```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

def generate_data():
    # Simulated monthly sales data
    months = np.array(range(1, 13)).reshape(-1, 1)  # Month 1 to 12
    sales = np.array([2500, 2600, 2700, 3000, 3200, 3300,
                      3500, 3700, 3900, 4200, 4500, 4700])
    return months, sales

def train_model(months, sales):
    model = LinearRegression()
    model.fit(months, sales)
    return model

def predict_future_sales(model, months):
    future_months = np.array(range(13, 19)).reshape(-1, 1)  # Predict next 6
months
    future_sales = model.predict(future_months)
    return future_months, future_sales

def visualize_sales(months, sales, future_months, future_sales):
    plt.plot(months, sales, marker='o', label='Actual Sales')
    plt.plot(future_months, future_sales, marker='x', linestyle='--',
label='Predicted Sales')
    plt.xlabel('Month')
    plt.ylabel('Sales')
    plt.title('Future Sales Prediction')
    plt.legend()
    plt.grid(True)
    plt.show()

def main():
    print("=== Future Sales Prediction using Linear Regression ===")
    months, sales = generate_data()
    model = train_model(months, sales)

    # Predict and show metrics
    predicted_sales = model.predict(months)
    print(f"R2 Score: {r2_score(sales, predicted_sales):.2f}")
    print(f"MSE: {mean_squared_error(sales, predicted_sales):.2f}")

    future_months, future_sales = predict_future_sales(model, months)
    visualize_sales(months, sales, future_months, future_sales)

if __name__ == "__main__":
    main()
5
```