PHASE 4 –FUTURE SALES PREDICTION

INTRODUCTION:

In today's dynamic business environment, the ability to forecast future sales accurately is paramount for informed decision-making. This project focuses on creating a robust future sales prediction model, a critical tool for businesses seeking a competitive edge. By harnessing the power of feature engineering, model training, and rigorous evaluation, we aim to equip organizations with a predictive solution that enhances resource allocation, inventory management, and strategic planning.

FEATURE ENGINEERING:

- Create relevant features that can help in sales prediction (e.g., time-related features, historical sales data, customer demographics).
- Scale or normalize numerical features.
- Feature selection if needed (using techniques like feature importance).

MODEL SELECTION:

- Choose a suitable algorithm for your regression task (e.g., Linear Regression, Random Forest, Gradient Boosting).
- Initialize the model and set hyper parameters.

MODEL TRAINING:

Train the selected model on the training data.

MODEL EVALUATION:

- Use appropriate evaluation metrics for regression tasks (e.g., Mean Absolute Error, Root Mean Squared Error, R-squared).
- Evaluate the model's performance on the testing data.

DATASOURCE:

https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction

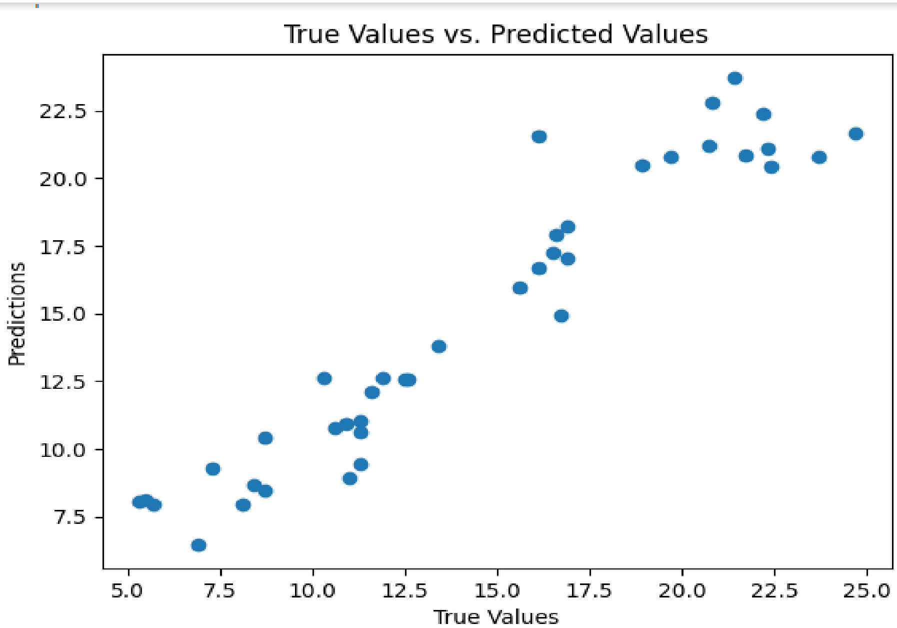
CODE:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
url = "your_dataset_url_here.csv"
df = pd.read_csv(url)

# Explore the dataset
print(df.head())

# Feature Engineering
X = df[['TV', 'Radio', 'Newspaper']]
y = df['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 and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Visualize the results
plt.scatter(y_test, y_pred)
```

```
plt.xlabel("True Values")
plt.ylabel("Predictions")
plt.title("True Values vs. Predicted Values")
plt.show()
```

OUTPUT:

```
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
Mean Squared Error: 2.9077569102710896
R-squared: 0.9059011844150826
```



CONCLUSION:

.In conclusion, building a future sales prediction model involves a series of steps, from data collection and preprocessing to model selection and evaluation. Feature engineering is crucial to capture the underlying patterns in the data, and the choice of the model should be based on the specific characteristics of the sales data. Regularly update and retrain the model to adapt to changing trends and external factors.