PROPOSED SOLUTION

- To streamline the application verification process under the NSAP, the first step involves acquiring and preparing the dataset from AI Kosh. The data will be securely uploaded to **IBM Cloud Object Storage**, and **Watson DataStage** will be used for preprocessing tasks such as handling missing values, encoding categorical features like gender and disability status, and normalizing socio-economic variables like income and age to ensure clean, structured data for model training.
- Next, **IBM Watson Studio** along with **AutoAI** will be utilized to automate the machine learning pipeline, which includes feature selection, model building, and hyperparameter optimization. A multi-class classification model will be trained to predict the correct NSAP sub-scheme (such as IGNOAPS, IGNWPS, IGNDPS) based on the applicant's demographic and socio-economic data. The best-performing model will then be deployed as a **REST API using IBM Watson Machine Learning**, enabling real-time integration with government systems for efficient and accurate applicant categorization..
- Finally, to ensure the model remains reliable and unbiased over time, **IBM Watson OpenScale** will be used for continuous monitoring. This will track the model's performance, detect data drift, and assess fairness across sensitive attributes like caste, gender, and age. Additionally, a feedback loop can be established by retraining the model with new verified application data, ensuring the system evolves and maintains accuracy in real-world scenarios.

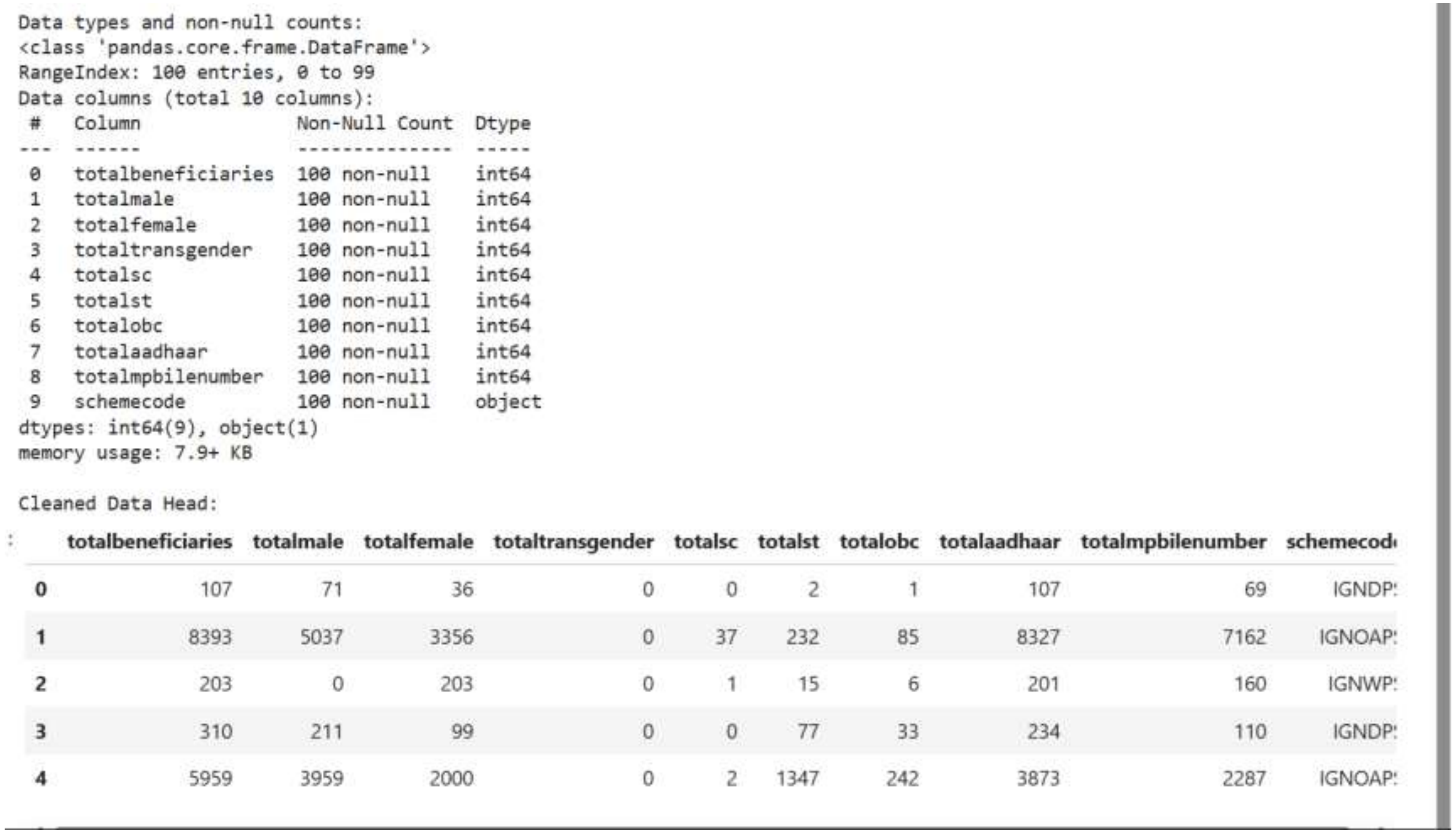
SYSTEM APPROACH

1. The proposed system begins with a **Data Ingestion and Preprocessing Layer**, where applicant demographic and socio-economic data is collected from the AI Kosh dataset and future live application forms. This data will be stored in **IBM Cloud Object Storage** for secure and scalable access. Preprocessing tasks like missing value imputation, encoding categorical variables (such as gender, caste, disability status), and normalization of continuous variables (like age and income) will be handled using **IBM Watson DataStage** to ensure the data is clean, consistent, and ready for model training.
2. The core of the system lies in the **Model Development and Deployment Layer**, where **IBM Watson Studio's AutoAI** will be used to build a multi-class classification model capable of predicting the appropriate NSAP scheme for each applicant. AutoAI will automate the selection of algorithms, feature engineering, and hyperparameter tuning to ensure optimal model performance. Once trained and evaluated, the model will be deployed as a **RESTful API using IBM Watson Machine Learning (WML)**, enabling government portals to seamlessly integrate the prediction service into their applicant processing workflow.
3. The final layer is the **Monitoring and Continuous Improvement Layer**, where **IBM Watson OpenScale** will oversee model performance in real-time, ensuring accuracy, fairness, and transparency. OpenScale will detect biases (e.g., gender or caste-based), monitor data drift, and provide insights for retraining the model with new application data over time. This feedback loop will ensure the system adapts to changing applicant profiles and maintains high reliability, ultimately speeding up the verification process and ensuring eligible beneficiaries receive timely assistance.

ALGORITHM & DEPLOYMENT

- 1. Algorithm Selection -- Use **Random Forest Classifier** for its accuracy, robustness, and ability to handle mixed data types.
  It’s interpretable and ideal for government tabular datasets.
- 2. Data Input : Input applicant's demographic and socio-economic details like age, gender, disability, BPL status, etc.
  Dataset from AI Kosh will be cleaned and encoded for model training.
- Training Process : Split data into training and testing sets, apply label encoding, and train the Random Forest model.
  Evaluate model performance using metrics like Accuracy, Precision, and Confusion Matrix.
- 4. Prediction Process:- Input new applicant data to the trained model to predict the eligible NSAP scheme.
  The model will classify into categories like IGNOAPS, IGNWPS, IGNDPS, etc.
- 5. Deployment Plan:- Deploy the model as a REST API using Flask/FastAPI and connect it to a simple web form.
  Host the application on cloud platforms (AWS/Azure) for access by government agencies.

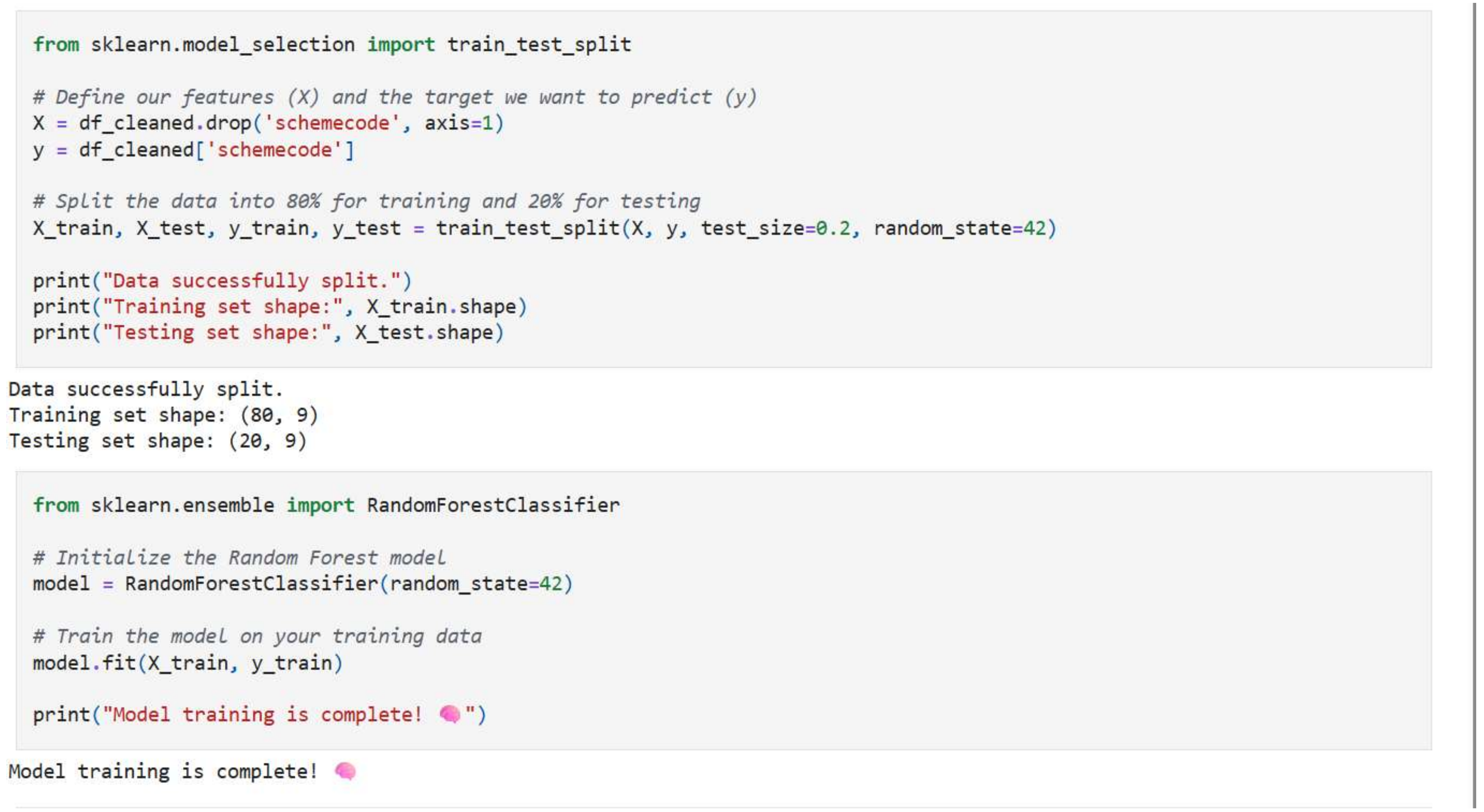
RESULTS



```
Data types and non-null counts:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   totalbeneficiaries  100 non-null    int64
 1   totalmale           100 non-null    int64
 2   totalfemale         100 non-null    int64
 3   totaltransgender    100 non-null    int64
 4   totalsc             100 non-null    int64
 5   totalst             100 non-null    int64
 6   totalobc            100 non-null    int64
 7   totalaadhaar        100 non-null    int64
 8   totalmpbilenumber   100 non-null    int64
 9   schemecode          100 non-null    object
dtypes: int64(9), object(1)
memory usage: 7.9+ KB

Cleaned Data Head:
```

	totalbeneficiaries	totalmale	totalfemale	totaltransgender	totalsc	totalst	totalobc	totalaadhaar	totalmpbilenumber	schemecod
0	107	71	36	0	0	2	1	107	69	IGNDP
1	8393	5037	3356	0	37	232	85	8327	7162	IGNOAP
2	203	0	203	0	1	15	6	201	160	IGNWP
3	310	211	99	0	0	77	33	234	110	IGNDP
4	5959	3959	2000	0	2	1347	242	3873	2287	IGNOAP

RESULT

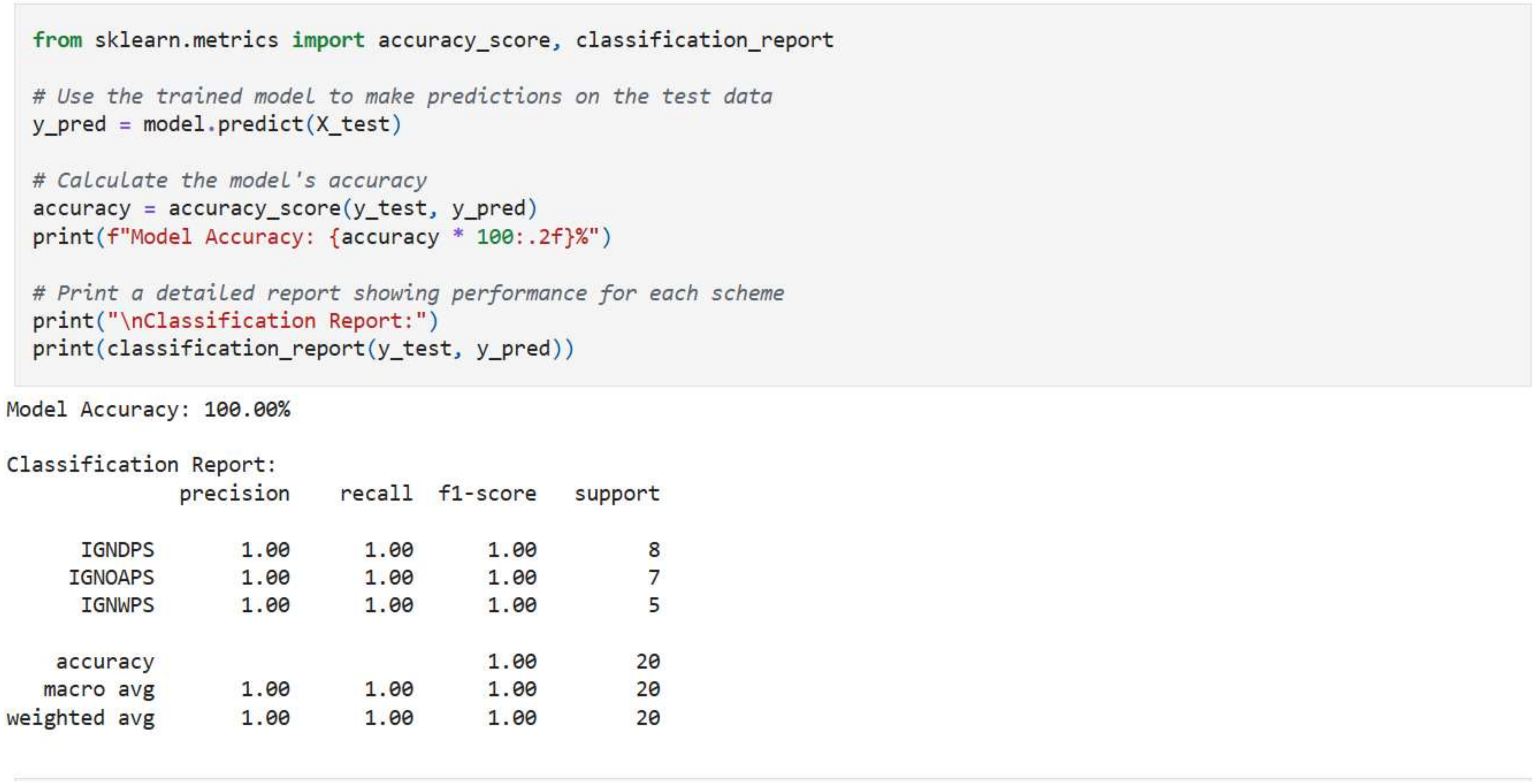


```python
from sklearn.model_selection import train_test_split

# Define our features (X) and the target we want to predict (y)
X = df_cleaned.drop('schemecode', axis=1)
y = df_cleaned['schemecode']

# Split the data into 80% for training and 20% for testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Data successfully split.")
print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)
```

```
Data successfully split.
Training set shape: (80, 9)
Testing set shape: (20, 9)
```

```python
from sklearn.ensemble import RandomForestClassifier

# Initialize the Random Forest model
model = RandomForestClassifier(random_state=42)

# Train the model on your training data
model.fit(X_train, y_train)

print("Model training is complete! 🧠")
```

```
Model training is complete! 🧠
```



```python
from sklearn.metrics import accuracy_score, classification_report

# Use the trained model to make predictions on the test data
y_pred = model.predict(X_test)

# Calculate the model's accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")

# Print a detailed report showing performance for each scheme
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

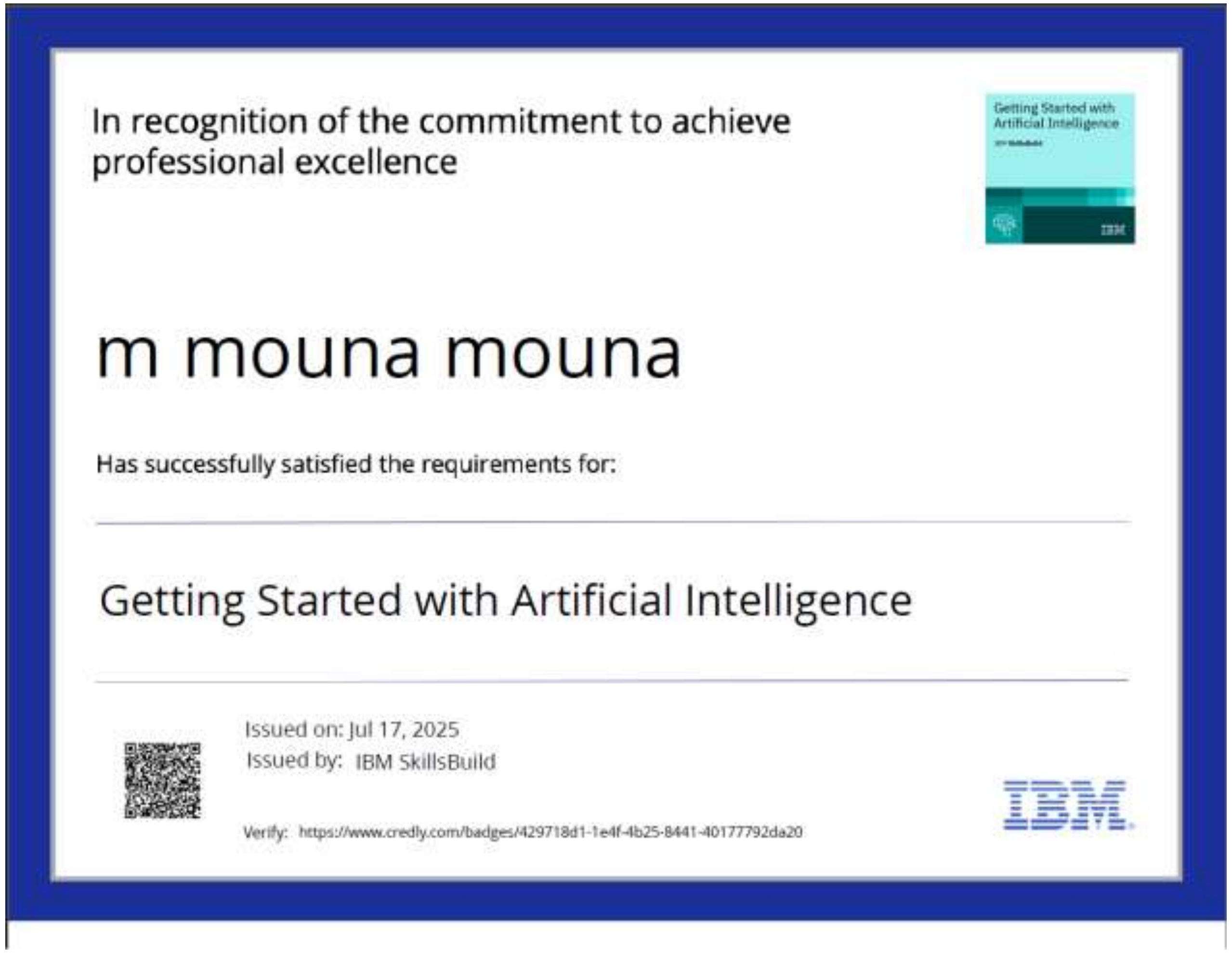
```
Model Accuracy: 100.00%

Classification Report:
              precision    recall  f1-score   support

      IGNDPS       1.00      1.00      1.00         8
     IGNOAPS       1.00      1.00      1.00         7
      IGNWPS       1.00      1.00      1.00         5

    accuracy                           1.00        20
   macro avg       1.00      1.00      1.00        20
weighted avg       1.00      1.00      1.00        20
```

In conclusion, by leveraging machine learning techniques for multi-class classification, we can significantly enhance the efficiency and accuracy of determining eligibility for various schemes under the National Social Assistance Programme (NSAP). Utilizing demographic and socio-economic data, the predictive model serves as an intelligent decision-support tool that can automate the process of categorizing applicants, reducing manual workload, minimizing errors, and ensuring that financial assistance reaches the rightful beneficiaries promptly. Deploying this solution on IBM Cloud Lite services not only ensures scalability and accessibility for government agencies but also demonstrates the practical application of AI in driving social welfare initiatives. This approach fosters transparency, accelerates benefit delivery, and contributes to the broader goal of inclusive development.

FUTURE SCOPE

- **Integration with Government Portals** – Automate real-time eligibility verification by integrating the model with official application portals.
- **Inclusion of Real-time Data Sources** – Enhance prediction accuracy by incorporating live socio-economic and demographic datasets.
- **Multilingual Chatbot Assistance** – Develop AI chatbots in regional languages to assist applicants in understanding their eligibility.
- **Fraud Detection Mechanisms** – Extend the model to identify and flag potentially fraudulent applications.
- **Scalable Deployment Across States** – Scale the solution for nationwide adoption, adapting to state-specific schemes and policies.

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

m mouna mouna

Has successfully satisfied the requirements for:

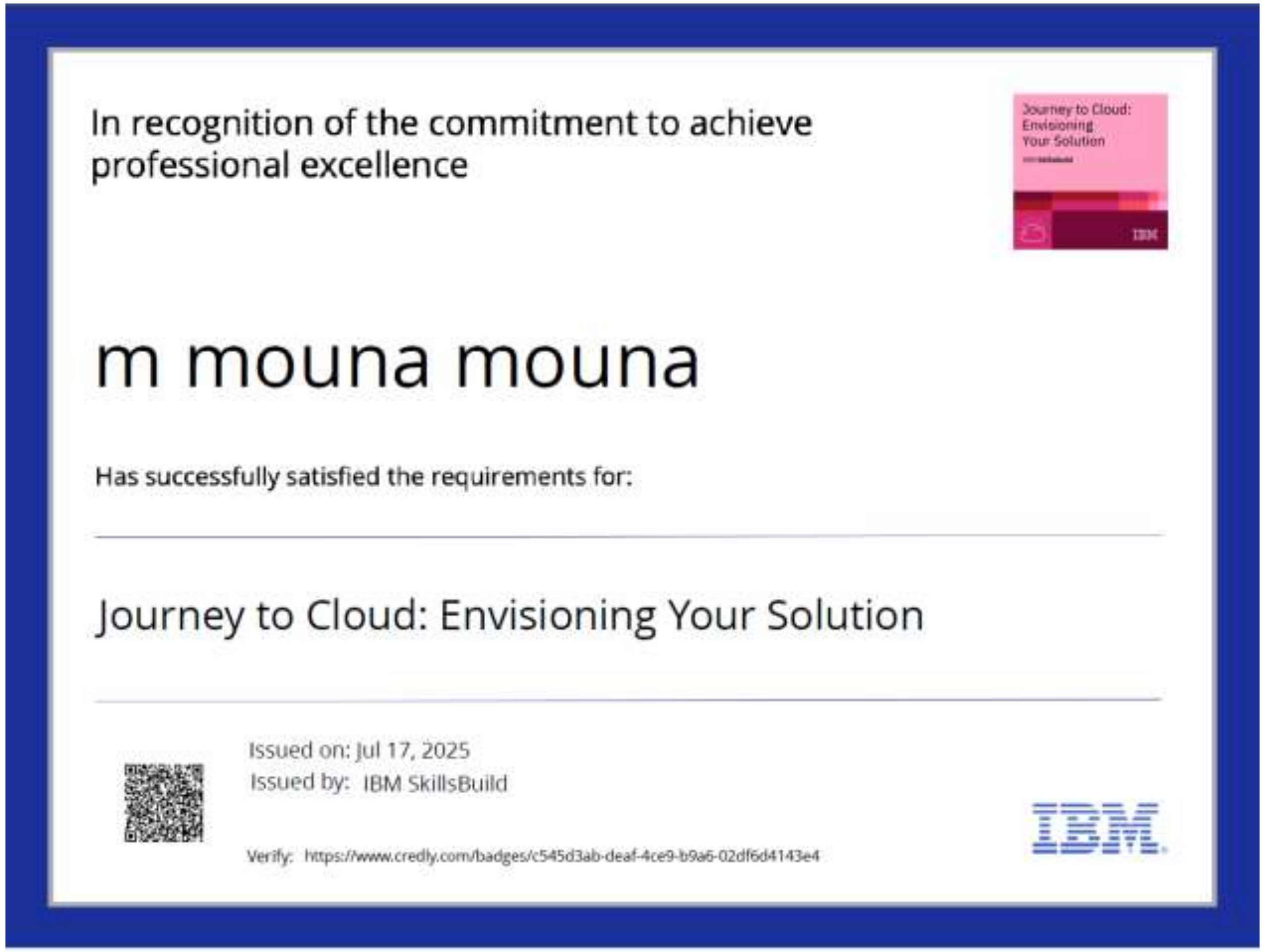
Getting Started with Artificial Intelligence

Issued on: Jul 17, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/429718d1-1e4f-4b25-8441-40177792da20

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

m mouna mouna

Has successfully satisfied the requirements for:

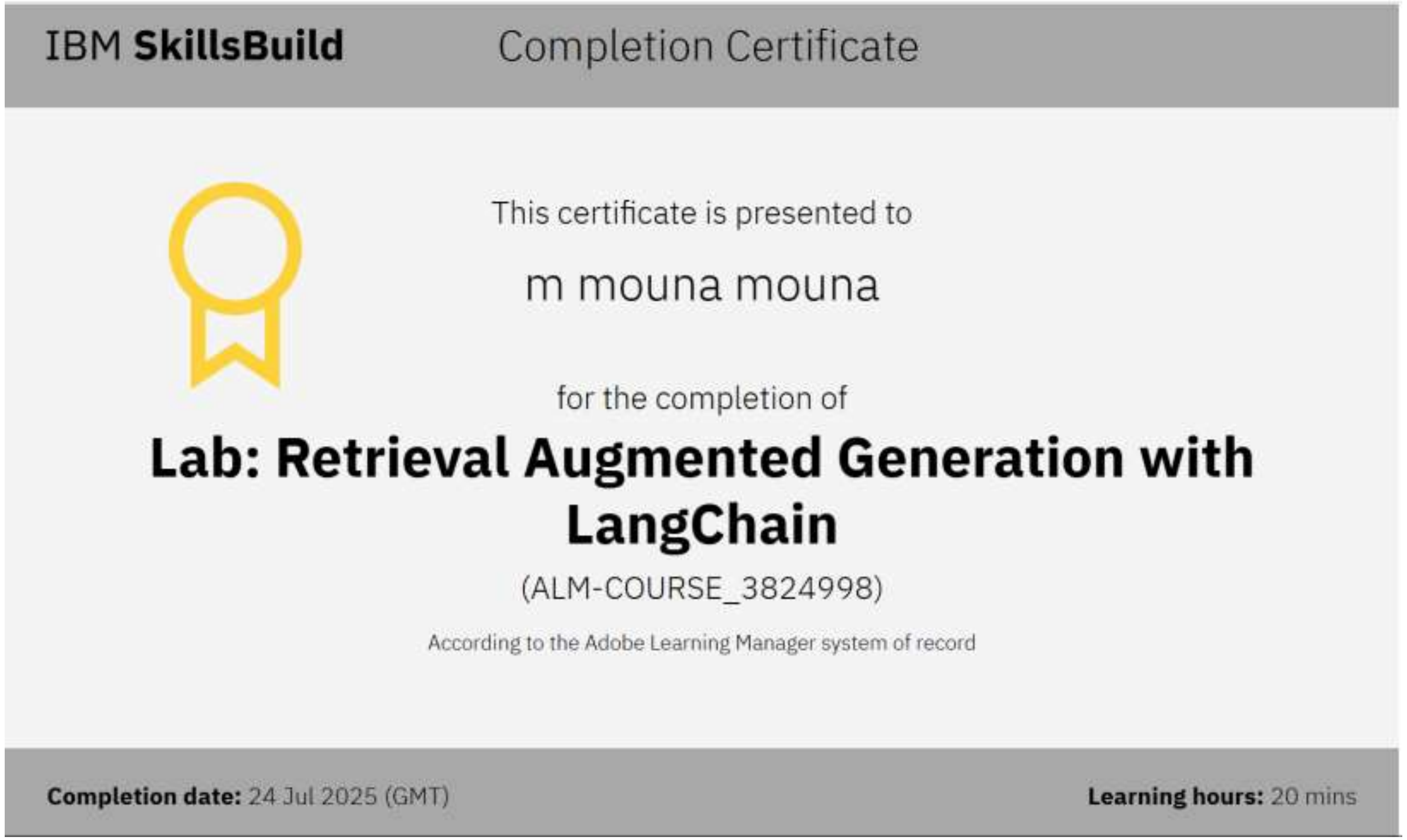
Journey to Cloud: Envisioning Your Solution

Issued on: Jul 17, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/c545d3ab-deaf-4ce9-b9a6-02df6d4143e4

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

m mouna mouna

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins

edunet foundation

GITHUB LINK

- https://github.com/Mouna-1122/NSAP_Eligibility_Project/tree/master

THANK YOU