Machine Maintenance Predictive Analysis

This project explores predictive maintenance using a dataset of machine operation parameters. Techniques include Exploratory Data Analysis (EDA), Random Forest (RF), Artificial Neural Networks (ANN), data balancing, and bagging.

Requirements

The project is implemented in **Python 3.9+** using Jupyter Notebook. Make sure the following libraries are installed before running the notebook:

- **pandas**
- **numpy**
- **matplotlib**
- **seaborn**
- **scikit-learn**
- **imbalanced-learn**
- **tensorflow**

Installing Dependencies

Use the following command in the jupyer notebook cell to install all required libraries:

```
!pip install pandas numpy matplotlib seaborn scikit-learn imbalanced-learn tensorflow
```

Running the Program

1. **Download the Files**:
   Ensure you have the `.ipynb` file and the dataset in the same directory.

2. **Launch the Notebook**:
   Open the notebook file `machine-maintenance-eda-rf-ann-balance-bagging.ipynb` using Jupyter Notebook.

3. **Load the Dataset**:
   Update the file path in the notebook's dataset loading cell if necessary.

4. **Execute the Notebook**:
   Run the cells sequentially to perform:
   - Data exploration (EDA)
   - Model training (Random Forest, ANN)
   - Data balancing and bagging
   - Model evaluation

5. **Outputs**:
   The notebook generates the following outputs:
   - Visualizations like heatmaps, pair plots, and boxplots
   - Confusion matrices and performance metrics for models

Dataset Overview

The dataset contains the following features:

Feature	**Description**
Product_ID	Identifier for the product
Type	Machine type
Air_temperature	Air temperature (Kelvin)
Process_temperature	Process temperature (Kelvin)
Rotational_speed	Rotational speed (RPM)
TorqueNm	Torque (Nm)
Tool_wear	Tool wear (minutes)
FailureType	Target variable (failure modes)

Key Techniques Used

1. **EDA**:
   - Visualized data distributions and relationships between features.
   - Detected outliers using boxplots.

2. **Models**:
   - Random Forest (RF) for baseline predictions.
   - Artificial Neural Network (ANN) for advanced modeling.

3. **Data Balancing**:
   - Addressed imbalanced data using `imbalanced-learn` techniques.

4. **Bagging**:
   - Improved model performance with ensemble methods.

Notes

- Make sure your Python environment is set up with the required dependencies.
- If the dataset file is not in the same directory, update the dataset path in the notebook.
- For better performance with ANN models, consider using a GPU-enabled TensorFlow installation.
