MOD10: Machine learning
Fall 2025
Instructor: Mohammed A. Shehab

Machine learning project information (30%)

Project Details

Objective:
Each project is focused on solving a specific real-world problem through machine learning, including tasks like data preprocessing, model training, evaluation, deployment, and tracking. Students will explore the complete model-building lifecycle, from initial data analysis to deployment.

Dataset:
Each project is associated with a suitable dataset:

- **Cybersecurity Threat Detection**: CICIDS 2017 Dataset (Kaggle).
- **Sentiment Analysis**: IMDb Movie Reviews Dataset.
- **Plant Disease Detection**: Plant Village Dataset.
- **Patient Health Risk Prediction**: Diabetes Health Indicators Dataset or Heart Disease UCI Dataset.

Project Components

1. **Feature Engineering and Exploratory Data Analysis (EDA)**
    - **Objective**: Understand the dataset, identify trends, and engineer new features to improve model performance.
    - **Tasks**:
        - Conduct an exploratory data analysis to observe feature distributions, correlations, and patterns.
        - Visualize data with histograms, box plots, and heatmaps.
        - Create additional features if they could enhance model accuracy.
    - **Deliverable**: Insights from EDA, supported by visualizations, included in the final report.
2. **Model Training and Comparison**
    - **Objective**: Train, compare, and select the best-performing model, exploring ensemble methods if applicable.
    - **Tasks**:

    - Train at least two different models (e.g., decision tree, random forest) and evaluate their performance.
    - Experiment with ensemble techniques (e.g., voting, stacking) to potentially improve performance.
  - **Deliverable**: A report section discussing model comparison, choice, and the effect of any ensemble methods.

3. **Evaluation Metrics**
  - **Objective**: Assess model performance through a variety of metrics to understand strengths and weaknesses.
  - **Tasks**:
    - For classification: Calculate metrics like ROC, AUC, accuracy, precision, recall, and F1-score.
    - For regression (if relevant): Use R-squared, MAE, and MSE.
  - **Deliverable**: Comparative analysis of metrics, documented in the report.

4. **Interpretability and Explainability**
  - **Objective**: Analyze the model's decision-making process to understand feature importance and enhance transparency.
  - **Tasks**:
    - Use interpretability tools (e.g., SHAP, LIME) to explain key features influencing predictions, especially for complex models.
  - **Deliverable**: A report section explaining the most influential features based on model interpretations.

5. **Deployment with Docker**
  - **Objective**: Containerize the model using Docker and deploy it with a REST API for real-time predictions.
  - **Tasks**:
    - Create a Dockerfile with all dependencies, enabling easy deployment.
    - Set up a REST API (using Flask or FastAPI) to serve predictions.
  - **Deliverable**: Docker image with instructions for running and testing the container.

6. **Performance Tracking with MLflow**
  - **Objective**: Log model metrics, parameters, and experiment versions using MLflow for performance tracking.

- **Tasks**:
  - Track model metrics during training and deployment for analysis and comparison.
  - Use MLflow to log all experiments, parameters, and versions.
- **Deliverable**: MLflow logs showing metrics and comparison across model versions.

Project Workflow

1. **Data Preprocessing and EDA**
   - Load and clean the dataset, handling missing values, normalizing or scaling numerical features, and encoding categorical variables.
   - Conduct EDA, visualizing feature distributions and identifying potential relationships to guide feature engineering.
2. **Model Training, Comparison, and Ensemble**
   - Train multiple models (e.g., logistic regression, decision trees) and evaluate them using relevant metrics.
   - Experiment with ensemble methods (e.g., voting, stacking) if it improves model performance.
3. **Evaluation Metrics and Analysis**
   - Evaluate models on various metrics, comparing results to understand model strengths and weaknesses, particularly for imbalanced datasets.
4. **Interpretability and Explainability**
   - Use tools like SHAP or LIME to explain model decisions and analyze feature importance, ensuring model transparency.
5. **Docker Deployment**
   - Build and deploy the model in a Docker container with a REST API, allowing for easy real-time predictions.
6. **MLflow Tracking**
   - Log all experiments, tracking parameters and metrics over multiple model versions with MLflow.

Deliverables

1. **Code**:
    - Scripts or Jupyter notebooks for EDA, model training, evaluation, deployment, and MLflow tracking.
2. **EDA Visualizations**:
    - Visualizations such as histograms, box plots, and heatmaps documenting data characteristics and feature correlations.
3. **Model Comparison and Interpretability Analysis**:
    - Report sections analyzing model performance and feature importance, including any ensemble techniques applied.
4. **Docker and MLflow**:
    - Dockerfile and container instructions for deployment, plus MLflow logs tracking model metrics.
5. **Final Report**:
    - A report summarizing project objectives, methodology, EDA findings, model evaluation, interpretability analysis, deployment steps, and key takeaways.

Evaluation Criteria

1. **Data Preprocessing and EDA (20%)**
    - **EDA Insights (10%)**: Effective visualizations and insights on feature distributions and correlations.
    - **Feature Engineering (10%)**: Creation of new features and analysis of their impact on model performance.
2. **Model Training, Comparison, and Ensemble (25%)**
    - **Model Implementations (15%)**: Successful training and tuning of at least two models, with clear model selection rationale.
    - **Ensemble Techniques (10%)**: Exploration and analysis of ensemble techniques, if applied.
3. **Evaluation Metrics (15%)**
    - **Metric Analysis (15%)**: Calculation and interpretation of metrics, with discussion on metric effectiveness for the project's objectives.

4. **Interpretability and Explainability (10%)**
   - **Feature Importance (10%)**: Clear analysis using SHAP, LIME, or similar tools to explain key features affecting model predictions.
5. **Deployment and Performance Tracking (20%)**
   - **Docker Deployment (10%)**: Complete Docker container with functional API for predictions.
   - **MLflow Tracking (10%)**: Accurate MLflow logging of parameters, metrics, and model versions.
6. **Presentation (10%)**
   - **Clarity and Organization (5%)**: Clear presentation, effectively communicating objectives, methodology, and results.
   - **Visuals and Engagement (5%)**: Use of visuals for data insights and effective collaboration among group members.