Future sales prediction

3.1 Dataset Selection and Description

For the "Future Sales Prediction" project, from kaggle dataset as a reference from future sales prediction by ML. The dataset contains crucial information, including historical sales data, product details, and time-related variables, all of which are pivotal in building an accurate prediction model.

* **Dataset Features**: Describe the dataset features, their data types, and their relevance to the project.
* **Data Source**: Provide a clear reference to the dataset source and its integrity.

3.2 Data Loading

we begin by loading the dataset into our Python environment using the Pandas library. The code snippet below demonstrates how to do this:

import pandas as pd

# Load the dataset

dataset = pd.read\_csv('your\_dataset.csv')

# Display the initial data view

print(dataset.head())

3.3 Data Preprocessing

Data preprocessing is a critical step to ensure the dataset is clean, complete, and ready for analysis and modelling. Here are some key preprocessing steps:

* **Handling Missing Values**: Eliminate or impute missing values, if any.
* **Categorical Encoding**: Transform categorical variables into numerical representations (e.g., one-hot encoding).
* **Numerical Scaling**: Normalize or standardize numerical features, if required.

Code:

# Handle missing values (if any)

dataset.dropna(inplace=True)

# Encode categorical variables (if any)

dataset = pd.get\_dummies(dataset, columns=['categorical\_column'])

# Scale numerical features (if necessary)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

dataset['numerical\_column'] = scaler.fit\_transform(dataset['numerical\_column'])

3.4 Data Analysis

To gain insights into the dataset and prepare for future sales prediction, we conduct various analyses:

**Exploratory Data Analysis (EDA)**: This phase involves understanding the data's distribution, trends, and central tendencies. Visualizations such as histograms and box plots can be effective in revealing data characteristics.

import seaborn as sns

import matplotlib.pyplot as plt

# Example: EDA with a histogram

sns.histplot(dataset['sales'])

plt.xlabel('Sales')

plt.ylabel('Frequency')

plt.title('Sales Distribution')

plt.show()

**Time Series Analysis**: If applicable, time series data can be decomposed to identify trends, seasonality, and noise. Autocorrelation and partial autocorrelation plots can help in understanding temporal dependencies.

from statsmodels.tsa.seasonal import seasonal\_decompose

# Decompose time series

result = seasonal\_decompose(dataset['sales'], model='additive')

result.plot()

plt.show()

**Feature Engineering**: Creating new features and lag features can be beneficial in capturing historical information and improving prediction models.

# Example: Create lag features

dataset['sales\_lag\_1'] = dataset['sales'].shift(1)

dataset['sales\_lag\_7'] = dataset['sales'].shift(7)

These phases and code examples lay the foundation for our Future Sales Prediction project. The specific modelling and forecasting steps will follow in subsequent project phases.