IBM HACKATHON PROJECT

PREDICTING ELIGIBILITY FOR THE CHALLENGE

Presented By:
Student name : Poorvi Mathur
College Name & Department : JECRC Foundation (AI&DS)

edunet foundation

OUTLINE

- Problem Statement
- Technology used
- Wow factor
- End users
- Result
- Conclusion
- Git-hub Link
- Future scope
- IBM Certifications

PROBLEM STATEMENT

Problem statement 34 - Predicting Eligibility for The Challenge: using Machine Learning 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.

TECHNOLOGY USED

IBM cloud lite services

Watsonx.ai Studio

Jupyter Notebook

Python libraries

IBM Granite model

IBM CLOUD SERVICES USED

- IBM Cloud Watsonx AI Studio
- IBM Cloud Watsonx AI runtime
- IBM Cloud Agent Lab
- IBM Jupyter Notebook
- IBM Granite foundation model

WOW FACTORS

This model serves as a vital tool that helps government agencies move from reactive policy to proactive, data-driven strategy, ensuring resources reach the right people more efficiently.

Unique features:

- Predictive Resource Planning: Anticipates future scheme needs by location and time.
- Actionable Policy Insights: Explains the reasons behind predictions to inform policy decisions.
- Automated Data Auditing: Flags data anomalies for quality control and investigation.
- Targeted Recommendations: Suggests which schemes to prioritize to increase program impact.

END USERS

The end users of this model would be government officials and policymakers working within the National Social Assistance Program (NSAP) and related social welfare departments.

- District and State-Level Administrators
- Data Analysts and Auditors
- Policymakers at the National Level

RESULTS

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | Poorvi Mathur's Account | Dallas | PM

Projects / NSAP Eligibility Prediction / NSAP_Data_Exploration_and_Preprocessing

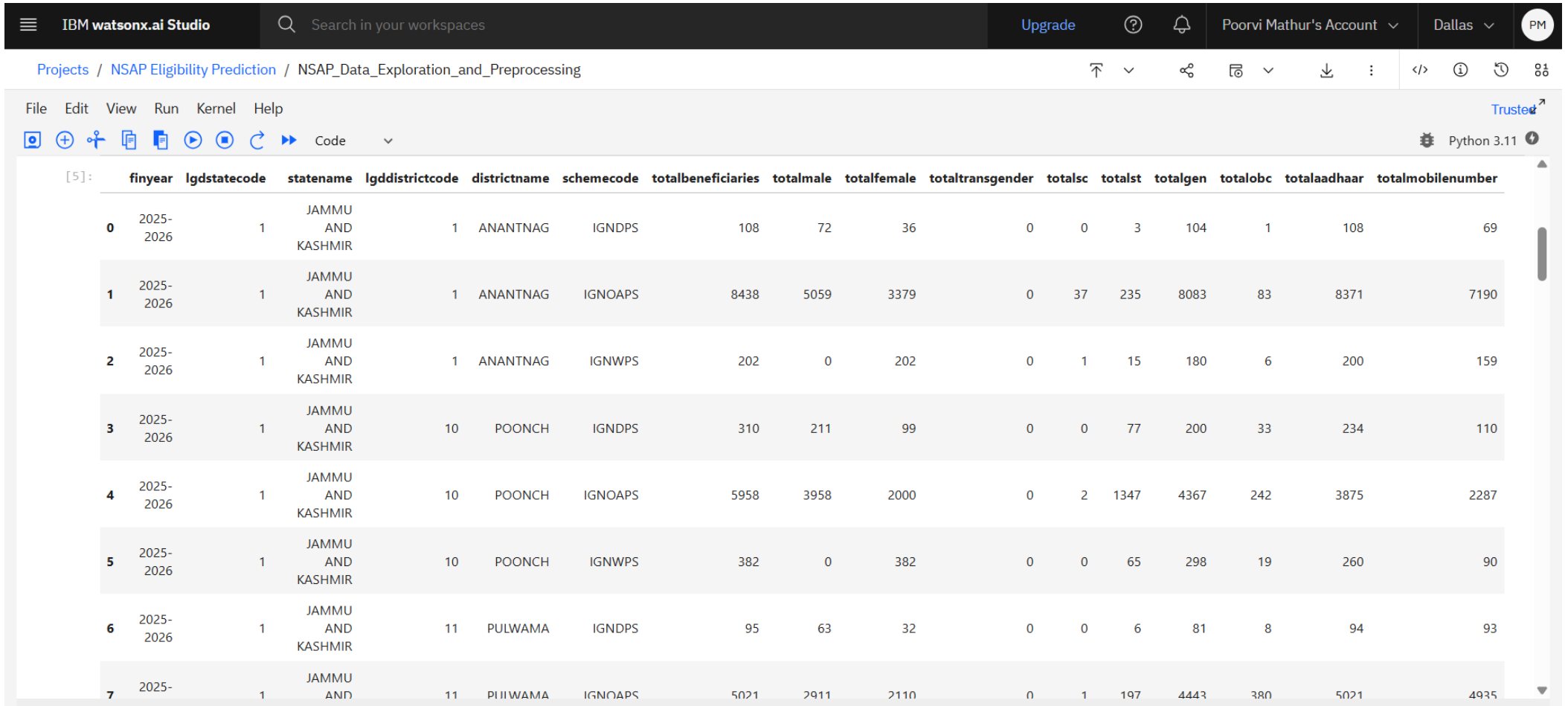
File Edit View Run Kernel Help | Trusted

Code | Python 3.11

[5]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.

cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='69_0BF7C7i0PwAva39vflbHemCMFrp8nGcadZcblok4y',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/identity/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.direct.us-south.cloud-object-storage.appdomain.cloud')

bucket = 'nsapeligibilityprediction-donotdelete-pr-f65sqc2h8s3goq'
object_key = 'nsapallschemes-1.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head(10)
```

RESULTS



	finyear	lgdstatecode	statename	lgddistrictcode	districtname	schemecode	totalbeneficiaries	totalmale	totalfemale	totaltransgender	totalsc	totalst	totalgen	totalobc	totalaadhaar	totalmobilenumber
0	2025-2026	1	JAMMU AND KASHMIR	1	ANANTNAG	IGNDPS	108	72	36	0	0	3	104	1	108	69
1	2025-2026	1	JAMMU AND KASHMIR	1	ANANTNAG	IGNOAPS	8438	5059	3379	0	37	235	8083	83	8371	7190
2	2025-2026	1	JAMMU AND KASHMIR	1	ANANTNAG	IGNWPS	202	0	202	0	1	15	180	6	200	159
3	2025-2026	1	JAMMU AND KASHMIR	10	POONCH	IGNDPS	310	211	99	0	0	77	200	33	234	110
4	2025-2026	1	JAMMU AND KASHMIR	10	POONCH	IGNOAPS	5958	3958	2000	0	2	1347	4367	242	3875	2287
5	2025-2026	1	JAMMU AND KASHMIR	10	POONCH	IGNWPS	382	0	382	0	0	65	298	19	260	90
6	2025-2026	1	JAMMU AND KASHMIR	11	PULWAMA	IGNDPS	95	63	32	0	0	6	81	8	94	93
7	2025-	1	JAMMU AND	11	PULWAMA	IGNOAPS	5021	2911	2110	0	1	197	4443	380	5021	4935

RESULTS

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | Poorvi Mathur's Account | Dallas | PM

Projects / NSAP Eligibility Prediction / NSAP_Data_Exploration_and_Preprocessing

File Edit View Run Kernel Help | Trusted | Code | Python 3.11

[7]:
```
df.describe()
```

[7]:

	lgdstatecode	lgddistrictcode	totalbeneficiaries	totalmale	totalfemale	totaltransgender	totalsc	totalst	totalgen	totalobc	totalaadhaar	totalmobilenumber
count	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000	2156.000000
mean	18.085807	366.853432	13864.290353	5518.308442	8344.925325	0.926716	2370.432282	1387.965213	5766.059369	4334.096011	12084.051948	9853.239332
std	9.770668	211.760095	22855.917172	11926.748715	12518.941254	6.692879	4760.239674	3850.474062	13814.145206	8513.111659	20514.175808	18277.289930
min	1.000000	1.000000	2.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	9.000000	185.000000	961.750000	0.000000	418.750000	0.000000	55.000000	12.000000	223.000000	54.000000	689.500000	466.750000
50%	19.000000	365.000000	4702.500000	503.000000	3415.000000	0.000000	481.000000	139.000000	1221.000000	692.500000	3199.500000	2419.500000
75%	24.000000	546.250000	17377.000000	4819.500000	11514.750000	0.000000	2432.750000	992.250000	5186.000000	4379.500000	14402.000000	10800.250000
max	38.000000	763.000000	278410.000000	130524.000000	147780.000000	165.000000	61247.000000	56759.000000	210999.000000	94046.000000	193015.000000	193091.000000

[8]:
```
df['schemecode'].value_counts()
```

[8]:
```
schemecode
IGNOAPS    731
IGNWPS     723
IGNDPS     702
Name: count, dtype: int64
```

[9]:
```
df['statename'].value_counts()
```

[9]: statename

edunet foundation

RESULTS

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | Poorvi Mathur's Account | Dallas | PM

Projects / NSAP Eligibility Prediction / NSAP_Data_Exploration_and_Preprocessing

```
preprocessor = ColumnTransformer(
    transformers=[
        ('num', 'passthrough', numerical_features), # Keep numerical features as they are
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features) # One-hot encode categorical features
    ])
```

In [58]:
```
handle_unknown='ignore'
```

In [59]:
```
from sklearn.model_selection import train_test_split
```

In [60]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

print(f"\nTraining set size: {X_train.shape[0]} rows")
print(f"Testing set size: {X_test.shape[0]} rows")
```

```
Training set size: 584 rows
Testing set size: 147 rows
```

In [61]:
```
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
```

In [62]:
```
model_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                 ('classifier', RandomForestClassifier(random_state=42, n_estimators=100))])
```

In [63]:
```
print("\nTraining the model...")
model_pipeline.fit(X_train, y_train)
print("Model training complete!")
```

```
Training the model...
Model training complete!
```

In [64]:
```
print("\n--- Model Evaluation ---")
y_pred = model_pipeline.predict(X_test)
```

```
--- Model Evaluation ---
```

In [66]:
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [67]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy on Test Set: {accuracy:.2f}")
```

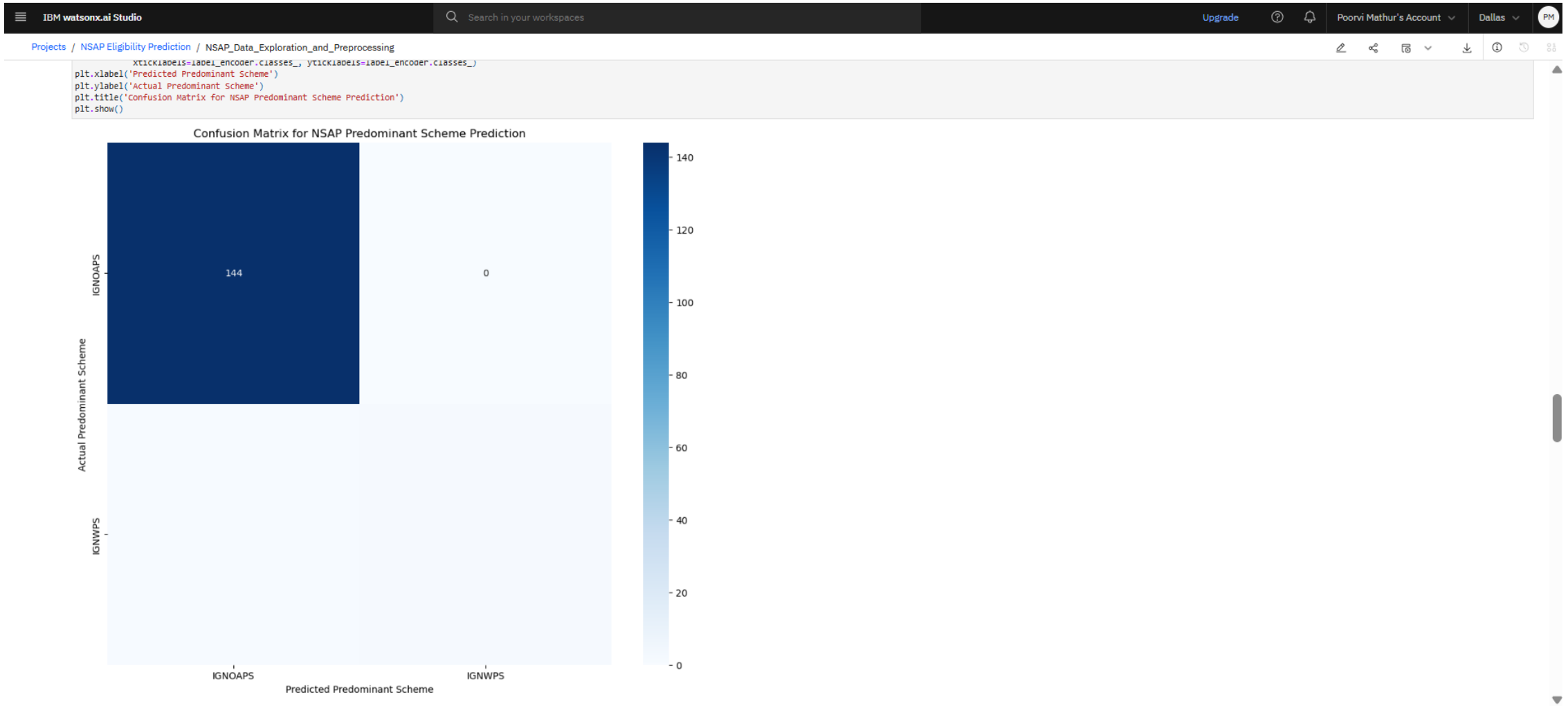
```
Model Accuracy on Test Set: 0.99
```

In [68]:
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))
```

```
Classification Report:
              precision    recall  f1-score   support

     IGNOAPS       0.99      1.00      1.00       144
      IGNWPS       1.00      0.67      0.80         3

    accuracy                           0.99       147
   macro avg       1.00      0.83      0.90       147
weighted avg       0.99      0.99      0.99       147
```

edunet foundation

RESULTS



CONCLUSION

- Problem: Predicted the predominant NSAP scheme for a district using an aggregated dataset.
- Solution: Developed a robust machine learning model using Python and IBM Watson Studio, covering the entire data science workflow.
- Impact: The model provides a practical tool for government agencies to make data-driven decisions, improving resource planning and program efficiency.

GITHUB LINK

https://github.com/poorvi-mathur/NSAP_PREDICTION.git

FUTURE SCOPE

- Individual Eligibility Prediction
- Geospatial Integration
- Real-Time Analytics
- Causal Inference
- Time-Series Forecasting
- Enhanced AI Capabilities

IBM CERTIFICATIONS

RAG LAB certificate

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

POORVI MATHUR

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

THANK YOU

edunet foundation