

Project 2 : Sales Performance Analysis

1. Load and Explore the DataSets.

```
-> import pandas as pd

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load and Inspect.....

```
df = pd.read_csv('sales_data.csv')
```

```
# Basic info
print(df.shape)
print(df.info())
print(df.describe())
print(df.isnull().sum())
```

2.Data Cleaning.

-> Remove Duplicates.....

```
df = df.drop_duplicates()
```

->Handling Missing Values.....

Fill numerical columns with mean or median

```
df['Sales'].fillna(df['Sales'].mean(), inplace=True)
df['Profit'].fillna(df['Profit'].median(), inplace=True)
df['Discount'].fillna(0, inplace=True) # Assuming 0 discount if missing
```

->convert Data Column.....

```
df['Date'] = pd.to_datetime(df['Date'])
```

3.Exploratory data Analysis (EDA).

->Sales trend Over Time

```

sales_trend = df.groupby('Date')['Sales'].sum()
plt.figure(figsize=(12,6))
sales_trend.plot()
plt.title('Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()

```

->Profit Vs Discount.....

```

plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='Discount', y='Profit', hue='Category')
plt.title('Profit vs. Discount')
plt.show()

```

->Sales by Region.....

```

region_sales = df.groupby('Region')['Sales'].sum()
region_sales.plot(kind='bar', title='Sales by Region', ylabel='Sales')
plt.show()

```

->Sales by category.....

```

category_sales = df.groupby('Category')['Sales'].sum()
category_sales.plot(kind='pie', autopct='%1.1f%%', title='Sales by Category')
plt.ylabel('')
plt.show()

```

4. Predictive modelling.

->Linear Regression Model.....

```
from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split


# Features and target

X = df[['Profit', 'Discount']]

y = df['Sales']


# Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


# Model training

model = LinearRegression()

model.fit(X_train, y_train)


# Model evaluation

print("Model Coefficients:", model.coef_)

print("Intercept:", model.intercept_)

print("R2 Score:", model.score(X_test, y_test))
```

5. Insights and Recommendations...

- **Optimal Discount Range:** If scatter plot shows diminishing profit beyond a certain discount, recommend capping discounts.
- **Top-Performing Regions:** Highlight regions with highest sales and suggest focusing marketing efforts there.

- **Category Performance:** Recommend expanding inventory or promotions for high-performing categories.