FUTURE SALES PREDICTION

PHASE I : PROBLEM DEFINITION AND DESIGN THINKING

PROBLEM DEFINITION

- The problem is to develop a predictive model that uses historical sales data to forecast future sales for retail company. The objective is to create a tool that enables the company to optimize Inventory management and make informed business decisions based on data driven sales prediction. This project involves data pre-processing, feature engineering, model selection, training and evaluation.

DESIGN THINKING : DATA SOURCE

- Historical Sales Data: This is essential. It includes past sales records, preferably at a granular level (daily, weekly, or monthly), product details, prices, and any relevant promotions or discounts
- Market and Economic Data: Factors like inflation rates, GDP growth, and consumer sentiment can impact sales. You can obtain this data from government agencies, financial institutions, or economic research organizations.
- Weather Data: Weather conditions can influence sales, especially for certain industries like retail (e.g., ice cream sales in summer, winter clothing in winter). APIs or datasets from meteorological organizations can provide this data.
- Demographic Data: Information about your target customers, such as age, gender, location, and income levels, can be valuable. Census data or surveys are common sources.

DATA PREPROCESSING

- Data Cleaning : Remove duplicate records, if any..
- Handle missing data: Decide whether to impute missing values or remove rows/columns with missing data, depending on the significance of the missing data. Check for outliers and decide how to handle them. Outliers can significantly affect predictions
- Data Transformation: Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding. Scale numerical features, such as using Min-Max scaling or standardization (z-score normalization), to ensure they have similar scales. Log-transform skewed data if necessary, especially for variables with highly skewed distributions.
- Feature Engineering: Create new relevant features that can enhance the predictive power of your model. For example, you can derive features like day-of-week, month, or quarter from date data. Consider lag features, which involve incorporating past sales data as predictors for future sales.
- Data Splitting: Split your dataset into training, validation, and test sets. The training set is used to train your model, the validation set helps tune hyperparameters, and the test set is used to evaluate the model's performance.

FEATURE ENGINEERING

- Historical Sales Data: Incorporate historical sales data, including date, sales volume, and revenue. Aggregations like monthly or yearly totals can help identify trends.
- Time-based Features: Create features like day of the week, month, quarter, or year to capture seasonality. Calculate moving averages or rolling statistics to smooth out noise and detect trends.
- Lagged Features: Include lagged sales data (e.g., sales from the previous month or year) as predictors to account for autocorrelation.
- Calendar Events: Incorporate information about holidays, special events, or promotions that could impact sales.
- Product Features: Extract attributes of the products, such as category, brand, price, and popularity. Calculate product-specific statistics like the average sales of a product over time.
- Store or Location Features: Include information about store locations, such as size, location type, and demographics of the area. Incorporate historical store-specific sales data.

MODEL SELECTION

- Understand the Problem: Clearly define your problem, including the type of sales data you have (e.g., time series, customer data), the time horizon you're predicting (e.g., daily, monthly sales), and the specific objectives of your prediction (e.g., forecasting overall sales or individual product sales).
- Data Preprocessing: Prepare your data by handling missing values, outliers, and encoding categorical variables. For time series data, consider smoothing techniques, detrending, and seasonality adjustments.
- Feature Engineering: Create relevant features that can help the model capture patterns in the data. This might involve lag features, rolling statistics, or domain-specific variables.
- Select a Subset of Models: Start with a few candidate models that are suitable for regression or time series forecasting tasks. Common choices include Linear Regression, Decision Trees, Random Forests, Gradient Boosting, ARIMA, and LSTM/GRU for deep learning-based approaches.
- Model Evaluation: Use appropriate evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to compare the performance of your candidate models. Also, consider business-specific metrics if available.

MODEL TRAINING

- Model Selection: Decide on the type of model to use. Common choices for time-series sales prediction include linear regression, decision trees, random forests, ARIMA, and more advanced models like LSTM or XGBoost.
- Data Splitting: Split the data into training and testing sets to evaluate the model's performance. Time-series data should be split chronologically to mimic real-world scenarios.
- Model Training: Train the chosen model on the training data. Tune hyperparameters to optimize model performance. You may also consider ensemble methods or deep learning techniques for improved accuracy.
- Validation and Testing: Validate the model's performance on the testing dataset to ensure it generalizes well to unseen data.
- Monitoring and Maintenance: Continuously monitor the model's performance in production. Retrain the model periodically with new data to ensure it stays accurate over time.

EVALUATION

- Model Selection: Select appropriate machine learning algorithms for regression or time series forecasting. Common choices include linear regression, decision trees, random forests, or deep learning models like LSTM for time series data.
- Training and Validation: Split your data into training and validation sets to train and assess the model's performance. Cross-validation can help in estimating how well your model might generalize to unseen data.
- Performance Metrics: Use appropriate metrics for evaluation, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or R-squared (R2) for regression tasks. These metrics quantify how well your model predicts future sales.
- Model Interpretability: Depending on the complexity of your model, consider using techniques like feature importance analysis or SHAP (Shapley Additive exPlanations) values to interpret how different features influence predictions.
- Backtesting: For time series forecasting, conduct backtesting by comparing predicted values to actual sales data in the past. This can provide insights into the model's accuracy in forecasting historical data.