510 DATA SCIENCE

Lecture 08

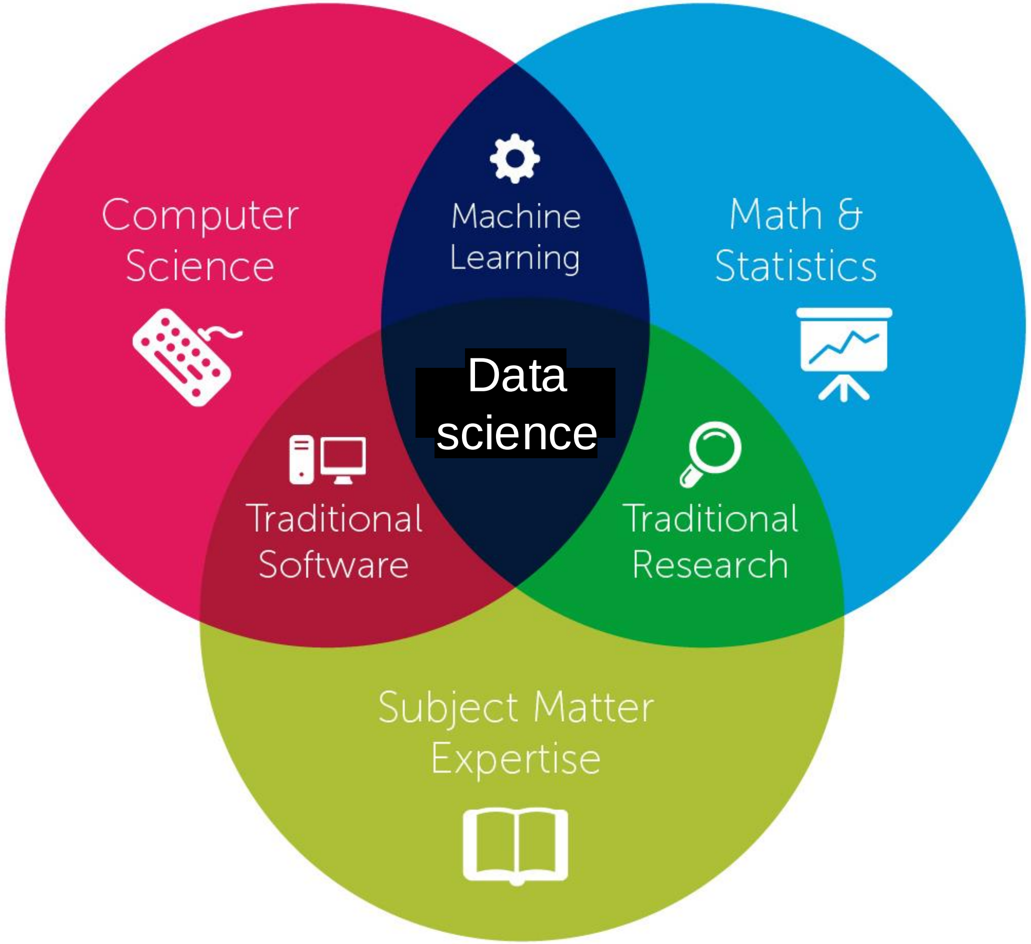
Fall 2024

Instructor: Assoc. Prof. Şener Özönder

Email: sener.ozonder@bogazici.edu.tr

Institute for Data Science & Artificial Intelligence

Boğaziçi University



Linear Regression

- We will use the advertisement data for 200 different companies (200 rows): How does TV, radio and newspaper ads affect sales?

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
…	…	…	…

Model:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

sales $= \beta_0 + \beta_1$ **TV** $+ \beta_2$ **radio** $+ \beta_3$ **newspaper**

We can use packages like sklearn, scipy and statmodels to find the model parameters (β_0, β_1, β_2, β_3).

statmodels outputs a nice table as shown here.

```
X = advertising[['TV', 'radio', 'newspaper']]
X = sm.add_constant(X)
y = advertising['sales']
model = sm.OLS(y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.896
Method:                 Least Squares   F-statistic:                     570.3
Date:                Sun, 05 Nov 2023   Prob (F-statistic):           1.58e-96
Time:                        17:47:12   Log-Likelihood:                -386.18
No. Observations:                 200   AIC:                             780.4
Df Residuals:                     196   BIC:                             793.6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.9389      0.312      9.422      0.000       2.324       3.554
TV             0.0458      0.001     32.809      0.000       0.043       0.049
radio          0.1885      0.009     21.893      0.000       0.172       0.206
newspaper     -0.0010      0.006     -0.177      0.860      -0.013       0.011
==============================================================================
Omnibus:                       60.414   Durbin-Watson:                   2.084
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              151.241
Skew:                          -1.327   Prob(JB):                     1.44e-33
Kurtosis:                       6.332   Cond. No.                         454.
==============================================================================
```

Linear Regression

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
…	…	…	…

- Model:

sales = β_0 + β_1 **TV** + β_2 **radio** + β_3 **newspaper**

p values testing:

H_0: $\beta=0$

H_a: $\beta\neq 0$

estimated coefficients

standard error SE

Confidence intervals

$\beta - 2*SE(\beta)$ $\beta + 2*SE(\beta)$

		coef	std err	t	P>|t|	[0.025	0.975]
β_0	const	2.9389	%10 0.312	9.422	0.000	2.324	3.554
β_1	TV	0.0458	0.001	32.809	0.000	0.043	0.049
β_2	radio	0.1885	0.009	21.893	0.000	0.172	0.206
β_3	newspaper	-0.0010	0.006	-0.177	0.860	-0.013	0.011

%600 error for **newspaper!** This is bad.

Since p-value for β_3 is not smaller than 0.05, we can't reject the null hypothesis $\beta_3=0$. In street language, don't include newspaper into the model.

Interval contains zero. So it's likely $\beta_3=0$, i.e., **newspaper** predictor does not affect **sales**.

- Look at the p values. Eliminate the predictors whose p>0.05. In this example, the best model is

sales = β_0 + β_1 **TV** + β_2 **radio** ~~+ β_3 newspaper~~

Linear Regression

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
…	…	…	…

- Model:

sales = β_0 + β_1 TV + β_2 radio + β_3 newspaper

estimated coefficients

standard error SE

p values testing:

H_0: $\beta=0$

H_a: $\beta \neq 0$

Confidence intervals

$\beta - 2*SE(\beta)$ $\beta + 2*SE(\beta)$

		coef	std err	t	P>|t|	[0.025	0.975]
β_0	const	2.9389	%10 0.312	9.422	0.000	2.324	3.554
β_1	TV	0.0458	0.001	32.809	0.000	0.043	0.049
β_2	radio	0.1885	0.009	21.893	0.000	0.172	0.206
β_3	newspaper	-0.0010	0.006	-0.177	0.860	-0.013	0.011

<u>Effect size:</u>
Radio is the most determining predictor. Invest on radio ads than TV ads if not too expensive.

<u>Error size:</u>
With more data, we can reduce error on β_0.

<u>Statistical significance:</u>
Effect of **newspaper** is not statistically significant since p>0.05 here.

<u>Business decision:</u>
For $1000 spending in TV ads, sales get a boost between 43 to 49 units. Is this acceptable or not. Your boss should decide!

Linear Regression

Remember, this is a very bad predictor!

- What if we use a simpler model: **sales** = β_0 + β_1 **newspaper**

		coef	std err	t	P>|t|
β_0	const	12.3514	0.621	19.876	0.000
β_1	newspaper	0.0547	0.017	3.300	0.001

newspaper effect size increased, its error decreased.

newspaper this time looks "significant", so the output telling us to keep the predictor **newspaper.**

Correlation matrix

```
correlation_matrix = advertising.corr()
print(correlation_matrix)
```

	TV	radio	newspaper	sales
TV	1.000000	0.054809	0.056648	0.782224
radio	0.054809	1.000000	0.354104	0.576223
newspaper	0.056648	0.354104	1.000000	0.228299
sales	0.782224	0.576223	0.228299	1.000000

- Why did **newspaper** as a predictor worked this time? Well, it's actually carrying the effect of **radio** predictor; it's a surrogate! Look at the correlation value in the matrix: 0.35. For some reason, people sometimes (corr=0.35) give ads to **newspaper** and **radio** at the same time.
- **newspaper** is imitating as if a good predictor because it's correlated with **radio.** Sales are actually affected by **radio** ads, not **newspaper** (see previous slide). **newspaper** *appears to be* significant because it is acting as a proxy for **radio**.
- However, when you fit a model with all the predictors (TV, radio, and newspaper), the unique contribution of each variable to the variance in sales is considered. If the effect of **newspaper** is no longer significant when **radio** is included, it suggests that the effect initially attributed to **newspaper** was actually due to its association with **radio** spending. In conclusion, eliminate **newspaper.**

Linear Regression

- Funny example: Data may suggest this model is good:

shark attacks = $\beta_0 + \beta_1$ **ice cream sales**

Would we conclude that **ice cream sales** should be banned at beaches to eliminate **shark attacks**?

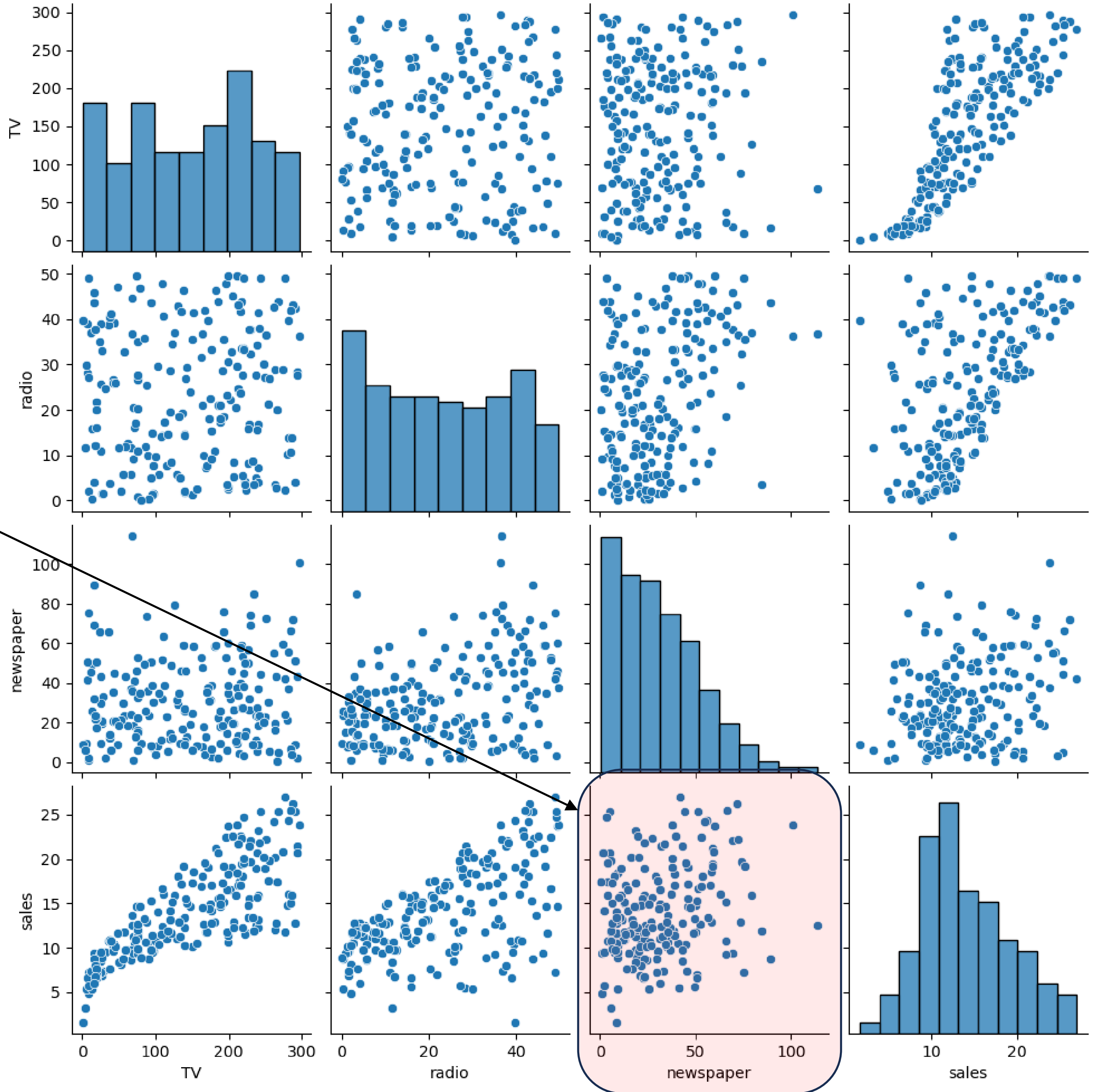
In reality, **higher temperatures** cause more people to visit the beach, which in turn results in more **ice cream sales** and more **shark attacks**. A multiple regression of **shark attacks** onto **ice cream sales** and **temperature** will reveal that, as intuition implies, **ice cream sales** is no longer a significant predictor after adjusting for **temperature.**

So, **shark attacks** = $\beta_0 + \beta_1$ **ice cream sales** seems to work because **ice cream sales** is acting as a surrogate for **temperature.** When we do the regression **shark attacks** = $\beta_0 + \beta_1$ **ice cream sales +** β_2 **temperature**, we'll see p-value for β_1 will turn out to be large and we'll find the correct model is **shark attacks** = $\beta_0 + \beta_2$ **temperature**

Linear Regression

- Back to advertising data:

Here it can be seen that the correlation between **newspaper** and **sales** is pretty weak or nonexistent based on the scatter diagram.



Linear Regression

- Okay, we found the best model as **sales** = 2.94 + 0.05 **TV** + 0.19 **radio**. But is it a good model? How well the this equation model the relationship between the predictors and the response? We assess it by using coefficient of determination R^2. It's given by

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}}$$

$$\mathrm{RSS} = \sum (y_i - \hat{y}_i)^2$$

Residual sum of squares. y_i is y value of data points, $\hat{y}_i$ is the model prediction on y values.

$$\mathrm{TSS} = \sum (y_i - \bar{y})^2$$

Total sum of squares. y_i is y value of data points, $\bar{y}$ is the mean of y values of data points.

- R^2 is between 0 and 1. The closer to 1, the better the model is.
- In physics where the underlying model is well-known to be linear, R^2=0.8 may be deemed bad model, but in biology, R^2=0.1-0.3 may be considered as evidence of an effect. So, a "good value" of R^2 depends on the field and context. For a given dataset, we should choose or eliminate predictors until we maximize R^2, i.e., until we find the best model.
- R^2 is also equal to the correlation squared corr$(Y, \hat{Y})^2$ between the response Y and fitted model predictions $\hat{Y}$.

Linear Regression

sales = 5.78 + 0.05 **TV** + 0.04 **newspaper**

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.646
Model:                            OLS   Adj. R-squared:                  0.642
Method:                 Least Squares   F-statistic:                     179.6
Date:                Mon, 06 Nov 2023   Prob (F-statistic):           3.95e-45
Time:                        00:26:13   Log-Likelihood:                -509.89
No. Observations:                 200   AIC:                             1026.
Df Residuals:                     197   BIC:                             1036.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.7749      0.525     10.993      0.000       4.739       6.811
TV             0.0469      0.003     18.173      0.000       0.042       0.052
newspaper      0.0442      0.010      4.346      0.000       0.024       0.064
==============================================================================
```

sales = 2.92 + 0.05 **TV** + 0.19 **radio**

- This is a better model as it has higher R^2 (or adjusted R^2) value.
- So we can look at R^2 or adjusted R^2 value while doing model selection.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.896
Method:                 Least Squares   F-statistic:                     859.6
Date:                Mon, 06 Nov 2023   Prob (F-statistic):           4.83e-98
Time:                        00:28:40   Log-Likelihood:                -386.20
No. Observations:                 200   AIC:                             778.4
Df Residuals:                     197   BIC:                             788.3
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.9211      0.294      9.919      0.000       2.340       3.502
TV             0.0458      0.001     32.909      0.000       0.043       0.048
radio          0.1880      0.008     23.382      0.000       0.172       0.204
==============================================================================
```

Linear Regression

- What if we add **radio²** term?

sales = β_0 + β_1 **TV** + β_2 **radio** + β_3 **radio²**

OLS Regression Results

Dep. Variable:	sales	R-squared:	0.898
Model:	OLS	Adj. R-squared:	0.896
Method:	Least Squares	F-statistic:	428.3
Date:	Mon, 06 Nov 2023	Prob (F-statistic):	2.31e-95
Time:	00:53:52	Log-Likelihood:	-385.60
No. Observations:	200	AIC:	781.2
Df Residuals:	195	BIC:	797.7
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.1869	0.389	8.201	0.000	2.421	3.953
TV	0.0458	0.001	32.831	0.000	0.043	0.049
radio	0.1562	0.031	4.962	0.000	0.094	0.218
newspaper	-0.0015	0.006	-0.262	0.794	-0.013	0.010
radio_squared	0.0007	0.001	1.069	0.286	-0.001	0.002

- Turns out, p-value for β_3 got larger (0.286), and it didn't increase the adjusted R^2, so no need to add the term **radio²** to the model.

(From previous slide)

sales = β_0 + β_1 **TV** + β_2 **radio**

OLS Regression Results

Dep. Variable:	sales	R-squared:	0.897
Model:	OLS	Adj. R-squared:	0.896
Method:	Least Squares	F-statistic:	859.6
Date:	Mon, 06 Nov 2023	Prob (F-statistic):	4.83e-98
Time:	00:28:40	Log-Likelihood:	-386.20
No. Observations:	200	AIC:	778.4
Df Residuals:	197	BIC:	788.3
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.9211	0.294	9.919	0.000	2.340	3.502
TV	0.0458	0.001	32.909	0.000	0.043	0.048
radio	0.1880	0.008	23.382	0.000	0.172	0.204

Linear Regression

Using adjusted R^2 instead of R^2 for model selection

- When you add new predictors like radio, or their powers like **radio²** or interaction terms like **TV** × **radio**, you want to know if adding this term made the model better by increasing R^2. However, R^2 may increase also merely due to adding a new term too. So, to remove this effect we use adjusted R^2 while comparing models with different number of predictors. Adjusted R^2 incorporates a penalty for adding predictors that do not improve the model.

sample size = # of records

$$\text{Adjusted } R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

# of predictors

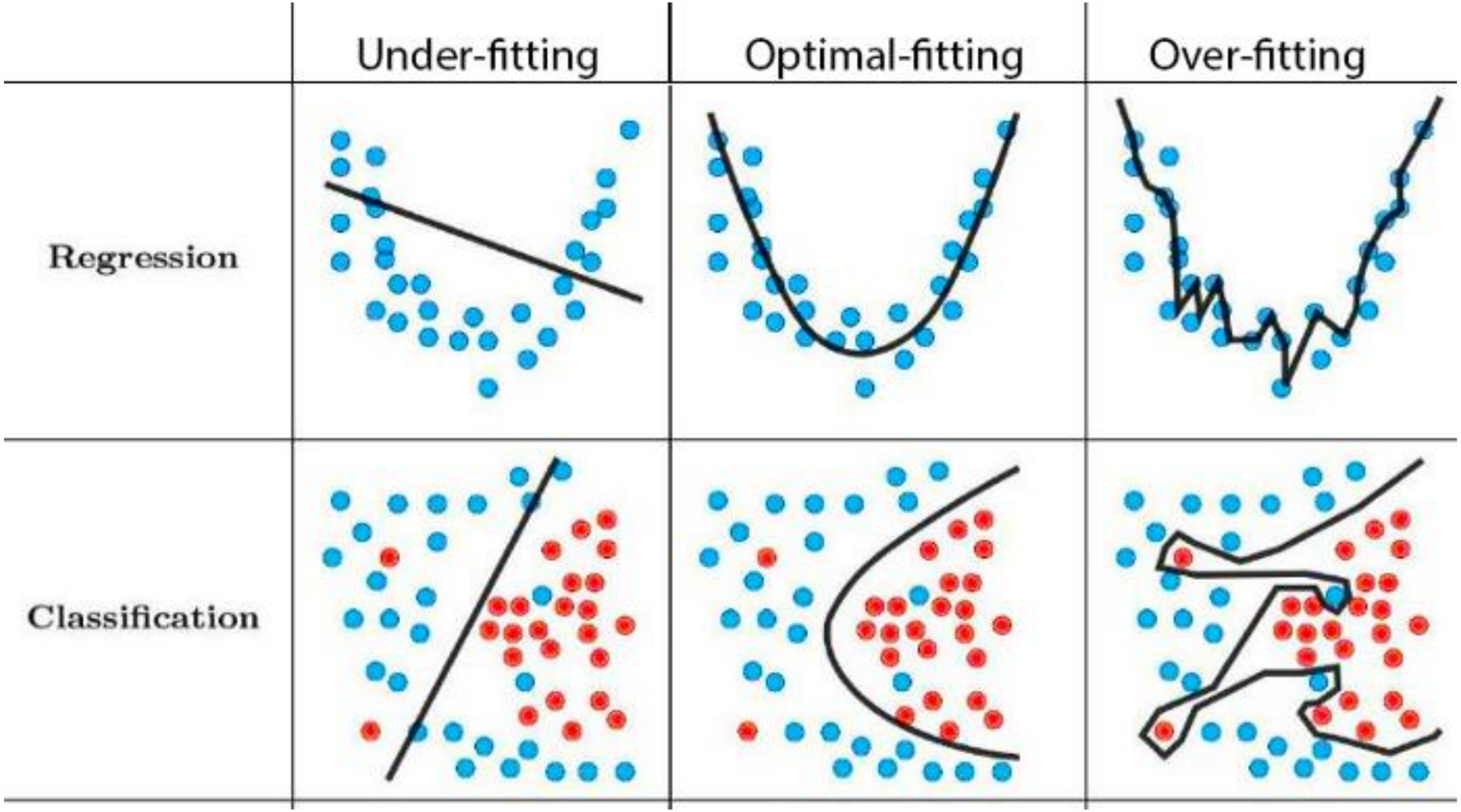
R^2 is not enough

- Okay, R^2 (or adjusted R^2) actually tells you how well your model fits the data data at hand very well. It doesn't tell you anything about how well your model will perform on new data that the model haven't seen yet.
- What if you have an overfitting model? If that's the case, it will not generalize and so underperform on new data.
- So, R^2 is not enough if you're interested in prediction (using model on new data). But before we move to train-test split of data, let's mention scenarios when R^2 may be enough:
  i. Descriptive Analysis (Explanatory Modeling): When the goal is to understand the relationships within the data, not to predict future or unseen outcomes (economics or sociology).
  ii. Small Datasets: Data set is too small to split as train (~%80) and test (~%20).

- Now let's discuss building generalizable models for prediction on new data.

Underfitting & Overfitting

- We need to avoid both underfitting and overfitting.



Performance metrics for regression

For $\hat{y}_i$=y_{pred} is the model prediction and y_i is the target from data.

1) MSE (mean squared error): $\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$

If y is in meters, MSE is in meter^2. So be careful while interpreting.

2) RMSE (root mean squared error): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$

RMSE and y have the same units. Easier to interpret.

3) MAE (mean absolute error): $\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$

MAE and y have the same units. Easier to interpret.

MAE is more robust against outliers wrt RMSE because RMSE has errors squared within it. So, if there are outliers (large $\hat{y}_i - y_i$), RMSE≫MAE.

Performance metrics for regression

4) MAPE (mean absolute percentage error): $\frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$

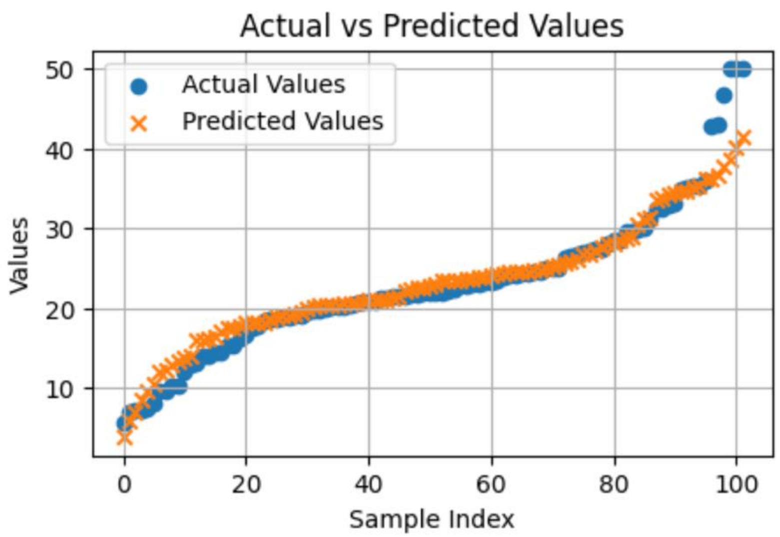
Seeing percentage error is nice, but MAPE sometimes may exceed 100%.

When some targets are zero, MAPE is infinite.

It's not symmetric since the denominator has only y_i. Using MAPE is risky!

Note: These metrics are average $\frac{1}{N}\Sigma$ quantities.

Error may be high and low in different parts of the data. So it may be a good idea to plot $\hat{y}_i$ and y_i on the same plot.



Train and Test sets

- Remember, R^2 doesn't tell us anything about the predictive power of the model. We have to use train-validation-test split to build a model for prediction.

Training	Validation	Test
Use it for **training**. (70%)	Uses it for **model selection** and **hyperparameter tuning**. (15%)	Never ever use it until you've done your best in **training**, **hyperparameter tuning** and **model selection** to obtain the best model. Only then, use it to calculate your final best model's test metrics. (15%)

- Be careful while choosing records for validation and test set. Your dataset may be sorted, so choosing from the beginning or end will create biased validation and test sets. Sample randomly or use Python build in data split functions.

Train and Test sets

Training	Validation	Test
Use it for **training**. (70%)	Uses it for **model selection** and **hyperparameter tuning**. (15%)	Never ever use it until you've done your best in **training**, **hyperparameter tuning** and **model selection** to obtain the best model. Only then, use it to calculate your final best model's test metrics. (15%)

- **Model selection example:** We want to determine which model would be better for prediction. In other words, we want to avoid an underfitted or overfitted model.

Model 1: $y = \beta_0 + \beta_1 x$

Model 2: $y = \beta_0 + \beta_1 x + \beta_2 x^2$

Assuming both has good R^2, we have to train them on the train set and then calculate error (MSE/RMSE/MAE/MAPE) on the validation set. Then, choose model that has the least error. Let's say it's model 1.

Then calculate the test error for model 1 using the test set and report your results.

Train and Test sets

Training	Validation	Test
Use it for **training**. (70%)	Uses it for **model selection** and **hyperparameter tuning**. (15%)	Never ever use it until you've done your best in **training**, **hyperparameter tuning** and **model selection** to obtain the best model. Only then, use it to calculate your final best model's test metrics. (15%)

- **Hyperparameter tuning example:** We will use the model below but we want to avoid overfitting,

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$

so we will use a loss function with regularization term

$$Loss = \sum_{i=1}^{N} (\underset{\text{pred}}{\hat{y}_i} - \underset{\text{target}}{y_i})^2 + \lambda \sum_{i=1}^{4} \beta_i^2$$

where the second term prevents some coeffs β's to be very large, which in turn reduces model complexity a bit to avoid overfitting.

But we don't know the hyperparameter λ. So we need to tune it to find the best model.

i. Make multiple model training on train set by using different values for λ (10^{-3} — 10^{+3}),

ii. Calculate error (MSE/RMSE/MAE/MAPE) on the validation set. Then, choose model that has the least error,

iii. Then calculate the test error for the chosen model using the test set and report your results.

Train and Test sets

Training	Validation	Test
Use it for **training**. (70%)	Uses it for **model selection** and **hyperparameter tuning**. (15%)	Never ever use it until you've done your best in **training**, **hyperparameter tuning** and **model selection** to obtain the best model. Only then, use it to calculate your final best model's test metrics. (15%)

- **Fixed model:** Assuming for some reason we determined the model to be

$y = \beta_0 + \beta_1 x + \beta_2 x^2$

We just want to find out how our model will perform on new data.

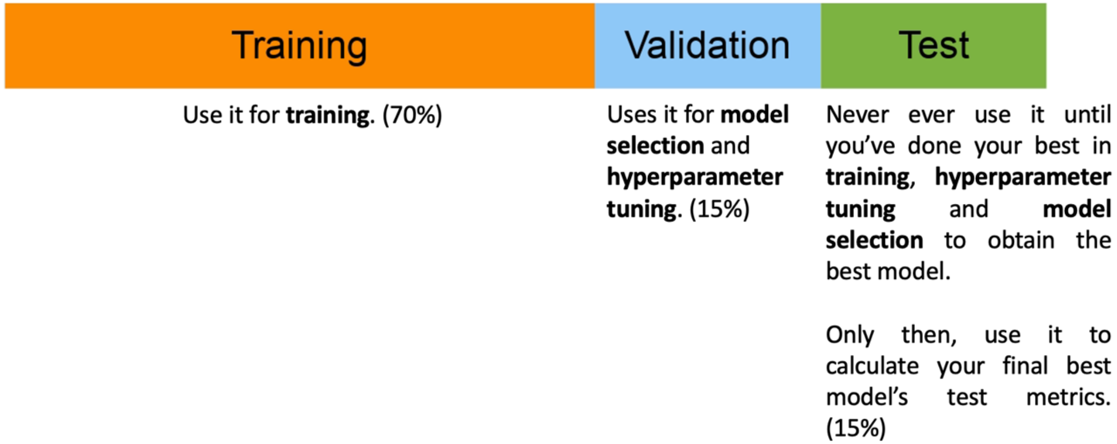
Since, no model selection here. Also no hyperparameter to tune. So, you don't need validation split. Train on training data (80%) and test with MSE/RMSE/MAE/MAPE on test data (20%).

Train and Test sets

Signs of overfitting:

- Low train error but high validation error.

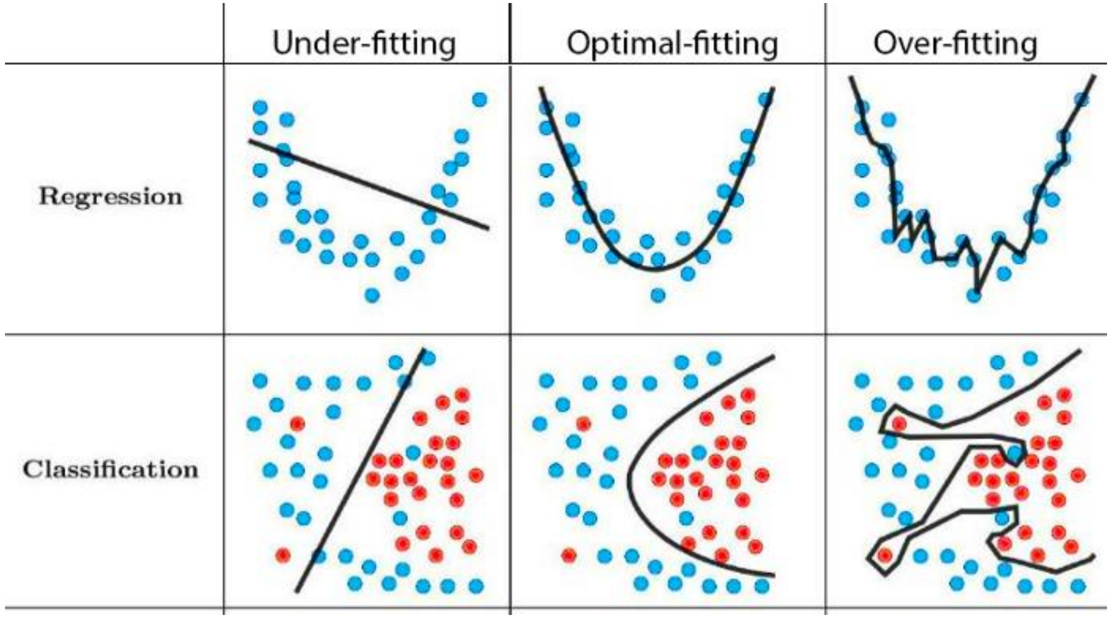
Needs: Less complex model, regularization.



Signs of underfitting:

- High train error.

Needs: More data, more complex model, tuning hyperparameters.



Signs of good fit:

- Low or moderate train error, and test error is close to train error (small gap between them.)