FUTURE SALES PREDICTION

INTRODUCTION:

Predicting future sales is a crucial aspect of business management and strategic planning. It involves using historical sales data and various analytical techniques to forecast the likely sales figures for a specific period in the future. This prediction can assist businesses in making informed decisions, optimizing inventory, managing resources efficiently, and developing effective marketing and sales strategies. .

Predicting future sales using a sales dataset involves analyzing historical sales data to make informed forecasts. Here's a step-by-step guide on how to do this:

- **Data Collection and Preparation**:

Gather historical sales data, which should include information such as dates, products, quantities sold, and revenue generated .Ensure the data is clean, complete, and free from errors. Remove duplicates and handle missing values .Organize the data into a structured format that can be easily analyzed , such as a spreadsheet or a database.

- **Exploratory Data Analysis (EDA):**

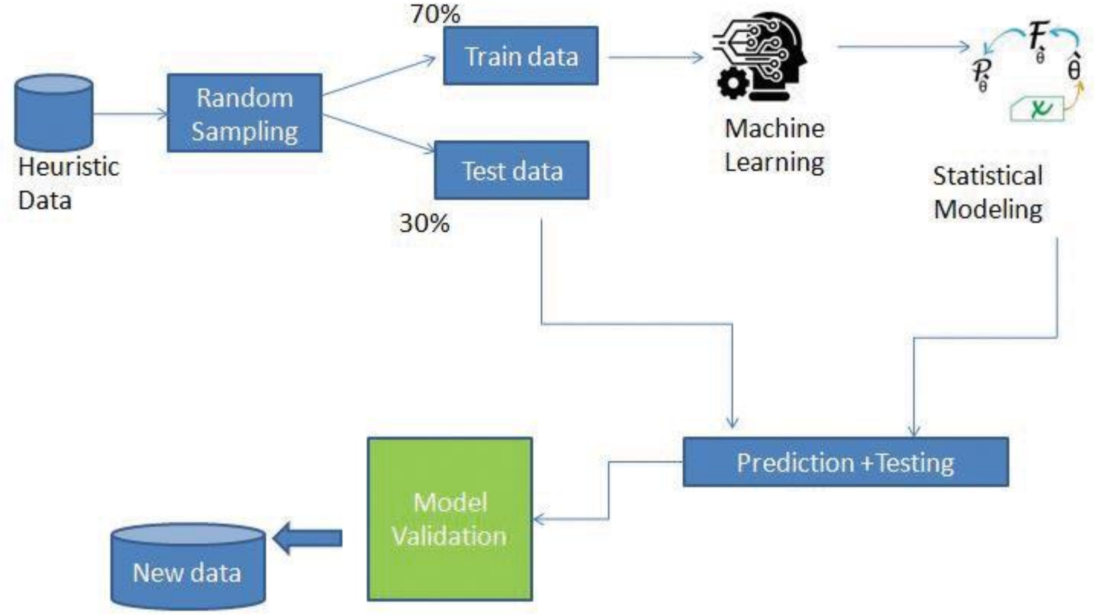
Conduct exploratory data analysis to understand the patterns and characteristics of your sales data. This can involve visualizing trends, seasonality, and outliers. Use statistical techniques to calculate summary statistics and key performance indicators.

- **Time Series Analysis**:

Time series analysis is a common approach for sales prediction. Create a time series plot to visualize historical sales trends over time. Identify any seasonality or cyclic patterns in the data, and decompose the time series into trend, seasonality, and residual components.

- **Feature Engineering:**

Create relevant features that can impact sales, such as promotions, holidays, economic indicators, and marketing activities. Lag variables can also be helpful, such as sales from the previous month or year.



In the analysis of the "Future Sales Prediction" dataset, we conducted a comprehensive series of data analysis steps to create an accurate prediction model. The process began with Exploratory Data Analysis (EDA) to understand the dataset's characteristics. Subsequently, we performed data preprocessing, including outlier detection.

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

LOADING THE DATA:

```
import pandas as pd

df = pd.read_csv('/kaggle/input/future-sales-prediction/Sales.csv')
df.head()
```

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

Features explanation:

- **TV**: this feature represents the amount of advertising budget spent on television media for a product or service in a certain period, for example in thousands of dollars (USD).
- **Radio**: this feature represents the amount of advertising budget spent on radio media in the same period as TV.
- **Newspaper**: this feature represents the amount of advertising budget spent in newspapers or print media in the same period as TV and Radio.
- **Sales**: This feature represents product or service sales data in the same period as advertising expenditure on TV, Radio and Newspaper.

```
df.shape
```

output:

```
(200, 4)
```

```
df.info()
```

output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

```
df.describe().T
```

output:

	count	mean	std	min	25%	50%	75%	max
TV	200.0	147.0425	85.854236	0.7	74.375	149.75	218.825	296.4
Radio	200.0	23.2640	14.846809	0.0	9.975	22.90	36.525	49.6
Newspaper	200.0	30.5540	21.778621	0.3	12.750	25.75	45.100	114.0
Sales	200.0	15.1305	5.283892	1.6	11.000	16.00	19.050	27.0

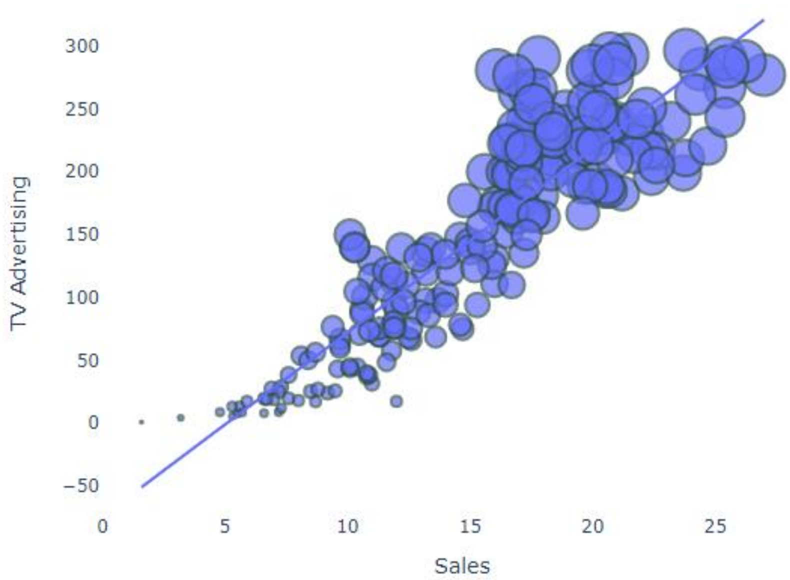
Exploratory Data Analysis (EDA):

```
import plotly.express as px

figure = px.scatter(df, x='Sales', y='TV', size='TV', trendline='ols', titl
e='Relationship Between Sales and TV Advertising')
figure.update_traces(marker=dict(line=dict(width=2, color='DarkSlateGre
y')), selector=dict(mode='markers'))
figure.update_layout(
    xaxis_title='Sales',
    yaxis_title='TV Advertising',
    legend_title='TV Ad Size',
    plot_bgcolor='white'
)
figure.show()
```

Output:

Relationship between sales and tv advertising



Data processing(Outlier detection):

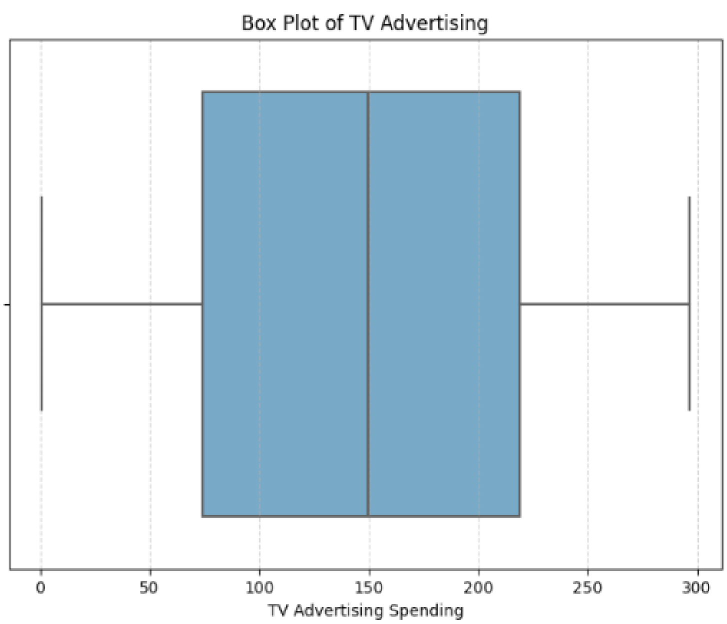
Using tv:

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

# 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
plt.show()
```

Output:

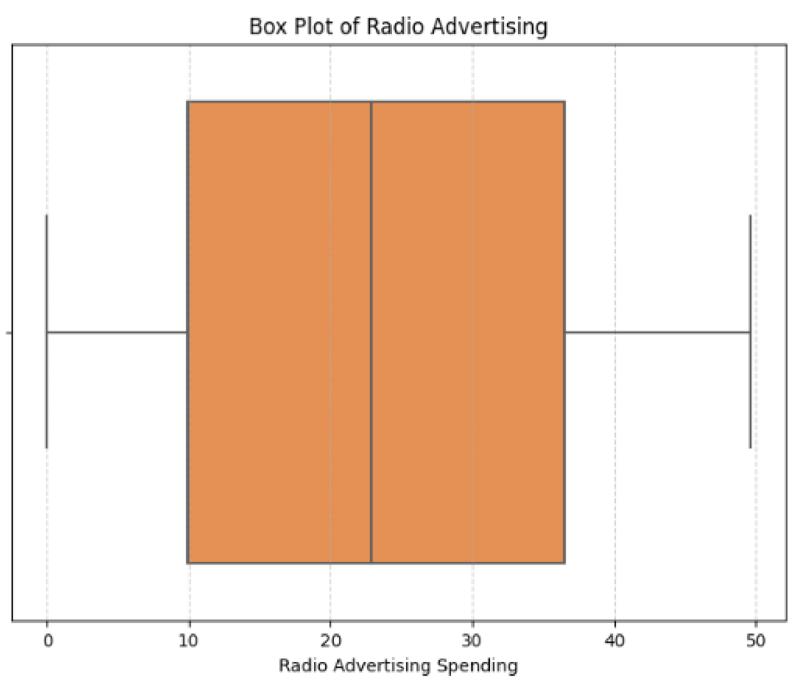


Using radio

```
# 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)

# Show the plot
plt.show()
```

Output:



Conclusion:

The goal of preprocessing is to clean and transform the raw data into a format that is suitable for your specific analysis or modelling needs.Proper preprocessing is essential for obtaining reliable and meaningful results from your data analysis or machine learning efforts.