FLOW PRESENTATION

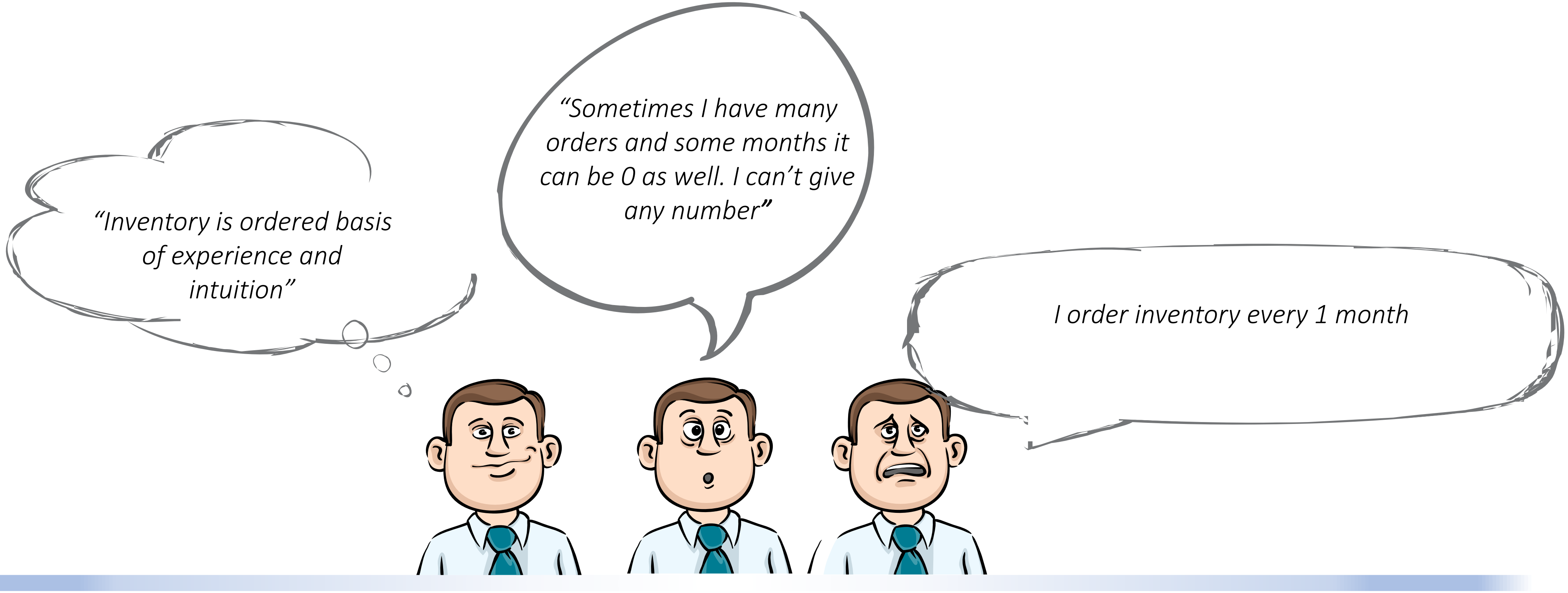
- CUSTOMER PAIN POINTS/GAPS FOR DIFFERENT BUSINESS TYPES/PERSONAS & SOLUTIONS ENVISAGED
- WHY DID WE CHOOSE THE MODELS THAT WE CHOSE?
- AI/ML MODELLING
  - DATA INTRODUCTION

    → ***DEMO: Inputs to the model***
  - DATA PREPARTION
  - MODEL EVALUATION & PREDICTION
  - INVENTORY REPLENISHMENT – REORDER POINT & SAFETY STOCK

    → ***DEMO: Output***
- BENEFITS – CUSTOMER & TBH

VOICE OF CUSTOMERS

Inventory is ordered using ***experience and intuition****.* ***Overstocking*** *seems to be a common trait. Use* ***time-based inventory replenishment*** *methodology*

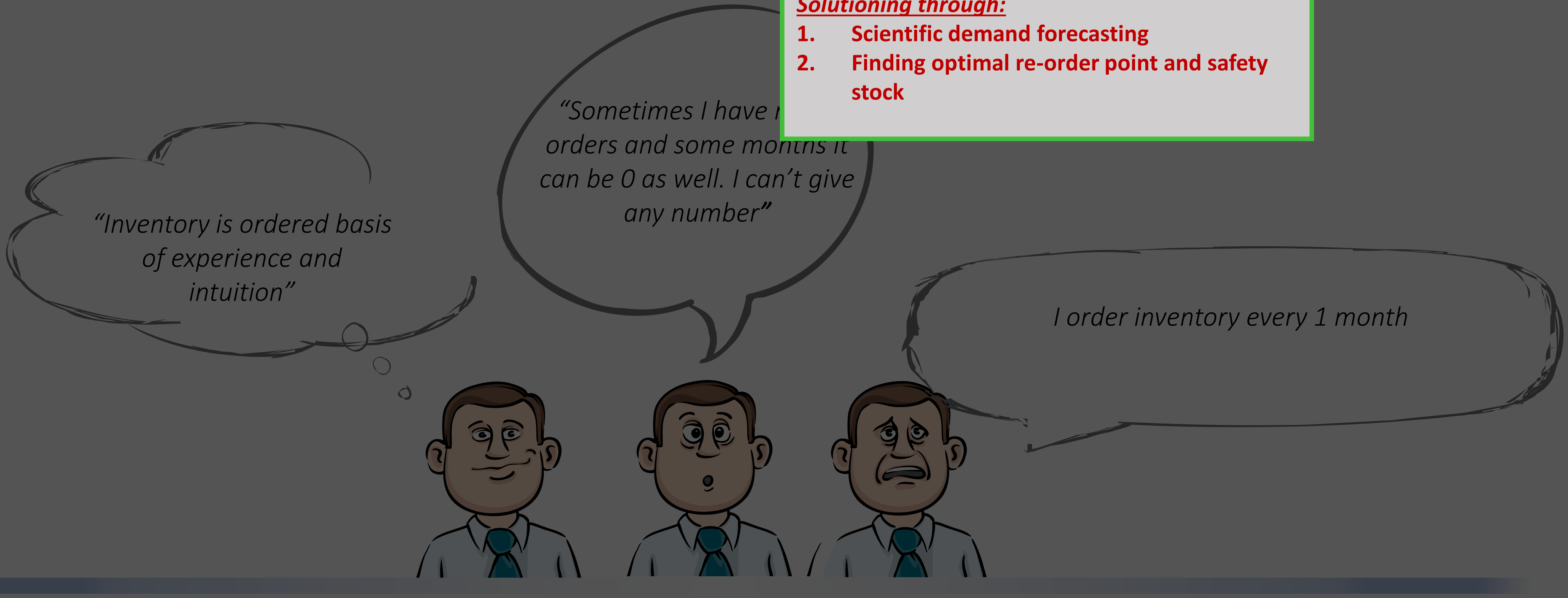


VOICE OF CUSTOMERS

*Inventory is ordered using **experience and intuition**. **Overstocking** seems to be a common trait. Use **time-based inventory replenishment** methodology*

Solutioning through:

1. **Scientific demand forecasting**
2. **Finding optimal re-order point and safety stock**



REPLENISHMENT OF INVENTORY

TIME BASED

INVENTORY BASED

INVENTORY BASED REPLENISHMENT is better because

- Better Adaptation to Demand Variability
- Reduced Risk of Stockouts
- Optimized Inventory Holding Costs
- Improved Supply Chain Responsiveness
- Customer Service Improvements

with limitations of

- Requires accurate demand forecast → *Addressed through our forecasting model*
- Requires regular monitoring of inventory → *Classify inventory and keep count in their system which they currently do*

DEMAND FORECASTING

DEMAND FORECASTING TECHNIQUES

AI/ML MODELS	RANDOM FOREST RECURENT NEURAL NETWORK	• Can handle non-linear and complex relationships and non-stationarity of data • Can handle trend and seasonality with feature engineering • Handles both sequential and features	SELECTED
TIME SERIES	SARIMAX	• Assumes data is stationary • Assumes linear relationship b/w variables	
	ARIMA/HOLT WINTER	• Doesn't account for external variables	
CAUSAL & SCENARIO	REGRESSION	• Only considers external variable and not trend and seasonality of data itself	
	SCENARIO ANALYSIS	• Only considers strategic scenarios and not trend and seasonality of data itself	
QUALITATIVE	DELPHI MARKET RESEARCH	• Done when data is not available	

DATASET STRUCTURE

	record_ID	week	store_id	sku_id	total_price	base_price	is_featured_sku	is_display_sku	units_sold
149381	211540	2013-07-09	9112	219844	244.3875	361.9500	0	1	41
149384	211543	2013-07-09	9112	222087	163.1625	208.0500	1	1	664
149389	211552	2013-07-09	9112	245387	355.5375	469.5375	1	0	22
149372	211525	2013-07-09	9092	245338	356.2500	467.4000	1	1	33
149363	211516	2013-07-09	9092	219029	278.5875	309.9375	0	0	31

Input Variables

- Week
- SKU ID
- Store ID

External/Exogenous Variables

- Total Price
- Base Price
- Is featured SKU
- Is display SKU

Target Variable

- Units Sold

Any external variable GDP, population can be added like mentioned in the use-case and the model will still work

Data Source: Kaggle

DATA PREPARATION

DATA CLEANING | DATA UNDERSTANDING | DATA PROCESSING

1. TRANSFORMED DATA TYPE
2. TREATED MISSING VALUES
3. REMOVED DUPLICATE ENTRIES
4. TREATED INCONSISTENT VALUES
5. SORTED DATA BASED ON DATE

If missing values > 1% then removed the rows else replaced with mean

```
# Calculate the total number of rows
total_rows = len(df)
# Calculate the number of rows with missing values
na_rows = df.isna().any(axis=1).sum()
if na_rows < total_rows * 0.01:
    df.dropna(inplace=True)
else:
    # Fill missing values with the average of store_id and sku_id combination
    df.fillna(df.groupby(['store_id', 'sku_id']).transform('mean'), inplace=True)
df.isnull().sum()
```

DATA PREPARATION

DATA CLEANING | DATA UNDERSTANDING | DATA PROCESSING

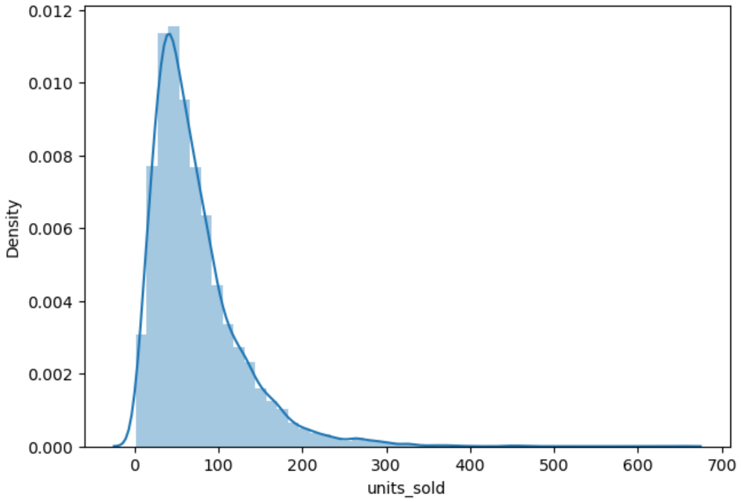
1 DATA SELECTION
Based on sku and store input

sku_id = 216419
store_id = 8091

2 UNDERSTANDING DISTRIBUTION

3 UNDERSTANDING COMPONENTS and DATA STATIONARITY USING DFT

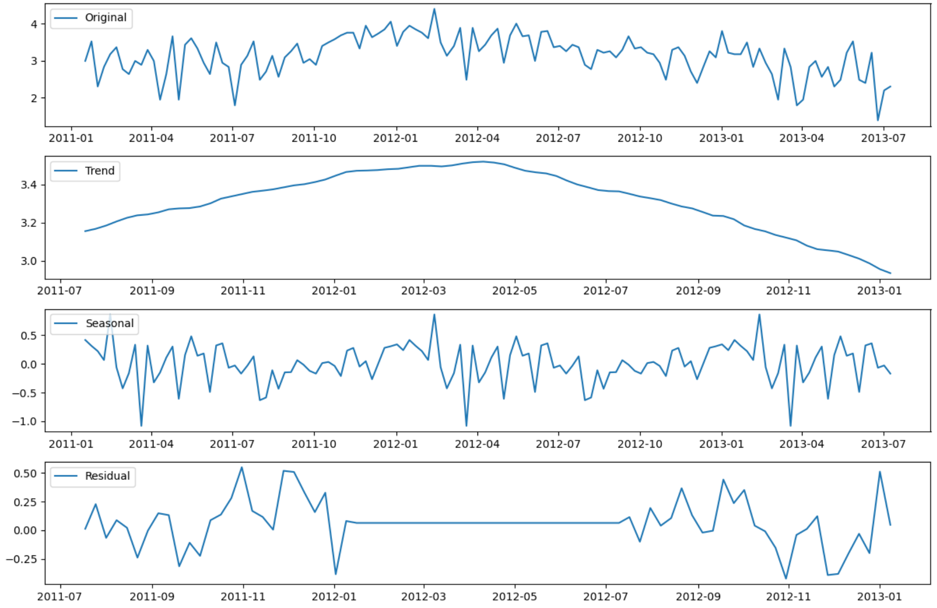
2



SKEWNESS: 1.7979

Observation: Data is highly skewed towards right

3



Observation: Data has both trend and seasonality

DATA PREPARATION

DATA CLEANING | DATA UNDERSTANDING | **DATA PROCESSING**

1
LOG TRANSFORMATION
Normalization for better predictions by treating skewness

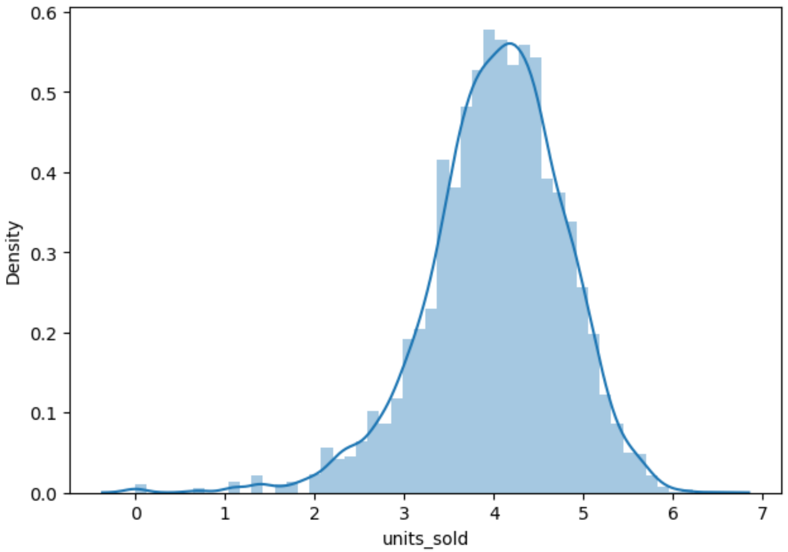
2
REMOVED OUTLIERS
μ +/- 3σ

3
INTRODUCED FEATURES
Accounting for seasonality in data and discount as another variable

4
DIVIDED INTO TRAIN & TEST
80% - train (4680 weeks);*
20% - test (1170 weeks)*

**For the selected sku and store*

1



SKEWNESS: -0.4571

Transformation: Normalized the curve

3

display_sku	units_sold	month	year	day_of_week	day_of_month	disc
0	162	7	2013	1	9	
0	140	7	2013	1	9	
0	74	7	2013	1	9	
0	45	7	2013	1	9	
0	63	7	2013	1	9	

Features introduced: month, year, day_of_week, day_of_month, discounts

MODELLING & PREDICTIONS

SELECTION OF MODEL | PREDICTION

Evaluated the models using RMSE (root mean squared error) and MAPE (Mean absolute percentage error) and selected the model with lower RMSE

RMSE: Avg. magnitude of errors

- **Sensitive to large errors**, therefore, useful for identifying outliers in the forecast.

Interpretation: A lower RMSE indicates that the model's predictions are closer to the actual values

Accuracy = 1 – MAPE: Percentage error

- Indicates accuracy in terms of %
- Asymmetry in Handling overestimation and underestimation
- Insensitive to error magnitude

Interpretation: A higher Accuracy indicates that the model's predictions are closer to the actual values

Evaluation Metrics

Param	Random Forest	Recursive Neural Network
RMSE	0.47	0.6
Accuracy	91.56%	88.89%

SELECTED*

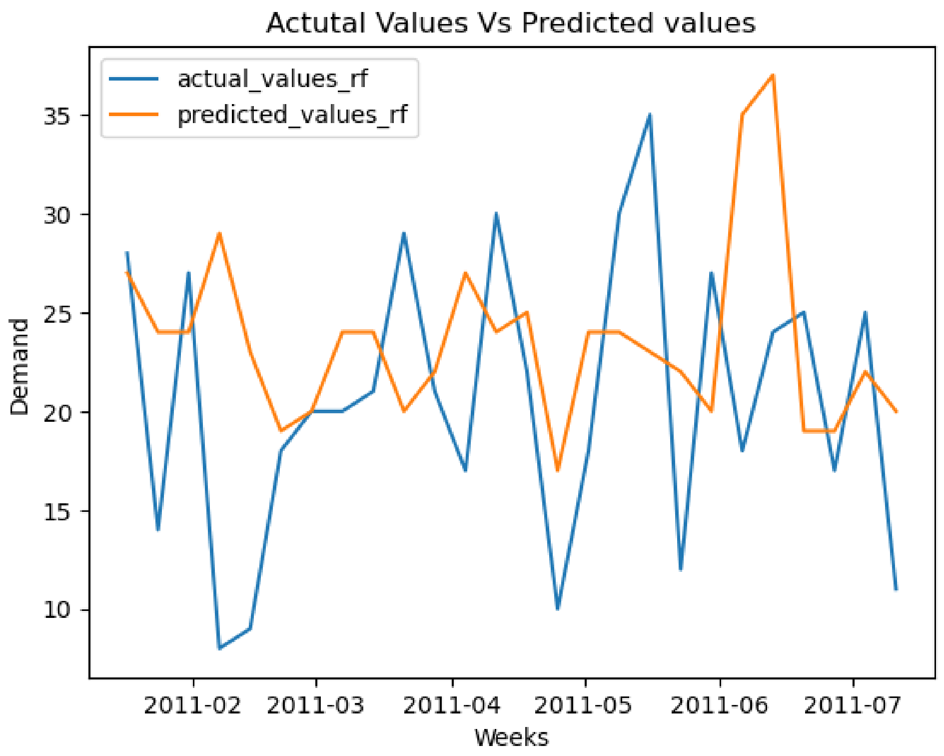
*For the selected sku and store

MODELLING & PREDICTIONS

SELECTION OF MODEL

PREDICTION

	week	store_id	sku_id	actual_values_rf	predicted_values_rf
	2011-07-11	8091	216419	11.0	20.0
	2011-07-04	8091	216419	25.0	22.0
	2011-06-27	8091	216419	17.0	19.0
	2011-06-20	8091	216419	25.0	19.0
	2011-06-13	8091	216419	24.0	37.0



6 most important features important to predicting demand for the sku and store:

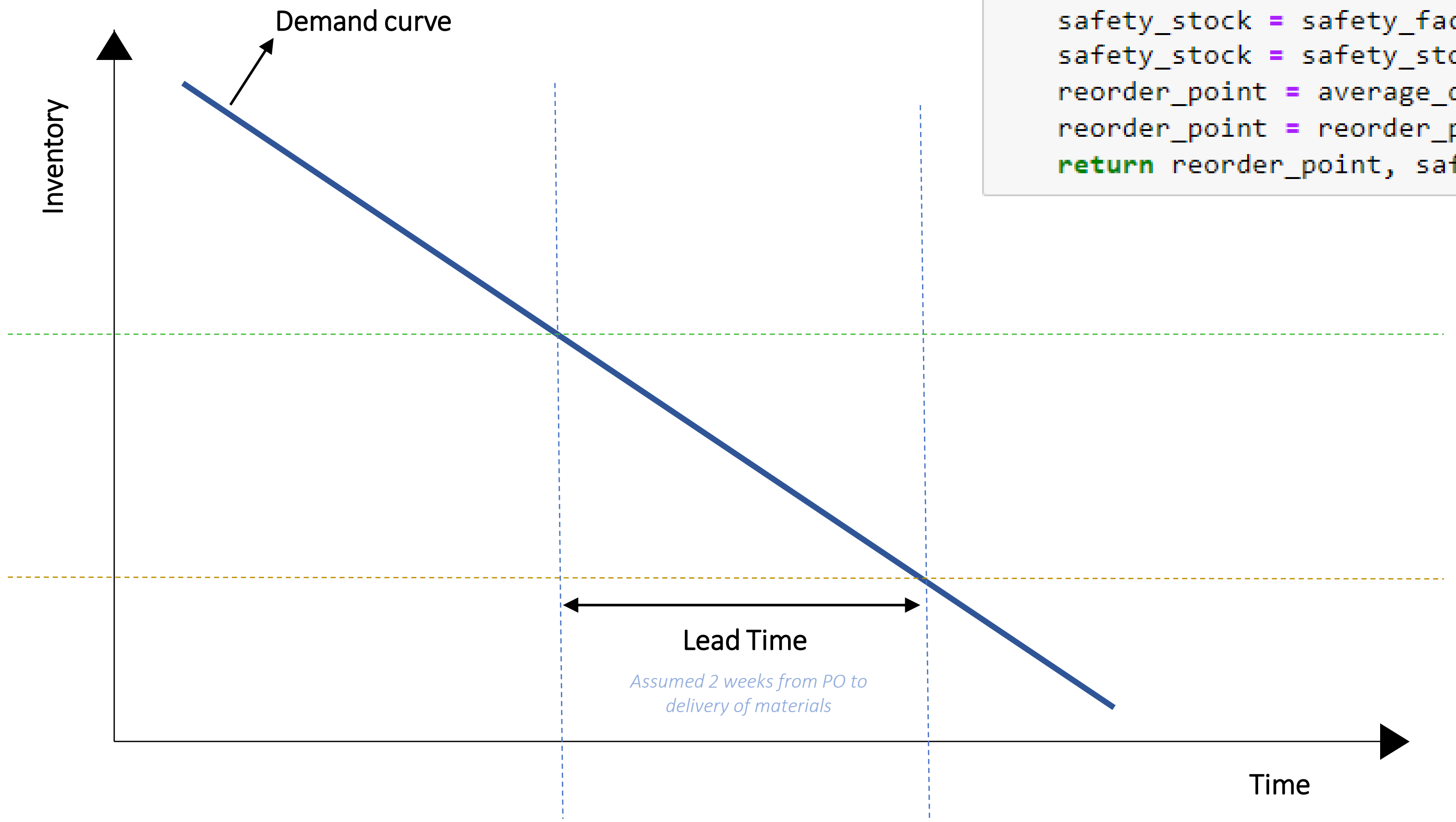
Selected Features: Index(['store_id', 'total_price', 'base_price', 'is_display_sku', 'month', 'day_of_month']

for the chosen sku and store

FINDING RE-ORDER POINT

Based on demand forecast

```
# Calculation of re-order point
def calculate_reorder_point (demand_forecast, lead_time, safety_factor):
    average_demand = np.mean(demand_forecast)
    demand_std = np.std(demand_forecast)
    safety_stock = safety_factor * demand_std
    safety_stock = safety_stock.round()
    reorder_point = average_demand * lead_time + safety_stock
    reorder_point = reorder_point.round()
    return reorder_point, safety_stock
```



Re-order point

Reorder point is the inventory level at which a new order should be placed to replenish stock before it runs out.

51 for the chosen sku and store

Safety stock

Safety stock acts as a buffer against variability in demand and lead time.

4 for the chosen sku and store

Sku_id - 216419
Reorder Point: 182.0
Safety Stock: 48.0

BENEFITS TO CUSTOMER

01

Identify Demand Patterns & Dependence on External Factors

Better adaptation to demand variability therefore improved production planning.

02

Reduce inventory holding cost

Reduce costs associated with storage, maintenance, and obsolescence

03

Improved cash flow

Maintain optimal inventory levels based on actual demand; better inventory turnover

04

Reduce stock outs

Setting optimal reorder point and maintaining safety stock ensures that businesses maintain adequate inventory levels