FUTURE SALES PREDICTION

PHASE 4 SUBMISSION DOCUMENT.

PHASE 4 DEVELOPMENT PART 2



Future sales prediction about Modern engineering :

Predicting future sales for a modern engineering company with program code involves creating a predictive model. Here's a brief outline of the steps involved:

Data Collection:

Gather historical sales data and other relevant information, such as market trends, economic indicators, and customer demographics.

Data Preprocessing:

Clean and prepare the data by handling missing values, outliers, and transforming it into a suitable format for analysis.

Feature Engineering:

Extract or create relevant features from the data, such as seasonality, product attributes, and marketing campaign data.

Model Selection: Choose an appropriate machine learning or statistical model for sales prediction. Common choices include linear regression, time series forecasting, or machine learning algorithms like Random Forest or XGBoost.

Training the Model: Use historical data to train the selected model. The code for this step typically involves splitting the data into training and testing sets, fitting the model to the training data, and evaluating its performance.

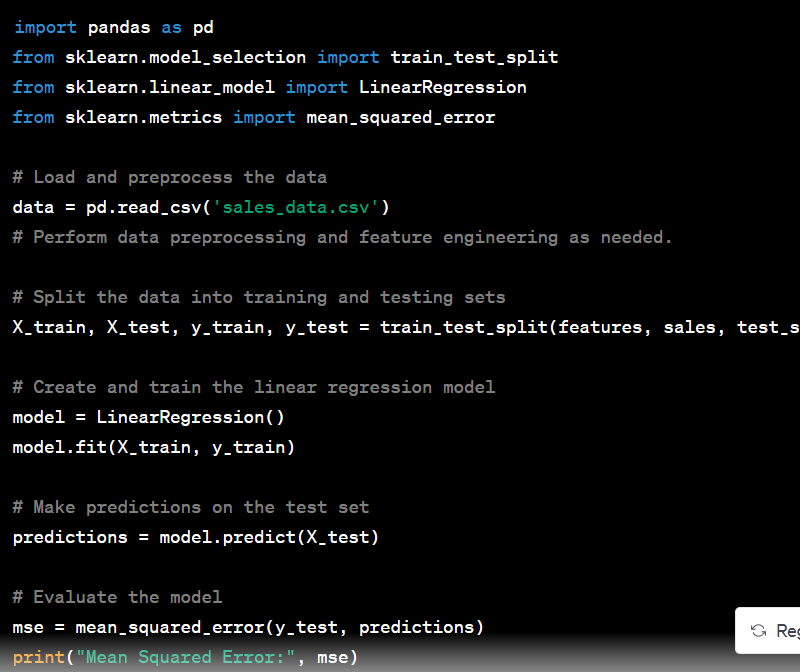
Hyperparameter Tuning: Optimize the model's hyperparameters to improve accuracy.

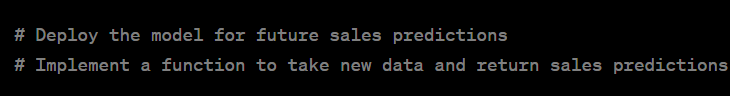
Deployment: Once the model performs well on the test data, deploy it for real-time predictions. This may involve integrating it into a web application or other software systems.

Continuous Monitoring: Continuously monitor the model's performance and update it as new sales data becomes available.

Here's a simplified example of Python code for a basic linear regression model:

Python code:

]

 simplified example. In practice, more sophisticated models and extensive data preprocessing may be required to make accurate sales predictions for a modern engineering company.

Model Training:

This is a simplified example. In practice, more sophisticated models and extensive data preprocessing may be required to make accurate sales predictions for a modern engineering company.

Predicting future sales involves using machine learning models to forecast future sales figures based on historical data and relevant features. Here's a brief overview of the process with examples:

Data Collection:

Gather historical sales data, including information such as date, product, location, marketing spend, and economic indicators.

Data Preprocessing:

Clean and prepare the data by handling missing values, outliers, and encoding categorical variables.

Create features like time of year, promotions, and external factors that can influence sales.

Model Selection:

Choose a suitable machine learning model for sales prediction, such as linear regression, time series models, decision trees, or neural networks.

Split Data:

Split the dataset into training and testing sets to evaluate the model's performance.

Model Training:

Train the selected model on the training data, optimizing parameters and hyperparameters for better performance.

Example: Using a simple linear regression model to predict future daily sales based on historical data, with features like advertising expenditure and day of the week.

Python code :

from sklearn.linear\_model import LinearRegression

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

from sklearn.metrics import mean\_squared\_error

# Assuming 'X' contains features (e.g., advertising spend and day of the week) and 'y' contains daily sales data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

model = LinearRegression()

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

# Predict future sales

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

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

Model Evaluation:

Assess the model's performance using evaluation metrics (e.g., Mean Squared Error, R-squared) on the test dataset.

Deployment:

Once you're satisfied with the model's performance, deploy it to make real-time predictions or automate future sales forecasts.

Continuous Monitoring and Updating:

Periodically retrain and fine-tune the model to adapt to changing trends and data patterns.

The choice of model and features can vary based on the specific business case. More advanced methods like time series forecasting using ARIMA, Prophet, or deep learning techniques can be employed for more accurate predictions when historical sales data has strong time dependencies.

Evaluation in future sales prediction:

Data Collection and Preprocessing:

Example: Imagine you have a dataset of monthly sales for an online clothing store over the past five years. Your data should include columns like date, product categories, and total sales.

Exploratory Data Analysis (EDA):

Example: Create line plots or bar charts to visualize the monthly sales trends for different product categories. You may identify a seasonal pattern where sales of winter clothes peak in November-December.

Feature Engineering:

Example: Create lag features to capture the impact of previous months' sales on the current month's sales. For instance, you can create a "Sales\_Lag\_1" column to represent sales from the previous month.

Model Selection:

Example: Choose a model such as an ARIMA model for time series data. Fit the model to your historical sales data and generate predictions for future months.

Train and Validate the Model:

Example:

Split your data into a training set (80%) and a validation set (20%). Train the ARIMA model on the training data and validate it using the validation set.

Evaluation Metrics:

Example:

Calculate evaluation metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess how well your predictions match the actual sales.

Time Series Specifics:

Example:

If you're using an ARIMA model, ensure that you've considered the seasonality and autocorrelation in your time series data.

External Factors:

Example: If your company ran marketing campaigns during certain months, include a binary feature like "Marketing\_Campaign" to account for this in your model.

Model Interpretability:

Example:

Analyze the ARIMA model's coefficients and residuals to understand how past sales and external factors influence future sales predictions.

Monitoring and Feedback Loop:

Example:

Continuously update your model as new data becomes available, retrain it, and assess its performance. If the model's performance deteriorates, it may be time to reevaluate and make adjustments.

Scenario Analysis:

Example: Create scenarios where you simulate the impact of different marketing strategies or external events on future sales. For instance, what if you run a summer sale in July?

Reporting and conclusion:

Example:

Present your sales predictions and insights to company stakeholders in a report or presentation, including visualizations and summaries of the evaluation metrics.

You can use data visualization tools like Matplotlib, Seaborn, or Tableau to create graphs and charts for your data analysis and reports. The specifics of your analysis will depend on your dataset, business goals, and the tools you are using.