**Development Part 2**

**Problem:**

In this technology you will continue building your project by performing feature engineering, model

training and evaluation.

Solution:

At the modelling stage, we use 5 algorithms for comparison, namely Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, and Random Forest And for evaluation using MSE, RMSE, MAE and R-Squared.

Program:

# import the needed packages or libraries  
import pandas as pd  
import numpy as np

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)  
print(df.shape)  
print(df.info())  
print(df.describe().T)

#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:

Linear Regression:

Average MSE: 18.90%

Average RMSE: 11.01%

Average MAE: 8.38%

Average R-squared: 89.53%

Ridge Regression:

Average MSE: 19.67%

Average RMSE: 11.20%

Average MAE: 8.54%

Average R-squared: 89.19%

Lasso Regression:

Average MSE: 115.55%

Average RMSE: 27.51%

Average MAE: 22.39%

Average R-squared: 35.98%

Decision Trees:

Average MSE: 16.73%

Average RMSE: 10.40%

Average MAE: 7.56%

Average R-squared: 90.65%

Random Forest:

Average MSE: 10.32%

Average RMSE: 8.09%

Average MAE: 5.99%

Average R-squared: 94.27%

Conclusion:

* + - Modelling and evaluating was done successfully.
    - Development part 2 completed.