JUNE 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										

May '22

Monday 02

09.00 RESAMPLING AND REGULARIZATION

Resampling - IMPORTANT FOR MODEL PERFOMANCE AND

10.00 VALIDATION. DONE IN EVERY DATA SCIENCE TASK.

EX- CROSS VALIDATION

11.00

REGULARIZATION - USED TO PREVENT OVERFITTING AND IMPROVE

12.00 PERFOMANCE.

EX- RIDGE REGRESSION AND LASSO

13.00

RESAMPLING -

14.00 • REPEATEDLY DRAW SAMPLES FROM A TRAINING SET AND REFIT

THE MODEL.

15.00 • GAIN MORE INFORMATION

: SEE HOW MODEL WOULD PERFORM ON NEW DATA WITHOUT

16.00 COLLECTING NEW DATA.

17.00 → CROSS VALIDATION IS WIDELY USED METHOD FOR RESAMPLING

→ IT IS USED TO EVALUATE MODEL'S BEST PERFOMANCE AND

18.00 FIND BEST PARAMETERS FOR THE MODEL.

3 APPROACHES TO DO - CROSS VALIDATION

- VALIDATION SET
- LEAVE ONE OUT CROSS VALIDATION
- K- FOLD CROSS VALIDATION

03 '22 May

Week 19 Day (123-242) | Tuesday

MAY 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					

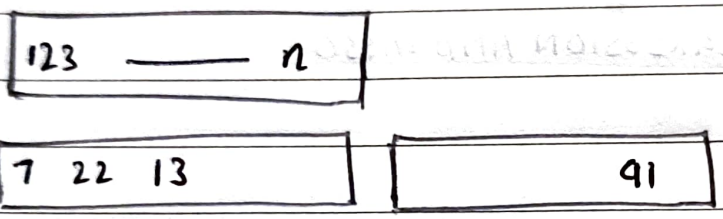
09.00 — VALIDATION SET

- VALIDATION SET IS THE MOST BASIC APPROACH.

10.00 • WE HAVE A DATASET OF 'n' POINTS AND WE RANDOMLY SPLIT DATA SET INTO TRAINING SET AND TEST SET.

11.00 • WE FIT THE MODEL ON TRAINING SET AND MAKE THE PREDICTIONS ON TEST SET.

12.00



13.00

14.00

15.00 SOME DRAWBACKS —

- TEST ERROR RATE IS VARIABLE DEPENDING ON WHICH

16.00 OBSERVATIONS WERE IN THE TRAINING AND VALIDATION SET. SINCE WE ARE SPLITTING RANDOMLY.

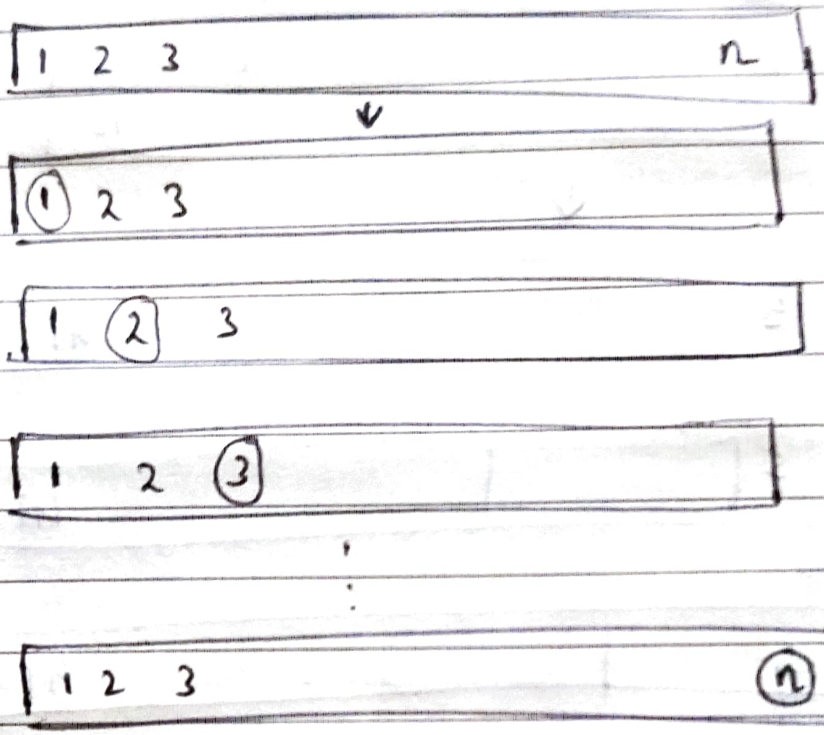
17.00 • ALSO ONLY SMALL SUBSET OF DATA IS USED FOR TRAINING. WHEN IDEALLY WE WANT AS MUCH DATA AS POSSIBLE.

18.00

09.00 LEAVE ONE OUT CROSS VALIDATION (LOOCV)

- ONLY ONE OBSERVATION IS USED FOR VALIDATION.

10.00



11.00

12.00

13.00

14.00

15.00

16.00 IN THIS METHOD, ONLY ONE DATA POINT IN CIRCLE IS USED FOR VALIDATION & REST IS USED FOR TRAINING.

17.00 WE REPEAT THE PROCESS AS MANY TIMES AS WE HAVE DATA POINTS

18.00 - THE ERROR RATE IS APPROXIMATED AS THE AVERAGE OF ERRORS FOR EACH RUN

BENIFIT OF LEAVE ONE OUT CROSS VALIDATION (LOOCV)

- NO RANDOMNESS,

DRAWBACK

- NOT A VIABLE OPTION FOR LARGE DATASETS (IT WILL TAKE A LOT OF TIME)

05 '22 May

Thursday

Week 19 Day (125-240)

MAY 2022

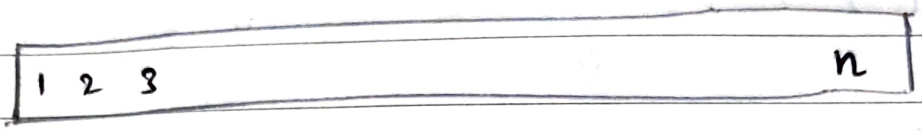
MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					

09.00 K-FOLD CROSS VALIDATION

- RANDOMLY DIVIDE THE DATASET INTO K GROUPS OR FOLDS

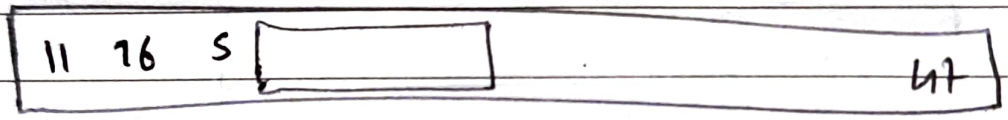
10.00 OF EQUAL SIZE

11.00



12.00

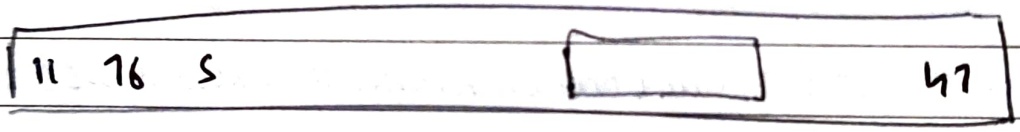
13.00



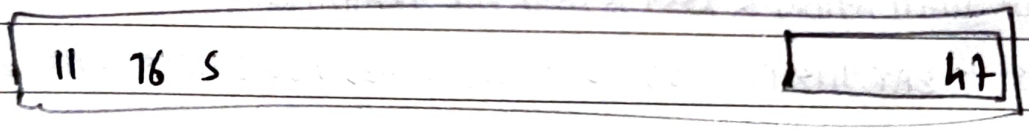
14.00

15.00

16.00



17.00



18.00 WE USE THE COMPLETE BOX FOR TRAINING, AND USE THE SMALL BOX FOR VALIDATION. AND WE REPEAT THE PROCESS 'K' TIMES.

- LOOCV is k-fold with K=n
- Usually set K to 5 or 10
- BEST & MOST WIDELY USED METHOD.

09.00 K-cross validation in PYTHON with LINEAR REGRESSION

10.00
```
lin_reg = LinearRegression()
MSEs = cross_val_score(lin_reg,
                       X,
                       y,
                       scoring =
'neg_mean_squared
_error',
                       cv = 5)
```
11.00

12.00

13.00
```
mean_MSE = np.mean(MSEs)
print(-mean_MSE)
```
14.00

15.00

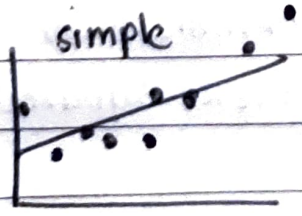
16.00

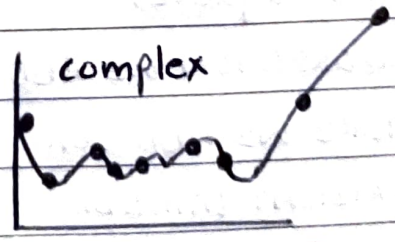
REGULARIZATION

17.00 - MODELS CAN OVERFIT, MEANING THAT THEY WILL NOT GENERATELIZE WELL

18.00 - THEY WILL NOT GENRALIZE WELL AND WILL FORM POORLY ON UNSEEN DATA.

→ BIAS VARIANCE TRADE OFF





HIGH BIAS
LOW VARIANCE
NOT VERY GOOD FIT

HIGH VARIANCE
LOW BIAS
OVERFITTING & VARYING ALOT - CAN'T GIVE GOOD PREDICTIONS

07 '22 May

Saturday

Week 19 Day (127-238)

MAY 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 WE HAVE TO FIND SOMETHING IN BETWEEN TO PREVENT MODELS FROM OVERFITTING.

10.00 AND THAT'S WHEN WE USE REGULARIZATION.

" REGULARIZATION WILL DE-COLERATE OUR MODELS TO

11.00 PREVENT OVERFITTING."

12.00 REGULARIZATION METHODS -

- RIDGE REGRESSION

13.00 - LASSO

- ALSO CALLED - SHRINKAGE METHODS (RIDGE & LASSO)

14.00

RIDGE REGRESSION -

15.00

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

16.00

RSS - Residual Sum of Square

17.00

Sunday 08

18.00

WE KNOW THAT TRADITIONAL LINEAR SETTING MINIMISES RSS (Residual Sum of Square)

WITH RIDGE REGRESSION WE ADD ANOTHER OPTIMIZATION FUNCTION, HERE WE ADD SUM OF PARAMETERS squared WITH A COEFFICIENT LAMBDA.

09.00 - LAMBDA IS CALLED TUNINGPARAMETER.

- TO FIND BEST VALUE OF LAMBDA WE WILL USE CROSS VALIDATION WITH RANGE OF LAMBDAS.

10.00 ★ BEST VALUE WILL BE THE ONE MINIMIZING THE TEST ERROR.

★ WITH THIS METHOD ALL PREDICTORS ARE KEPT.

11.00 ★ ALSO CALLED AS L2 REGULARIZATION.

12.00

- RIDGE REGRESSION IN PYTHON -

13.00

```
ridge = Ridge()
parameters = {'alpha': [1e-15, 1e-10, 1e-8,
1e-4, 1e-3, 1e-2, 1, 5, 10, 20]}
ridge_regressor = GridSearchCV(ridge,
parameters, scoring='neg_mean_squared
_error', cv=5)
ridge_regressor.fit(X, Y);
```

14.00

15.00

16.00

17.00

18.00

IN PYTHON TO USE IT WE FIRST INITIALIZE THE RIDGE MODEL THEN LAMBDA PARAMETER IS ACTUALLY ALPHA, WITH THE LIBRARY THAT WE USE IN PYTHON.

IN SNIPPET WE PROVIDE AN ARRAY WITH DIFFERENT VALUES OF ALPHA.

09.00 THEN WE USE 5 FOLD CROSS VALIDATION TO MAKE USE OF BEST VALUE OF ALPHA FOR OUR MODEL

10.00

LASSO

11.00 - ADD A NEW TERM TO THE OPTIMIZATION FUNCTION.
- LASSO IS ALSO CALLED L1 REGULARIZATION.

12.00

$$RSS + \lambda \sum_{j=1}^{p} |B_j|$$

13.00 HERE WE ADD SUM OF ABSOLUTE VALUES OF COEFFICIENTS.

14.00 λ - OUR TUNING PARAMETER

15.00

16.00 • IF LAMBDA IS LARGE ENOUGH, SOME BETA WILL GO TO ZERO SOME FEATURES WILL DISAPPEAR.

17.00 → THEN FEATURE SELECTION CAN BE DONE.

18.00

```
lasso = Lasso(tol=0.05)
parameters = {'alpha': [1e-15, 1e-10, 1e-8,
1e-4, 1e-3, 1e-2, 1, 5, 10, 20]}
lasso_regressor = GridSearchCV(lasso,
parameters, scoring='neg_mean_squred_
error', cv=5)
lasso_regressor.fit(X,Y)
```

of them.

JUNE 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										

May '22 11

Wednesday

Week 20 Day (131-234)

09.00 RESAMPLING AND REGULARIZATION IN PYTHON -

10.00

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%. matplotlib inline
```

11.00

12.00

13.00

14.00

15.00

16.00

import numpy as np - Library importing for analyses

17.00 import pandas as pd - " " "

import matplotlib.pyplot - pyplot module is collection of

18.00 functions that provide MATLAB like interface for creating plots and charts.

%. matplotlib inline - magic command that set up Jupyter notebook environment to display Matplotlib plot inline in notebook.

- Advertising.csv used

12 '22 May

Thursday

MAY 2022

```
def scatter_plot (feature):
    plt.figure (figsize = (10,5))
    plt.scatter (data [feature], data ['sales'],
                 c='black')
    plt.xlabel (f"Money Spent on [features] ads")
    plt.ylabel ("Sales in $")
    plt.show

scatter_plot ('Tv')
scatter_plot ('radio')
scatter_plot ('newspaper')
```

1) def scatter_plot(feature): This defines a function called scatter_plot that takes a single argument called feature
2) This creates a new figure with size of 10 inches by 5 inches using the figure() function from pyplot module.
3) x values are taken from feature column, y values are taken from sales column, c specifies color of points
4) Add label to the x-axis
5) Add label to the Y-axis
6) plotting the Graph

Define Baseline Model & See How Regularization Improves it

```
from sklearn.model_selection
import cross_val_score                    — 1
from sklearn.linear_model
import LinearRegression                   — 2
```

1) This function is used to perform cross-validation on model. WHICH HELP TO ASSESS HOW MODEL WILL GENERATE TO NEW DATA.

2) USED TO CREATE LINEAR REGRESSION MODEL, WHICH IS TYPE OF SUPERVISED LEARNING Used for predicting continous values.

```
X = data.drop(['sales'],axis=1)           — 1
y = data['sales'].values.                 — 2
    reshape(-1,1)
```

1) creates a new column new dataframe X by removing the column named 'sales' from original data set. This axis=1 argument specifies that the column should be removed. This line assumes the 'sales' column contains target variable or output that model is trying to predict, and remaining columns in data are features or inputs that model will use to make predictions.

14 '22 May

Week 20 Day (134-231) Saturday

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					

09.00 2) The reshape (-1,1) method reshapes the array into a column vector with the same number of rows as

10.00 orignal array. but with only one column.

11.00

```
lin_reg = LinearRegression()                    — 1
MSEs = cross_val_score(lin_reg, X, y,           — 2
scoring='neg_mean_squared_error',
 cv=5)
mean_MSE = np.mean(MSEs)                        — 3
print(-mean_MSE)                                — 4
```

14.00

15.00

16.00

17.00

Sunday 15

1) create an instance of LinearRegression class

18.00 2) uses the cross_val_score function from scikit-learn to perform 5-fold cross validation on LR Model.

3) calculates the mean of negative MSE Values across all folds of cross validation.

4) print negative values of MSE Value.

Regularization

Ridge Regression -

```
From sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
```

GridSearch is scikit-learn that performs exhaustive search over specified parameter grid to find set of hyperparameters.
Ridge is class in scikit-learn that implements Ridge regression Model.

```
ridge = Ridge()
parameters = {'alpha': [1e-15, 1e-8, 1e-10, 1e-4, 1e-3, 1e-2, 1, 5, 10, 20]}

ridge_regressor = GridSearchCV(ridge, parameters, scoring='neg_mean_squared_error', cv=5)

ridge_regressor.fit(X,y);
```

17 '22 May

Week 21 Day (137-228) | Tuesday

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					

09.00 Ridge() - instance of Ridge class

Second line - defines dictionary of hyperparameters to be

10.00 tuned using grid search. We are only tuning the

Hyper parameter from 1e-15 to 20, mix of small &

11.00 Large values.

Create instance of GridSearchCV

12.00 Fits the ridge-regressor object to input data X and target-

variable y.

13.00

Lasso (Regularization)

14.00

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Lasso
```

15.00

16.00

17.00

18.00

```
lasso = Lasso(tol=0.05)
parameters = {'alpha':[1e-15, 1e-10, 1e-8, 1e-4, 1e-3, 1e-2, 1, 5, 10, 20]}
lasso_regressor = GridSearchCV(lasso, parameters, scoring='neg_mean_squared_error', cv=5)
lasso_regressor.fit(X, y)
print(lasso_regressor.best_params_)
print(lasso_regressor.best_score)
```