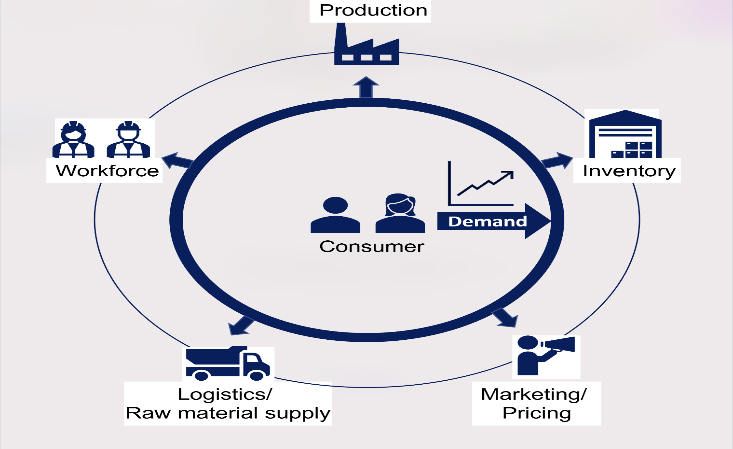
**Project 10: Product Demand Analysis**

**PHASE 1 : DOCUMENT SUBMISSION**

**410721243039 : JANASTHUTHI M S**

**Problem Definition and Design Thinking**

**Problem Definition**: You may want to consider more specific business goals for your model, such as reducing inventory costs or improving customer satisfaction. This will help you to focus your data collection and feature engineering efforts.

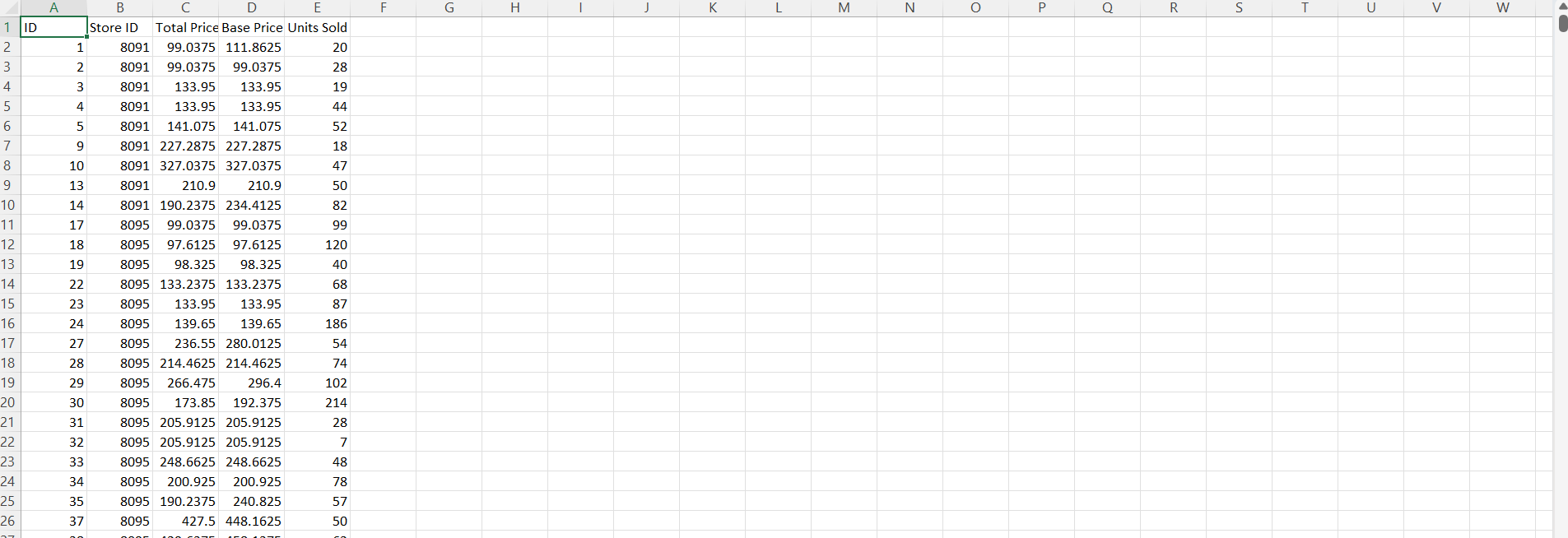


**Design Thinking**: It is important to keep the end user in mind when designing your model. Who will be using the model and how will they be using it? What information do they need from the model? Once you have a good understanding of the user's needs, you can start to think about how to design and implement your model in a way that is both accurate and user-friendly.

**Data Collection and Preprocessing**

**Data Collection**: Be sure to collect a comprehensive dataset that includes all of the relevant features that may influence demand. You may also want to consider collecting data from multiple sources to get a more complete picture of the market.

**Dataset** : [**https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning**](https://www.kaggle.com/datasets/chakradharmattapalli/product-demand-prediction-with-machine-learning)

****

Data Preprocessing: It is important to carefully clean and preprocess your data before training your model. This includes handling missing values, converting categorical features into numerical representations, and scaling your data.

**Feature Engineering**

**Feature Engineering**: Feature engineering is the process of creating new features from existing data. This can be a powerful way to improve the performance of your model. For example, you could create features that capture seasonal patterns, trends, and external influences on product demand.

**Model Selection and Training**

**Model Selection**: There are many different regression algorithms that can be used for demand forecasting. Some popular choices include Linear Regression, Random Forest, and XGBoost. You may want to experiment with different algorithms to see which one performs best on your data.

**Model Training**: Once you have selected a model, you need to train it on your preprocessed data. This involves feeding the data to the model and allowing it to learn the relationships between the features and the target variable (product demand).

**Model Evaluation and Deployment**

**Model Evaluation**: Once your model is trained, you need to evaluate its performance on a held-out test set. This will give you an idea of how well the model will generalize to new data.

**Model Deployment**: Once you are satisfied with the performance of your model, you can deploy it to production. This may involve integrating the model into a software application or making it available as a web service.

**Source Code:**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

# Load the dataset

df = pd.read\_csv('product\_demand\_prediction.csv')

# Split the data into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df.drop('demand', axis=1), df['demand'], test\_size=0.25, random\_state=42)

# Create a Linear Regression model

model = LinearRegression()

# Train the model on the training data

model.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = model.predict(X\_test)

# Calculate the mean absolute error

mae = mean\_absolute\_error(y\_test, y\_pred)

# Print the MAE

print('MAE:', mae)

**Conclusion:**

This code will train a Linear Regression model to forecast product demand based on the historical sales data and external factors in the dataset. You can experiment with different regression algorithms and hyperparameters to see which one performs best on your data.