22 '22 March

Tuesday

MARCH 2022

09.00 LINEAR REGRESSION

10.00 SIMPLE LINEAR REGRESSION -

11.00 $y = \beta_0 + \beta_1 x$

12.00 target y, constant, parameter

13.00 To find the parameter we need to find ERROR function.

14.00 $\ell_i = y_i - \hat{y}_i$

15.00 real target, predicted outcome

16.00

17.00 FOR LINEAR REGRESSION WE MINIMISES SUM OF SQUARED ERRORS.

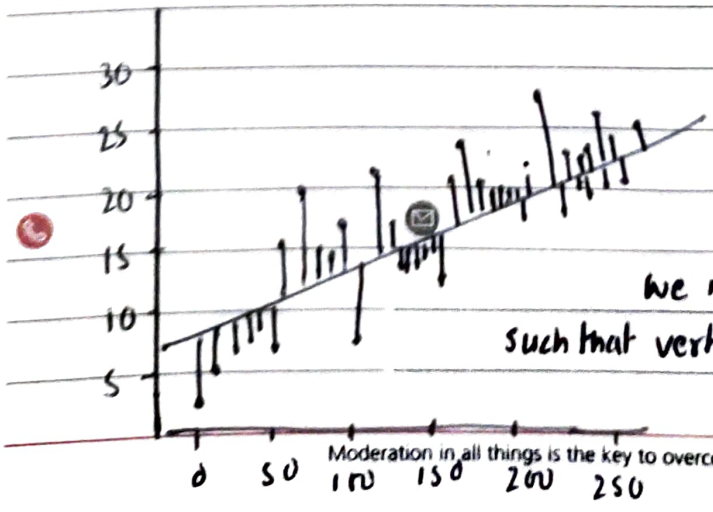
18.00

ERROR VISUALIZATION -

• - DOTS REPRESENT DATA

— - LINE IS FITTED STRAIGHT LINE

EACH VERTICAL LINE IS MAGNITUDE OF ERROR



we need to fit the straight line such that vertical lines are small as possible.

09.00

WHY WE SQUARE THE ERRORS?

10.00 - ERRORS CAN BE POSITIVE OR NEGATIVE (ABOVE OR BELOW THE LINE)

↳ so the results could be positive or negative.

11.00 ↳ if we did not square the errors, we would be adding bunch of negative errors and reduce the sum of errors.

12.00 ↳ it would trick us in thinking we are fitting a good straight line, but in reality we are not.

13.00

14.00 - Negative values could decrease the error

15.00 - Penalize the errors

16.00 Finding Coefficient –

17.00

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

18.00

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$\bar{x}$ - mean of independent variable

$\bar{y}$ - mean of the target.

24 '22 March

Week 13 Day (083-282) | Thursday

MARCH 2022

09.00 FINDING THE COEFFICIENT IN PRACTISE - + PYTHON

10.00 reg = LinearRegression() // initialize the model
reg.fit(X, y) // fit on X and y, then we can
11.00 retrieve both intercept and coeff

12.00 intercept = reg.intercept_[0]
coef = reg.coef_[0][0]

13.00 Estimate the relavance of coefficients

14.00 • p-value : quantify statistical significance

15.00 • Tells us if we can reject the null hypothesis or not

16.00 In PYTHON, WE CAN ANALYSE pvalue FOR EACH COEFFICIENT
LIKE THIS —

17.00

```
exog = sm.add_constant(X)
est = sm.OLS(y, exog).fit()
print(est.summary())
```

// we statistical model package that allows us to use summary of the model.

sm model provides
- statistical test
- data exploration

ow we live them.

09.00 NULL HYPOTHESIS -

10.00

	coef	std err	t	P>|t|	[0.025	0.975]
	7.0326	0.458	15.360	0.00	6.130	7.935
11.00	0.0475	0.003	11.668	0.00	0.042	0.053

12.00 ↑ actually it is not zero

13.00 but it is so small, that it appears to be zero.

14.00 ONCE WE KNOW ARE PARAMETERS ARE RELAVANT

15.00 ↳ WE MUST ACCESS THE MODEL ITSELF

16.00 Accuracy of Linear Model -

17.00 Residual Standard Error (RSE) = $\sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$

18.00 • smaller the value - the better it is

since

★★★ ↳ difference b/w actual and predicted is small

Do not say all that you know, but always know what you say.

MARCH 2022

MON	TUE	WED	THU	FRI	SAT	SUN
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

09.00

R^2 Value - It measures the proportion of variability explained by a feature x.
As it approaches 1 - it means that we are explaining a lot of variablity in our target.

10.00

11.00

12.00

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}, \quad TSS = \sum (y_i - \hat{y}_i)^2$$

13.00

14.00 • Measure of proportion of variability explained by feature X

15.00 In Python, we can use the same process as we did before to find the p-value using the same package.

16.00

17.00

```
exog = sm.add_constant(X)
est = sm.OLS(y,exog).fit()
print(est.summary())
```

Sunday 27

18.00

APRIL 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30								

March '22

28

Monday

Week 14 Day (087-278)

09.00 ACCURACY OF LINEAR MODEL

10.00 OLS REGRESSION RESULTS

11.00	Dep. Variable	sales	R-squared	0.612
	Model	OLS	Adj. R-squared	0.610
12.00	Method	Least Square	F-statistic	312.1
	Date	Sat, 09 May 20	Prob (F-statistic)	1.47e-42
13.00	Time	15:19:51	Log-Likelihood	-519.05
	No. Obs.	200	AIC	1042
14.00	Df Residuals	198	BIC	1049
	Df Model	1		
15.00	Covariance Type	nonrobust		

		coef	std err	t	P>|t|	[0.025	0.975]
16.00	const	7.0326	0.458	15.360	0.00	6.130	7.935
17.00	TV	0.0475	0.003	17.668	0.00	0.042	0.053

18.00	Omnibus :	0.531	Durbin-Watson	1.935
	Prob (Omnibus)	0.767	Jarque-Bera (JB)	0.669
	Skew :	-0.089	Prob (JB)	0.716
	Kurtosis :	2.779	Cond. No	338.

29 '22 March

Week 14 Day 088-277 | Tuesday

MARCH 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

09.00 MULTIPLE LINEAR REGRESSION

10.00 $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \_\_\_ + \beta_p X_p$

```
req = LinearRegression()
req.fit(X, y)

beta_0 = req.intercept_[0]
beta_1 = req.intercept_[0][0]
beta_2 = req.intercept_[0][1]
beta_3 = req.intercept_[0][2]
```

TO ACCESS THE MODEL OF MULTIPLE LINEAR REGRESSION

18.00 ↳ we use 'F' statistics.

$$F = \frac{\frac{TSS - RSS}{p}}{\frac{RSS}{(n-p-1)}}$$

p - no. of predictors

n - no. of data points

APRIL 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30								

March '22

Wednesday 30

Week 14 Day (089-276)

09.00 In Python, we would use same package for 'F' statistics

10.00 ★ Whole Result Report as previous

11.00 • How RELAVANT IS THE MODEL

12.00 → F >> 1, there is a strong relationship

13.00 → for small dataset, F has to be way larger than 1

14.00 Hands On -

15.00 • On Google Colab / Could also be done on Jupyter Notebook

16.00

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import
LinearRegression
import statsmodels.api
%matplotlib inline
```

17.00

18.00

31 22 March

Thursday

MARCH 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

09.00

READING THE CSV FILE

10.00

```
data = pd.read_csv('path or url')
```

11.00

12.00

SIMPLE LINEAR REGRESSION

13.00

14.00

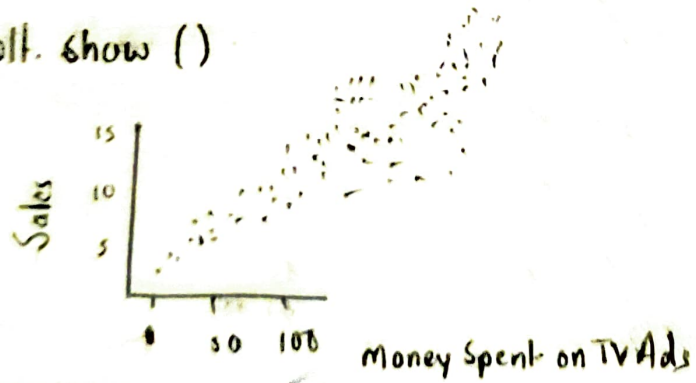
```
plt.figure(figsize = (16, 8))
plt.scatter(data['TV'], data['sales'],
            c='black')
plt.xlabel('Money Spent on TV Ads ($)')
plt.ylabel('Sales (k$)')
plt.show()
```

15.00

16.00

17.00

18.00



Learn to manage yourself, then you are qualified to manage others.

MAY 2022

MON	TUE	WED	THU	FRI	SAT	SUN
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

April '22 02

Saturday

Week 14 Day (092-273)

09.00 SPECIFYING OUR FEATURE

```
X = data['TV'].values.reshape(-1,1)
y = data['sales'].values.reshape(-1,1)
reg = LinearRegression()
reg.fit(X,y)
print(f"The Linear Model is \n
      Y = (reg.intercept_[0]) +
      (reg.coef_[0][0] * TV")
```

specifying the feature

specifying the target

call reg model & fit it

printing our constant and coef. for TV

OUTPUT-

The Linear Model is:
Y = (reg.intercept_[0]) + (reg.coef_[0][0] * TV

Sunday 03

→ NEXT STEP IS PLOTTING THE LINE ←

04 '22 April

Week 15 Day (094-271) | Monday

APRIL 2022

MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

09.00

PLOTTING THE LINE

10.00

```
predictions = reg.predict(X)
plt.figure(figsize=(16,8))
plt.scatter(X,y,c="black")
plt.plot(X, predictions, c="blue",
  linewidth=2)
plt.xlabel('Money Spent on TV Ads')
plt.ylabel('Sales (K$)')
plt.show()
```

calling predict method on X

black is color of dots

Y in this case is predictions

blue is line color

17.00 Access the Quality of Our Model

18.00 #CATCH - FIRST IN ORDER TO USE ANY API, IT SHOULD BE PROPERLY EXPORTED.

In our case, import statsmodels.api as sm — ✓

import statsmodels.api — X

Your prayer must be for a sound mind in a sound body.

09.00

ACCESS THE QUALITY OF OUR MODEL

10.00

```
import statsmodels.api as sm

X = data['TV']
Y = data['Sales']

exog = sm.add_constant(X)
est = sm.OLS(Y, exog).fit()

print(est.summary())
```

11.00

12.00

13.00

14.00

15.00

OLS Regression Results

16.00

17.00 X = data['TV'] - selects the TV coloumn of the dataset and assigns it to the variable 'X'

18.00 y = data['Sales'] - select the sales coloumn of the dataset and assigns it to variable 'y'

exog = sm.add_constant(X) - adds coloumn of ones to the X variable to serve as intercept term and assign the results to exog

'est = sm.OLS(y, exog).fit()'- Fits an ordinary least square(OLS) regression model using 'y' as the response variable and 'exog' as the predictor variables, then assign result to 'est'.

06 '22 April

Week 15 Day (096-269) | Wednesday

APRIL 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30								

09.00 print(est.summary()) - print a summary of the regression results, which include information such as coefficient, standard errors, t-values, p-values, of predictor variables, as well as goodness of fit statistics like R-squared and F-statistic

13.00 Overall, this code is fitting a simple linear regression model to the data and printing out summary of results.

15.00 — Multi Linear Regression —

16.00

17.00
```
Xs = data.drop(['sales'], axis=1)
y = data['sales'].values.reshape(-1,1)
```

18.00
```
reg = LinearRegression()
reg.fit(Xs, y)
print(f"The Linear Model is: \n Y =
{reg.intercept_[0]} + {reg.coef_[0][0]}
*TV + {reg.coef_[0][1]}* radio +
{reg.coef_[0][2]} * newspaper")
```

09.00 LINEAR REGRESSION CLASS FROM SCIKIT LEARN

10.00 Xs = data.drop ('sales'], axis=1) - select all colouums of the dataset except for 'sales' and assigns them to the

11.00 variable 'Xs'

Y = data ['sales'] .values.reshape (-1,1) - select the sales

12.00 coloumn of dataset and convert it to a two-dimensional Numpy array with one coloumn using the 'reshape()'

13.00 method. This is assigned to the variable 'y'.

reg = LincarRegression()' - create an instance of the LR class

14.00 and assign it to variable 'reg'.

reg.fit (Xs,y) - Fits the linear regression model using 'Xs' as

15.00 the predictor variable and 'y' as the responsible variable

print (f" the Linear ____ neswpaper") - prints the

16.00 equation of Linear model, including intercept term and coefficient of the predictor variable

17.00

Overall, code is performing a multiple LR using TV, radio

18.00 and newspaper coloumn of dataset as predictor variable and sales coloum as the response variable.

It then print outs the equation of Linear model that was fitted

08 '22 April

Week 15 Day (098-267) | Friday

APRIL 2022

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
				1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30								

09.00

This code fits a multi Linear regression model to data, with 'TV', 'radio', and 'newspaper' as predictor variables and 'sales' as the response variable.

10.00

11.00

12.00

```
X = np.column_stack((data['TV'],
    data['newspaper']))                    — 1
y = data['sales'].values.reshape(-1,1)     — 2

exog = sm.add_constant(X)                  — 3
est = sm.OLS(y, exog).fit()                — 4

print(est.summary())                       — 5
```

13.00

14.00

15.00

16.00

17.00

18.00

1- creates matrix X with 3 Coloumns, corresponding to the TV, radio, and newspaper variables. the line stacks the coloums horizontally using np.coloum_stack() function from Numpy.

2- creates a coloumn vector y containing the sales variables, reshaped to have one column and many rows as orignal data.

09.00 The line uses the '.values' attribute of pandas DataFrame
to get the the underlying data as Numpy array and
10.00 then applies the '.reshape()' method to convert it
to the desired shape.
11.00

3) exog = sm.add_constant(X) adds a constant column
12.00 to the matrix X using the 'add_constant()' function
from statsmodels. The constant term needed to fit
13.00 the intercept term in regression model.

14.00 4) est = sm.OLS(y, exog).fit() - fits a multiLinear regression
model to the data, using the ordinary least Square (OLS)
15.00 method from statsmodel.
The line creates new object 'est' of the
16.00 OLS class and applies its '.fit()' method to the data.
This method estimates the regression coefficient by
17.00 minimizing the sum of squared residuals b/w the Sunday 10
predicted values and actual values and returns an object
18.00 of the 'RegressionResults' class that contains various
statistics and diagnostic about the model.

5) print summary of regression results to the console.