Project Title	**Industrial Copper Modeling**
Skills take away From This Project	**Python scripting, Data Preprocessing, EDA, Streamlit**
Domain	**Manufacturing**

Problem Statement:

The copper industry deals with less complex data related to sales and pricing. However, this data may suffer from issues such as skewness and noisy data, which can affect the accuracy of manual predictions. Dealing with these challenges manually can be time-consuming and may not result in optimal pricing decisions. A machine learning regression model can address these issues by utilizing advanced techniques such as data normalization, feature scaling, and outlier detection, and leveraging algorithms that are robust to skewed and noisy data.

Another area where the copper industry faces challenges is in capturing the leads. A lead classification model is a system for evaluating and classifying leads based on how likely they are to become a customer . You can use the STATUS variable with WON being considered as Success and LOST being considered as Failure and remove data points other than WON, LOST STATUS values.

The solution must include the following steps:

1) Exploring skewness and outliers in the dataset.
2) Transform the data into a suitable format and perform any necessary cleaning and pre-processing steps.
3) ML Regression model which predicts continuous variable ‘Selling_Price’.
4) ML Classification model which predicts Status: WON or LOST.
5) Creating a streamlit page where you can insert each column value and you will get the Selling_Price predicted value or Status(Won/Lost)

Data: Data-set

Approach:

1) Data Understanding: Identify the types of variables (continuous, categorical) and their distributions. Some rubbish values are present in ‘Material_Reference’ which starts with ‘00000’ value which should be converted into null. Treat reference columns as categorical variables. INDEX may not be useful.
2) Data Preprocessing:
    - Handle missing values with mean/median/mode.
    - Treat Outliers using IQR or Isolation Forest from sklearn library.
    - Identify Skewness in the dataset and treat skewness with appropriate data transformations, such as log transformation(which is best suited to transform target variable-train, predict and then reverse transform it back to original scale eg:dollars), boxcox transformation, or other techniques, to handle high skewness in continuous variables.
    - Encode categorical variables using suitable techniques, such as one-hot encoding, label encoding, or ordinal encoding, based on their nature and relationship with the target variable.
3) EDA: Try visualizing outliers and skewness(before and after treating skewness) using Seaborn’s boxplot, distplot, violinplot.
4) Feature Engineering: Engineer new features if applicable, such as aggregating or transforming existing features to create more informative representations of the data. And drop highly correlated columns using SNS HEATMAP.
5) Model Building and Evaluation:
    - Split the dataset into training and testing/validation sets.
    - Train and evaluate different classification models, such as ExtraTreesClassifier, XGBClassifier, or Logistic Regression, using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, and AUC curve.
    - Optimize model hyperparameters using techniques such as cross-validation and grid search to find the best-performing model.
    - Interpret the model results and assess its performance based on the defined problem statement.
    - Same steps for Regression modelling.(note: dataset contains more noise and linearity between independent variables so itll perform well only with tree based models)
6) Model GUI: Using streamlit module, create interactive page with

    (1) task input(Regression or Classification) and

(2) create an input field where you can enter each column value except ‘Selling_Price’ for regression model and except ‘Status’ for classification model.

(3) perform the same feature engineering, scaling factors, log/any transformation steps which you used for training ml model and predict this new data from streamlit and display the output.

7) Tips: Use pickle module to dump and load models such as encoder(onehot/ label/ str.cat.codes /etc), scaling models(standard scaler), ML models. First fit and then transform in separate line and use transform only for unseen data

```
Eg: scaler = StandardScaler()

scaler.fit(X_train)

scaler.transform(X_train)

scaler.transform(X_test_new) #unseen data
```

The learning outcomes of this project are:

1. Developing proficiency in Python programming language and its data analysis libraries such as Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, and Streamlit.
2. Gaining experience in data preprocessing techniques such as handling missing values, outlier detection, and data normalization to prepare data for machine learning modeling.
3. Understanding and visualizing the data using EDA techniques such as boxplots, histograms, and scatter plots.
4. Learning and applying advanced machine learning techniques such as regression and classification to predict continuous and binary target variables, respectively.
5. Building and optimizing machine learning models using appropriate evaluation metrics and techniques such as cross-validation and grid search.
6. Experience in feature engineering techniques to create new informative representations of the data.
7. Developing a web application using the Streamlit module to showcase the machine learning models and make predictions on new data.
8. Understanding the challenges and best practices in the manufacturing domain and how machine learning can help solve them.

Overall, this project will equip you with practical skills and experience in data analysis, machine learning modeling, and creating interactive web applications, and provide you with a solid foundation to tackle real-world problems in the manufacturing domain.