Pump Failure Prediction Report

1. Introduction:-

Predictive maintenance is a crucial aspect of industrial operations, where identifying potential equipment failures before they occur can save significant time and costs. In this context, machine learning models are often employed to predict failures based on sensor data. However, challenges such as class imbalance and small sample sizes can significantly impact model performance, leading to issues like overfitting. This report discusses the application of various machine learning models to predict pump failures, addressing the challenges posed by imbalanced data. The report also explains the process of model selection, optimization, and deployment of a predictive model using a Streamlit app. Also I had used various Data Visualizations to better interpret data so we could have a great understanding how our data looks like (can be seen in Notebook file).

2. Dataset and Preprocessing:-

The dataset used in this project contains sensor readings and failure indicators for a hypothetical pump system. The features included in the dataset are:

vibration_level: The level of vibration in the pump.

temperature_C: The temperature of the pump in degrees Celsius.

pressure_PSI: The pressure within the pump in pounds per square inch (PSI).

flow_rate_m3h: The flow rate through the pump in cubic meters per hour.

The target variable is **failure**, where 0 indicates no failure and 1 indicates a failure in the pump. A significant challenge with this dataset is the class imbalance, with far fewer instances of pump failure (1) compared to no failure (0).

To prepare the data for modeling, the following steps were taken:

Data Splitting: The dataset was split into training and testing sets using an 80-20 ratio. This means that 80% of the data was used to train the models, while the remaining 20% was reserved for testing the models' performance.

Feature Scaling: Given that the features are on different scales (e.g., temperature in degrees Celsius vs. flow rate in cubic meters per hour), a StandardScaler was applied to transform the features. This step ensures that each feature contributes equally to the model's predictions, avoiding bias towards any particular feature due to its scale.

3. Model Selection and Parameter Optimization

Several machine learning models were considered for predicting pump failures:

Logistic Regression: A linear model commonly used for binary classification problems.
Support Vector Machine (SVM): A model that attempts to find the optimal hyperplane that separates the classes.
Multilayer Perceptron (MLP): A type of neural network that can capture complex patterns in the data.
Random Forest Classifier: An ensemble method that uses multiple decision trees to improve predictive performance.
Gradient Boosting Classifier: Another ensemble method that builds trees sequentially, where each tree tries to correct the errors of the previous ones.
Parameter Optimization: To ensure the models performed optimally, a grid search technique was applied to each model. Grid search is a method that systematically works through multiple combinations of hyperparameters to find the best model configuration. This process was repeated for all models, ensuring that the best possible parameters were used in the final model.

4. Addressing Class Imbalance and Overfitting

One of the major challenges encountered during model training was the significant class imbalance in the dataset. The minority class (1 - indicating failure) was underrepresented, leading to potential overfitting where models could predict the majority class (0 - indicating no failure) with high accuracy but fail to correctly predict the minority class. Overfitting occurs when a model learns the noise in the training data rather than the true underlying patterns, leading to poor generalization on unseen data.

Despite the class imbalance, the Random Forest and Gradient Boosting classifiers exhibited better precision and recall than other models. These models are known for their robustness to overfitting, especially when combined with techniques such as grid search for hyperparameter tuning. Precision refers to the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positive predictions among all actual positive instances. In this

context, a higher precision and recall indicate that the models are better at correctly identifying pump failures without being overly influenced by the majority class.

5. Model Selection for Deployment

Given the challenges and the performance of the models, the Random Forest classifier was selected for deployment. It provided a good balance between precision, recall, and overall model stability. The final model was trained on the scaled features and optimized using grid search. The following steps were taken:

Model Saving: The trained Random Forest model was saved using joblib, ensuring that it could be easily loaded and used for making predictions on new data.
Feature Input: The model takes four input features: vibration_level, temperature_C, pressure_PSI, and flow_rate_m3h. These features are first scaled using the previously fitted StandardScaler before being passed to the model for prediction.
Prediction Output: The model outputs a binary prediction indicating whether there is a failure in the pump or not (0 for no failure, 1 for failure).

6. Deployment Using Streamlit

To make the model accessible and user-friendly, a web application was developed using Streamlit. Streamlit is a popular Python library that allows for the rapid development of web applications. The app allows users to input the four sensor readings and get a real-time prediction on whether the pump is likely to fail.

Streamlit App Features:

User Input: The app takes input for the four sensor features.
Prediction: Upon submission, the app processes the input through the trained Random Forest model and returns a prediction of whether the pump is likely to fail.
Deployment: The app can be deployed on various platforms, including Streamlit Cloud or Heroku, making it accessible to users across different locations.

7. Conclusion

The process of predicting pump failures using machine learning models involved addressing significant challenges related to class imbalance and overfitting. By applying techniques such as feature scaling, data splitting, and grid search for parameter optimization, a robust Random Forest model was developed and deployed. The Streamlit app further enhances the usability of this model, allowing for real-time predictions based on sensor data. While the current model performs well, future improvements could include collecting more data, especially instances of pump failures, to further improve model accuracy and reliability.