ERIC MWANIKI
AI INNOVATION

Team Name : Meaniki AI Lab

Team Leader Name : Eric Mwaniki

Which domain does your idea address? (Agriculture / Healthcare / Skilling / Education): AI/Technology

What is the problem you are solving? (50 words max)

The problem being solved is automating the classification of grant proposals into "funded" or "not funded" categories using AI. The solution also incorporates anomaly detection to identify potentially fraudulent proposals, improving the efficiency and accuracy of grant evaluation processes for organizations and funding agencies

Describe your solution. How different is it from any of the other existing ideas? How will it be able to solve the problem? USP of the proposed solution? What is the intended impact of your solution (max 350 words).

The **Grant Proposal Classification Pipeline** is an AI-driven solution that automates the classification of grant proposals into "funded" or "not funded" categories. Using machine learning models like Logistic Regression, Random Forest, XGBoost, and SVM, along with text analysis (TfidfVectorizer) and numerical feature processing, the solution provides accurate predictions. An **anomaly detection** model (IsolationForest) identifies potentially fraudulent proposals, enhancing security in the evaluation process.

What sets this solution apart is its combination of multiple machine learning techniques, fraud detection, and text processing, making it a robust, scalable, and transparent solution for grant evaluation. The **intended impact** is to streamline the grant review process, saving time and reducing human bias, while ensuring more accurate and fair funding decisions.

Who is the primary user of your solution, and explain how your solution will leverage open-source AI to address the aspects mentioned in the Key Design Guidelines (max 200 words).

The solution targets **grant funding organizations** (e.g., government and research institutions) to automate the classification of grant proposals and detect fraud, improving efficiency and accuracy. It uses **open-source AI** tools like **scikit-learn** for machine learning, **nltk** for text processing, and **joblib** for model saving, ensuring scalability and cost-effectiveness.

Key design features:

- **Scalability**: Handles large datasets with models like XGBoost and SVM.
- **Fairness**: Reduces human bias through automation.
- **Efficiency**: Speeds up the review process by automating proposal classification.
- **Security**: Detects fraudulent proposals using anomaly detection.

This solution offers a comprehensive, efficient, and secure grant evaluation process using
open-source AI tools

How is this solution scalable? (100 words max)

This solution is scalable due to its use of **machine learning models** like **XGBoost** and **SVM**, which can efficiently handle large datasets and diverse proposals. The use of **TfidfVectorizer** for text data and **StandardScaler** for numerical features ensures that the solution can process high volumes of data while maintaining accuracy. Additionally, the modular nature of the pipeline allows easy integration of more data sources, models, or features as the volume of proposals increases, ensuring the system can grow with the needs of the funding organization. This makes it suitable for small to large-scale applications.

List of features offered by the solution

- **Automated Proposal Classification**: Classifies grant proposals as "funded" or "not funded" using machine learning models.
- **Text Processing**: Uses **TfidfVectorizer** for converting text data (titles and abstracts) into numerical features.
- **Numerical Feature Processing**: Scales numerical features like budget, research impact using **StandardScaler**.
- **Fraud Detection**: Identifies potentially fraudulent proposals using **anomaly detection** (IsolationForest).
- **Multiple Model Support**: Utilizes several models, including **Logistic Regression**, **Random Forest**, **XGBoost**, and **SVM**.
- **Hyperparameter Tuning**: Optimizes model performance with **GridSearchCV**.
- **Model Evaluation**: Evaluates model performance using **F1-Score**, **classification report**, and **ROC-AUC**.
- **Model Saving**: Saves trained models using **joblib** and **pickle** for later use.
- **Scalability**: Efficiently handles large datasets and diverse proposals.

What open-source AI tools and technologies will you use to design the solution? (Please list all.)

- **scikit-learn**: For machine learning models (Logistic Regression, Random Forest, XGBoost, SVM), model evaluation, and hyperparameter tuning.
- **nltk**: For text processing, including tokenization, lemmatization, and stopword removal.
- **TfidfVectorizer**: For transforming text data (titles and abstracts) into numerical features using TF-IDF.
- **StandardScaler**: For scaling numerical features like budget and research impact.
- **IsolationForest**: For anomaly detection to flag potentially fraudulent proposals.
- **GridSearchCV**: For hyperparameter optimization and model tuning.
- **joblib**: For saving and loading machine learning models and preprocessing tools.

Why are these open-source technologies the most appropriate for your solution? (150 words max)

These open-source technologies are the most appropriate for the solution because they offer scalability, flexibility, and proven effectiveness.

- **scikit-learn** provides a wide range of efficient machine learning models like Logistic Regression, Random Forest, XGBoost, and SVM, which are ideal for classification tasks and handling large datasets.
- **nltk** and **TfidfVectorizer** are robust for text processing, converting textual data into actionable features, essential for grant proposal classification.
- **StandardScaler** ensures numerical feature scaling for better model performance.
- **IsolationForest** excels at detecting anomalies, helping identify potentially fraudulent proposals.
- **GridSearchCV** allows for easy optimization of hyperparameters to fine-tune model performance.
- **joblib** and **pickle** facilitate seamless model saving and loading, enabling future scalability.

These tools are well-supported, widely used, and cost-effective, making them highly suitable for creating a flexible, scalable, and

Describe the Solutions Architecture (500 words)

The **Grant Proposal Classification Pipeline AI** leverages a modular architecture, utilizing several components that work together to automate the process of classifying grant proposals, detecting anomalies, and ensuring secure evaluations. The architecture follows a data-driven approach, integrating data processing, machine learning, and model evaluation in a seamless pipeline. Below is an outline of the key components of the architecture:

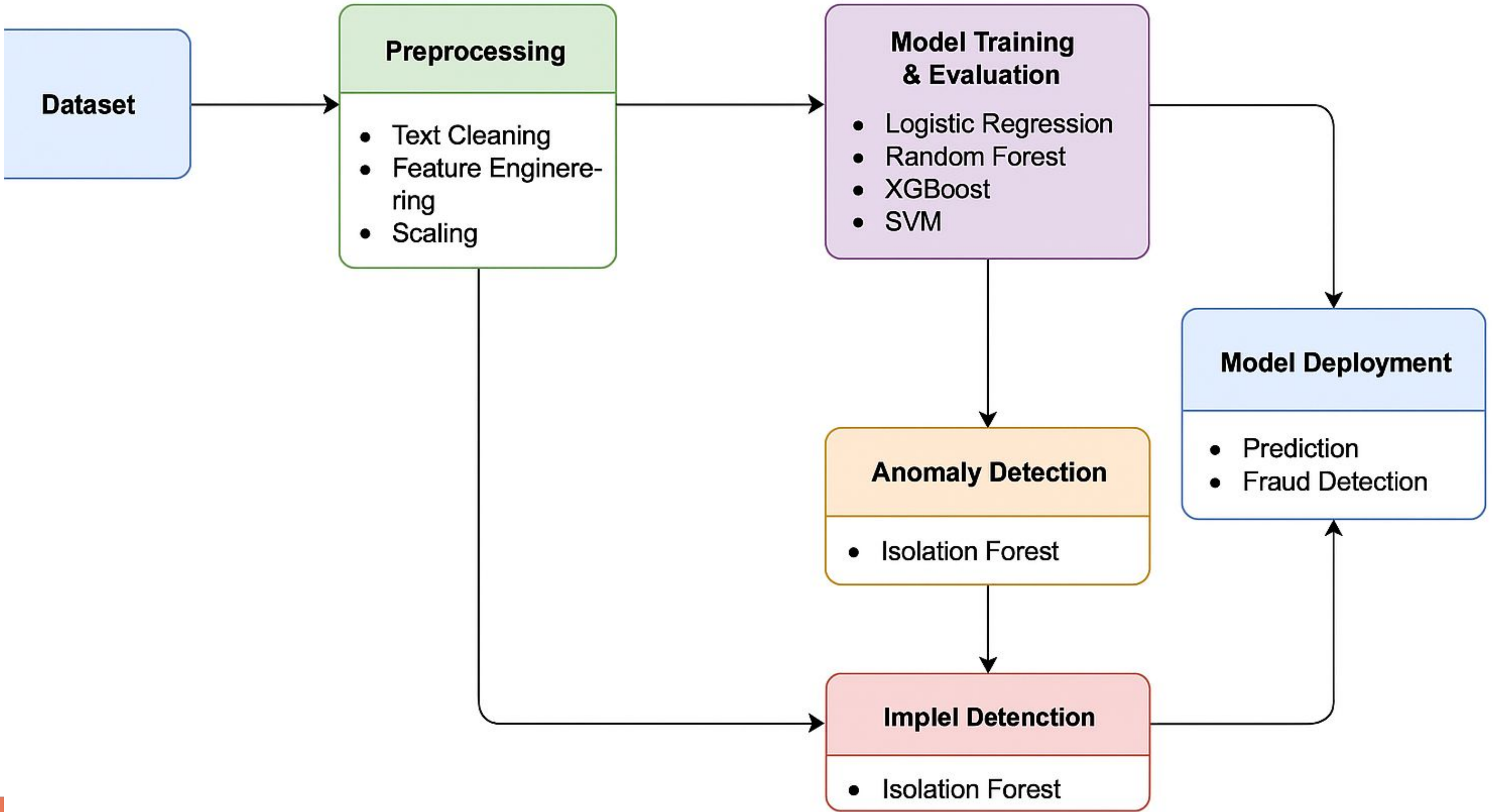
1. Data Ingestion and Preprocessing:

- **Data Source**: The primary data source is a CSV file (`grant_proposals_synthetic.csv`), containing grant proposals with fields like titles, abstracts, budgets, and research impact.
- **Data Cleaning**: The first step involves cleaning and preparing the data. Missing values are handled, and categorical variables (e.g., institution names) are encoded.
- **Text Processing**: The title and abstract columns undergo text cleaning, including:
    - Converting text to lowercase.
    - Removing non-alphabetic characters.
    - Tokenizing text and lemmatizing to reduce words to their root form.

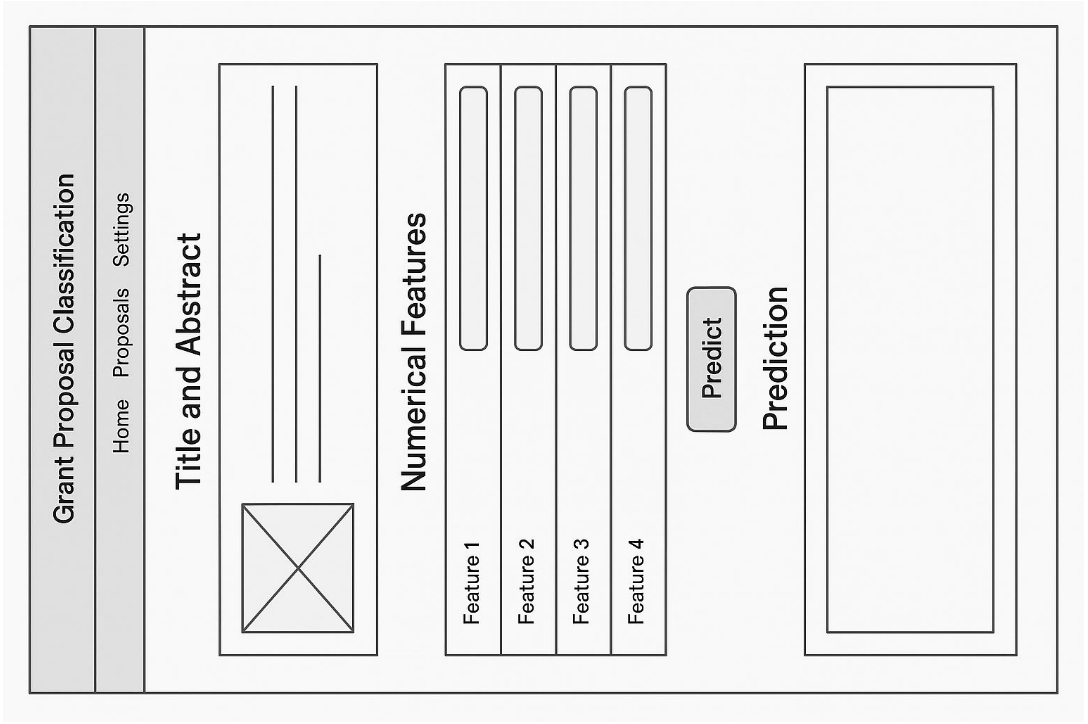
THE

Pro d
sol

Grant Proposal Classification Pipeline AI Innovation by Eric Meaiki



Please share the wireframes/Mock diagrams of the proposed solution (optional)



What datasets will your solution use? Are they publicly available, synthetic, or user-generated?

The solution uses the following datasets:

1. **Grant Proposals Dataset**:
   - **Description**: This dataset contains synthetic grant proposal data, including fields like title, abstract, budget, research impact, and funding status (e.g., funded or not funded).
   - **Type**: **Synthetic** dataset, specifically designed to simulate real-world grant proposal data for classification tasks.
   - **Availability**: The dataset is not publicly available but is generated specifically for this solution.
2. **Additional Datasets (optional)**:
   - **Description**: Depending on the specific project or use case, the solution can be adapted to work with additional datasets like real-world grant proposals, institutional data, or proposal outcome datasets.

Does your solution require cloud-based computation, or can it work with on-device processing? If cloud-based, how do you plan to address connectivity challenges and cost constraints?

The solution can be implemented both with **cloud-based computation** or with **on-device processing**, depending on the scale and use case. Here's a breakdown of both approaches:

Cloud-based Computation:

- **Why Cloud-based**: For large-scale data processing, model training, and real-time predictions, cloud infrastructure is preferred. It enables easier scaling, storage of large datasets, and the ability to train complex models (like XGBoost or Random Forest) without being limited by local hardware constraints.
- **Handling Connectivity Challenges**:
  - **Data Storage**: The dataset can be stored on cloud storage services (e.g., AWS S3, Google Cloud Storage), ensuring it's accessible from anywhere. To mitigate intermittent connectivity issues, data can be downloaded in chunks and processed locally if needed.
  - **Offline Mode**: The models can be designed to operate in an offline mode with periodic updates from the cloud. This allows the solution to continue functioning even when connectivity is temporarily disrupted.
  - **Edge Computing**: For certain parts of the solution (like real-time predictions), edge devices (e.g., IoT devices) can

Thank You

Grant Proposal Classification Pipeline

— By Eric Mwaniki