DS Tutorial-1

Question 1:
Based on advertising data, find out the residual standard error(RSE), R^2 and F-statistics with respect to TV, radio, newspaper advertising budgets. Comment on the values.

TV	
Beta1	0.04753664
Beta0	7.03259355
RSS	2102.53856
RSE	3.26
TSS	5417.14875
R^2	0.61187358
F-statistic	312.143059

Radio	
Beta1	0.202495783
Beta0	9.311638095
RSS	3618.479549
RSE	4.27
TSS	5417.14875
R^2	0.332032455
F-statistic	98.42158757

Newspaper	
Beta1	0.054693098
Beta0	12.35140707
RSS	5134.804544
RSE	5.09
TSS	5417.14875
R^2	0.052120445
F-statistic	10.88729908

Answer:
TV: The strongest predictor, with the lowest **RSE (3.26),** high **R^2 (0.6119), and** a very significant **F-statistic (312.14).**
Radio: A moderate predictor, with a higher **RSE (4.27),** moderate **R^2 (0.3320),** and a significant **F-statistic (98.42).**
Newspaper: The weakest predictor, with the highest **RSE (5.09),** very low **R^2 (0.0521),** and the lowest **F-statistic (10.89).**

Question 2:

Create a dataset of your own choice, explain the dataset and using logistic regression predict the value for unknown inputs.

Answer:
Let's create a dataset to predict whether a student will pass or fail an exam based on their study hours.
Data:

Study Hours (X)	Pass (Y)
2	0
4	0
6	1
8	1
10	1
12	1
14	1
16	1

X: Study Hours (independent variable)

Y: Pass (dependent variable, binary: 0 = Fail, 1 = Pass)

Logistic Regression:

Logistic regression is suitable for this binary classification problem. We'll model the probability of passing **(P(Y=1))** as a function of study hours using the sigmoid function:

P(Y=1) = 1 / (1 + exp(-(b0 + b1*X)))

where:

- **b0:** Intercept
- **b1:** Coefficient for study hours

To predict whether a student with unknown study hours will pass, we plug the study hours value into the fitted model and calculate the probability. If the probability is above a certain threshold (e.g., 0.5), we predict that the student will pass; otherwise, we predict that they will fail.

Implementation:

Python

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
X = np.array([[2], [4], [6], [8], [10], [12], [14], [16]])
y = np.array([0, 0, 1, 1, 1, 1, 1, 1])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
intercept = model.intercept_
coef = model.coef_
```

BY
KEERTHANA SURESH.D
S5 IT
ROLL NUMBER:33