Variables { Predictor → X value or Feature
Target or Response → Y value or Labels.

MULTIPLE LINEAR REGRESSION WITH SCIKIT LEARNING

Here we have multiple variables affecting our outcome. So instead of univariate model of Simple Linear Reg. we use multiple linear reg.

Simple → $Y = \beta_0 + \beta_1 X + \epsilon$
or $y = \theta_0 + \theta_1 x$ (Like we already did)

If we use multiple linear regression on simple linear one feature will be used & other will be completely ignored

Multiple = $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$

In our case sales = $\beta_0 + \beta_1 \times TV + \beta_2 \times radio + \beta_3 \times newspaper + \epsilon$.

TASK-1 Import LIBRARIES

TASK-2 LOAD THE DATA

df = pd.read_csv('/content/Advertising.csv')
df.head() → first 5 rows

df.info() → info about dataset

df.isna().sum() → to null we can also use df.isnull().any()

Duplicates df.duplicated(subset=['TV', 'radio', 'newspaper']).values.any()

__/__/____

TASK - 3 Relationship between Features & Response

Sns.~~pairplot~~ pairplot(df, x_vars = ['TV', 'radio', 'newspaper'], y_vars = 'sales', height = 7, aspect = 0.5)

Seaborn.

↳ It helps to draw the relationship ~~to~~ in one line.

TASK-4 Multiple Linear Regression - Estimating Coefficients.

Feature: x = df[['TV', 'radio', 'newspaper']]

Response: y = df.sales

can be written as x = df.iloc[:, :-1]

lr = LinearRegression() from sklearn.linear_model import LinearRegression

lr.fit(x,y) → It will return only the coefficients our model estimates.

Coefficient Returned. {

print(lr.intercept_)

print(lr.coef_) → List of 3 values one unit.

the lr.coef_ tells us that if we increase one feature by what effect does it have on the Response or y

Pair each of value with coefficient.

list(zip([['TV', 'radio', 'newspaper'], lr.coef_]))

Correlation

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

What does heat map describes?

__/__/____

TASK-5 Feature Selection.

Q1. Is atleast one of predictor which is useful or is it actually a group of predictors.

Q2. We can also ask our model how well it fits our data.

Q3 Given a set of predictor values (Features) what response value should we predict and how accurate is our prediction.

We will answer these questions.

→ This process is known as feature selection.

How to determine which model is best.?

We will use r2-score of scikit learn.

First we will check for TV and Radio together.
fitting the model.

```
lr2 = LinearRegression().fit(x[['TV','radio']],y)
lr2_predictions = lr2.predict(x[['TV','radio']])
print(r2_score(lr2_predictions,y))
```

Predicting on training set so not a good practice

Predicted value

Original value.

This is WRONG

```
r2_score(y, lr2_predictions)
```

True value first ←

→ Predicted value afterwards

__/__/____

TASK-6 Model Evaluation Using Train/Test split and Metrics.

```
x = df[['TV', 'radio', 'newspaper']]
y = df.sales

x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)

lr3 = LinearRegression().fit(x_train, y_train)

lr3_predictions = lr3.predict(x_test)
```

RMSE → `print(np.sqrt(mean_squared_error(y_test, lr3_pred)))`

(y_test: original, lr3_pred: predicted)

R2 score → `print(r2_score(y_test, lr4_predictions))`

Plotting Prediction Error of our Result.

```
visualizer = PredictionError(lr_final).fit(x_train, y_train)

visualizer.score(x_test, y_test)

visualizer.poof()  # draw/show/poof the data
```

__/__/____

TASK → Interaction Effect (Synergy) in Regression Analysis.

Make a model with one more feature called Interaction where we assume both TV and radio together as one unit with same weightage.

```
df['interaction'] = df['TV'] * df['radio']

x = df[['TV', 'radio', 'interaction']]

y = df.sales

# split
# fit
# predict.
```

Such model in this case gives better result from previous we have seen in this case. which means 50% budget for TV 50% for radio till now is better.

~~I think~~

I think there can be a way to find if instead of 50-50 maybe 75-25 budget model reveals better results.