CAPSTONE PROJECT

INTELLIGENT CLASSIFICATION OF RURAL INFRASTRUCTURE PROJECTS

Presented By:

1. Student Name-KAVISHA SHARMA
2. College Name-Pandit Deendayal Energy University
3. Department-Information and Communication Technology

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- Conclusion
- Future Scope
- References
- Github Link:- https://github.com/kavisha2035/IBM_Rural_Infrastructure

PROBLEM STATEMENT

- **Rural infrastructure** plays a pivotal role in connecting remote communities to mainstream economic and social development. In India, the **Pradhan Mantri Gram Sadak Yojana (PMGSY)** aims to provide all-weather road connectivity to unconnected rural habitations. Over the years, the program has evolved into multiple schemes like **PMGSY-I, PMGSY-II, RCPLWEA**, each with distinct objectives, guidelines, and financial structures.
- With **thousands of projects** sanctioned across states, manually categorizing each project into its respective scheme based on financial and physical parameters has become a **tedious and error-prone task**. This affects timely audits, efficient monitoring, and budget allocation.
- Therefore, the need arises for an **intelligent, automated classification system** that can accurately and instantly identify the correct scheme for each road or bridge project. This would empower policymakers and planners with **transparent and scalable infrastructure tracking**, improving both **governance** and **execution efficiency** at the grassroots level.

PROPOSED SOLUTION

- **Data Collection**
- Utilize government project datasets containing details like:
  - State, district, sanctioned and completed road/bridge counts
  - Length of roads, cost estimates, expenditures, and project statuses
- **Data Preprocessing**
- Clean and handle missing or inconsistent values from raw project records.
- Apply feature engineering techniques to create meaningful input variables for classification.
- Normalize or encode categorical fields (e.g., state/district names) as needed for model input.
- **Machine Learning Algorithm**
- Train a classification model such as Random Forest, XGBoost, or Logistic Regression to predict the PMGSY scheme.
- Analyze feature importance to understand the factors influencing classification decisions.
- Use cross-validation and hyperparameter tuning to improve generalization performance.
- **Deployment**
- Develop an interactive Streamlit web interface to allow users to input project data and get real-time classification results.
- Host the model on IBM Cloud using services like Cloud Object Storage, Machine Learning Deployment, and Streamlit deployment via Cloud Foundry.
- **Evaluation**
- Evaluate model performance using classification metrics such as accuracy, precision, recall, and F1-score.
- Monitor prediction consistency and periodically update the model with new project data for improved accuracy.

SYSTEM APPROACH

System Development Approach
The system approach for classifying rural infrastructure projects is structured into well-defined stages to ensure robust model development and deployment. It includes data preparation, model training, and deployment through a user-friendly web interface.
System Requirements
- **Hardware Requirements:**
  - Minimum 8 GB RAM and Intel i5 processor (or equivalent)
  - GPU (optional, for faster training in large datasets)
- **Software Requirements:**
  - Operating System: Windows / Ubuntu / macOS
  - Python 3.8 or higher
  - IBM Cloud Lite Account for deployment

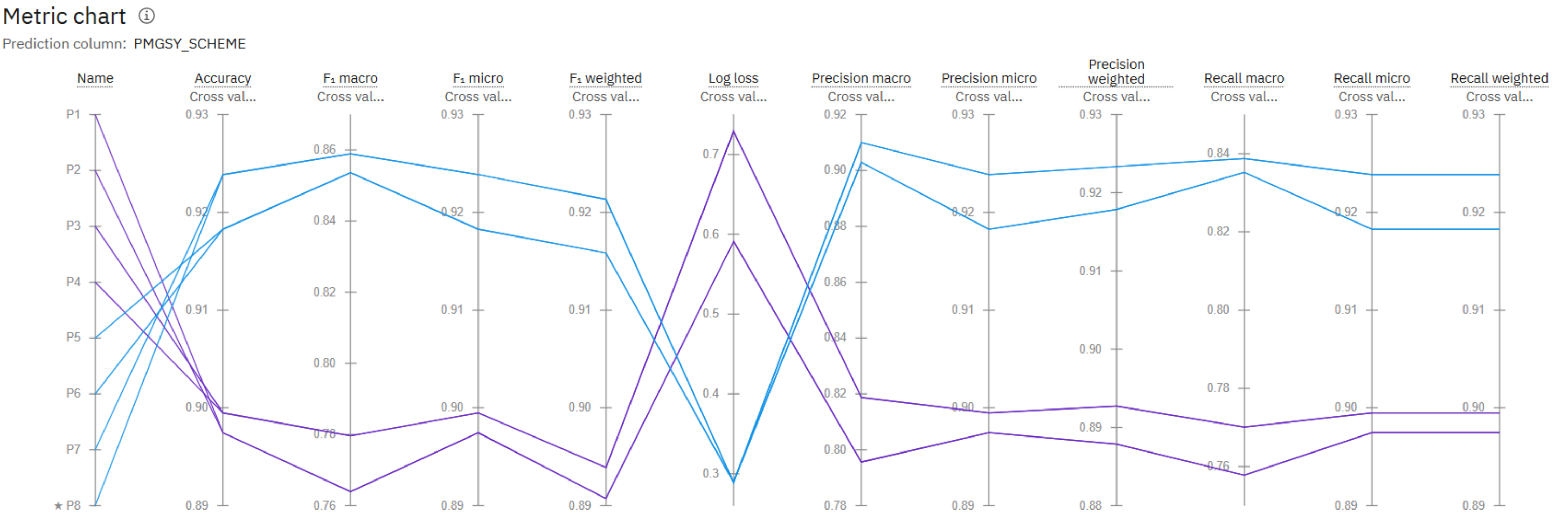
Libraries Required to Build the Model
- **Data Handling & Processing:**
  - pandas – for reading and manipulating tabular project data
  - numpy – for numerical operations
  - scikit-learn – for preprocessing, training, evaluation, and classification models
- **Visualization:**
  - matplotlib and seaborn – for EDA and feature distribution plots
  - plotly – for interactive charts in the dashboard
- **Web Application:**
  - streamlit – for creating the user-friendly frontend interface
  - requests – for integrating with deployed ML APIs
- **Model Deployment:**
  - ibm-watson-machine-learning – for uploading, deploying, and scoring models on IBM Cloud
  - joblib – for model serialization

ALGORITHM & DEPLOYMENT

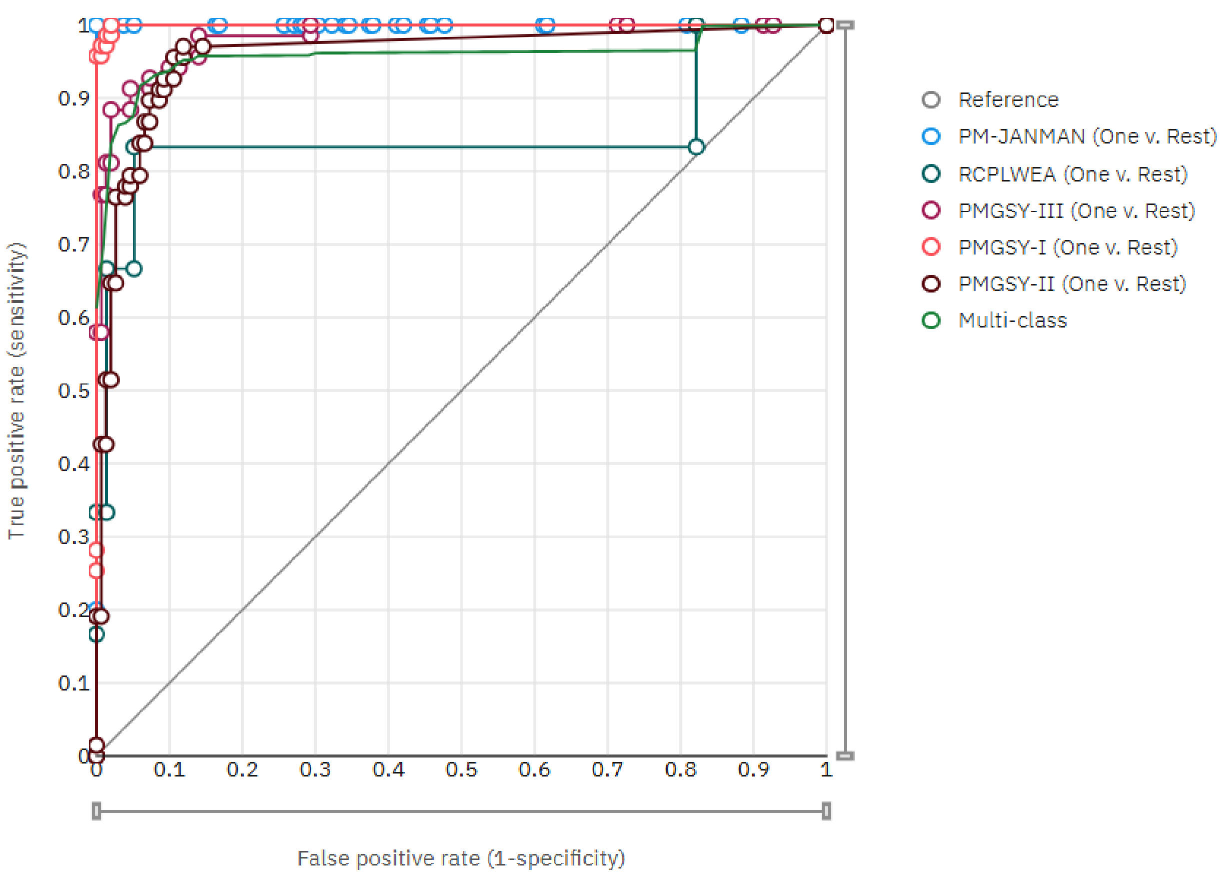
Algorithm Selection:
For the classification of rural infrastructure projects into their respective PMGSY schemes, the XGB Classifier was selected due to its robustness, ability to handle categorical and numerical features, and effectiveness in multi-class classification tasks.
Alternative algorithms like Random Forest and Logistic Regression were also evaluated, but XGB Classifier provided a good trade-off between performance and interpretability.
Data Input:
The model was trained using the following features extracted from the dataset:
•State & District Names (encoded)
•No. of Road/Bridge Works Sanctioned
•Length of Road Work Sanctioned
•Cost of Works Sanctioned
•No. of Works Completed and Remaining
•Expenditure Incurred
•Total Road and Bridge Metrics
These features capture both the physical and financial attributes of each project, which are key indicators for categorizing the scheme type.
Training Process:
•The dataset was preprocessed with:
    •Label encoding for categorical features (state, district)
    •Handling of missing values
    •Feature scaling (if required by model choice)
•The data was split into training and test sets using an 80:20 ratio.
•Hyperparameter tuning (number of trees, depth, etc.) was done using GridSearchCV to optimize accuracy.
Prediction Process:
•Once trained, the model can predict the scheme label (e.g., PMGSY-I, PMGSY-II, RCPLWEA) for any new infrastructure project record based on its input features.
•Predictions are made in real-time through a deployed API on IBM Cloud, where the backend ML model is hosted.
•The Streamlit-based web interface accepts user input via dropdowns or CSV upload and displays the predicted scheme with a confidence score and visualizations.

RESULT

The machine learning model, developed using IBM AutoAI, successfully classified rural infrastructure projects into their respective PMGSY schemes with high accuracy. The system automatically selected and optimized the best model pipeline using historical project data. The final output displayed the predicted scheme (e.g., PMGSY-I, PMGSY-II, RCPLWEA) for each project, helping streamline classification and improve policy monitoring efficiency.



ROC



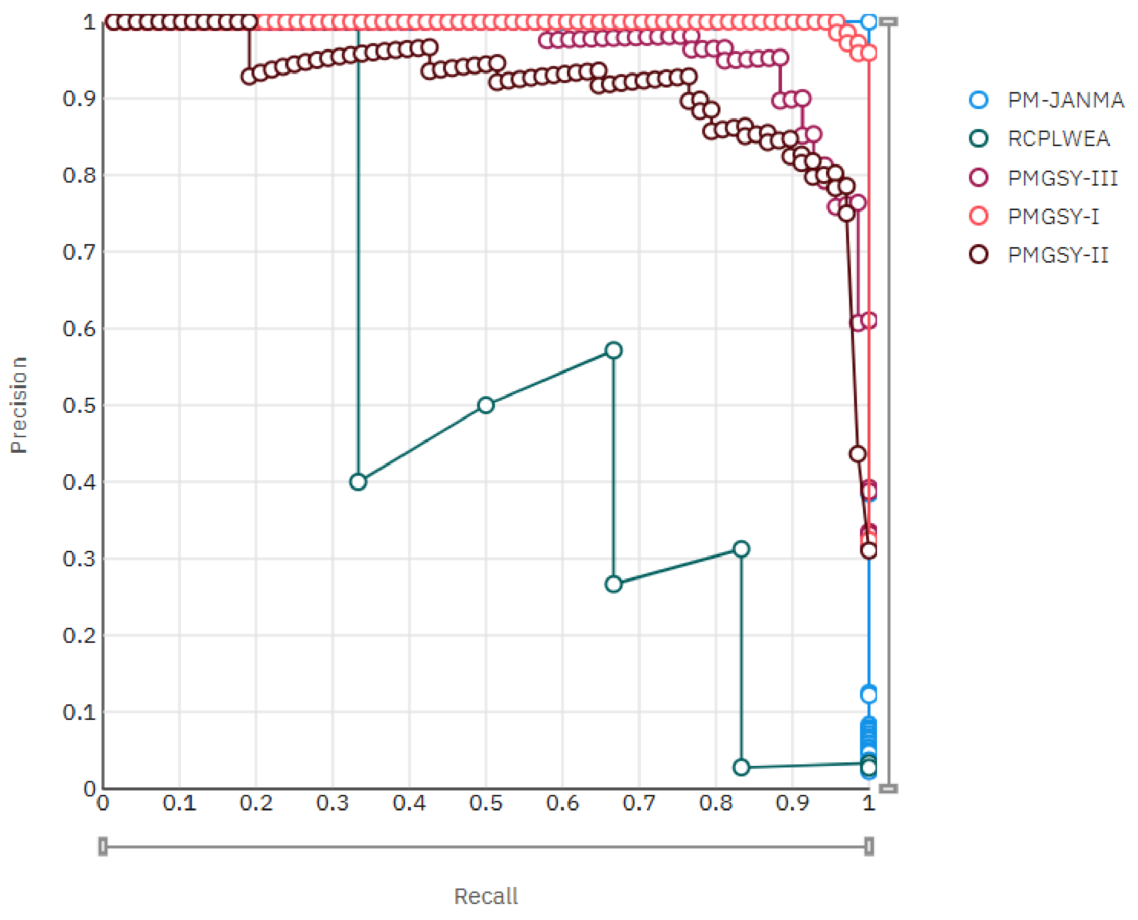
CONFUSION MATRIX

Observed	Predicted					
	PM-JANMAN	PMGSY-I	PMGSY-II	PMGSY-III	RCPLWEA	Percent correct
PM-JANMAN	5	0	0	0	0	100.0%
PMGSY-I	0	69	1	1	0	97.2%
PMGSY-II	0	1	64	3	0	94.1%
PMGSY-III	0	0	9	60	0	87.0%
RCPLWEA	0	0	2	1	3	50.0%
Percent correct	100.0%	98.6%	84.2%	92.3%	100.0%	91.8%

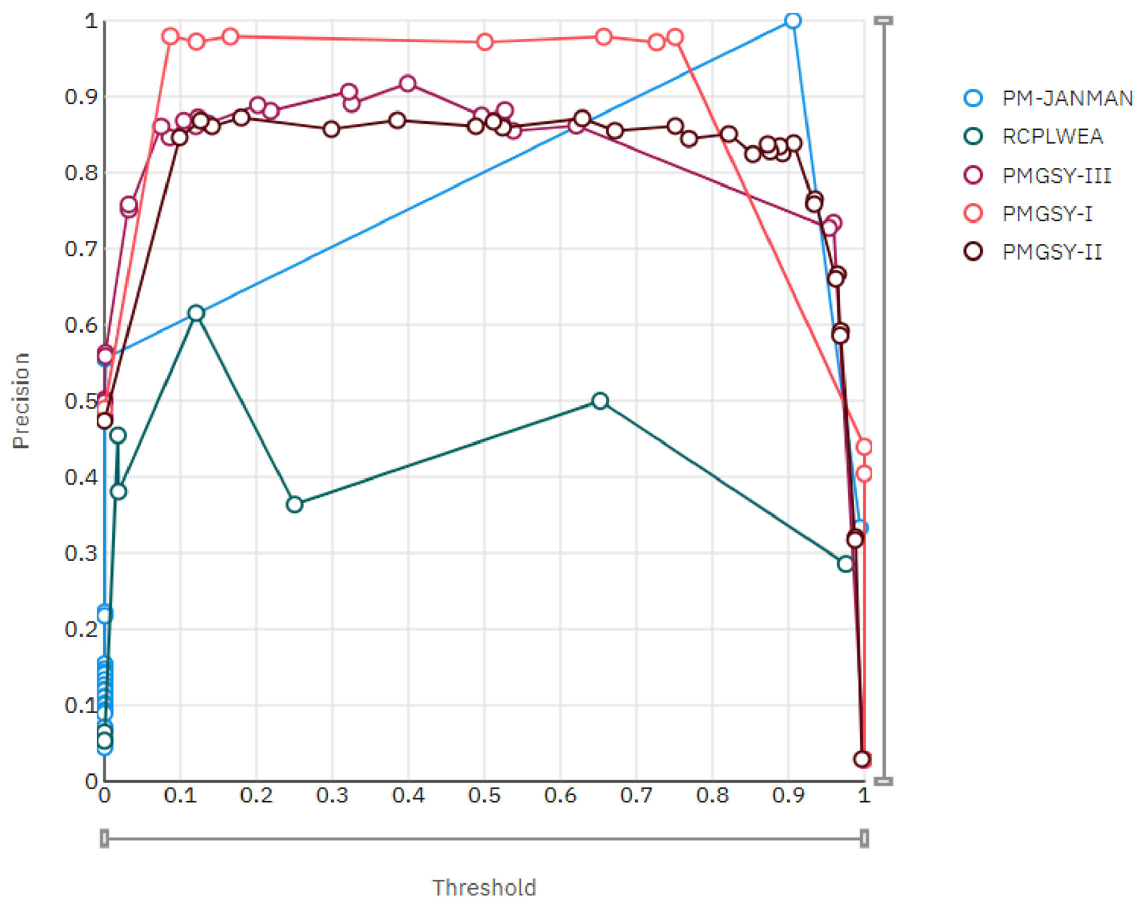
Less correct — More correct

edunet foundation

PRECISION RECALL



THRESHOLD CHART



TESTING THE DEPLOYED MODEL

Scheme_prediciton Deployed Online

API reference | **Test**

Enter input data

Text | JSON

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

Download CSV template | Browse local files | Search in space | Clear all

	STATE_NAME (other)	DISTRICT_NAME (other)	NO_OF_ROAD_WORK_SANCTIONED (double)	LENGTH_OF_ROAD_WORK_SANCTIONED (double
1	Arunachal Pradesh	Tirap	91	577.61
2				
3				
4				

1 row, 14 columns

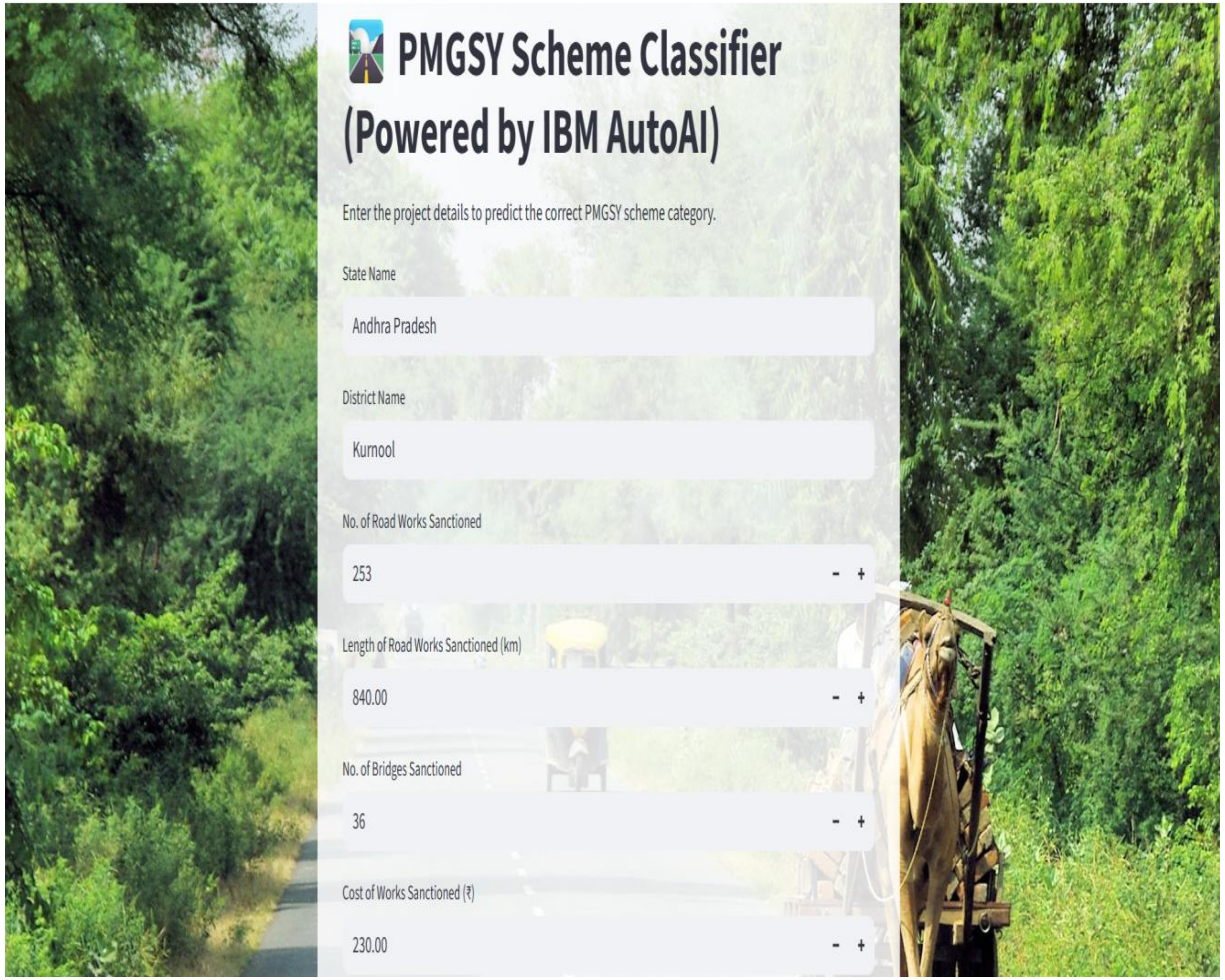
Prediction results

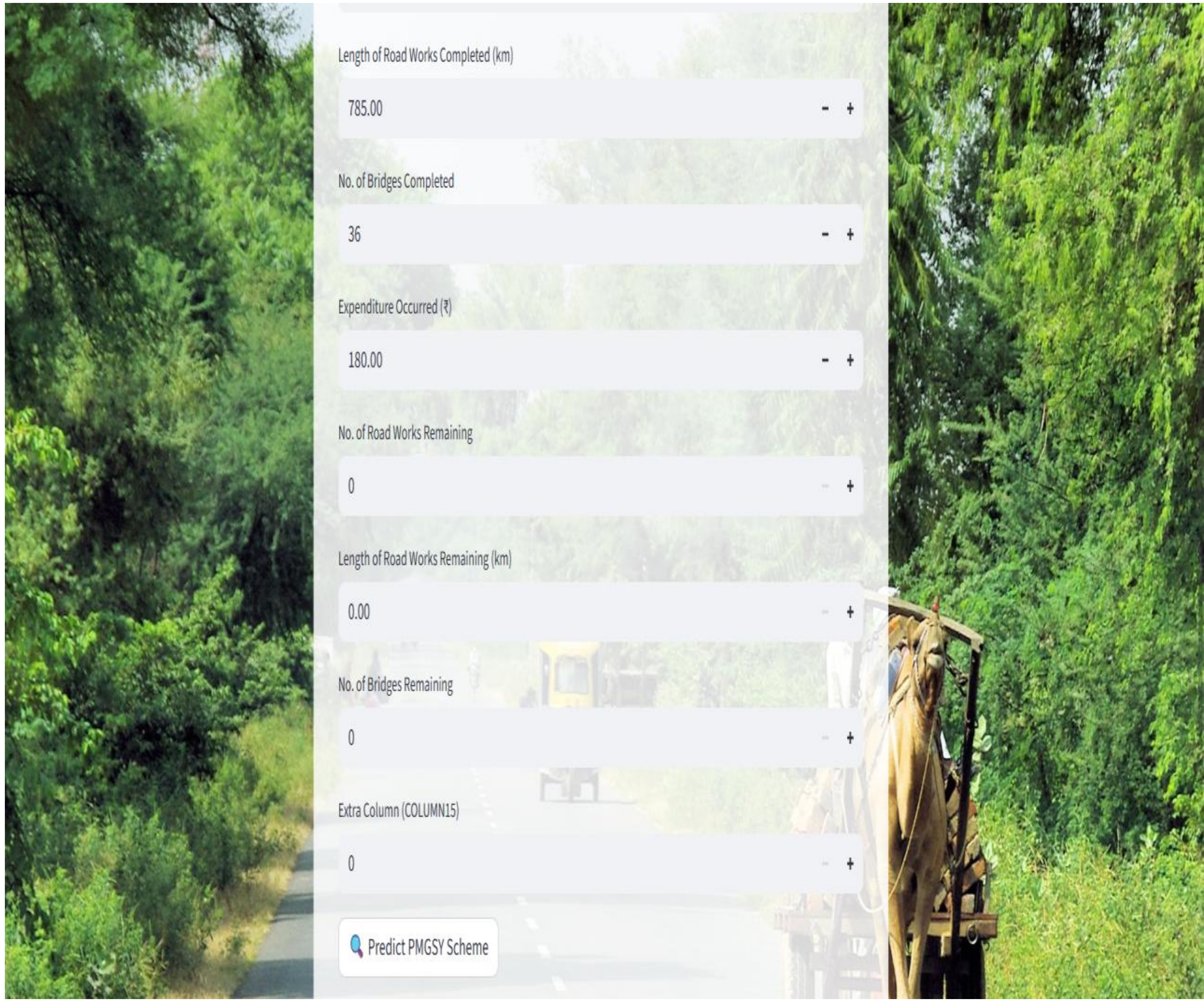
Display format for prediction results

Table view | JSON view | Show input data

	prediction	probability
1	PMGSY-I	[0.0000028691283660009503,0.999664306640625,0.00006983347702771425,0.00...
2		
3		

STREAMLIT APP

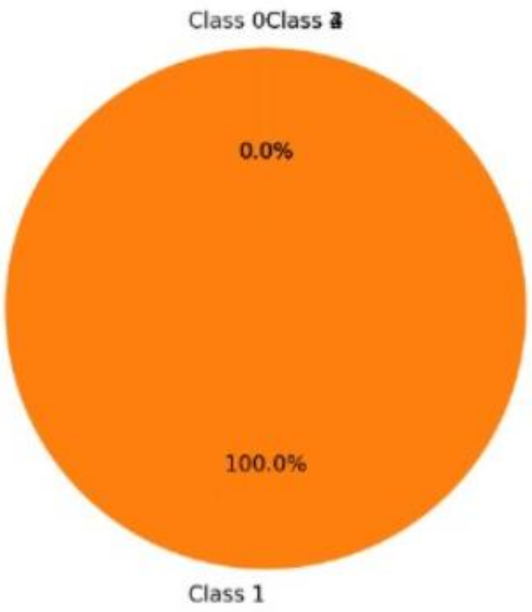




edunet foundation

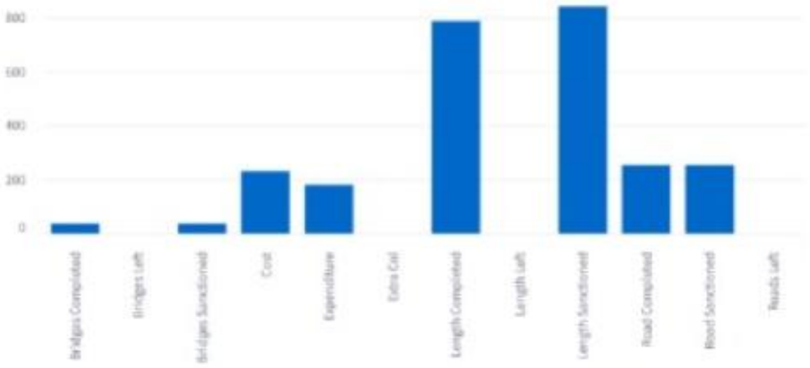
RESULTS

Class Probabilities



🎯 Predicted Scheme Category: **PMGSY-I**

Input Features Overview



CONCLUSION

- The proposed system successfully automates the classification of rural infrastructure projects into appropriate PMGSY schemes using IBM Watson AutoAI. This helps reduce manual effort, ensures consistency, and improves transparency in project monitoring.
- Despite deployment constraints due to limited compute capacity, the AutoAI model demonstrated strong classification performance and valuable insights into key features influencing scheme selection.
- This approach highlights the potential of AI-driven tools in supporting efficient policy implementation and rural infrastructure planning.
- Github Link- https://github.com/kavisha2035/IBM_Rural_Infrastructure

FUTURE SCOPE

- **AI-Powered Policy Feedback Loop**
Use classification outputs to inform and dynamically adapt infrastructure policy decisions based on real-time trends in project types, delays, or regional imbalances.
- **Integration with Satellite and Drone Data**
Merge ML-based classification with satellite imagery or drone surveillance to automatically validate physical progress, detect anomalies, or cross-check reported vs. actual work.
- **Voice/Chatbot Interface for Rural Officers**
Deploy a multilingual chatbot or voice-based interface to help on-ground engineers or officers interact with the system and report project metadata directly using mobile apps.
- **Explainable AI (XAI) for Government Audits**
Implement explainability features so auditors and officials can understand *why* a particular scheme was predicted—building transparency and trust in the classification system.
- **Blockchain for Immutable Project Logs**
Integrate blockchain to maintain tamper-proof logs of project classifications, fund allocations, and status updates to ensure integrity and prevent manipulation.
- **Federated Learning for State-Level Customization**
Train models locally on state-specific project data using federated learning, enabling regional customization without sharing sensitive data.
- **Integration with PMGSY e-Governance Dashboards**
Seamlessly connect the ML system to official government dashboards (e.g., OMMS, GeoPMGSY) for automated project tagging and progress visualization.

REFERENCES

•**World Bank PMGSY Report**
“Performance Monitoring and Evaluation of PMGSY”
https://documents.worldbank.org
•**Ministry of Rural Development, Government of India**
Official PMGSY Scheme Guidelines & Data Portal
https://pmgsy.nic.in
•**AutoAI: IBM Watson Studio**
IBM Documentation - *Automated Machine Learning with AutoAI*
https://www.ibm.com/cloud/watson-studio/autoai
•**Zhang, Y., Wang, H., & Wang, X. (2021).**
“Machine Learning in Infrastructure Project Classification: A Review”
IEEE Access, 9, 45623–45635.
DOI:10.1109/ACCESS.2021.3067512
•**Chen, T., & Guestrin, C. (2016).**
“XGBoost: A Scalable Tree Boosting System”
Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
https://doi.org/10.1145/2939672.2939785
•**Pedregosa, F. et al. (2011).**
“Scikit-learn: Machine Learning in Python”
Journal of Machine Learning Research, 12, 2825–2830.
http://jmlr.org/papers/v12/pedregosa11a.html
•**Brownlee, J. (2020).**
“Data Preparation for Machine Learning”
Machine Learning Mastery
https://machinelearningmastery.com/data-preparation-for-machine-learning/
•**IBM Cloud Documentation**
Watson Machine Learning Deployment and Scoring API
https://cloud.ibm.com/docs/watson-machine-learning

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild

kavisha sharma

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 16, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/d3ff6adc-bbb2-462e-805f-379493bf689b

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

kavisha sharma

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 18, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/dcdd905e-6926-4dd7-bb76-204b5cc8872c

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Kavisha Sharma

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 17 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU