**Mini Project: Predictive Maintenance Model for an Industrial Pump**

**Project Overview:**

The goal of this mini project is to assess your ability to build, evaluate, and simulate the deployment of a machine learning model for predictive maintenance. You'll work with a dataset that simulates sensor readings from an industrial pump. Your task is to develop a model that predicts when the pump is likely to fail based on these readings.

**Project Tasks:**

1. **Data Preprocessing:**
   * **Dataset**: You will work with a dataset containing sensor data, such as vibration levels, temperature, pressure, and flow rate, along with a timestamp and a binary label indicating pump failure (1 = failure, 0 = no failure).
   * **Task**: Clean the dataset by handling missing values, normalizing/standardizing the features, and applying feature engineering techniques that could improve the model’s performance.
2. **Exploratory Data Analysis (EDA):**
   * **Task**: Perform exploratory data analysis to understand the distribution of the data, identify correlations between features, and uncover patterns that could inform your modeling approach. Visualize these insights using charts and graphs (e.g., histograms, correlation matrices, box plots).
3. **Model Development:**
   * **Task**: Build a predictive model using Python and relevant libraries (e.g., Scikit-learn, TensorFlow, Keras). You may choose a classification model (e.g., Random Forest, Logistic Regression) or a time-series model (e.g., LSTM, ARIMA), depending on your approach.
   * **Output**: Train the model on the provided dataset, evaluate its performance using appropriate metrics (e.g., accuracy, precision, recall, F1-score, ROC/AUC), and fine-tune the model to optimize these metrics.
4. **Model Evaluation:**
   * **Task**: After training the model, evaluate its performance on a test set or through cross-validation. Provide a brief report on the model’s performance, discussing its strengths, limitations, and potential areas for improvement.
5. **Data Visualization and Reporting:**
   * **Task**: Create visualizations to help explain the model’s predictions. For example, you might visualize the relationship between specific sensor readings and pump failures or demonstrate the model’s predictive accuracy.
   * **Output**: Present your findings and the model’s performance in a Jupyter Notebook, including all code, visualizations, and explanations. Ensure the report is clear and understandable to both technical and non-technical stakeholders.
6. **Model Deployment Simulation:**
   * **Task**: Simulate a deployment scenario by creating a simple script or API that takes new sensor data as input and outputs a failure prediction. While actual deployment on a cloud platform is not required, you should demonstrate how the model could be integrated into a real-world application.
7. **Bonus Task (Optional):**
   * **Advanced Feature Engineering**: If time allows, create additional features (e.g., rolling averages, trend indicators) that might enhance the model’s predictive power.

**Project Deliverables:**

* **Jupyter Notebook**: Containing your code, analysis, model development, evaluation, and visualizations.
* **Report**: Summarizing your approach, findings, and the model’s performance.
* **Simulation Script**: A Python script that simulates how the model could be used in a real-world scenario to predict pump failures.

**Timeline:**

* Please complete the project and submit your deliverables by [specific date/time, e.g., "Monday, September 2nd, 5:00 PM"]. This will give us time to review your work before proceeding with the next steps in the interview process.

**Submission:**

* You can submit your work via email, either as attachments or by providing a link to a GitHub repository containing your project files.

**Assessment Criteria:**

* **Technical Proficiency**: Your ability to preprocess data, build and evaluate a model, and handle challenges such as imbalanced data or overfitting.
* **Problem-Solving**: How effectively you approach the problem, including your choice of model and methods for improving its accuracy.
* **Communication**: Clarity and thoroughness of your Jupyter Notebook, including explanations of your approach and results.
* **Creativity**: Any innovative approaches or additional steps taken to enhance the model or improve its interpretability.

**Additional Instructions:**

1. **Dataset Details:**
   * The dataset will include features like timestamp, vibration level, temperature, pressure, flow rate, and a binary target variable indicating pump failure. You can simulate or generate your own dataset if necessary, ensuring it reflects these features.
2. **Assumptions:**
   * Document any assumptions you make during the project, such as the importance of certain features or thresholds used in feature engineering.
3. **Libraries and Tools:**
   * Feel free to use any Python libraries you're comfortable with (e.g., Scikit-learn, TensorFlow, Keras, Pandas, NumPy) but avoid using pre-built solutions that bypass the core tasks.
4. **Documentation:**
   * Ensure that the Jupyter Notebook is well-documented, with comments explaining key steps and markdown cells describing your approach, reasoning, and conclusions.
5. **Communication of Results:**
   * Focus on making your results and explanations understandable, especially for stakeholders who may not have a deep technical background.
6. **Time Management:**
   * Prioritize tasks that demonstrate your core skills and knowledge, managing your time to deliver results within the given timeframe.
7. **Reproducibility:**
   * Ensure that your code is reproducible on another machine, specifying any dependencies or library versions used.
8. **Professional Presentation:**
   * Present the final deliverable professionally, as if it were being delivered to a client or senior management.

**Final Thoughts:**

We encourage you to view this mini project not just as a test, but as an opportunity to showcase your creativity, technical skills, and approach to solving real-world problems. There is no single “right” way to approach this project—what matters most is how you apply your skills and thinking to tackle the task.