FUTURE SALES PREDICTION

Applied Data Science

(Phase 5)

1. INTRODUCTION

In the ever-evolving world of business, staying ahead of the competition and making informed decisions is crucial for long-term success. One key aspect of this is predicting future sales accurately. Data science and predictive analytics have emerged as invaluable tools for businesses to gain insights into their sales trends and make data-driven decisions. In this project, we embark on a journey to predict future sales using a dataset that includes information about advertising expenditure on three major mediums: TV, radio, and newspaper, and the corresponding sales data.

1.1 Project Preview

The primary objective of this project is to develop a strong predictive model that can accurately forecast future sales based on advertising investments in tv, radio, and newspaper. by doing so, we aim to empower businesses with actionable insights to optimize their advertising budgets and strategies. this project not only showcases the power of data science but also highlights the significance of understanding the impact of different advertising channels on sales.

1.2 Purpose

The primary purpose of this project is to help businesses optimize their advertising budgets. By accurately predicting future sales based on historical advertising data, companies can allocate their resources more efficiently. This ensures that they invest in the advertising channels that yield the best return on investment (ROI), reducing wastage of marketing budgets.

2. Problem Statement

2.1 Problem Statement Definition

The goal of this project is to develop an accurate sales prediction model to predict future sales for a business. Accurate sales forecasting is crucial for optimizing inventory management, resource allocation, and overall business planning.

3.Design Thinking Process:

Design thinking is a problem-solving approach that encourages creative and user-centric solutions. While it's often associated with product or service design, you can adapt its principles to a data science project like future sales prediction.

* Data Source:
  + - This includes the data of TV, radio, newspaper, and sales. Historical data can help that identify patterns, trends, seasonality, and anomalies in sales revenue generation
    - This data can be used to see where the biggest profit margins lie and strategize accordingly and the loss.
* Data Preprocessing:
  + - Raw data often needs to be cleaned and prepared for analysis. This involves handling missing values, outliers, and converting data into a suitable format for machine learning algorithms. Time-series data may also require special treatment.
    - there is an we can Converting the unformat data to usable data format.
* Feature Engineering:
  + - it is the process of transforming raw data into features that are suitable for machine learning models. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.
* Model Selection:
  + It is the task of selecting *a model from among various candidates on the basis of performance criterion* to choose the best one
  + Choose suitable time series forecasting algorithms for predicting future sales
* Model Training:
  + it is the process in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range
  + it is the process was Train the data set and selected model using the preprocessed data.
* Evaluation:
  + this process to we can evaluate the appropriate output of the prediction.
  + Evaluate the performance of your predictive models using appropriate evaluation metrics, such as Mean Squared Error (MSE) or R-squared. Determine which model provides the most accurate sales predictions
* Empathize:
* Dive deep into the provided dataset of TV, radio, newspaper, and sales. Understand the data's quality, completeness, and any potential biases that might affect the predictions. Consider the historical context of the data.

4.Data Set (Used):

In This project focused on future sales prediction in applied data science, the dataset typically includes information on advertising expenditures across different mediums (TV, radio, newspaper) and corresponding sales figures.

1. TV:

* This column in the dataset represents the amount spent on advertising through television. It typically includes financial figures, such as dollars or the local currency unit, indicating the budget allocated for TV advertising during a specific period (e.g., monthly, quarterly, or annually).

2.Radio:

* This column corresponds to the amount spent on radio advertising. It records the budget allocated for radio ads during the same time periods as the TV data. This column typically includes financial values as well.

3.Newspaper:

* Similar to the TV and radio columns, the newspaper column contains data related to advertising spending on newspaper advertisements. It reflects the budget designated for newspaper advertising during the chosen time intervals.

4.sales:

* This column is the primary outcome variable of interest in the dataset. It records the total sales achieved during the corresponding time periods, aligning with the TV, radio, and newspaper advertising expenditures. Sales are typically measured in the same currency unit as the advertising budget columns.

5.**Development Part 1:**

**5.1** Pre-Processing the Dataset**:**

Data Pre-processing is a critical step in preparing the dataset for future sales prediction in applied data science. Pre-processing involves cleaning, transforming, and organizing the data to ensure that it is suitable for analysis and modelling.

**1. Handling Missing Data:**

* **Check for missing values in the dataset, and decide on an appropriate strategy to address them. You can choose to drop rows with missing values, fill them using statistical measures (e.g., mean, median), or use more advanced imputation methods if applicable.**

**2. Data Cleaning:**

* **Inspect the data for any anomalies or outliers that might be errors. If found, consider either removing or correcting these data points.**

**3. Data Transformation:**

* **Normalize or standardize the data if the scales of the features are significantly different. This ensures that all variables have equal weight in the analysis.**

**4. Encoding Categorical Data:**

* **If there are categorical variables in the dataset (e.g., marketing channel categories), you may need to encode them into numerical values using techniques such as one-hot encoding.**

**5. Feature Engineering:**

* **Create new features if they can provide additional insights. For example, you might calculate the total advertising budget as a sum of TV, radio, and newspaper expenditures. Feature engineering can improve the model's performance.**

**6. Data Splitting:**

* **Split the dataset into a training set and a testing set (or validation set). The training set is used to build and train your predictive model, while the testing set is used to evaluate the model's performance.**

**7. Time Series Considerations:**

* **If the dataset contains a time component (e.g., monthly sales data), you might want to consider time series analysis. This can involve lag features or rolling statistics to capture temporal patterns.**

**8. Exploratory Data Analysis (EDA):**

* **Conduct EDA to visualize relationships between features, identify trends, and gain a better understanding of the data. EDA helps you identify patterns and potential predictive variables.**

**9. Outlier Detection:**

* **Perform outlier detection to identify and handle extreme data points that may adversely affect model training. You can use statistical methods or machine learning algorithms to detect outliers.**

**10. Correlation Analysis:**

* **Calculate correlation coefficients to understand the relationships between variables. This can help you identify which advertising channels are most strongly correlated with sales.**

**11. Splitting the Target Variable:**

* **If your project involves time series analysis, consider splitting the dataset into train and test periods, where you use past data for training and future data for testing.**

**12. Data Scaling and Normalization:**

* **Depending on the modelling techniques you plan to use (e.g., regression, neural networks), you may need to scale or normalize the data to ensure consistent model performance.**

**13. Handling Seasonality and Trends:**

* **If applicable, address seasonality and trends in the data by detrending the time series data.**

**14. Data Visualization:**

**Create visualizations to better understand the relationships between advertising expenditures and sales. Visualization can be a powerful tool for gaining insights.**

**6**. **Development Part 2:**

**6.1** Model Training Process:

The model training process for a project on future sales prediction in applied data science, using the dataset of TV, radio, newspaper, and sales, involves building and training predictive models to estimate future sales based on historical data.

**1. Data Preparation:**

* **Load and pre-process the dataset, following the data pre-processing steps mentioned earlier.**
* **Split the data into a training set and a testing set (typically, an 80/20 or 70/30 split).**
* **Prepare the features (TV, radio, newspaper) and the target variable (sales) for training.**

**2. Model Selection:**

* **Choose appropriate machine learning algorithms or models for regression. In this case, regression models are most suitable for predicting sales.**
* **Common regression models include linear regression, decision trees, random forests, support vector regression, and neural networks.**
* **Consider trying multiple models to determine which one performs best for your specific dataset.**

**3. Feature Selection and Engineering:**

* **Decide which features to include in the model. You can experiment with different combinations of TV, radio, and newspaper advertising expenditures.**
* **Perform feature engineering, if necessary, to create new features that may improve model performance. For example, you can calculate the total advertising budget, add lag features, or incorporate external factors that influence sales.**

**4. Model Training:**

* **Train your selected model(s) on the training dataset. The model learns from historical data to make predictions about future sales.**
* **Depending on the complexity of the model and the size of the dataset, training may take a variable amount of time.**

**5. Hyperparameter Tuning:**

* **Fine-tune the model's hyperparameters to optimize its performance. Use techniques like grid search or random search to identify the best hyperparameters for your chosen algorithm.**
* **Cross-validation can be used to assess the model's generalization performance and avoid overfitting.**

**6. Model Evaluation:**

* **Assess the model's performance using appropriate evaluation metrics for regression tasks. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).**
* **Compare the model's performance on the testing dataset against the baseline and other models you may have tried.**

**7. Interpretability and Insights:**

* **Understand the importance of features in the model. Some models, like linear regression, provide coefficients that indicate the impact of each feature on sales.**
* **Interpret the model's predictions and identify which advertising channels contribute the most to sales.**

**8. Visualization:**

* **Create visualizations to better understand the model's predictions and compare them to the actual sales figures.**

**6.2** Evaluation Metrics:

When evaluating the performance of a future sales prediction project in applied data science using the dataset of TV, radio, newspaper, and sales, you typically use regression-specific evaluation metrics. The choice of the most appropriate metric depends on the characteristics of the data and the specific goals of the project. (We can use any five metrices)

**1. Mean Absolute Error (MAE):**

* **MAE measures the average absolute difference between the predicted values and the actual values. It gives equal weight to all errors, making it less sensitive to outliers.**
* **Formula: MAE = Σ |Actual - Predicted| / n**

**2. Mean Squared Error (MSE):**

* **MSE calculates the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily and is sensitive to outliers.**
* **Formula: MSE = Σ (Actual - Predicted) ^2 / n**

**3. Root Mean Squared Error (RMSE):**

* **RMSE is the square root of the MSE. It's in the same units as the target variable, making it more interpretable.**
* **Formula: RMSE = √(MSE)**

**4. R-squared (R^2) or Coefficient of Determination:**

* **R-squared measures the proportion of the variance in the target variable that is explained by the model. It ranges from 0 to 1, with 1 indicating a perfect model fit.**
* **Formula: R^2 = 1 - (MSE / Variance of the Target)**

**5. Mean Absolute Percentage Error (MAPE):**

* **MAPE calculates the average percentage difference between predicted and actual values. It is useful when you want to understand the relative magnitude of errors.**
* **Formula: MAPE = Σ (|Actual - Predicted| / Actual) / n**

**6. Adjusted R-squared:**

* **Adjusted R-squared adjusts the R-squared value by the number of predictors in the model. It helps account for the complexity of the model, penalizing the inclusion of irrelevant features.**
* **Formula: Adjusted R^2 = 1 - [(1 - R^2) \* (n - 1) / (n - k - 1)]**
* **n = Number of data points**
* **k = Number of predictors**

**7. Percentage of Variance Explained (PVE):**

* **PVE measures the proportion of variance in the target variable that the model can explain. It is an alternative to R-squared.**
* **Formula: PVE = 1 - (MSE / Variance of the Target)**

**8. Explained Variance Score (EVS):**

* **EVS quantifies the proportion of variance that the model's predictions explain in the target variable.**
* **Formula: EVS = 1 - (Var (Predicted - Actual) / Var (Actual))**

**9. Maximum Error (Max Error):**

* **Max Error calculates the largest absolute difference between any predicted and actual value in the dataset. It can help identify the worst-case prediction error.**
* **Formula: Max Error = max (|Actual - Predicted|)**

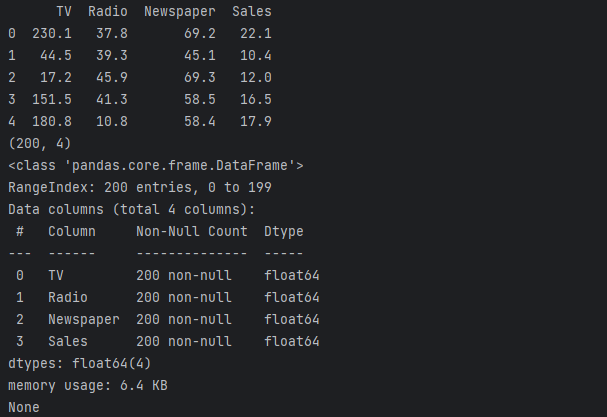
7. program & Output:

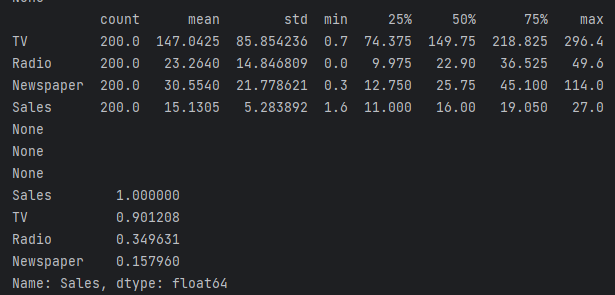
**Program:**

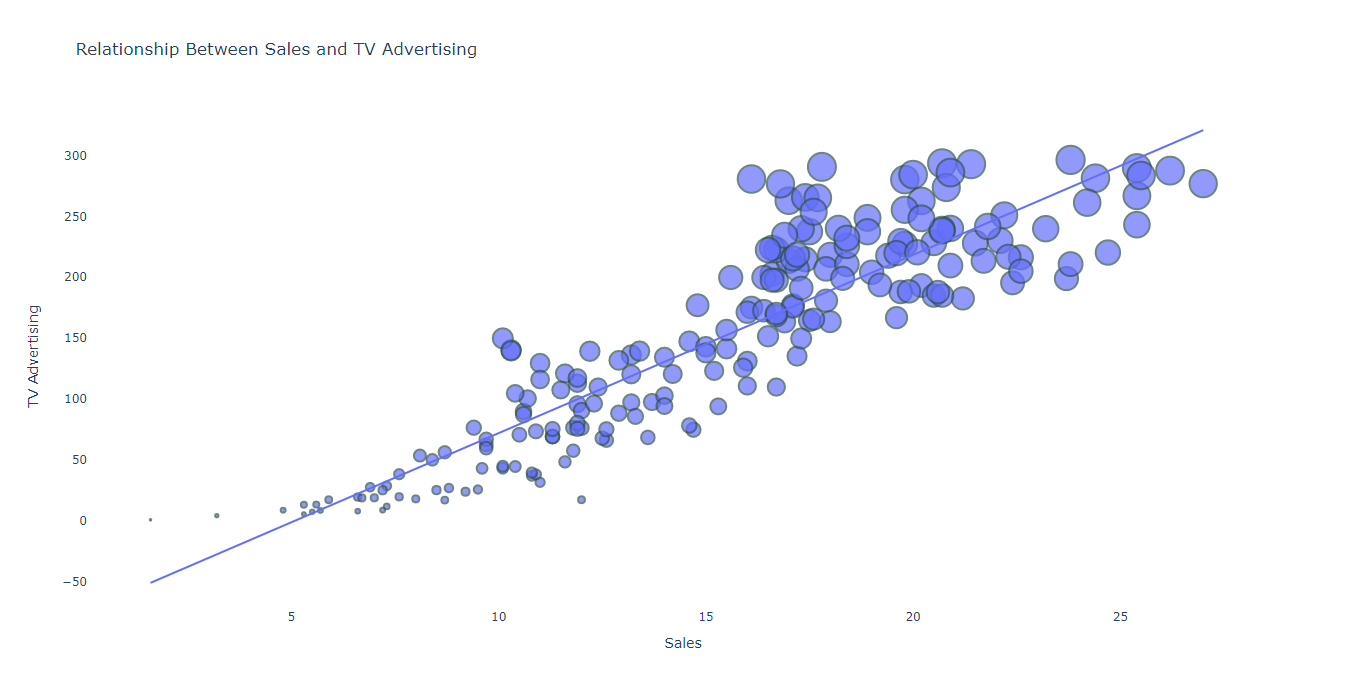
# import the needed packages or libraries  
import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
import plotly.express as px  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.model\_selection import cross\_val\_score  
from sklearn.linear\_model import LinearRegression, Ridge, Lasso  
from sklearn.tree import DecisionTreeRegressor  
from sklearn.ensemble import RandomForestRegressor  
# loading a data set in the project  
df=pd.read\_csv(r"C:\Users\TOBI\Desktop\IBM PROJECT\data set\Sales.csv")  
print(df.head())  
print(df.shape)  
print(df.info())  
print(df.describe().T)  
#using data analysis (EDA)  
figure = px.scatter(df, x='Sales', y='TV', size='TV', trendline='ols', title='Relationship Between Sales and TV Advertising')  
figure.update\_traces(marker=dict(line=dict(width=2, color='DarkSlateGrey')), selector=dict(mode='markers'))  
figure.update\_layout(  
 xaxis\_title='Sales',  
 yaxis\_title='TV Advertising',  
 legend\_title='TV Ad Size',  
 plot\_bgcolor='white'  
)  
print(figure.show())  
figure1 = px.scatter(df, x='Sales', y='Newspaper', size='Newspaper', trendline='ols', title='Relationship Between Sales and Newspaper Advertising')  
figure1.update\_traces(marker=dict(line=dict(width=2, color='DarkSlateGrey')), selector=dict(mode='markers'))  
figure1.update\_layout(  
 xaxis\_title='Sales',  
 yaxis\_title='Newspaper Advertising',  
 legend\_title='Newspaper Ad Size',  
 plot\_bgcolor='white'  
)  
print(figure1.show())  
figure2 = px.scatter(df, x='Sales', y='Radio', size='Radio', trendline='ols', title='Relationship Between Sales and Radio Advertising')  
figure2.update\_traces(marker=dict(line=dict(width=2, color='DarkSlateGrey')), selector=dict(mode='markers'))  
figure2.update\_layout(  
 xaxis\_title='Sales',  
 yaxis\_title='Radio Advertising',  
 legend\_title='Radio Ad Size',  
 plot\_bgcolor='white'  
)  
print(figure2.show())  
# Calculate the correlation  
correlation = df.corr()  
sales\_correlation = correlation["Sales"].sort\_values(ascending=False)  
  
# Format and style the correlation values  
styled\_sales\_correlation = sales\_correlation.apply(lambda x: f'{x:.2f}')  
styled\_sales\_correlation = styled\_sales\_correlation.reset\_index()  
styled\_sales\_correlation.columns = ["Feature", "Correlation with Sales"]  
styled\_sales\_correlation.style.background\_gradient(cmap='coolwarm', axis=0)  
print(sales\_correlation)  
#data preprocessing  
#Outlier detection  
#Box Plot of TV Advertising  
# Create the box plot  
plt.figure(figsize=(8, 6))  
sns.boxplot(x='TV', data=df, palette='Blues')  
plt.title('Box Plot of TV Advertising')  
plt.xlabel('TV Advertising Spending')  
plt.grid(axis='x', linestyle='--', alpha=0.6)  
# Show the plot  
print(plt.show())  
#Box Plot of Radio Advertising  
# Create the box plot  
plt.figure(figsize=(8, 6))  
sns.boxplot(x='Radio', data=df, palette='Oranges')  
plt.title('Box Plot of Radio Advertising')  
plt.xlabel('Radio Advertising Spending')  
plt.grid(axis='x', linestyle='--', alpha=0.6)  
#Newspaper Advertising Spending  
# Create the box plot  
plt.figure(figsize=(8, 6))  
sns.boxplot(x='Newspaper', data=df, palette='YlGnBu')  
plt.title('Box Plot of Newspaper Advertising')  
plt.xlabel('Newspaper Advertising Spending')  
plt.grid(axis='x', linestyle='--', alpha=0.6)  
  
# Show the plot  
print(plt.show())  
  
  
#threshold  
upper\_threshold = 2 \* np.std(df['Newspaper']) + np.mean(df['Newspaper'])  
df['Newspaper'] = np.where(df['Newspaper'] > upper\_threshold, upper\_threshold, df['Newspaper'])  
print(upper\_threshold)  
# Create a MinMaxScaler object  
scaler = MinMaxScaler()  
  
# Columns to be normalized  
columns\_to\_normalize = ['TV', 'Radio', 'Newspaper']  
  
# Apply Min-Max normalization to the selected columns  
df[columns\_to\_normalize] = scaler.fit\_transform(df[columns\_to\_normalize])  
print(df.head())  
  
#Modelling and Evaluation  
X = df[['TV', 'Radio', 'Newspaper']]  
y = df['Sales']  
# Performing 5-fold cross-validation (can be adjusted to the desired number of folds)  
num\_folds = 5  
# Function to perform cross-validation and calculate metrics in percentage  
def perform\_cross\_validation(model, X, y, num\_folds):  
 mse\_scores = -cross\_val\_score(model, X, y, cv=num\_folds, scoring='neg\_mean\_squared\_error')  
 rmse\_scores = np.sqrt(mse\_scores)  
 mae\_scores = -cross\_val\_score(model, X, y, cv=num\_folds, scoring='neg\_mean\_absolute\_error')  
 r2\_scores = cross\_val\_score(model, X, y, cv=num\_folds, scoring='r2')  
 return mse\_scores, rmse\_scores, mae\_scores, r2\_scores  
# Linear Regression  
linear\_model = LinearRegression()  
linear\_mse, linear\_rmse, linear\_mae, linear\_r2 = perform\_cross\_validation(linear\_model, X, y, num\_folds)  
print("Linear Regression:")  
print(f"Average MSE: {np.mean(linear\_mse) / np.mean(y) \* 100:.2f}%")  
print(f"Average RMSE: {np.mean(linear\_rmse) / np.mean(y) \* 100:.2f}%")  
print(f"Average MAE: {np.mean(linear\_mae) / np.mean(y) \* 100:.2f}%")  
print(f"Average R-squared: {np.mean(linear\_r2) \* 100:.2f}%")  
print("\n")  
# Ridge Regression  
ridge\_model = Ridge(alpha=1.0) # You can adjust alpha as needed  
ridge\_mse, ridge\_rmse, ridge\_mae, ridge\_r2 = perform\_cross\_validation(ridge\_model, X, y, num\_folds)  
print("Ridge Regression:")  
print(f"Average MSE: {np.mean(ridge\_mse) / np.mean(y) \* 100:.2f}%")  
print(f"Average RMSE: {np.mean(ridge\_rmse) / np.mean(y) \* 100:.2f}%")  
print(f"Average MAE: {np.mean(ridge\_mae) / np.mean(y) \* 100:.2f}%")  
print(f"Average R-squared: {np.mean(ridge\_r2) \* 100:.2f}%")  
print("\n")  
# Lasso Regression  
lasso\_model = Lasso(alpha=1.0) # You can adjust alpha as needed  
lasso\_mse, lasso\_rmse, lasso\_mae, lasso\_r2 = perform\_cross\_validation(lasso\_model, X, y, num\_folds)  
print("Lasso Regression:")  
print(f"Average MSE: {np.mean(lasso\_mse) / np.mean(y) \* 100:.2f}%")  
print(f"Average RMSE: {np.mean(lasso\_rmse) / np.mean(y) \* 100:.2f}%")  
print(f"Average MAE: {np.mean(lasso\_mae) / np.mean(y) \* 100:.2f}%")  
print(f"Average R-squared: {np.mean(lasso\_r2) \* 100:.2f}%")  
print("\n")  
# Decision Trees  
tree\_model = DecisionTreeRegressor(max\_depth=None, random\_state=0) # You can adjust parameters as needed  
tree\_mse, tree\_rmse, tree\_mae, tree\_r2 = perform\_cross\_validation(tree\_model, X, y, num\_folds)  
print("Decision Trees:")  
print(f"Average MSE: {np.mean(tree\_mse) / np.mean(y) \* 100:.2f}%")  
print(f"Average RMSE: {np.mean(tree\_rmse) / np.mean(y) \* 100:.2f}%")  
print(f"Average MAE: {np.mean(tree\_mae) / np.mean(y) \* 100:.2f}%")  
print(f"Average R-squared: {np.mean(tree\_r2) \* 100:.2f}%")  
print("\n")  
# Random Forest  
forest\_model = RandomForestRegressor(n\_estimators=100, random\_state=0) # You can adjust parameters as needed  
forest\_mse, forest\_rmse, forest\_mae, forest\_r2 = perform\_cross\_validation(forest\_model, X, y, num\_folds)  
print("Random Forest:")  
print(f"Average MSE: {np.mean(forest\_mse) / np.mean(y) \* 100:.2f}%")  
print(f"Average RMSE: {np.mean(forest\_rmse) / np.mean(y) \* 100:.2f}%")  
print(f"Average MAE: {np.mean(forest\_mae) / np.mean(y) \* 100:.2f}%")  
print(f"Average R-squared: {np.mean(forest\_r2) \* 100:.2f}%")  
print("\n")

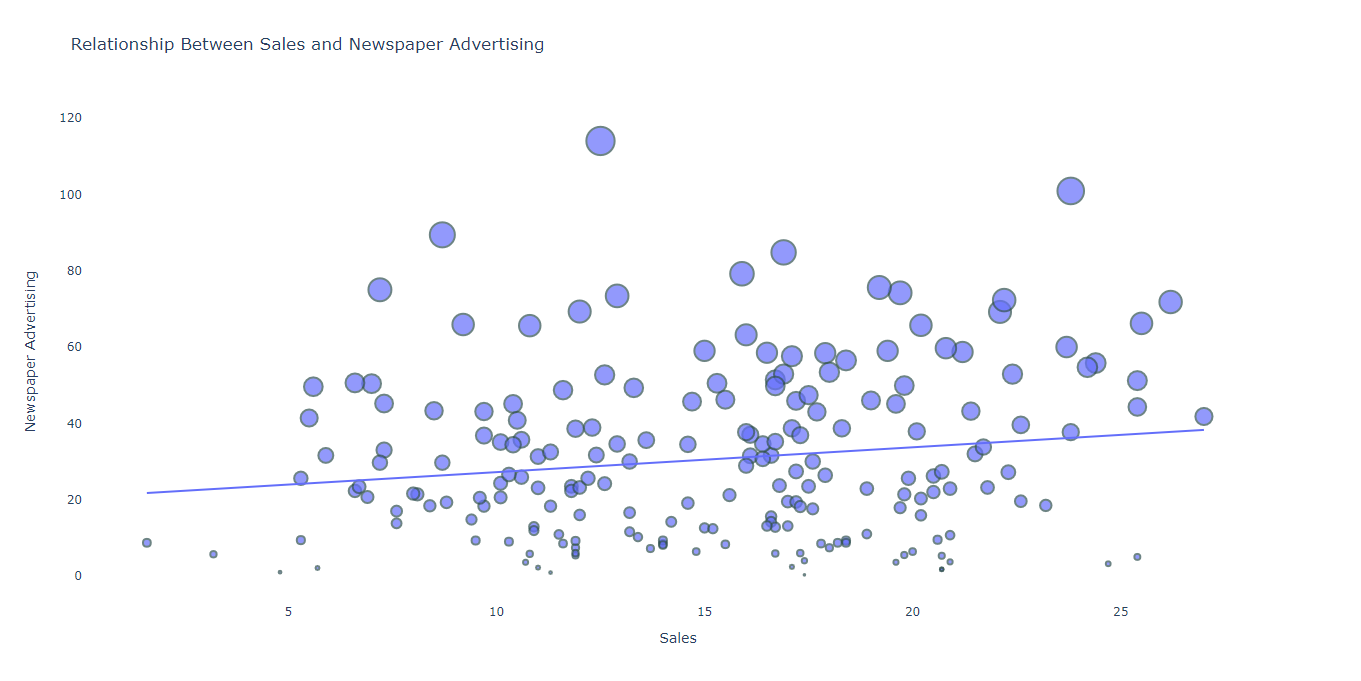
**Output:**

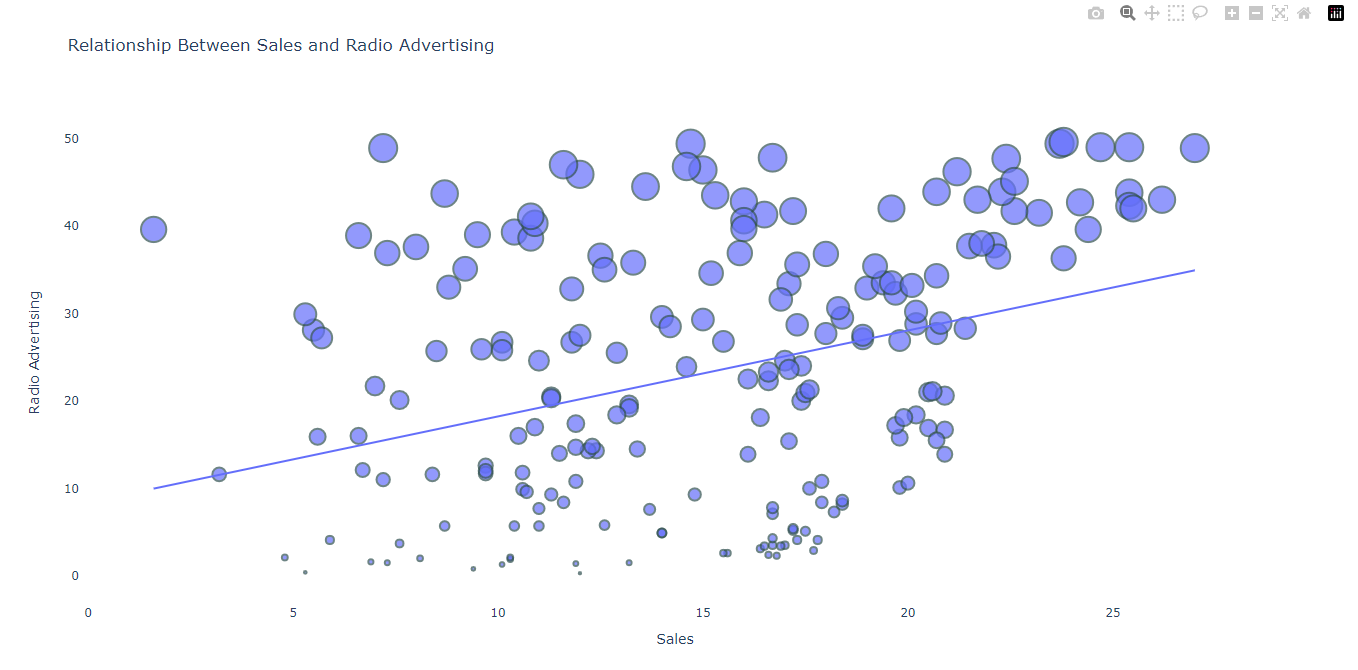
**C:\Users\TOBI\PycharmProjects\ADS-FSP\venv\Scripts\python.exe C:\Users\TOBI\PycharmProjects\ADS-FSP\L&P.py**

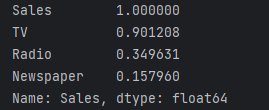
****

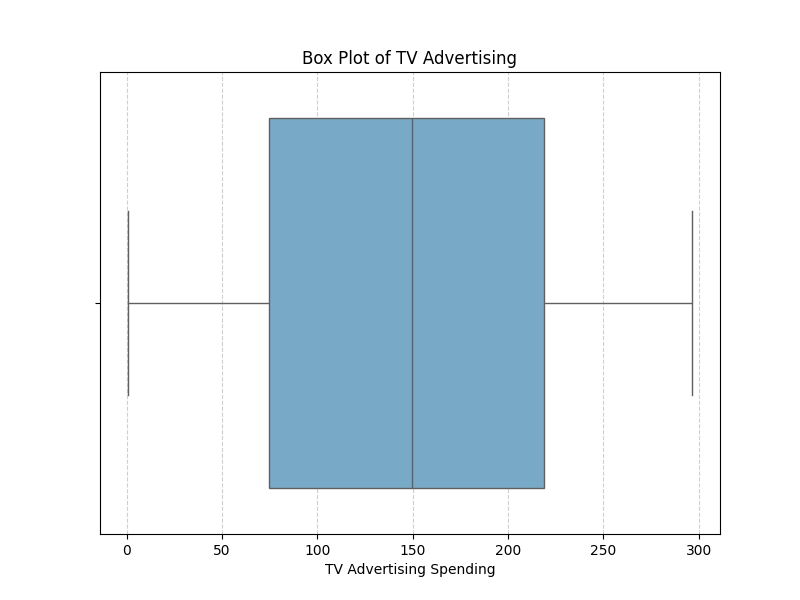
****

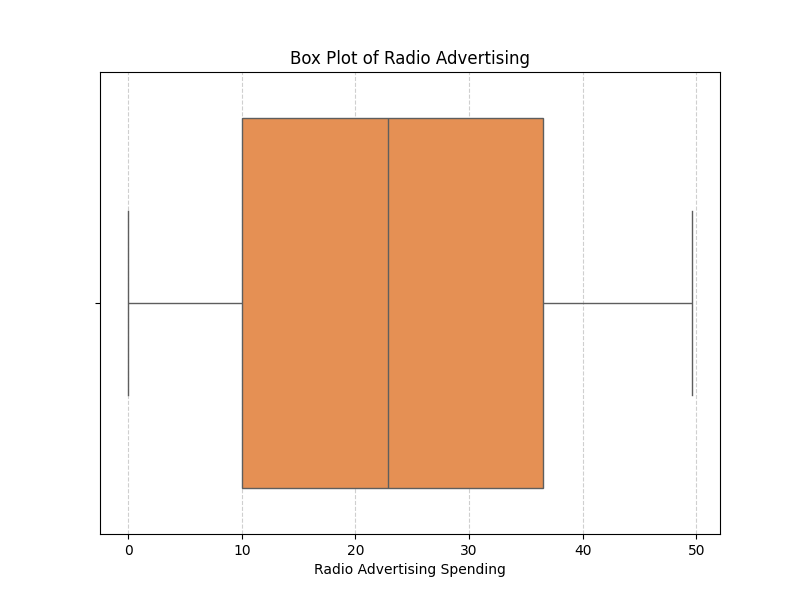


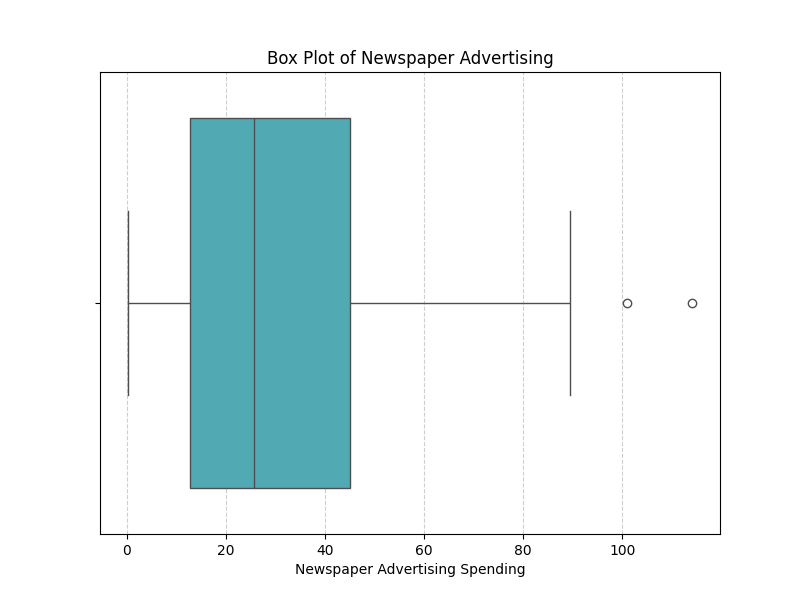


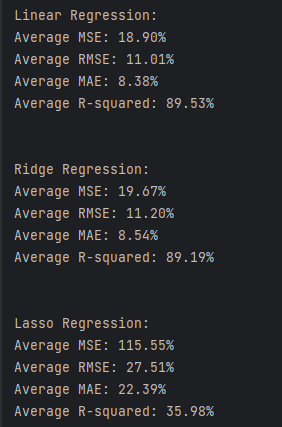


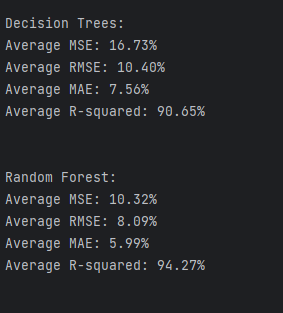
****









****

**8. Conclusion:**

* This Project on Future Sales Prediction Through Applied Data Science, Using the Dataset Of TV, Radio, Newspaper, And Sales, Offers Businesses A Data-Driven Pathway To Enhance Marketing Strategies, Optimize Advertising Budgets, And Make More Accurate Sales Forecasts.
* The project completed successfully.