**Proposal for Predictive Model Development**

**Introduction**

In this project, the objective is to predict whether a truck part will fail within 28 days. However, the dataset lacks a direct feature indicating failure (the target variable). Therefore, we must engineer a target variable based on available maintenance and failure history data. Once the target variable is defined, we can preprocess the data and select appropriate machine learning models for accurate failure prediction.

**1. Preprocessing Data for Machine Learning**

Data preprocessing is a crucial step in machine learning, as it ensures that the data fed into the models is clean, consistent, and suitable for accurate predictions. The following steps are recommended for preprocessing the data effectively:

**1.1 Handling Missing Data**

* **Identifying Missing Data**: The first step is to identify columns with missing data using methods like df.isna().sum(). Columns with excessive missing values (e.g., more than 50%) should be removed, as they will likely add noise to the model.
* **Imputation for Minor Missing Data**: For columns with a smaller percentage of missing values, imputation methods such as mean, median, or mode imputation should be used for numeric columns. For categorical columns, the mode (most frequent value) can be used to fill in missing data.
* **Dropping Irrelevant Columns**: Columns that contain only one unique value or are entirely made up of missing values should be dropped from the dataset. These columns do not offer valuable information and can negatively affect the model's performance.

**1.2 Handling Date-Time Features**

Many predictive models for industrial maintenance rely on time-related features such as failure history, service dates, and part usage. These date-time columns must be processed appropriately:

* **Conversion to Datetime**: Use methods like pd.to\_datetime() to convert date-related columns into a standardized datetime format, ensuring consistency.
* **Feature Engineering**: Extract useful features from date-time columns, such as year, month, day, day of the week, and hour. These features can capture temporal patterns that are often useful in predicting failures. For example, certain failure patterns may be related to specific seasons or times of the year.

**1.3 Categorical Data Encoding**

Many datasets include categorical variables, such as truck part types or failure status. Since most machine learning models cannot process raw categorical data directly, encoding is necessary:

* **Label Encoding**: For ordinal categorical variables (where categories have an inherent order, such as severity levels), label encoding can be used. This converts each category into a numerical value.
* **One-Hot Encoding**: For nominal categorical variables (where no order is implied), one-hot encoding is recommended. This creates a new binary column for each category, allowing the model to treat the categories as distinct values.

**1.4 Feature Scaling and Normalization**

Feature scaling ensures that numerical features are on a similar scale, which can improve the performance of certain machine learning models (e.g., those based on distance metrics, such as k-nearest neighbors or linear models). The following steps should be followed:

* **Scaling**: Use Min-Max Scaling or Standardization (Z-score normalization) for numerical columns. Min-Max scaling scales data to a specified range (e.g., 0 to 1), while standardization normalizes data by centering it around zero and scaling to unit variance.
* **Log Transformation**: In case of highly skewed data, applying a log transformation can help normalize the distribution. This can be particularly useful for variables like the age of truck parts, where the values may vary across a wide range.

**1.5 Handling Imbalanced Data**

Imbalanced datasets, where the target variable (e.g., failure or no failure) has significantly more instances of one class, can cause models to be biased toward the majority class. To address this:

* **Resampling**: Use techniques like SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic samples for the minority class, or apply undersampling to reduce the number of instances from the majority class.
* **Class Weight Adjustment**: Many machine learning algorithms, like logistic regression and decision trees, allow for class weight adjustment. This gives more importance to the minority class during training, helping to balance the model’s prediction ability across both classes.

**2. Model Development for Predicting Truck Part Failures**

Once the data is preprocessed, machine learning models can be trained to predict whether a truck part will fail within the next 28 days. The following models are recommended for this predictive task:

**2.1 Model Selection**

* **Logistic Regression**: Logistic regression is a simple yet effective algorithm for binary classification problems (failure vs. no failure). It provides a good baseline model and is easily interpretable, helping to identify which features contribute the most to predictions.
* **Random Forest**: Random Forest is an ensemble learning method that works well for both classification and regression tasks. It can handle complex relationships between variables and is particularly robust when dealing with a mix of numeric and categorical data. Random forests can also provide valuable insights into feature importance, helping to understand which variables are most predictive of failures.
* **Gradient Boosting Machines (GBM)**: Algorithms like XGBoost or LightGBM offer high predictive accuracy and are ideal for imbalanced datasets. These models build weak learners iteratively to correct errors made by previous models, leading to improved performance on complex data. They are especially well-suited for predicting events like part failures that may occur irregularly.
* **Neural Networks**: For larger datasets with complex patterns, deep learning models, such as feedforward neural networks or recurrent neural networks (RNNs), can be considered. These models are more computationally intensive but may uncover intricate patterns that simpler models might miss. They are particularly useful for time-series data, where sequential dependencies may exist.

**2.2 Model Evaluation**

After training the models, the following metrics should be used to evaluate their performance:

* **Accuracy**: Measures the overall percentage of correct predictions. However, in the case of imbalanced data, accuracy alone may not be sufficient.
* **Precision and Recall**: Precision measures the percentage of correct positive predictions, while recall measures the percentage of actual positives correctly identified. These metrics are crucial when dealing with imbalanced datasets.
* **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of performance, especially when dealing with imbalanced data.
* **ROC-AUC**: The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) offer a way to assess model performance across various thresholds, highlighting the trade-off between true positive rate and false positive rate.

**2.3 Hyperparameter Tuning**

Each machine learning model has several hyperparameters that can significantly impact its performance. Use Grid Search or Random Search to optimize these parameters, such as the number of trees in a random forest, the learning rate in gradient boosting models, or the number of layers and units in a neural network. Proper hyperparameter tuning can lead to substantial improvements in model accuracy.

**Conclusion**

This proposal outlines the necessary steps for preprocessing the data and selecting the most suitable models for predicting truck part failures within 28 days. By addressing issues such as missing data, categorical encoding, feature scaling, and handling imbalanced data, we can ensure that the dataset is optimized for machine learning. Additionally, the selection of appropriate models, including logistic regression, random forests, gradient boosting machines, and neural networks, will allow for accurate predictions and help identify the most important factors contributing to truck part failures. With proper evaluation and hyperparameter tuning, this approach will provide actionable insights to predict failures and optimize maintenance schedules effectively.