**Developing an Intelligent Employee Access Control System using Machine Learning**

**22K-4778 – Ahmed Hussain**

**22K-4771 – Mehtab Ali**

**22K-4709 – Zakyas Ali**

**1. Introduction**

Managing employee access to various organizational resources (applications, files, systems, etc.) is a critical aspect of information security. Traditional access control methods often rely on predefined, static rule sets and policies (e.g., Role-Based Access Control - RBAC, Access Control Lists - ACLs). While effective to a degree, these methods can be rigid, complex to manage, slow to adapt to changing organizational structures or employee roles, and may not effectively detect unusual or potentially unauthorized access patterns.

This project proposes an innovative approach using machine learning to create a more intelligent and adaptive access control system. By analyzing historical data of access requests, employee attributes, resource types, and past actions (granted or denied), a machine learning model can learn complex patterns that dictate appropriate access. This learned model can then be used to make real-time access decisions, potentially offering greater accuracy in distinguishing legitimate requests from suspicious ones and adapting permissions based on evolving usage patterns.

**2. Problem Statement**

Traditional access control mechanisms struggle with:

Complexity: Managing intricate rule sets for a large number of employees and resources.

Lack of Adaptability: Policies are slow to update as roles or organizational needs change.

Inability to Detect Anomalies: Static rules are poor at identifying access requests that deviate from typical behavior but don't explicitly violate a rule.

Maintenance Overhead: Manually updating and reviewing policies is time-consuming and error-prone.

This project addresses these issues by proposing a machine learning-based solution that can learn and adapt access control logic directly from historical data, aiming for a more dynamic, accurate, and potentially less burdensome system than solely relying on static policy enforcement.

**3. Project Objectives**

The primary objectives of this project are:

To explore and understand the characteristics of the dataset, including class distribution of access actions.

To split the dataset into appropriate training and testing sets.

To develop and train one or more machine learning classification models (e.g., RandomForest, LightGBM) capable of predicting whether a given access request should be authorized or unauthorized based on historical patterns.

To evaluate the performance of the trained model(s) using relevant metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

To demonstrate the potential of the developed model to serve as a basis for replacing or augmenting traditional access control policies.

To provide a clear report on the model's performance and suitability for the task.

**4. Scope of Work**

The scope of this project includes:

**Data Loading and Preparation:** Loading the train.csv dataset from Kaggle, handling data types, and initial inspection.

**Exploratory Data Analysis (EDA):** Analyzing features, understanding data distribution, visualizing the target variable distribution.

**Data Preprocessing:** Scaling numerical features (StandardScaler).

**Data Splitting:** Dividing the dataset into training and testing subsets.

**Model Selection and Training:** Implementing and training at least two classification algorithms (RandomForestClassifier, LGBMClassifier as indicated in code) on the prepared data.

**Model Evaluation:** Calculating and reporting key performance metrics (Accuracy, Precision, Recall, F1-Score, Confusion Matrix, Classification Report) on the test set.

**Reporting:** Documenting the process, findings, model performance, and potential implications.

The following are **out of scope** for this project phase:

Deployment of the model into a production environment.

Integration with existing organizational IT infrastructure or identity management systems.

Development of a user interface for access requests or administration.

Collecting or incorporating real-time organizational data.

Advanced feature engineering beyond basic preprocessing if not explicitly required by initial model performance.

**5. Methodology and Approach**

**Data Preparation:**

Separate features (X) and the target variable (y).

Apply feature scaling using StandardScaler on training data and transform both training and testing data.

Split the data into training and testing sets using train\_test\_split (e.g., 80% training, 20% testing).

**Model Selection:** Choose appropriate classification algorithms suitable for the binary classification task (access granted/denied). Based on the provided code, RandomForestClassifier and LGBMClassifier will be initial candidates.

**Model Training:** Train the selected models on the scaled training data (X\_train\_scaled, y\_train).

**Model Evaluation:**

Use the trained models to make predictions on the scaled test data (X\_test\_scaled).

Evaluate model performance using the metrics calculated in the model\_train function: Accuracy, Classification Report (including Precision, Recall, F1-Score per class), Confusion Matrix, and derived metrics like True Positive Rate (Recall) and False Positive Rate.

**Reporting:** Document the steps, results, and analysis in a final project report.

**6. Data Source**

The project will exclusively use the publicly available "Employee Access" dataset from Kaggle, specifically the train.csv file. This dataset contains historical access requests with anonymized features representing employee attributes, resource attributes, and the outcome ('ACTION' - 1 for authorized, 0 for unauthorized).

**7. Deliverables**

Upon completion, the project will deliver the following:

A comprehensive report detailing the methodology, exploratory data analysis findings, model performance metrics for each trained model, and a discussion of the results and potential implications for access control.

A presentation summarizing the project's findings (Optional).

**8. Timeline**

Assuming dedicated effort, the project timeline is estimated as follows:

**Week 1:** Data Acquisition, Initial Exploration, and Preprocessing. (Loading data, understanding columns, EDA, plotting distribution, splitting data, scaling).

**Week 2:** Model Development and Training. (Implementing model training function, training RandomForest and LGBM models).

**Week 3:** Model Evaluation and Analysis. (Running evaluation metrics, interpreting results, comparing models).

**Week 4:** Reporting and Documentation. (Writing the project report, finalizing code, preparing deliverables)

**9. Resources Required**

**Personnel:** One Data Scientist / Machine Learning Engineer.

**Software:** Python 3.x, essential libraries (pandas, numpy, scikit-learn, lightgbm, matplotlib, seaborn), Integrated Development Environment (IDE) or Jupyter Notebook environment.

**Hardware:** A standard computer with sufficient processing power and RAM to handle the dataset and model training (typical for medium-sized datasets like the Kaggle one).

**Data:** Access to download the Kaggle Employee Access dataset (train.csv).

**10. Evaluation Metrics**

The success of the model will be evaluated based on its performance on the test set using the following metrics:

**Accuracy:** Overall correct predictions.

**Confusion Matrix:** Detailed breakdown of True Positives, True Negatives, False Positives, and False Negatives.

**Classification Report:** Provides Precision, Recall, and F1-score for each class (0 and 1).

**Precision (for class 0 - Unauthorized):** Out of all predictions of unauthorized access, how many were actually unauthorized? (Minimizing False Positives).

**Recall (for class 0 - Unauthorized / True Positive Rate):** Out of all actual unauthorized access instances, how many were correctly identified? (Minimizing False Negatives - crucial for security).

**F1-Score (for class 0):** Harmonic mean of Precision and Recall, balancing both metrics.

**False Positive Rate (FPR):** The rate at which authorized access is incorrectly flagged as unauthorized (can impact usability).

Emphasis will be placed on metrics related to correctly identifying unauthorized access (Recall for class 0) while managing the rate of falsely denying authorized access (FPR).

**11. Risks and Mitigation**

**Risk:** The Kaggle dataset may not fully represent the complexity and nuances of real-world organizational access patterns.

**Mitigation:** Acknowledge this limitation in the report. Focus evaluation on the model's capability *on this dataset* and discuss potential challenges or needs when transitioning to real-world data.

**Risk:** Model performance may not be sufficiently high to reliably replace static policies.

**Mitigation:** Explore different models, hyperparameter tuning (within scope), and potentially more advanced preprocessing or feature engineering if time permits. Clearly report achieved performance and discuss implications.

**Risk:** The model might be a "black box," making it hard to explain why a specific access decision was made.

**Mitigation:** While full explainability is out of scope, consider discussing techniques like SHAP or LIME in the report as future work. Focus on demonstrating *predictive performance* in this phase.

**Risk:** Class Imbalance (if present in the dataset, though the plot suggests it's not severely imbalanced).

**Mitigation:** Evaluate using metrics robust to imbalance (Precision, Recall, F1-Score) and consider techniques like weighted classes or over/undersampling if imbalance is significant (not explicitly needed based on the provided plot, but good practice to consider). The provided plot shows ~60% for 1 and ~40% for 0, which is manageable imbalance.