Data preprocessing is a crucial step in preparing a dataset for a future sales prediction project. Here are the key steps you should consider:

1. Data Collection:

- Gather relevant data sources, including historical sales data, product information, customer data, and any other pertinent information.

2. Data Cleaning:

- Handle missing values by imputing them or removing rows with missing data.
- Remove duplicates if they exist in the dataset.
- Correct any inconsistent or erroneous data entries.

3. Data Transformation:

- Convert date and time information into a consistent format.
- Encode categorical variables using techniques like one-hot encoding or label encoding.
- Scale or normalize numerical features to ensure they have similar scales.

4. Feature Engineering:

- Create new features that could be relevant for sales prediction, such as seasonality indicators, average order value, or customer segmentation.
- Extract meaningful information from text data, if applicable.

5. Data Split:

- Split the dataset into training, validation, and test sets to evaluate model performance.

6. Time Series Handling (if applicable):

- If your sales data is time-dependent, ensure proper time series handling, including lag features and rolling statistics.

7. Outlier Detection and Handling:

- Identify and address outliers in the dataset that may affect predictions.

8. Data Balancing (if applicable):

- If your dataset is imbalanced, consider techniques like oversampling or undersampling to balance it.

9. Data Visualization:

- Create visualizations to gain insights into the data and understand its distribution.

10. Feature Selection (if needed):

- Choose relevant features for modeling to reduce dimensionality.

11. Data Preprocessing Pipeline:

- Create a data preprocessing pipeline to ensure consistency when preparing new data for predictions.

Remember that the specific steps and techniques may vary depending on your dataset and the machine learning algorithms

you plan to use for sales prediction. It's essential to adapt your preprocessing based on the unique characteristics of your data and project goals.

Program :

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt


# Load your sales data into a Pandas DataFrame
data = pd.read_csv('sales_data.csv')  # Replace 'sales_data.csv' with your data file


# Data preprocessing and feature engineering (customize as needed)
data['Date'] = pd.to_datetime(data['Date'])
data['Month'] = data['Date'].dt.month
data['Day Week'] = data['Date'].dt.dayofweek


# Define features and target variable
X = data[['Month', 'Days Of Week']]
y = data['Sales']


# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Create a linear regression model and train it
model = Linearregression()
model.fit(X_train, y_train)


# Make predictions
y_pred = model.predict(X_test)


# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
```

歩いて行くという森の旅

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f'Mean Absolute Error: {mae}')
print(f'Root Mean Squared Error: {rmse}')


# Visualize the predictions
plt.scatter(X_test['Month'], y_test, label='Actual Sales', color='blue')
plt.scatter(X_test['Month'], y_pred, label='Predicted Sales', color='red')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.legend()
plt.show()


# Make future sales predictions for a given set of features (Month and DayOfWeek)
future_features = np.array([[10, 4]])  # Customize the values
future_sales = model.predict(future_features)
print(f'Predicted Sales for the Future: {future_sales[0]}')
```

Project :

Future sale prediction _phase 3

歩いて行くという森の旅