CROSS VALIDATING A MOTORCARS DATASET

WITH PYTHON

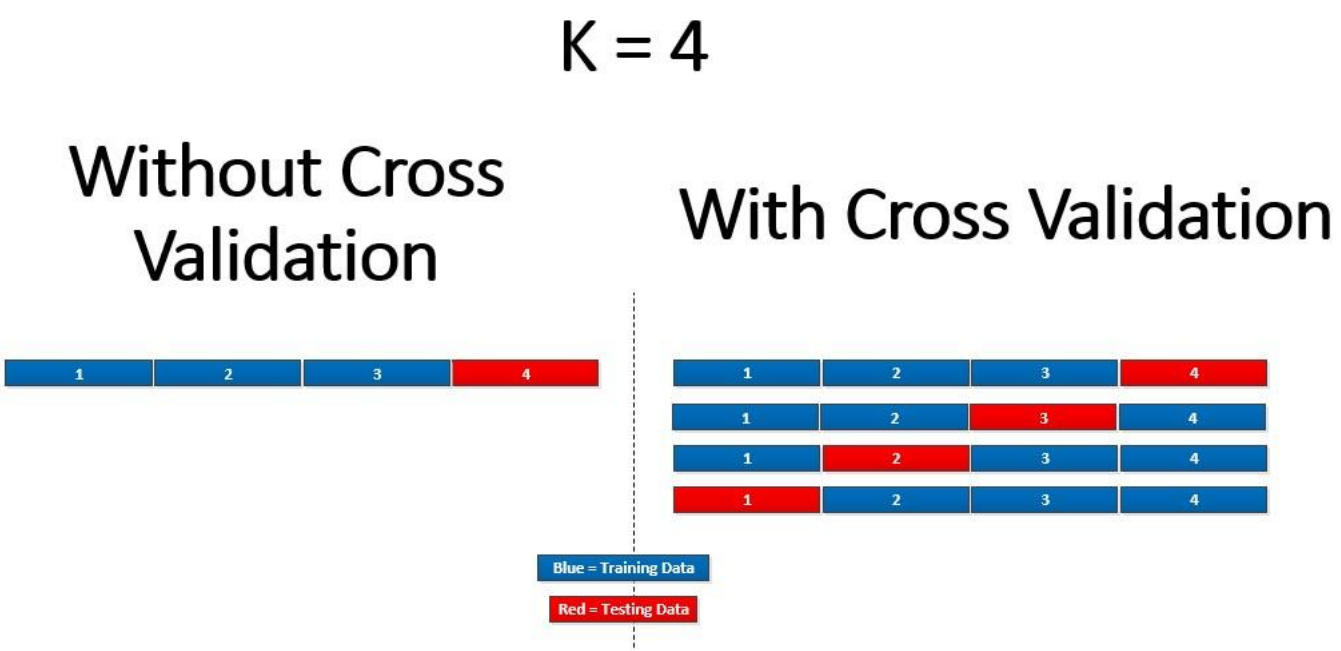
DR. ALVIN ANG

CONTENTS

I. WITHOUT CROSS VALIDATION

https://www.alvinang.sg/s/Cross_Validating_a_European_and_Japanese_Car_Dataset_by_Dr_Alvin_Ang.ipynb

https://www.alvinang.sg/s/cleansed_autocsv.csv



A. STEP 1: READ IN DATASET

1. 1A) IMPORT LIBRARIES

```
WITHOUT CROSS VALIDATION

Step 1: Read in Dataset

1a) Import Libraries

[84] import pandas as pd
     import numpy as np
```

2. IMPORT CSV

```
1b) Import CSV

[ ] df = pd.read_csv('https://www.alvinang.sg/s/cleansed_autocsv.csv')
```

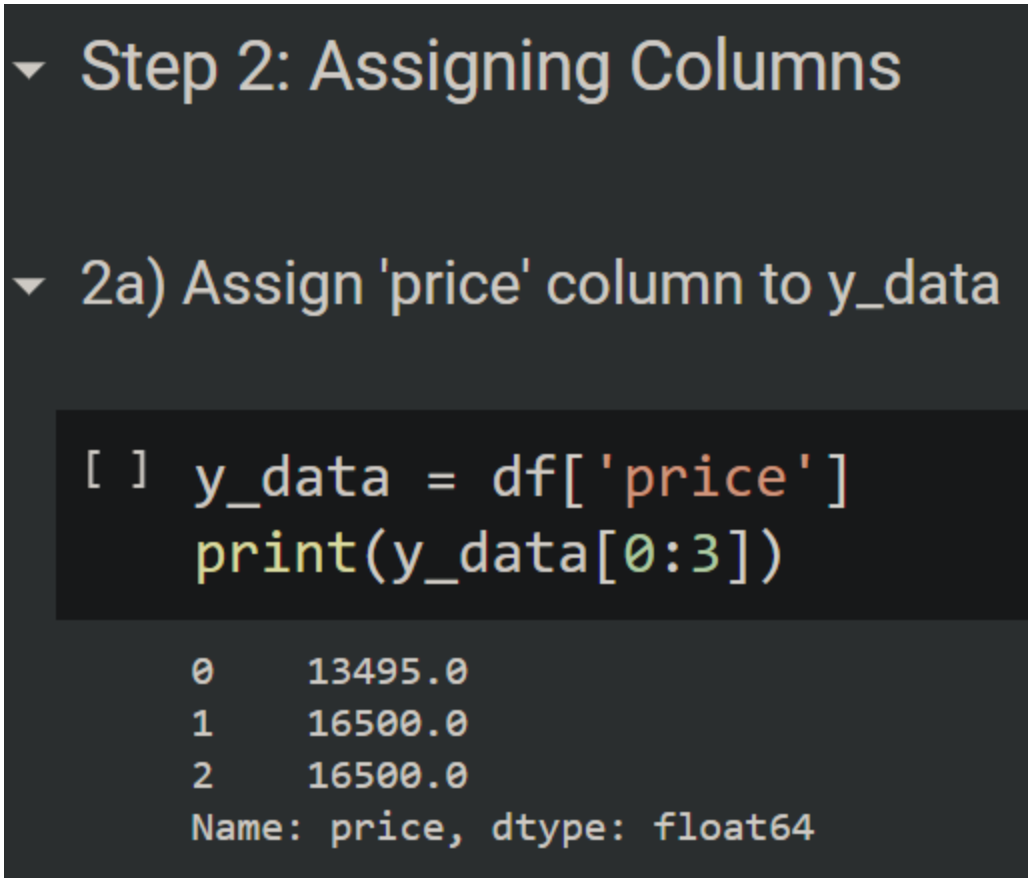
```
[ ] df=df._get_numeric_data()
    df.head()

    #we only want to get the Numeric Data from the dataset
```

	Unnamed: 0	Unnamed: 0.1	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	...	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	city-L/100km	diesel	gas
0	0	0	3	122	88.6	0.811148	0.890278	48.8	2548	130	...	2.68	9.0	111.0	5000.0	21	27	13495.0	11.190476	0	1
1	1	1	3	122	88.6	0.811148	0.890278	48.8	2548	130	...	2.68	9.0	111.0	5000.0	21	27	16500.0	11.190476	0	1
2	2	2	1	122	94.5	0.822681	0.909722	52.4	2823	152	...	3.47	9.0	154.0	5000.0	19	26	16500.0	12.368421	0	1
3	3	3	2	164	99.8	0.848630	0.919444	54.3	2337	109	...	3.40	10.0	102.0	5500.0	24	30	13950.0	9.791667	0	1
4	4	4	2	164	99.4	0.848630	0.922222	54.3	2824	136	...	3.40	8.0	115.0	5500.0	18	22	17450.0	13.055556	0	1

5 rows × 21 columns

B. STEP 2: ASSIGNING COLUMNS

1. ASSIGN 'PRICE' COLUMN TO Y_DATA



2. ASSIGNING ALL OTHER COLUMNS TO X_DATA

2b) Assigning all other columns to x_data

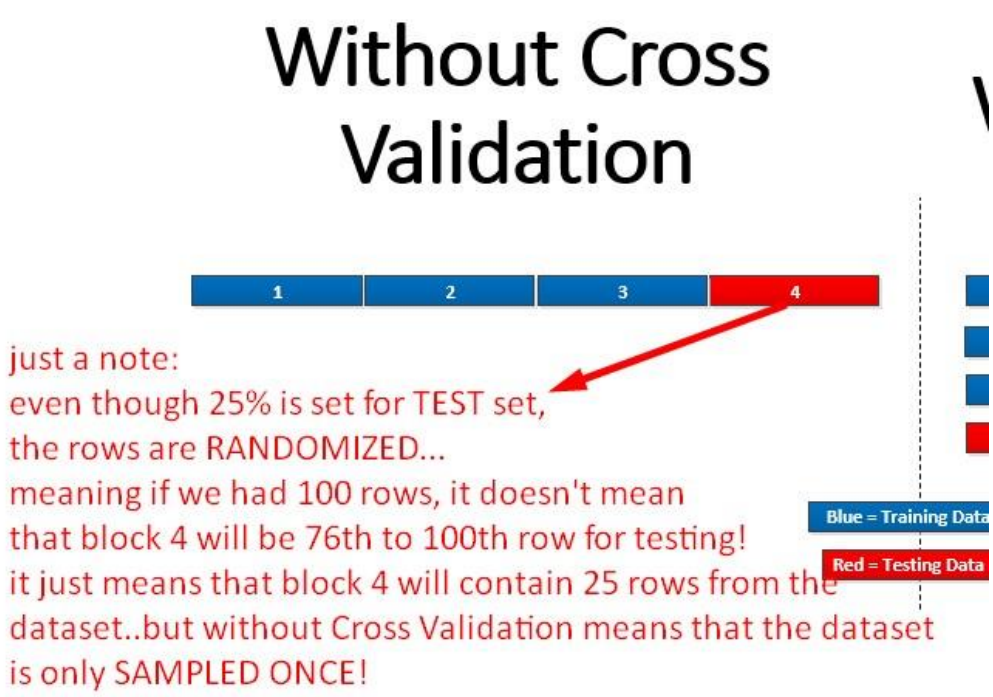
```
x_data=df.drop('price',axis=1)
print(x_data[0:3])
```

```
   Unnamed: 0  Unnamed: 0.1  symboling  normalized-losses  wheel-base  \
0           0             0          3                122        88.6
1           1             1          3                122        88.6
2           2             2          1                122        94.5

     length     width  height  curb-weight  engine-size  bore  stroke  \
0  0.811148  0.890278    48.8         2548          130  3.47    2.68
1  0.811148  0.890278    48.8         2548          130  3.47    2.68
2  0.822681  0.909722    52.4         2823          152  2.68    3.47

   compression-ratio  horsepower  peak-rpm  city-mpg  highway-mpg  \
0                9.0       111.0    5000.0        21           27
1                9.0       111.0    5000.0        21           27
2                9.0       154.0    5000.0        19           26

   city-L/100km  diesel  gas
0     11.190476       0    1
1     11.190476       0    1
2     12.368421       0    1
```

C. STEP 3: TRAIN TEST SPLIT

Step 3: Train Test Split

```
[ ] from sklearn.model_selection import train_test_split
    x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.25, random_state=1)

    #Test size = 25%
    #Train size = 75%
    #Random State is a seed for random dataset splitting.
```



D. STEP 4: USING LINEAR REGRESSION

```
Step 4: Using Linear Regression

[ ] from sklearn.linear_model import LinearRegression
    lre=LinearRegression()
```

1. FITTING LINEAR REGRESSION MODEL TO TRAIN DATASET

```
4a) Fitting Linear Regression Model to Train Dataset

[ ] lre.fit(x_train[['horsepower']], y_train)

    #we fit a linear model to 'price' vs 'horsepower'
    #(of the TRAIN dataset)

    LinearRegression()
```

2. GETTING R2 SCORE

```
4b) Getting R2 score

[ ] lre.score(x_test[['horsepower']], y_test)

    #great fit!
    #linear regression fits 63% of 'price' to 'horsepower'
    #on the TEST dataset!

    0.6296166860021476
```

3. DO A PREDICTION

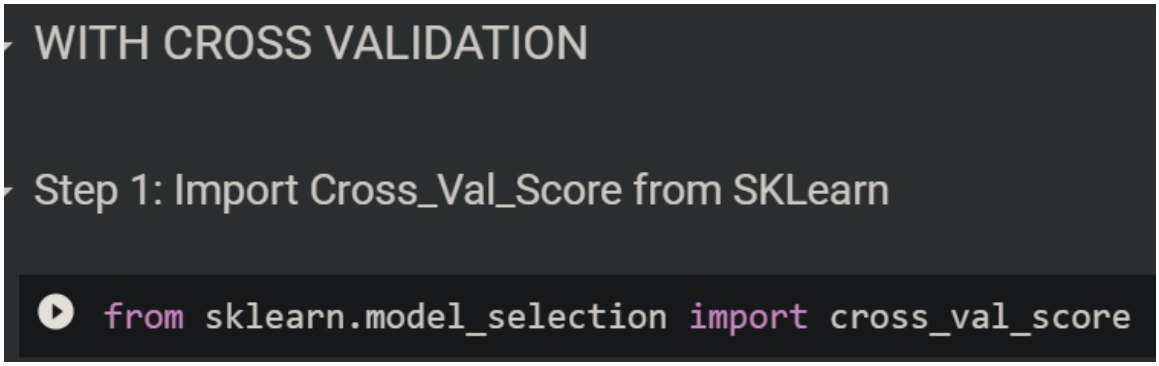
4c) Do a Prediction

```
[ ] yhat = lre.predict(x_test[['horsepower']])
    yhat[0:3]

    #just for fun, we use the LR model we created
    #to predict and preview the first 3 'pricing' rows
    #(labelled as yhat)
```

```
array([12159.57270173,  7182.0124868 ,  9928.25260538])
```

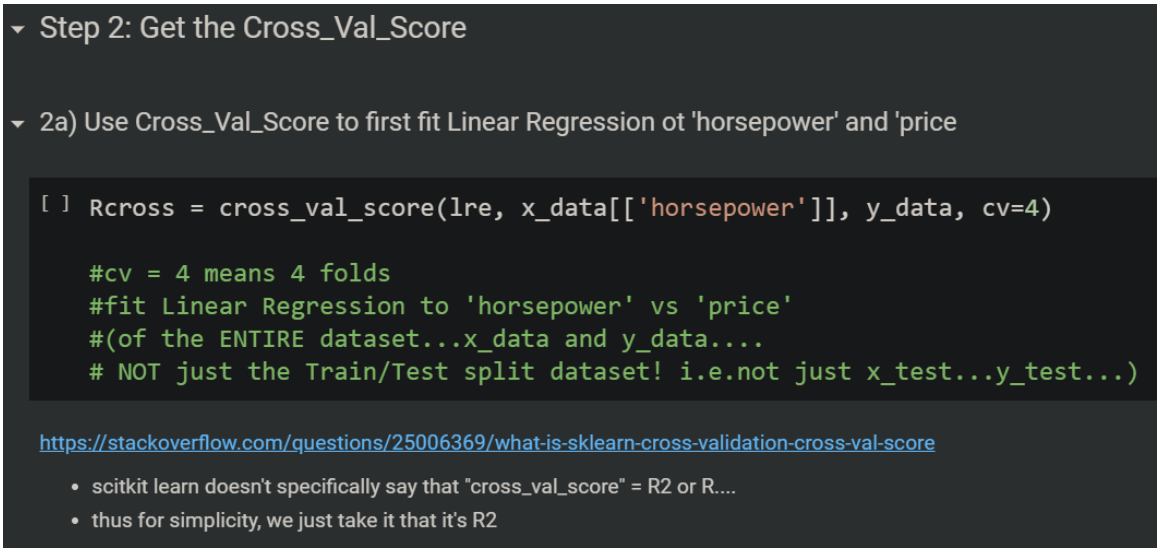
II. WITH CROSS VALIDAION

A. STEP 1: IMPORT CROSS_VAL_SCORE

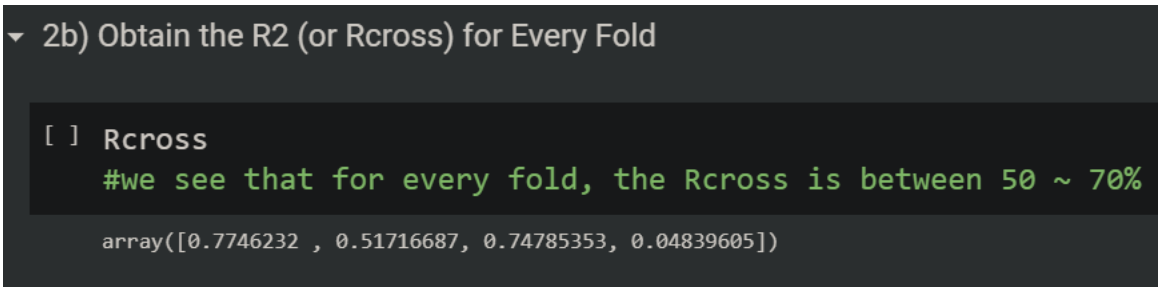


B. STEP 2: GET THE CROSS_VAL_SCORE

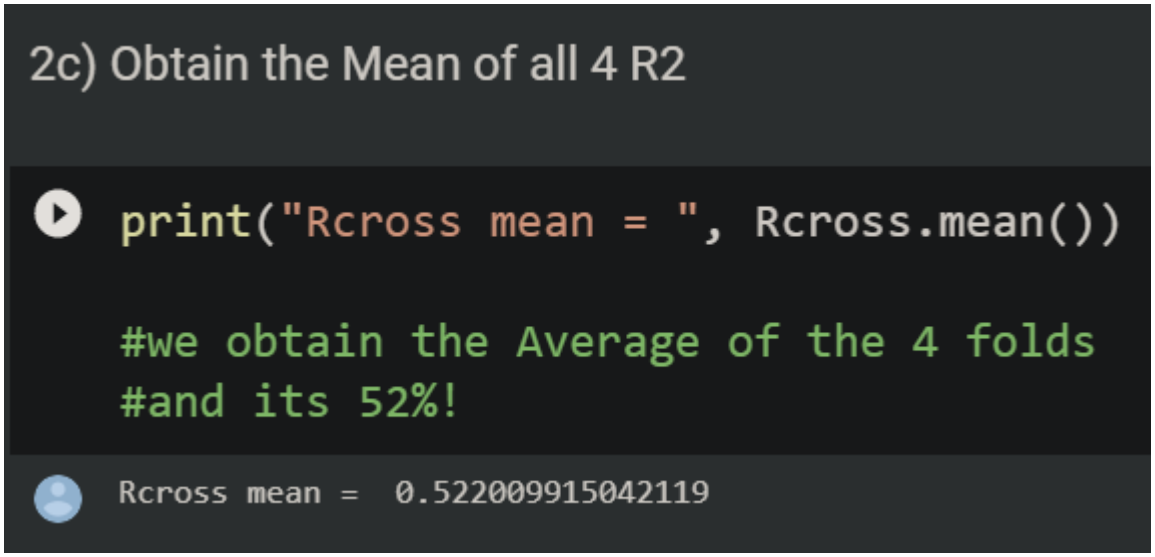
1. USE CROSS_VAL_SCORE TO FIRST FIT LINEAR REGRESSION OF 'HORSEPOWER' AND 'PRICE'



2. OBTAIN THE R2(OR RCROSS) FOR EVERY FOLD

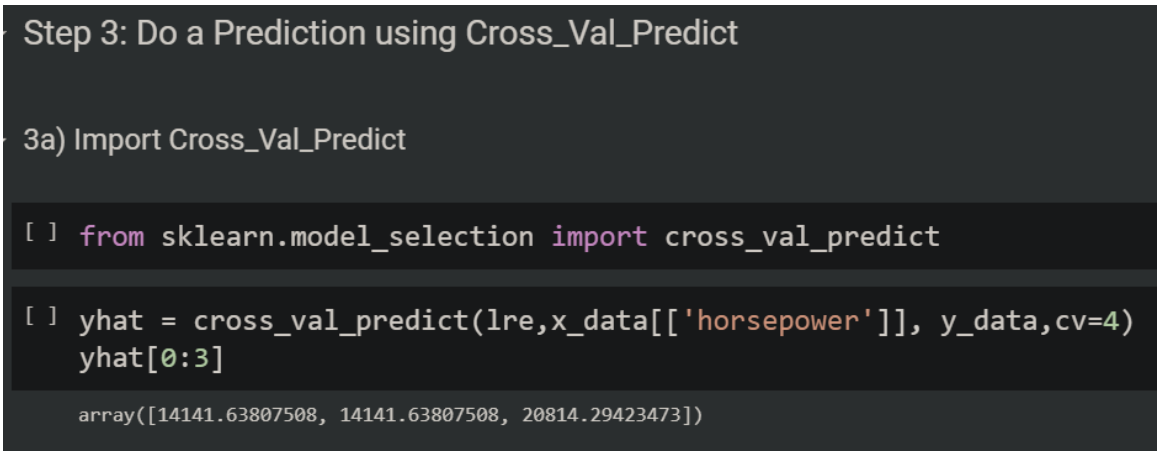


3. OBTAIN THE MEAN OF ALL 4 R2



C. STEP 3: DO A PREDICTION USING CROSS_VAL-PREDICT

1. IMPORT CROSS_VAL_PREDICT



Comparing WITHOUT vs WITH Cross Validation

Without Cross Validation

- R2 = 63%
- We might be fooled that Linear Regression fits the dataset well (63% fitting!)

- Y1 = $12,159
- Y2 = $7,182
- Y3 = $9,928

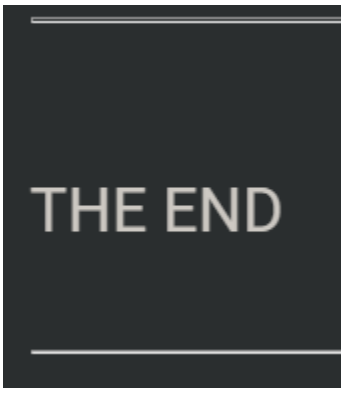
Purpose of Cross Validation = Highlight signs of OVERFITTING

With Cross Validation

- R2 = 52% (only fits at 52%!)

- Y1 = $14,141
- Y2 = $14,141
- Y3 = $20,814

- The 3 sample predictions show how far off the predictions can get
- The values from "Without Cross Validation" bluffs us that the model works well.
- The values from With Cross Validation shows us the true stuff → Model is not working well with predictions quite far off.



IV. ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.