DS ASSIGNMENT 1

Name : ABHINAV.P

Roll : 21481A5401

METHOD 1 : LINEAR REGRESSION(for Advertising.csv)

In [2]:
```
#importing all the packages

import pandas as pd
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
import numpy as np
```

In [4]:
```
# Load the dataset

df = pd.read_csv("C:\\Users\\Administrator\\Downloads\\Advertising.csv")

# Extract features (X) and target variable (y)

X = df['TV'].values.reshape(-1, 1)
y = df['Sales'].values
```

In [5]:
```
# 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_
```

In [6]:
```
# Create a linear regression model
model = LinearRegression()
```

In [8]:
```
# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)
```

In [9]:
```
# Calculate the accuracy metrics

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = model.score(X_test, y_test)
```

In [10]:
```
# Display accuracy metrics

print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R-squared (R2): {r2}')
```

```
Mean Squared Error (MSE): 6.101072906773963
Root Mean Squared Error (RMSE): 2.470035001123256
R-squared (R2): 0.802561303423698
```

In [11]:
```
# Displaying accuracy for this particular method in percentage

accuracy = r2 * 100
print(f'Accuracy = {accuracy:.2f}%')
```

```
Accuracy = 80.26%
```

METHOD 2 : DECISION TREE(for Advertising.csv)

In [13]:
```
#importing the packages

import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score
```

In [14]:
```
# Load the data
data = pd.read_csv("C:\\Users\\Administrator\\Downloads\\Advertising.csv")

# Define features and target variables
features = ["TV", "Radio", "Newspaper"]
target = "Sales"
```

In [15]:
```
# Split data into training and testing sets

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    data[features], data[target], test_size=0.2, random_state=42
)
```

In [17]:
```
# Create the decision tree model
model = DecisionTreeRegressor(max_depth=3)  # Adjust max_depth as needed

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)
```

In [18]:
```
# Evaluate the model performance using R-squared

r2 = r2_score(y_test, predictions)

# Print the Accuracy

print("Accuracy:", f"{r2:.2%}")
```

```
Accuracy: 83.18%
```

METHOD 3 : RANDOM FOREST(for Advertising.csv)

In [20]:
```
#importing all the packages

import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

In [22]:
```
# Load data

data = pd.read_csv("C:\\Users\\Administrator\\Downloads\\Advertising.csv")
```

In [23]:
```
# Define features and target

features = ["TV", "Radio", "Newspaper"]
target = "Sales"
```

In [24]:
```
# Split data

X_train, X_test, y_train, y_test = train_test_split(
    data[features], data[target], test_size=0.2, random_state=42
)
```

In [25]:
```
# Create the Random Forest model
model = RandomForestRegressor(n_estimators=100, random_state=42)
```

In [26]:
```
# Train the model
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)
```

In [27]:
```
# Evaluate performance using R-squared
r2 = r2_score(y_test, predictions)

# Print Accuracy
print(f"Accuracy : {r2:.2%}")
```

```
Accuracy : 95.35%
```

METHOD 4 : POLYNOMIAL REGRESSION (for Advertising.csv)

In [33]:
```
#importing all the packages

import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

In [31]:
```
# Load data

data = pd.read_csv("C:\\Users\\Administrator\\Downloads\\Advertising.csv")
```

In [32]:
```
# Define features and target
```

```
features = ["TV", "Radio", "Newspaper"]
target = "Sales"
```

In [34]:
```
# Split data

X_train, X_test, y_train, y_test = train_test_split(
    data[features], data[target], test_size=0.2, random_state=42
)
```

In [35]:
```
# Create polynomial quadratic features
poly = PolynomialFeatures(degree=2)
```

In [36]:
```
# Transform training and testing features
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)
```

In [38]:
```
# Create the model (use linear regression internally)
model = LinearRegression()

# Train the model
model.fit(X_train_poly, y_train)

# Make predictions
predictions = model.predict(X_test_poly)
```

In [39]:
```
# Evaluate performance using R-squared
r2 = r2_score(y_test, predictions)

# Print Accuracy
print(f"Accuracy: {r2:.2%}")
```

```
Accuracy: 95.33%
```

DISPLAYING DATA IN BAR PLOT

In [48]:
```
import matplotlib.pyplot as plt

# Define data
Models = ["Linear Regression", "Decision Tree", "Random Forest", "Polynomial Reg
Accuracy = [80.26, 83.18, 95.35, 95.33]

# Create the bar chart
plt.figure(figsize=(8, 6))  # Set figure size for better readability
bars = plt.bar(Models, Accuracy, color='red', edgecolor='black')

# Add labels and title
plt.xlabel("Models")
plt.ylabel("Accuracy (%)")
plt.title("Comparison of Accuracy")

# Rotate x-axis labels for better visibility
plt.xticks(rotation=45, ha='right')

# Display the percentages on top of each bar
for bar, acc in zip(bars, Accuracy):
    plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, bar.get_height() + 0.5, f"
```

```
# Display the chart
plt.tight_layout()
plt.show()
```