|  |  |
| --- | --- |
| **Date** | **17-10-2023** |
| **Team ID** | **3869** |
| **Project Name** | **Future Sales Prediction** |

## 

## Importing Dependencies

import numpy as np  
import pandas as pd  
import plotly as px  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import MinMaxScaler

Import important Python libraries for data science and machine learning:

* numpy for scientific computing
* pandas for data analysis and manipulation
* plotly for interactive data visualization
* seaborn for statistical data visualization
* matplotlib.pyplot for plotting data
* sklearn.preprocessing.MinMaxScaler for scaling numerical features

This code provides the basic foundation for performing a variety of data science and machine learning tasks, such as:

* Loading and cleaning data
* Exploratory data analysis
* Feature engineering
* Model building
* Model evaluation
* Data visualization

# 

# Load Dataset

This code reads the Sales.csv file into a Pandas DataFrame and prints the first five rows.

data = pd.read\_csv('Sales.csv')  
data.head()

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

This code returns a tuple of two integers, representing the number of rows and columns in the DataFrame.

data.shape

(200, 4)

The data.info() method in Pandas prints a concise summary of the DataFrame, including the number of rows, columns, data types, memory usage, and range index.

data.info()

<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

The data.describe().T code in Pandas transposes the output of the describe() method, which summarizes the central tendency, dispersion, and shape of the numerical values in the DataFrame. Transposing the output means that the columns and rows are swapped.

data.describe().T

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

The data.columns attribute in Pandas returns a list of the column names in the DataFrame.

data.columns

Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')

# Visualisation of Data

Creates a joint plot with a regression line, showing the relationship between sales and TV advertising. Colors the plot sky blue and the regression line dark blue.

sns.set(style="white")  
sns.jointplot(data=data, x='Sales', y='TV', kind='reg', height=7, color='skyblue', line\_kws={'color':'darkblue'})  
plt.xlabel('Sales', fontsize=14)  
plt.ylabel('TV Advertising', fontsize=14)  
plt.suptitle('Relationship Between Sales and TV Advertising', y=1.02) plt.show()

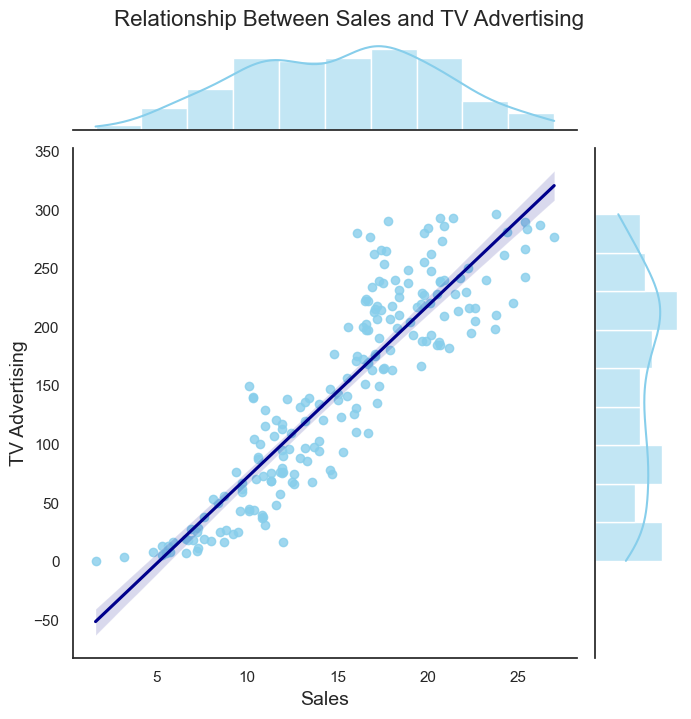


Fig.1 Relationship Between Sales and TV Advertising

This code uses the Seaborn library to create a joint plot with a regression line, showing the relationship between sales and newspaper advertising.

sns.jointplot(data=data, x='Sales', y='Newspaper', kind='reg', height=7, color='skyblue', line\_kws={'color':'darkgreen'})  
plt.xlabel('Sales', fontsize=14)  
plt.ylabel('Newspaper Advertising', fontsize=14)  
plt.suptitle('Relationship Between Sales and Newspaper Advertising', y=1.02, fontsize=16)  
plt.show()

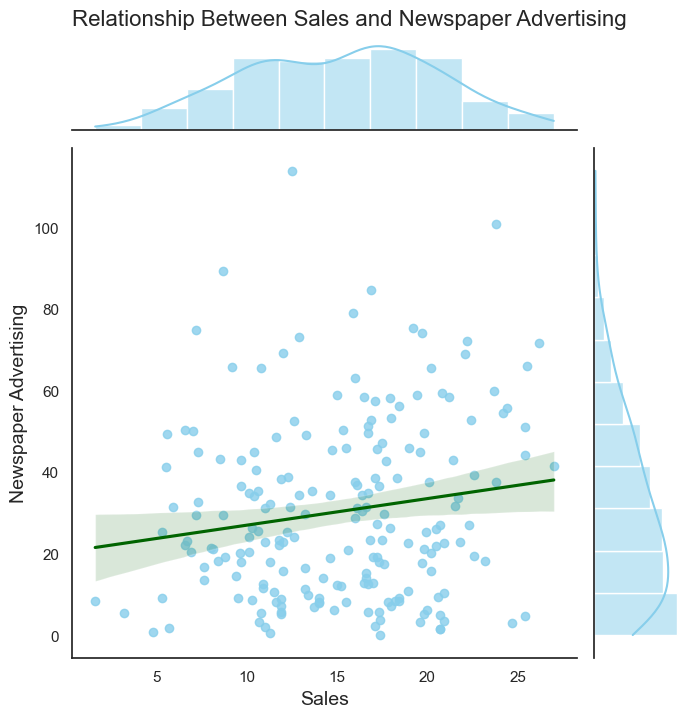


Fig 2. Relationship Between Sales and Newspaper Advertising

Creates a joint plot with a regression line, showing the relationship between sales and radio advertising. Colors the plot sky blue and the regression line dark orange.

sns.set(style="white")  
sns.jointplot(data=data, x='Sales', y='Radio', kind='reg', height=7, color='skyblue', line\_kws={'color':'darkorange'})  
plt.xlabel('Sales', fontsize=14)  
plt.ylabel('Radio Advertising', fontsize=14)  
plt.suptitle('Relationship Between Sales and Radio Advertising', y=1.02, fontsize=16)  
plt.show()

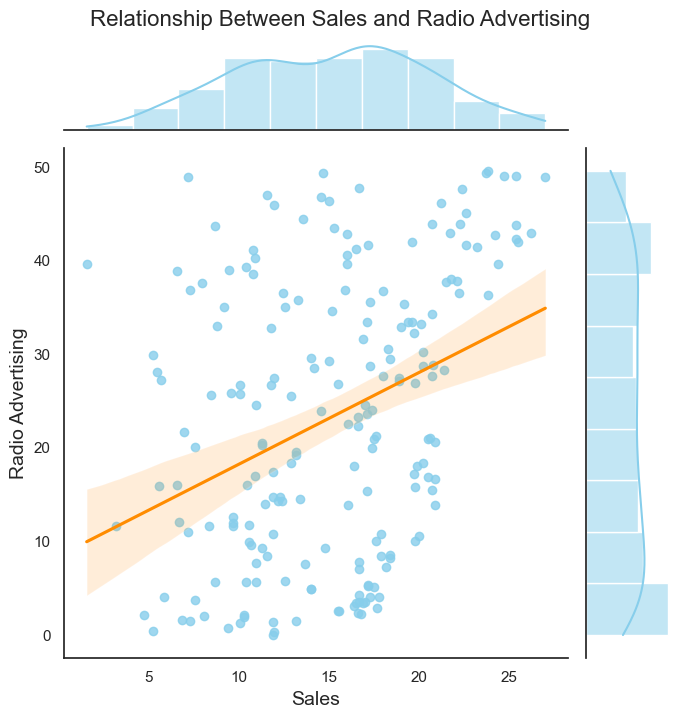


Fig 3. Relationship between Sales and Radio Advertising

Calculates the correlation between all features and sales, sorts by correlation, and creates a styled DataFrame with color gradient to highlight strongest correlations with sales.

correlation = data.corr()  
sales\_correlation = correlation["Sales"].sort\_values(ascending=False)  
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)

<pandas.io.formats.style.Styler at 0x1b1017ba250>

The code plt.figure(figsize=(12,8)) creates a new figure with a size of 12 inches by 8 inches. The code sns.pairplot(data) uses the Seaborn library to create a pairplot of the data DataFrame. A pairplot is a visualization that shows the pairwise relationships between all of the variables in a dataset.

plt.figure(figsize=(12,8))  
sns.pairplot(data)

<seaborn.axisgrid.PairGrid at 0x1b17da6b410>

<Figure size 1200x800 with 0 Axes>

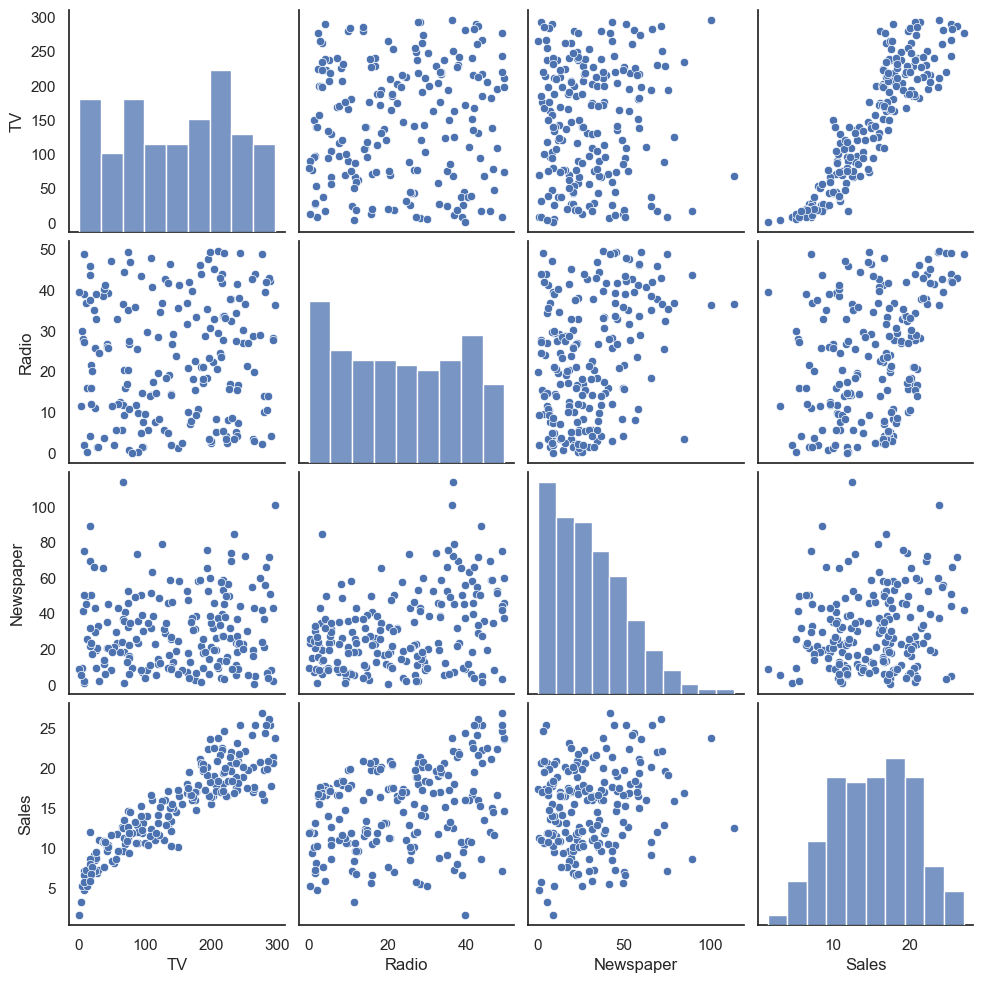


Fig 4. Pair Plot of TV, Radio, Newspaper, and Sales

# Data Preprocessing

To use the hist() method, you simply need to pass in the DataFrame that you want to create a histogram for. The hist() method will automatically create a histogram for each numerical column in the DataFrame. You can also specify the number of bins in the histogram and the color of the bars.

data.hist(figsize=(10,8))

array([[<Axes: title={'center': 'TV'}>,  
 <Axes: title={'center': 'Radio'}>],  
 [<Axes: title={'center': 'Newspaper'}>,  
 <Axes: title={'center': 'Sales'}>]], dtype=object)

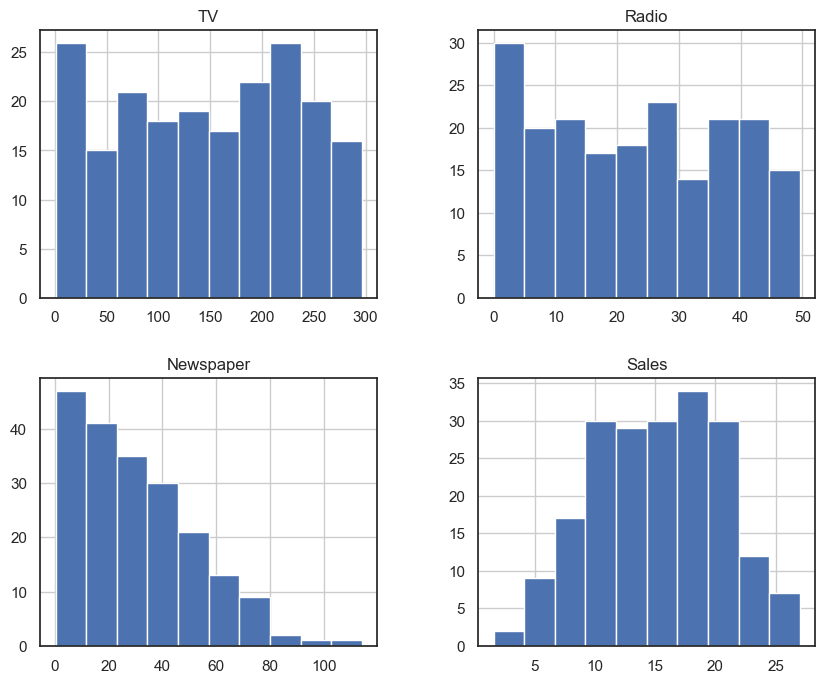


Fig 5. Histograms of TV, Radio, Newspaper, and Sales

# Data normalization

Normalization of numerical features is a common data preprocessing step that is used to scale the values of features to a common range. This can be helpful for machine learning algorithms, as it helps to prevent them from being biased towards features with larger values.

In the given code, the MinMaxScaler() object is used to scale the TV, Radio, and Newspaper columns in the data DataFrame to a range of 0 to 1. This is done by subtracting the minimum value from each column and then dividing by the difference between the maximum and minimum values.  
scaler = MinMaxScaler()  
columns\_to\_normalize = ['TV', 'Radio', 'Newspaper']  
data[columns\_to\_normalize] = scaler.fit\_transform(data[columns\_to\_normalize])  
data.head()

TV Radio Newspaper Sales  
0 0.775786 0.762097 0.605981 22.1  
1 0.148123 0.792339 0.394019 10.4  
2 0.055800 0.925403 0.606860 12.0  
3 0.509976 0.832661 0.511873 16.5  
4 0.609063 0.217742 0.510994 17.9

sns.heatmap(data.corr(),annot=True)

<Axes: >

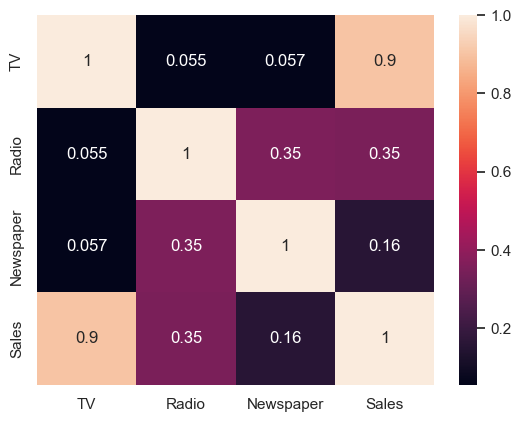


Fig 6. Correlation Heatmap