Week 1 Report

Sales Forecasting Based on Past Sales Statistics

Nguyen Minh Duc - 20225437

Week 1 Objectives:

- Inspect dataset (schema, datatypes, missing values).
- Theoretical review (time-series basics, ARIMA/Prophet, LSTM).
- Data preprocessing (datetime parsing, missing values, outliers).

1 Dataset Inspection

- The dataset contains a total of 17,049 entries and 18 features.
- There are no missing values and no duplicate entries.
- The dataset includes 5,000 unique customers, each having up to 10 recorded orders.

A summary of the dataset schema is provided in Table 1.

Column Name	**Data Type**	**Description**	**Example**
Order_ID	String	Unique order identifier	ORD_001337
Customer_ID	String	Unique customer identifier	CUST_01337
Date	DateTime	Transaction date	2023-06-15
Age	Integer	Customer age	35
Gender	String	Customer gender	Female
City	String	Customer city	Istanbul
Product_Category	String	Product category	Electronics
Unit_Price	Float	Price per unit	1299.99
Quantity	Integer	Units purchased	2
Discount_Amount	Float	Discount applied	129.99
Total_Amount	Float	Final amount paid	2469.99
Payment_Method	String	Payment method	Credit Card
Device_Type	String	Device used for purchase session	Mobile
Session_Duration_Minutes	Integer	Session duration (minutes)	15
Pages_Viewed	Integer	Number of pages viewed	8
Is_Returning_Customer	Boolean	Whether the customer has purchased before	True
Delivery_Time_Days	Integer	Delivery duration in days	3
Customer_Rating	Integer	Customer satisfaction rating	5

2 Theoretical Review

2.1 Time-series basics

Definition and notation. A *time series* is a sequence of observations $\{y_t\}_{t\in T}$ indexed over time t. Time-series analysis aims to describe the stochastic properties of y_t to produce forecasts $\hat{y}_{t+h}$ for $h \geq 1$.

Typical components.

- **Trend** T_t: a long-term change in the level of the series.
- **Seasonality** S_t: systematic periodic fluctuations with known period s (e.g., weekly, yearly).
- **Cycle** C_t: non-periodic oscillations associated with economic/business cycles.
- **Irregular / Noise** ε_t: unpredictable variation.

Two common decompositions are additive and multiplicative:

$$y_t = T_t + S_t + C_t + \varepsilon_t \quad (additive), \qquad y_t = T_t \times S_t \times C_t \times \varepsilon_t \quad (multiplicative).$$

Stationarity. A stochastic process $\{y_t\}$ is *weakly (second-order) stationary* if its mean $E[y_t] = \mu$ is constant over time and its autocovariance $\gamma(h) = Cov(y_t, y_{t+h})$ depends only on lag h, not on t. In practice, non-stationary series are often rendered stationary by differencing or detrending:

$$\Delta y_t = y_t - y_{t-1}$$

for integer $t \geq 1$.

Autocovariance and autocorrelation. The autocovariance at lag h is

$$\gamma(h) = Cov(y_t, y_{t+h}) = E[(y_t - \mu)(y_{t+h} - \mu)],$$

and the autocorrelation function (ACF) is

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \quad \rho(0) = 1.$$

The partial autocorrelation function (PACF) at lag h measures the correlation between y_t and y_{t+h} after removing linear effects of intermediate lags $1, \ldots, h-1$. ACF and PACF plots are routine diagnostic tools for identifying model structure.

2.2 ARIMA and Prophet

ARIMA family (statistical time-series models). The ARIMA(p, d, q) family models a univariate time series by combining autoregressive (AR) and moving-average (MA) terms with differencing of order d to remove non-stationarity. Let Δ^d denote dth difference. The ARMA(p, q) model for the stationary series $z_t = \Delta^d y_t$ is

$$z_t = \phi_1 z_{t-1} + \ldots + \phi_p z_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q},$$

or, using lag operator L,

$$\Phi(L) z_t = \Theta(L)\, \varepsilon_t, \qquad \Phi(L) = 1 - \phi_1 L - \ldots - \phi_p L^p, \quad \Theta(L) = 1 + \theta_1 L + \ldots + \theta_q L^q.$$

ARIMA(p, d, q) simply means apply Δ^d to y_t and model the result as ARMA(p, q). Seasonal extensions (SARIMA) add seasonal AR/MA terms with period s, commonly written SARIMA$(p, d, q) \times (P, D, Q)_s$.

Advantages and limitations.

- *Advantages:* parsimonious parametric form, well-understood inference, clear diagnostics, good short-horizon forecasts for stationary-like series.
- *Limitations:* cannot easily incorporate many external regressors or complex nonlinear relationships; seasonal extensions require manual identification of seasonal periods; performance degrades when complex seasonality patterns exist.

Prophet (additive regression with automatic seasonality). Prophet is a practical forecasting procedure that models a time series as a sum of interpretable components:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t,$$

where $g(t)$ is a piecewise-linear (or logistic) trend with automatic *changepoints*, $s(t)$ is a flexible seasonality term often modelled via truncated Fourier series, and $h(t)$ captures user-specified holidays or events. Seasonality with period s is represented using Fourier terms:

$$s(t) = \sum_{k=1}^{K} \left[a_k \cos(2\pi k t s) + b_k \sin(2\pi k t s) \right].$$

Prophet fits the model by regularized (MAP) estimation, placing priors on trend changepoints and seasonal coefficients to avoid overfitting.

Practical behavior and use-cases.

- Prophet is designed for business time series with multiple seasonalities, holidays, and missing data; it requires minimal manual tuning and provides interpretable components (trend, seasonality, holidays).
- It handles irregularly spaced observations, automatically detects trend changepoints (with a tunable prior scale), and allows inclusion of external regressors.
- Limitations: being additive and regression-based, Prophet may underperform on series with strong nonlinearity not captured by its components, or when careful statistical inference is required.

2.3 Long Short-Term Memory (LSTM) Networks

Overview. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data, addressing the vanishing gradient problem common in vanilla RNNs. They are widely used for time-series forecasting when nonlinear patterns, complex seasonality, or long-term dependencies exist.

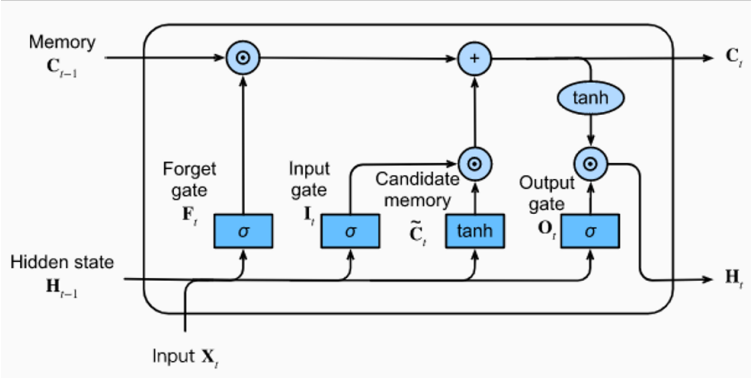

Figure 1: Structure of an LSTM cell.

LSTM cell structure. An LSTM cell maintains a *cell state* C_t that acts as memory, and a hidden state h_t that is used for output and passing information to the next time step. Each cell contains three main gates:

- **Forget gate** f_t: decides what information to discard from the cell state.
- **Input gate** i_t: decides which new information to add to the cell state.
- **Output gate** o_t: decides what part of the cell state to output as the hidden state.

The cell updates follow the equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C),$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t,$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o),$$

$$h_t = o_t * \tanh(C_t),$$

where x_t is the input at time t, σ is the sigmoid activation, and $*$ denotes element-wise multiplication.

Advantages for time-series forecasting.

- Capable of learning long-term dependencies.
- Handles nonlinear relationships between past observations and future values.
- Robust to sequences with varying lengths and irregular patterns.

3 Data Preprocessing

3.1 Date Parsing and Feature Extraction

The dataset contains a `Date` column recorded in the `YYYY-MM-DD` format. This field was converted into a `datetime` object, after which several temporal components were extracted to support time-series modeling and exploratory analysis:

- Day of month
- Month
- Year
- Weekday (0 = Monday, 6 = Sunday)
- Week of year

Extracting these components enables the identification of seasonality, weekly purchasing patterns, and monthly fluctuations in customer behavior, while also improving the interpretability of temporal dynamics.

3.2 Handling Missing Values

As mentioned above, the dataset contains no missing/null values or duplicate entries. Therefore, no imputation or deletion procedures were required.

3.3 Outlier Detection Using the IQR Method

Outlier detection was conducted using the Interquartile Range (IQR) method. For each numerical variable, values lying outside the interval $[Q1 - 1.5 \cdot IQR,\ Q3 + 1.5 \cdot IQR]$ were classified as outliers. Table 1 summarizes the number of detected outliers, alongside notes on the distributional characteristics of each variable.

Variable	**Outliers**	**Distribution Notes**
Unit_Price	1757	Extremely right-skewed
Quantity	0	No outliers
Discount_Amount	2789	Extremely right-skewed
Total_Amount	1943	Extremely right-skewed
Session_Duration_Minutes	85	Approximately normal
Pages_Viewed	1	Approximately normal
Delivery_Time_Days	475	Slightly right-skewed
Customer_Rating	0	No outliers

Table 1: Outlier counts for numerical variables using the IQR method.

Interpretation

Monetary Variables (Unit_Price, Discount_Amount, Total_Amount). These variables exhibit a large number of outliers due to their highly right-skewed nature. In e-commerce settings, it is common for a portion of transactions to involve expensive items, large discounts, or unusually high final amounts. Thus, the detected outliers represent genuine business behavior rather than data quality issues.

Behavioral Metrics (Session_Duration_Minutes, Pages_Viewed). These variables follow near-normal distributions, consistent with typical user browsing behavior. The small number of outliers likely corresponds to unusually long browsing sessions or atypical user interactions.

Delivery_Time_Days. A moderate number of outliers reflects natural logistical variation. The slight right skew suggests delivery delays occur more frequently than unusually fast deliveries.

No-Outlier Variables (Quantity, Customer_Rating). Quantity displays controlled variation, as customers rarely purchase excessively large quantities per order. Customer rating is bounded by design (e.g., a fixed 1–5 scale), making outliers impossible.