Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

AI-Based Bureaucracy Reducer: An NLP Framework for Automated Grievance Classification and Priority Detection

[1]Shubham Sharma, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]AMITY UNIVERSITY, CHHATTISGARH
[1]hereshubhamsharma.ss@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

The increasing number of public grievances in urban governance systems has created a need for efficient and automated complaint management solutions. Traditional grievance systems rely on manual processing, resulting in delays, incorrect classification, and lack of prioritization. This paper presents an AI-Based Bureaucracy Reducer system that uses Natural Language Processing (NLP) and machine learning techniques to automate complaint classification and priority detection.

The proposed system utilizes TF-IDF vectorization and a Logistic Regression model to classify municipal complaints into relevant departments such as sanitation, water supply, and public works. A rule-based mechanism is used to identify urgent complaints and assign priority levels. The system is implemented using a Streamlit frontend, Flask backend, and SQLite database.

The framework provides real-time complaint processing with low latency and improved administrative efficiency. The proposed system reduces manual effort, enhances transparency, and offers a scalable solution for intelligent grievance management systems.

.

Keywords

Natural Language Processing, TF-IDF, Logistic Regression, Flask API, Streamlit, SQLite, Grievance Management, Complaint Classification, Real-Time Processing.

I. INTRODUCTION

In contemporary governance systems, efficient grievance management plays a vital role in ensuring public satisfaction and administrative transparency. Rapid urbanization, increasing population density, and expanding digital communication platforms have significantly increased the number of complaints submitted to municipal corporations and civic authorities. These complaints include issues related to sanitation, drainage overflow, road damage, water leakage, garbage accumulation, and infrastructure maintenance.

Traditional grievance redressal systems are primarily dependent on manual classification and routing mechanisms. Administrative personnel are required to read individual complaints, determine the relevant department, assign priority levels, and forward the issue for resolution. As complaint volume increases, these systems become inefficient, error-prone, and difficult to scale.

One of the major challenges associated with grievance management is the unstructured nature of complaint data. Citizens express issues in natural language using different writing styles,

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

vocabulary patterns, and sentence structures. For example, road-related complaints may be described using terms such as “potholes,” “damaged road,” “broken street,” or “uneven surface.” Traditional rule-based systems fail to interpret such linguistic variations effectively.

Additionally, existing systems often lack automated priority detection mechanisms. Critical complaints involving public safety or emergency situations are frequently treated similarly to minor issues, leading to delayed responses and reduced administrative efficiency.

Artificial Intelligence (AI), specifically Natural Language Processing (NLP), provides a promising solution to these challenges. NLP enables machines to analyze and interpret human language, allowing automated text classification and information extraction. Machine learning techniques further improve the ability of systems to identify patterns and make intelligent predictions based on textual data.

This paper presents an AI-Based Bureaucracy Reducer system designed to automate municipal grievance classification and priority detection using NLP and machine learning. The system integrates TF-IDF feature extraction, Logistic Regression classification, Flask-based backend APIs, SQLite database management, and a Streamlit frontend interface into a unified full-stack architecture.

The proposed system aims to:

- Reduce manual administrative workload
- Improve complaint routing accuracy
- Enable real-time grievance analysis
- Enhance transparency and scalability in governance systems

II. LITERATURE REVIEW

Early grievance management systems relied heavily on manual administrative workflows and human operators for complaint analysis. While effective for small-scale operations, these systems suffered from severe scalability limitations and inconsistent complaint handling. As digital communication platforms expanded, researchers began exploring computational approaches for automated text classification.

One of the most widely adopted feature extraction techniques in NLP is TF-IDF (Term Frequency–Inverse Document Frequency). TF-IDF converts textual documents into numerical vector representations by assigning weights to words based on their relative importance within a corpus. This technique has proven highly effective for text classification tasks involving sparse datasets.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

Several machine learning algorithms have been applied to text classification problems, including Naïve Bayes, Decision Trees, Support Vector Machines (SVM), and Logistic Regression. Among these, Logistic Regression demonstrated strong performance in handling high-dimensional sparse textual data while maintaining computational efficiency and interpretability.

Recent advancements in NLP include transformer-based architectures such as BERT, RoBERTa, and GPT models. These deep learning systems provide improved contextual understanding but require extensive computational resources and large-scale datasets, making them less suitable for lightweight municipal governance systems.

Existing e-governance platforms such as online municipal portals and grievance submission websites provide digital complaint registration but still rely heavily on manual classification workflows. Most systems lack intelligent prioritization mechanisms and real-time NLP-based analysis capabilities.

The proposed system combines lightweight NLP preprocessing techniques with machine learning classification and rule-based priority detection to create an efficient, scalable, and low-latency grievance management framework suitable for real-time deployment.

III. PROBLEM STATEMENT

The rapid increase in public complaints related to municipal services has created significant challenges in administrative complaint management systems. Traditional grievance redressal mechanisms depend heavily on manual processing, leading to delays, incorrect routing, and inefficient handling of complaints.

One of the primary issues is the inability of existing systems to process unstructured natural language efficiently. Citizens describe similar issues using different vocabulary and writing styles, making it difficult for static rule-based systems to interpret complaints accurately.

Additionally, the absence of automated prioritization mechanisms causes critical complaints to be processed alongside routine issues without urgency differentiation. Complaints involving public safety hazards, infrastructure collapse, or severe sanitation problems may therefore experience delayed responses.

Another major challenge is the lack of scalable digital infrastructure capable of handling large complaint volumes in real time. Manual workflows become increasingly inefficient as complaint frequency rises.

The central problem addressed in this research is therefore the design and implementation of an intelligent grievance classification framework capable of:

- Processing unstructured textual complaints

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

- Automatically classifying complaints into departments
- Detecting urgency and assigning priority levels
- Providing real-time responses with minimal latency
- Maintaining structured complaint storage for future analysis

IV. OBJECTIVES

1. Develop an AI-driven grievance classification system using Natural Language Processing techniques.
2. Automate complaint routing to relevant municipal departments.
3. Implement a real-time priority detection mechanism for urgent complaints.
4. Design a responsive and user-friendly web interface for complaint submission and analysis.
5. Ensure low-latency complaint processing using lightweight machine learning models.
6. Integrate database storage for complaint tracking and retrieval.
7. Create a scalable architecture suitable for future government integration.

V. SYSTEM ARCHITECTURE

The AI-Based Bureaucracy Reducer system follows a layered client-server architecture designed for modularity, scalability, and efficient data processing.

Frontend Presentation Layer

The frontend interface is developed using Streamlit and provides users with an intuitive complaint submission portal. The UI follows a clean government-portal-inspired minimalist design. Users can submit textual complaints, view analysis results, access complaint history, and observe analytical visualizations.

Backend Application Layer

The backend is implemented using the Flask framework and operates as a REST API server. The Flask server receives user requests, processes complaint data, communicates with the machine learning module, and interacts with the database layer.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

Machine Learning Layer

The NLP processing layer contains the TF-IDF vectorizer and Logistic Regression classification model. Serialized .pkl files are loaded into memory during server initialization to reduce runtime latency.

Database Layer

SQLite is utilized as the primary database system for storing complaint records. Each complaint entry contains complaint text, predicted department, priority level, confidence score, and timestamps.

The layered separation ensures maintainability, efficient communication between modules, and scalability for future enhancements

VI. SYSTEM IMPLEMENTATION

A. Process Involved

The implementation of the AI-Based Bureaucracy Reducer system was executed in multiple phases, beginning with dataset preparation and ending with full-stack deployment.

Phase 1: Dataset Preparation (dataset.csv)

The project required a structured dataset containing municipal complaints categorized into multiple departments such as sanitation, water supply, drainage, and infrastructure. Complaint records were cleaned, labeled, and formatted for machine learning training.

Phase 2: Model Training (train_model.py)

The training pipeline preprocesses complaint text by converting input to lowercase, removing punctuation, and cleaning unnecessary symbols. TF-IDF vectorization converts textual data into sparse numerical matrices.

The Logistic Regression model is trained on the transformed dataset and exported alongside the TF-IDF vectorizer as serialized .pkl files using the pickle library.

Phase 3: Backend API Integration (server.py)

The Flask backend initializes the machine learning model and exposes RESTful prediction endpoints. Incoming complaint text is vectorized and passed to the classification model for prediction.

The backend also performs:

- Confidence score calculation

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

- Priority detection
- Database insertion
- JSON response formatting

Phase 4: Frontend Development (app.py)

The frontend application is developed using Streamlit and integrates:

- Complaint submission interface
- Analytics visualization
- Complaint history display
- Real-time result rendering

The interface communicates with the Flask API through HTTP requests.

A. Input / Output Screen Design

Input Screen Design

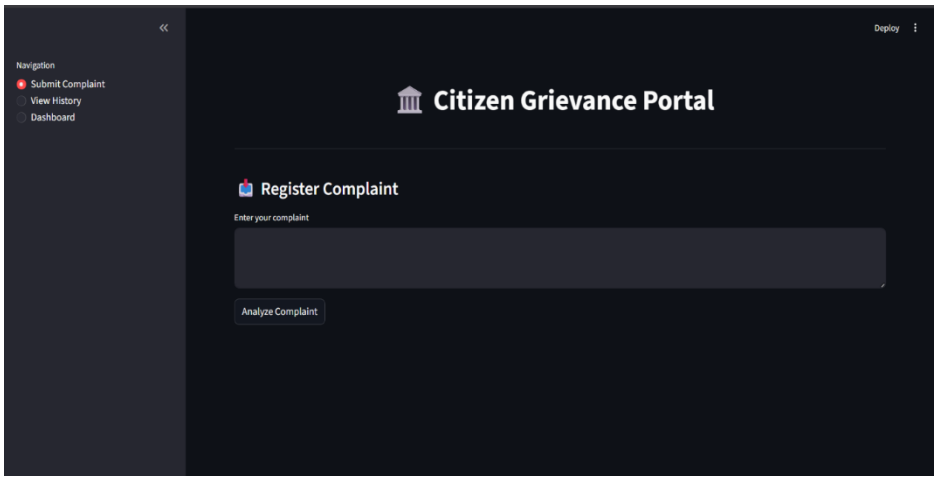
The input screen follows a clean and responsive government-portal-inspired UI design. Users can enter complaints into a large text area and submit them using the “Analyze” button.

Additional interface components include:

- Complaint history viewer
- Analytics section
- Interactive charts
- Responsive layout

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

Fig. 1. Input Screen Design

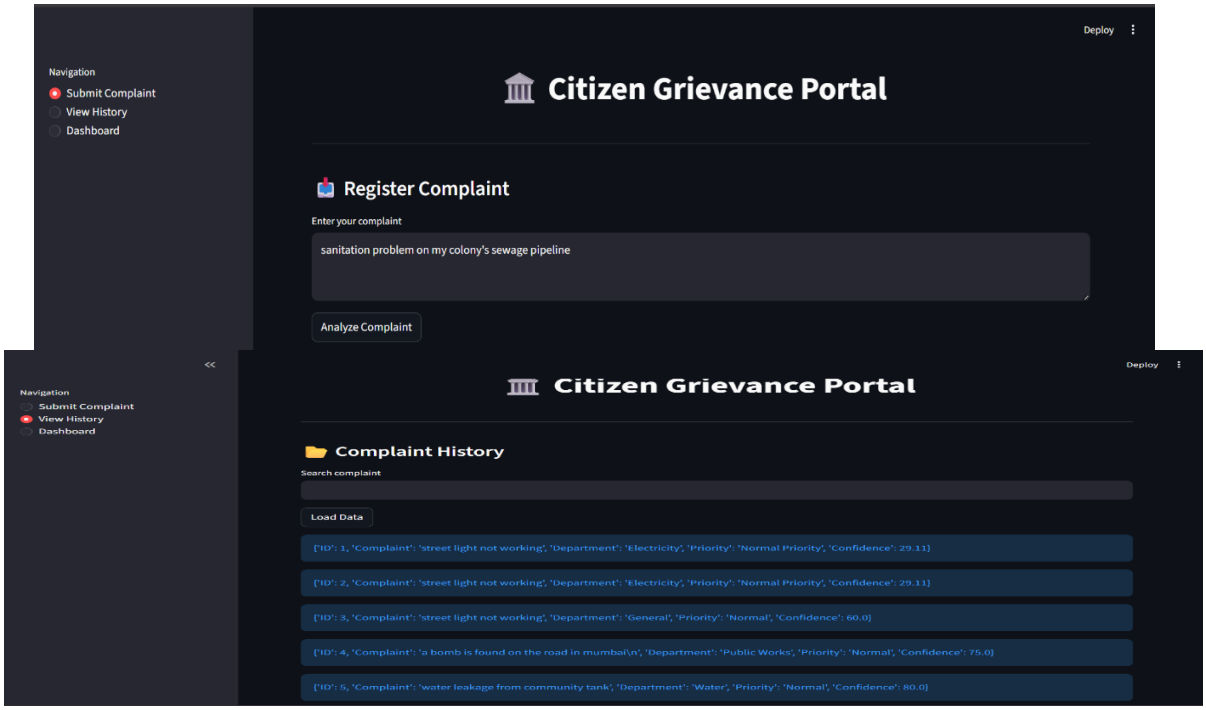


Output Screen Design:

Upon complaint submission, the system displays:

- Predicted department
- Complaint priority
- Confidence score
- Keywords detected
- Suggested action
- Analytical visualization

The output is rendered dynamically with structured formatting to improve readability and interpretability



Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

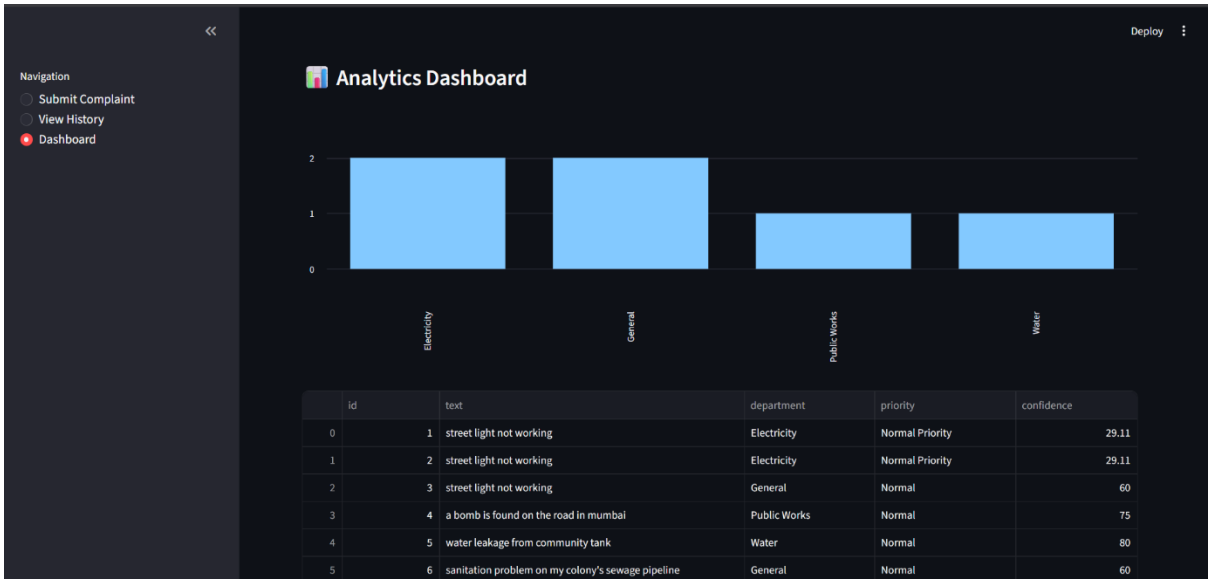


Fig.2.Output screen design

VII. METHODOLOGY

The AI-Based Bureaucracy Reducer system follows a hybrid methodology combining Natural Language Processing (NLP), machine learning, and full-stack web integration to automate grievance classification and priority detection.

The system processes unstructured municipal complaints and converts them into structured outputs through preprocessing, classification, and database storage.

The overall workflow of the system is:

User Complaint → Text Preprocessing → TF-IDF Vectorization → Logistic Regression Classification → Priority Detection → Database Storage → Result Display

- **A. Data Collection and Preprocessing**

The dataset consists of municipal complaints categorized into departments such as sanitation, drainage, water supply, and public works. The data was stored in CSV format and manually labeled for training.

Before model training, preprocessing operations were performed, including:

- Lowercase conversion
- Removal of punctuation
- Removal of unnecessary symbols

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

These operations improved data consistency and classification accuracy.

- **B. Feature Extraction using TF-IDF**

TF-IDF (Term Frequency–Inverse Document Frequency) was used to convert complaint text into numerical vectors.

This technique identifies important words within complaints while reducing the impact of frequently occurring but less meaningful words. The generated vectors were used as input for the machine learning model.

- **C. Complaint Classification using Logistic Regression**

The classification module uses the Logistic Regression algorithm to classify complaints into appropriate departments.

The model was trained using TF-IDF-transformed complaint data and corresponding department labels. During prediction, the model generates:

- Predicted department
- Confidence score

Logistic Regression was selected because of its efficiency, simplicity, and good performance on textual data.

- **D. Priority Detection Mechanism**

A rule-based priority detection mechanism was implemented to identify urgent complaints.

Keywords such as:

- “urgent”
- “danger”
- “emergency”

trigger High Priority classification, while other complaints are assigned Normal Priority.

- **E. Backend and Database Integration**

The backend was developed using Flask, which handles:

- API requests

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

- Model prediction
- Database operations

SQLite was used as the database system to store complaint records, including:

- Complaint text
- Department
- Priority
- Confidence score
- Timestamp

- **F. Frontend Interface**

The frontend was developed using Streamlit and provides:

- Complaint submission interface
- Real-time output display
- Complaint history
- Analytics visualization

The frontend communicates with the Flask backend through HTTP requests to generate real-time results.

- **G. System Integration**

All components including:

- Streamlit frontend
- Flask backend
- Machine learning model
- SQLite database

were integrated into a unified architecture to provide a scalable and efficient grievance management system.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

VIII. TESTING AND VALIDATION

Testing and validation were performed to evaluate the performance, reliability, and functionality of the AI-Based Bureaucracy Reducer system. The system was tested using various municipal complaints related to sanitation, drainage, roads, and water supply.

- **A. Functional Testing**

Functional testing was conducted to verify:

- Complaint submission
- API communication
- Complaint classification
- Priority detection
- Database storage
- Result display

The system successfully processed complaints and generated structured outputs in real time.

- **B. Machine Learning Validation**

The Logistic Regression model was tested using sample complaint data to evaluate classification performance.

Sample Outputs

Complaint	Department	Priority
“Garbage not collected”	Sanitation	Normal
“Water leakage near road”	Water Department	High
“Road full of potholes”	Public Works	High

The model demonstrated satisfactory classification accuracy for common municipal complaints.

- **C. API and Database Testing**

The Flask API was tested for:

- JSON request handling
- Prediction generation
- Error handling

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

SQLite database testing verified:

- Complaint storage
- Record retrieval
- Data consistency

- **D. User Interface Testing**

The Streamlit frontend was tested for:

- Responsiveness
- Ease of use
- Real-time output rendering

The interface functioned smoothly and provided a user-friendly experience.

- **E. Performance Evaluation**

The system demonstrated:

- Real-time complaint processing
- Low response latency
- Stable frontend-backend communication

The testing results confirmed that the proposed system is reliable and suitable for automated grievance management applications.

IX. TECHNOLOGY USED

The AI-Based Bureaucracy Reducer system was developed using various open-source technologies for frontend development, backend integration, machine learning, and database management.

A. Hardware Requirements

- Intel i3/i5 Processor or higher
- Minimum 4 GB RAM

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

- 500 MB free disk space

B. Software Requirements

Technology	Purpose
Python 3.x	Core programming language
Streamlit	Frontend user interface
Flask	Backend API development
SQLite	Database management
Scikit-learn	Machine learning implementation
Pandas	Data processing
NumPy	Numerical computation
Pickle	Model serialization
VS Code	Development environment

C. Libraries Used

1. Scikit-learn

Used for TF-IDF vectorization and Logistic Regression model training.

2. Streamlit

Used for building the interactive frontend interface.

3. Flask

Used for backend API integration and request handling.

4. SQLite

Used for storing complaint records and complaint history.

The combination of these technologies enabled the development of a lightweight, scalable, and efficient grievance management system.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

X. ADVANTAGES

A. Automated Complaint Classification

The system automatically classifies complaints into relevant departments using NLP and machine learning techniques, reducing manual administrative effort.

B. Real-Time Processing

The lightweight architecture enables fast complaint analysis and real-time result generation with minimal latency.

C. Priority Detection

Urgent complaints are automatically identified and assigned higher priority levels, improving response efficiency.

D. User-Friendly Interface

The Streamlit-based frontend provides a clean and intuitive interface suitable for users with minimal technical knowledge.

E. Database Integration

The integration of SQLite enables structured complaint storage, retrieval, and historical analysis.

F. Scalability

The modular architecture allows future expansion and integration with advanced AI models and government systems.

G. Cost-Effective Implementation

The system utilizes open-source technologies, reducing deployment and maintenance costs.

XI. LIMITATIONS

A. Limited Dataset

The classification model was trained on a relatively small dataset, which may affect performance for highly diverse complaint patterns.

B. Rule-Based Priority Detection

The priority detection mechanism relies on predefined keywords and lacks deep contextual understanding.

C. Limited Multilingual Support

The current implementation primarily focuses on English-language complaints.

D. Absence of Deep Learning Models

The system does not currently utilize transformer-based architectures such as BERT or GPT for contextual semantic analysis.

E. Dependency on Internet for Some Features

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

Certain external integrations and package installations require internet connectivity.

XII. FUTURE SCOPE

The AI-Based Bureaucracy Reducer system provides a strong foundation for future advancements in intelligent grievance management systems.

One of the major future enhancements includes the integration of advanced transformer-based NLP models such as BERT and GPT. These models can significantly improve contextual understanding and semantic analysis of complaints.

The system can also be extended to support multilingual complaint processing, enabling users to submit grievances in regional languages. This enhancement would improve accessibility and inclusiveness.

Another important future improvement is the development of a dedicated mobile application for Android and iOS platforms. Mobile integration would enhance usability and allow users to submit complaints directly from smartphones.

Future versions of the system may also include:

- Real-time complaint tracking
- Push notifications
- Administrative dashboards
- Geographic complaint visualization
- Government API integration

Additionally, cloud deployment and distributed database architectures can improve scalability and support large-scale municipal deployments.

The proposed framework therefore has significant potential for expansion into a comprehensive AI-powered governance platform.

XIII. CONCLUSION

This paper presented an AI-Based Bureaucracy Reducer system designed to automate grievance classification and priority detection using Natural Language Processing and machine learning techniques.

The proposed system integrates TF-IDF vectorization, Logistic Regression classification, Flask-based backend APIs, SQLite database management, and a Streamlit frontend interface into a unified full-stack architecture.

The system successfully processes unstructured municipal complaints and generates structured outputs including department classification, confidence scores, and priority levels.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1157-1172

Experimental testing demonstrated reliable real-time performance and efficient complaint processing.

The lightweight and modular architecture ensures scalability, maintainability, and cost-effective deployment. Furthermore, the use of open-source technologies makes the framework practical for real-world governance systems.

The research highlights the potential of AI-driven automation in improving administrative efficiency, reducing manual workload, and enhancing transparency in grievance management workflows.

The proposed system therefore represents a significant step toward intelligent digital governance and scalable public grievance management systems.

XIV. REFERENCES

[1] Python Software Foundation, “Python Documentation,”
[2] Streamlit Inc., “Streamlit Documentation,”
[3] Pallets Projects, “Flask Documentation,”
[4] SQLite Consortium, “SQLite Documentation,”
[5] Scikit-learn Developers, “Scikit-learn Machine Learning Library,”
[6] Jurafsky, D., and Martin, J. H., *Speech and Language Processing*, Pearson Education.
[7] Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, 2011.
[8] Kaggle, “Natural Language Processing Datasets,”
[9] Bird, S., Klein, E., and Loper, E., *Natural Language Processing with Python*, O’Reilly Media.
[10] Manning, C. D., Raghavan, P., and Schütze, H., *Introduction to Information Retrieval*, Cambridge University Press.