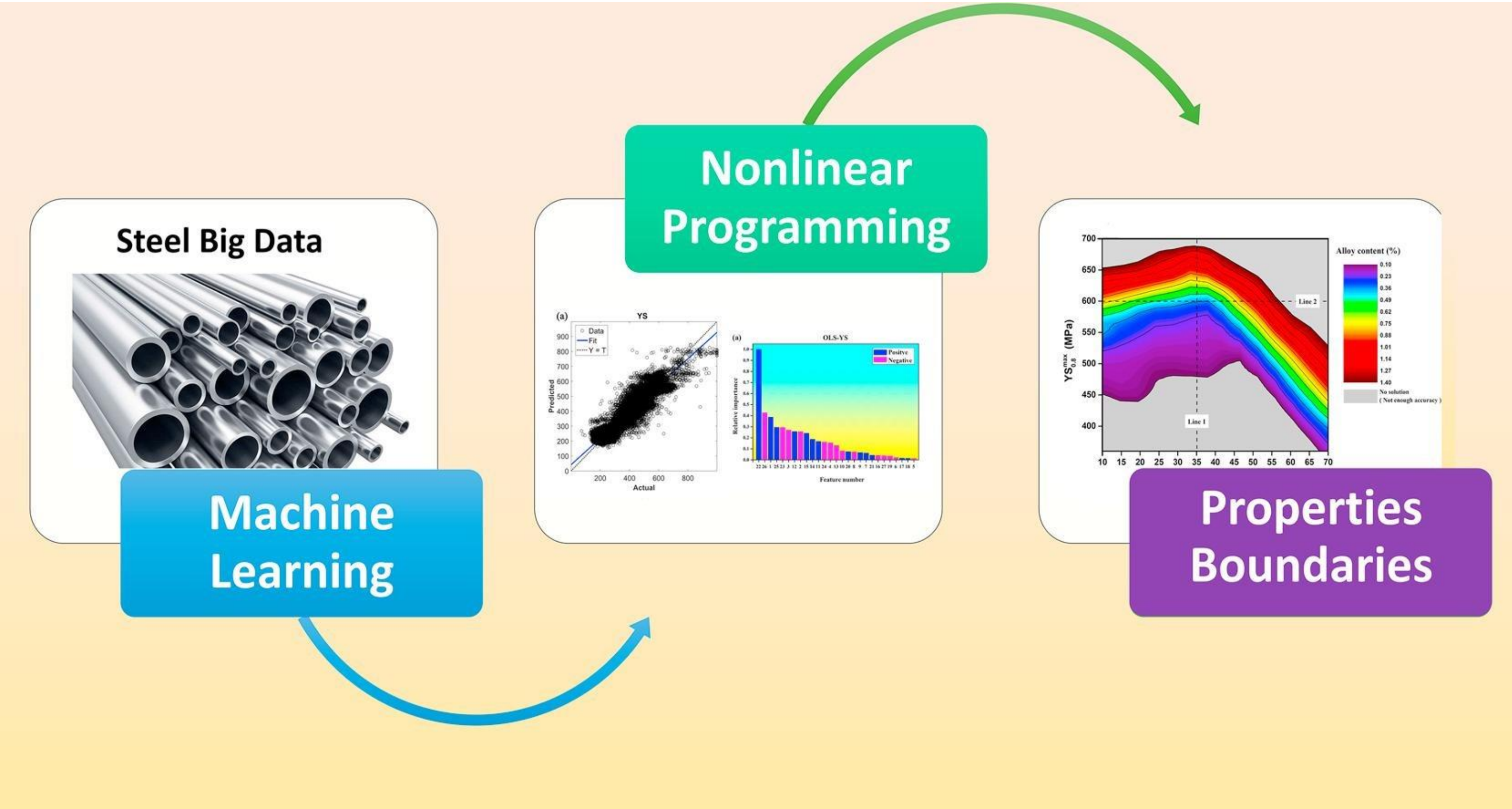
MSE1065: Lab 3

Case study #1: Processing-property relations for steel using machine learning



Steel Big Data

Machine Learning

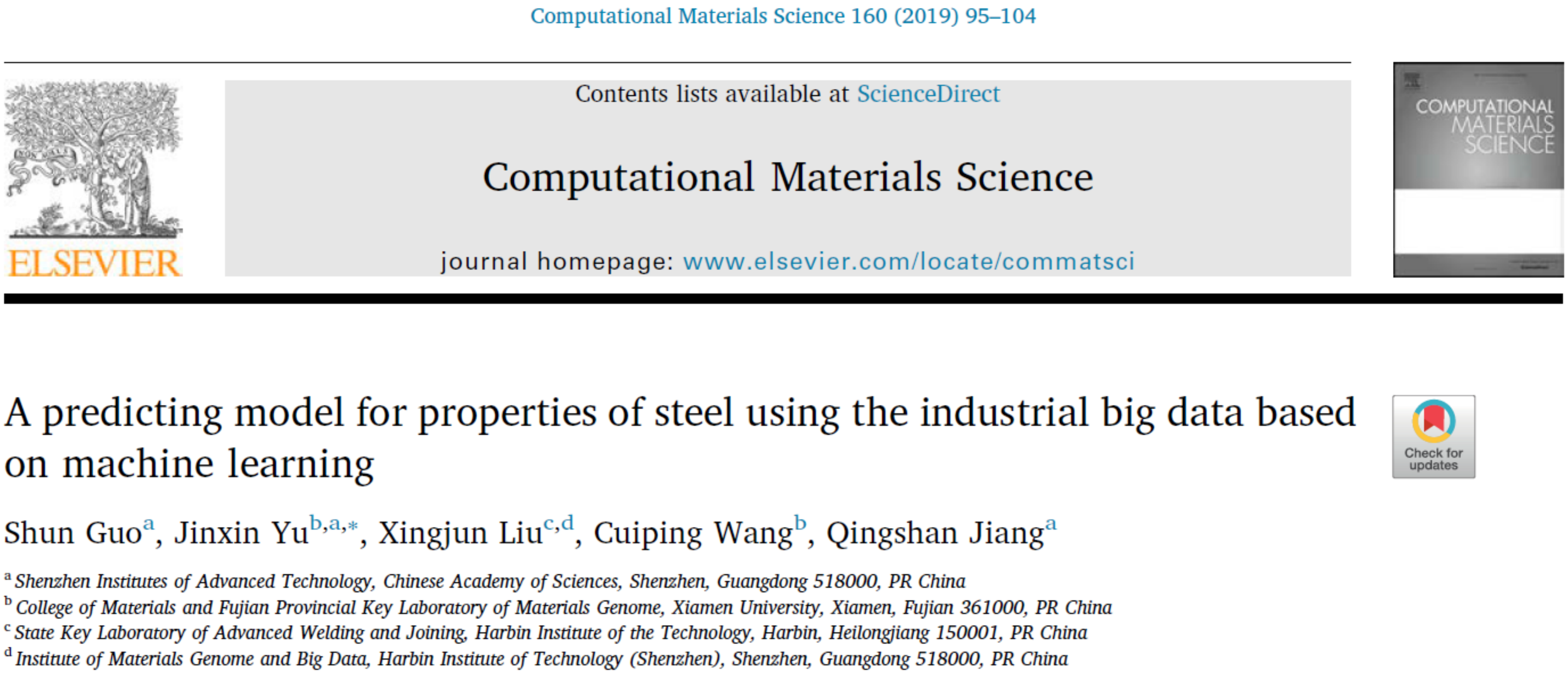
Nonlinear Programming

Properties Boundaries

Learning objectives for this lab

Utilize ML based regression analysis to develop processing-property relationships for steel

Source



Computational Materials Science 160 (2019) 95–104

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

A predicting model for properties of steel using the industrial big data based on machine learning

Shun Guo[a], Jinxin Yu[b,a,*], Xingjun Liu[c,d], Cuiping Wang[b], Qingshan Jiang[a]

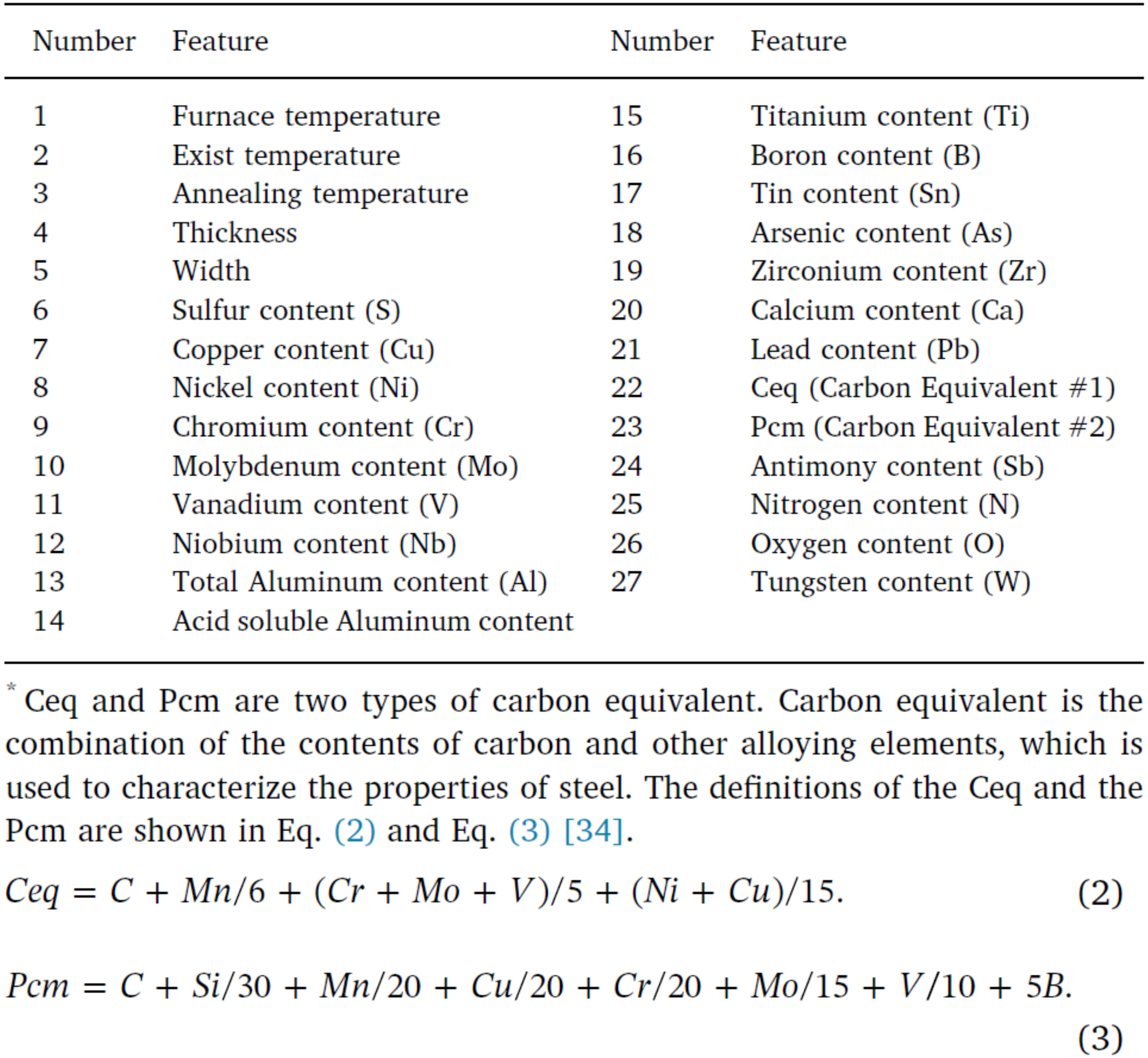
[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518000, PR China
[b] College of Materials and Fujian Provincial Key Laboratory of Materials Genome, Xiamen University, Xiamen, Fujian 361000, PR China
[c] State Key Laboratory of Advanced Welding and Joining, Harbin Institute of the Technology, Harbin, Heilongjiang 150001, PR China
[d] Institute of Materials Genome and Big Data, Harbin Institute of Technology (Shenzhen), Shenzhen, Guangdong 518000, PR China

- Data availability: https://data.mendeley.com/datasets/msf6jzm52g/1
  - **DOI:** 10.17632/msf6jzm52g.1

Dataset

- Steel production data, collected by the Shanghai Meishan Iron and Steel Corporation Ltd. Of Bao Steel Group.
- Original data: 65,288 samples, while
- Processed data: 63,137 samples
  - 27 influence factors (features), including process parameters and chemical compositions
  - 3 Properties: yield strength (YS), the tensile strength (TS), and the elongation (EL) (plasticity)

Feature engineering



Number	Feature	Number	Feature
1	Furnace temperature	15	Titanium content (Ti)
2	Exist temperature	16	Boron content (B)
3	Annealing temperature	17	Tin content (Sn)
4	Thickness	18	Arsenic content (As)
5	Width	19	Zirconium content (Zr)
6	Sulfur content (S)	20	Calcium content (Ca)
7	Copper content (Cu)	21	Lead content (Pb)
8	Nickel content (Ni)	22	Ceq (Carbon Equivalent #1)
9	Chromium content (Cr)	23	Pcm (Carbon Equivalent #2)
10	Molybdenum content (Mo)	24	Antimony content (Sb)
11	Vanadium content (V)	25	Nitrogen content (N)
12	Niobium content (Nb)	26	Oxygen content (O)
13	Total Aluminum content (Al)	27	Tungsten content (W)
14	Acid soluble Aluminum content		

* Ceq and Pcm are two types of carbon equivalent. Carbon equivalent is the combination of the contents of carbon and other alloying elements, which is used to characterize the properties of steel. The definitions of the Ceq and the Pcm are shown in Eq. (2) and Eq. (3) [34].

$$Ceq = C + Mn/6 + (Cr + Mo + V)/5 + (Ni + Cu)/15. \tag{2}$$

$$Pcm = C + Si/30 + Mn/20 + Cu/20 + Cr/20 + Mo/15 + V/10 + 5B. \tag{3}$$

Performance of different ML models (for YS)

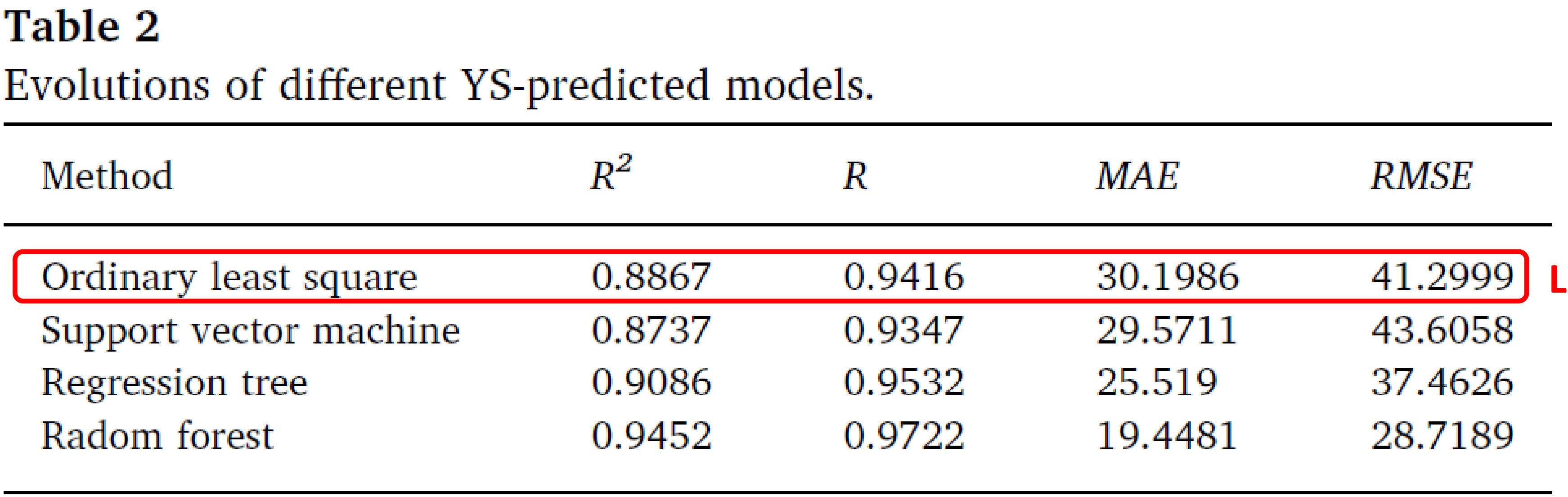


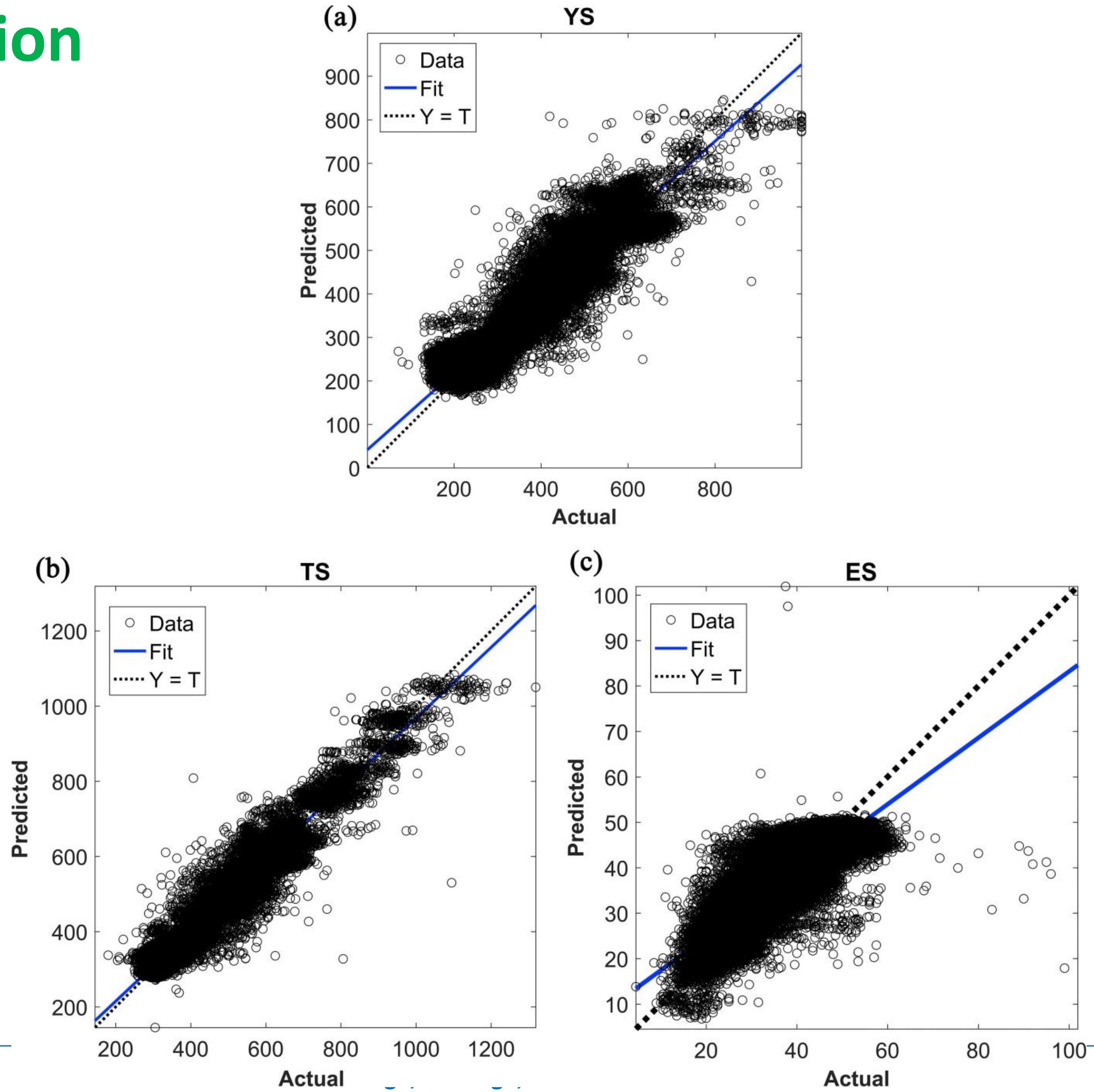
Table 2

Evolutions of different YS-predicted models.

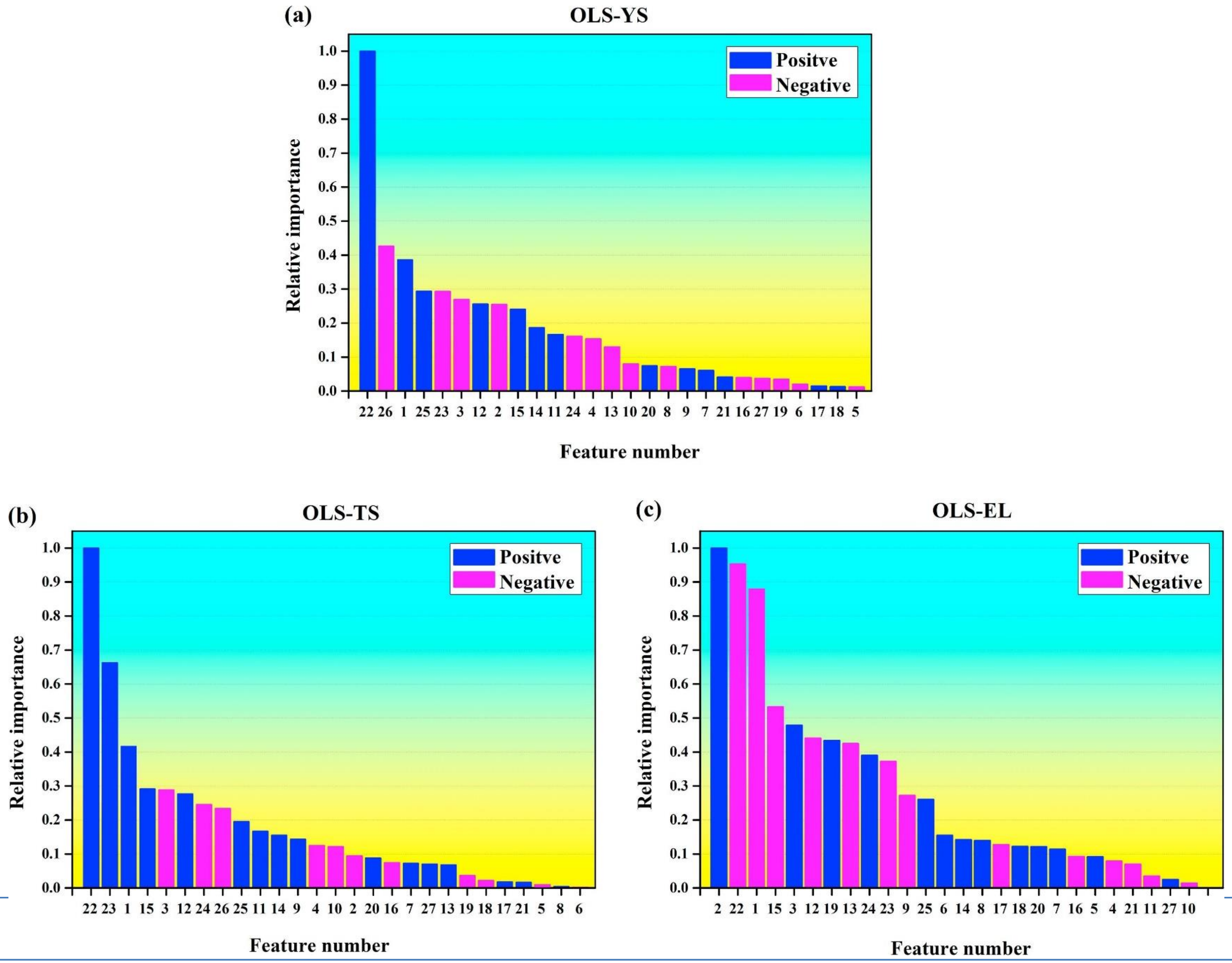
Method	R^2	R	MAE	$RMSE$
Ordinary least square	0.8867	0.9416	30.1986	41.2999
Support vector machine	0.8737	0.9347	29.5711	43.6058
Regression tree	0.9086	0.9532	25.519	37.4626
Radom forest	0.9452	0.9722	19.4481	28.7189

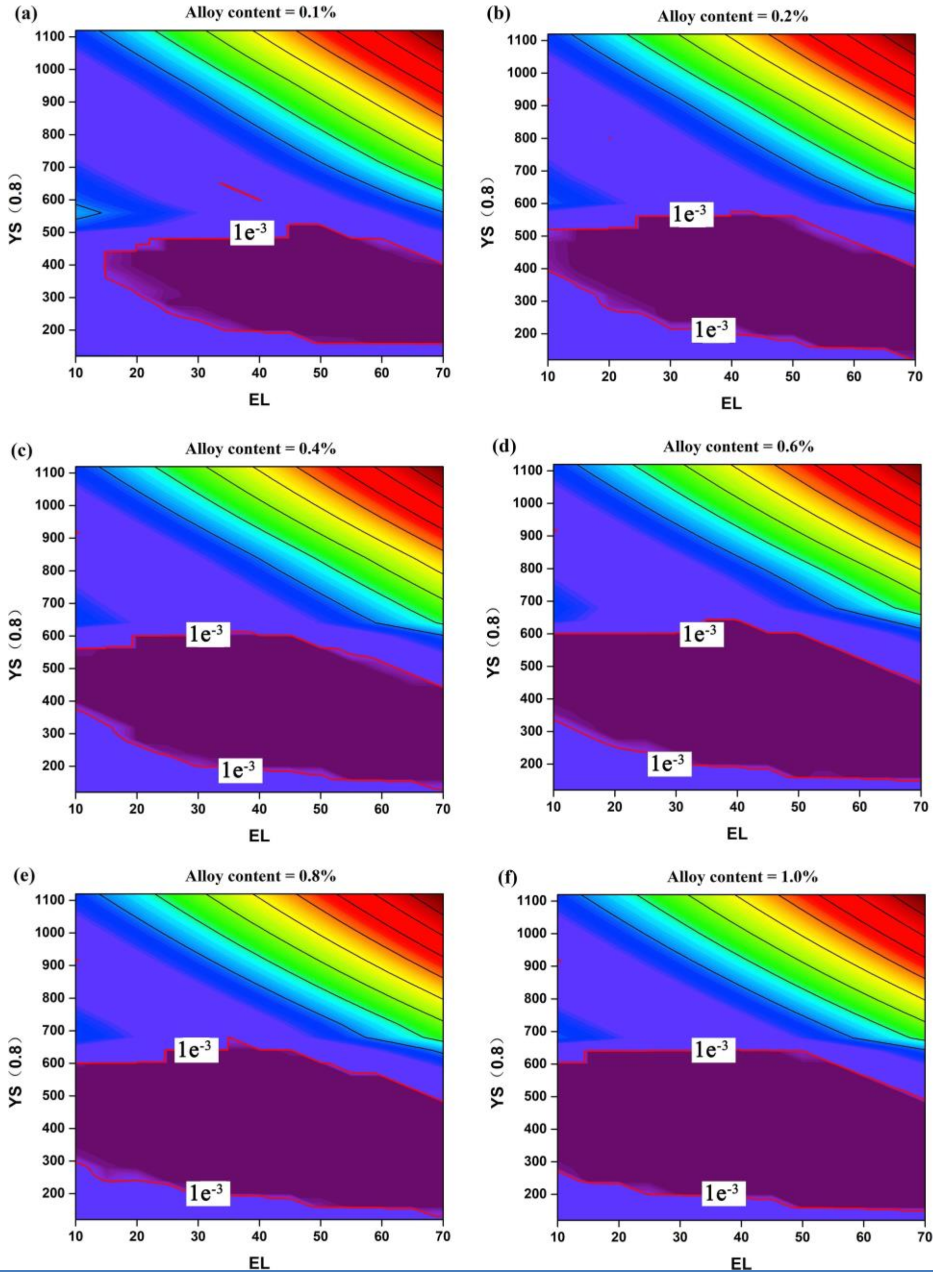
Lab 3

Cross-validation plots



Relative feature importance

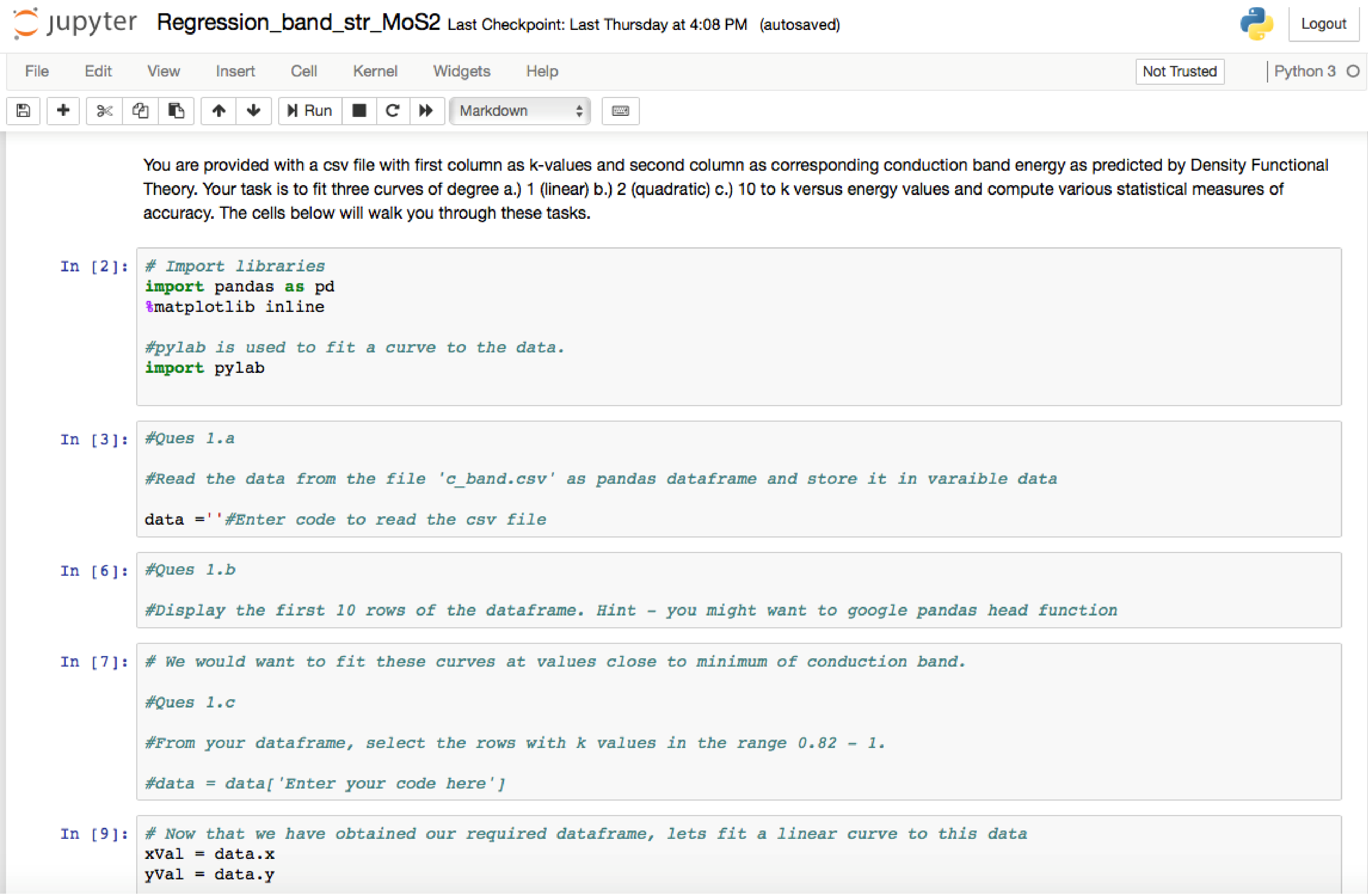



(a)
Alloy content = 0.1%
(b)
Alloy content = 0.2%
(c)
Alloy content = 0.4%
(d)
Alloy content = 0.6%
(e)
Alloy content = 0.8%
(f)
Alloy content = 1.0%
YS（0.8）
EL
1e-3
1100
1000
900
800
700
600
500
400
300
200
10
20
30
40
50
60
70

What will you accomplish in the lab?

- Under this lab, linear regression-based models will be applied for two datasets.
- We will wet our feet by applying simple curve fitting models on 1-D problem.
  - Data is Energy versus k-values for MoS2
  - The goal of this exercise is to make you comfortable loading various ML libraries, data visualization, basic curve fitting and model analysis.
  - We will also introduce different error metrics that will help you to assess different models and choose the one appropriately.

What will you accomplish in the lab?



Regression_band_str_MoS2 Last Checkpoint: Last Thursday at 4:08 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

You are provided with a csv file with first column as k-values and second column as corresponding conduction band energy as predicted by Density Functional Theory. Your task is to fit three curves of degree a.) 1 (linear) b.) 2 (quadratic) c.) 10 to k versus energy values and compute various statistical measures of accuracy. The cells below will walk you through these tasks.

In [2]:
```
# Import libraries
import pandas as pd
%matplotlib inline

#pylab is used to fit a curve to the data.
import pylab
```

In [3]:
```
#Ques 1.a

#Read the data from the file 'c_band.csv' as pandas dataframe and store it in varaible data

data =''#Enter code to read the csv file
```

In [6]:
```
#Ques 1.b

#Display the first 10 rows of the dataframe. Hint - you might want to google pandas head function
```

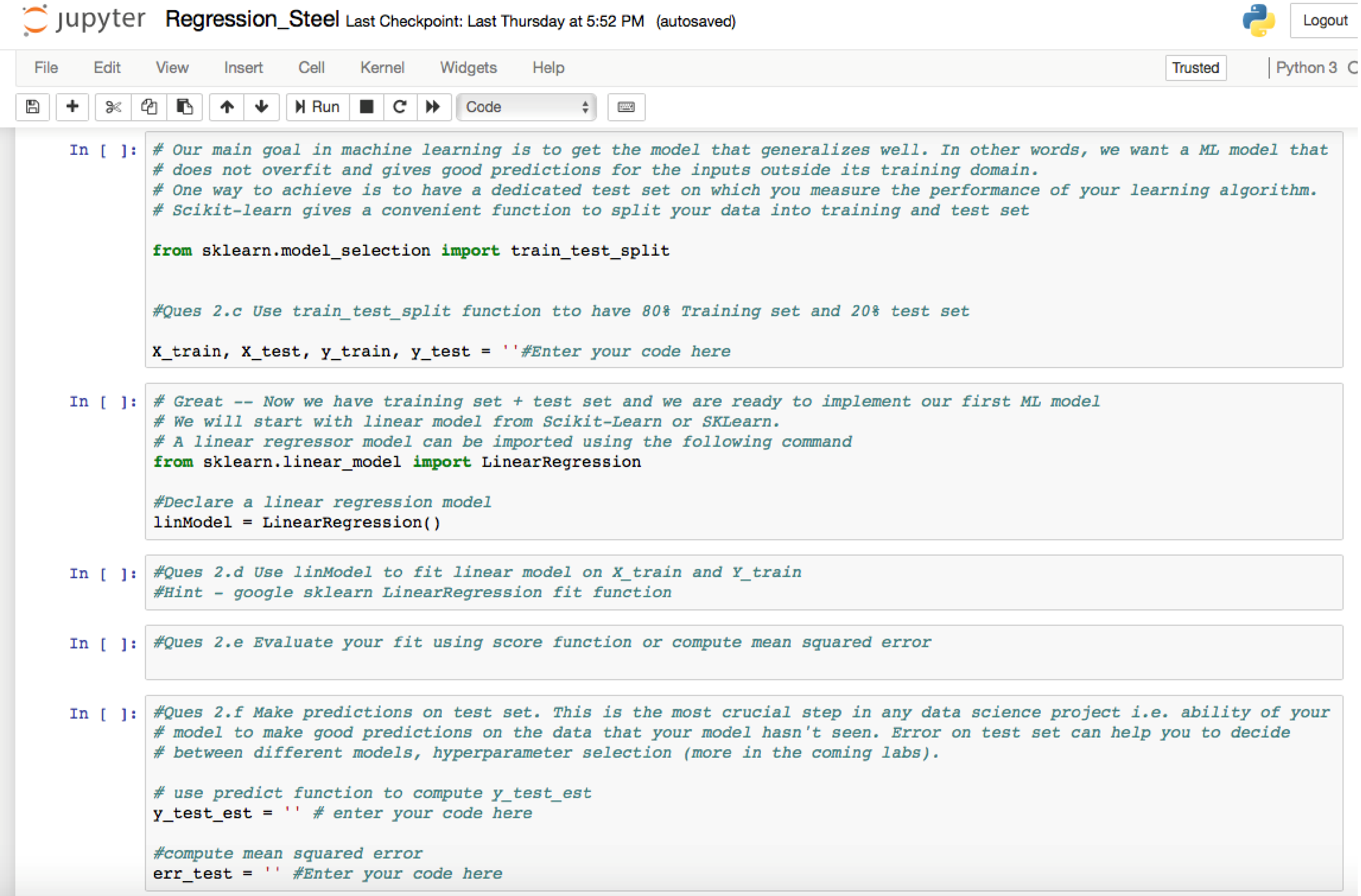
In [7]:
```
# We would want to fit these curves at values close to minimum of conduction band.

#Ques 1.c

#From your dataframe, select the rows with k values in the range 0.82 - 1.

#data = data['Enter your code here']
```

In [9]:
```
# Now that we have obtained our required dataframe, lets fit a linear curve to this data
xVal = data.x
yVal = data.y
```

What will you accomplish in the lab?

- Once finished with the 1-D problem, we will start with multi-dimensional data as described in the paper.
- We will learn how to use different ML libraries to perform tasks such as
  - Data Splitting
  - Linear regression
  - Assessing the model performance through various error metrics.

What will you accomplish in the lab?



```
In [ ]: # Our main goal in machine learning is to get the model that generalizes well. In other words, we want a ML model that
        # does not overfit and gives good predictions for the inputs outside its training domain.
        # One way to achieve is to have a dedicated test set on which you measure the performance of your learning algorithm.
        # Scikit-learn gives a convenient function to split your data into training and test set

        from sklearn.model_selection import train_test_split


        #Ques 2.c Use train_test_split function tto have 80% Training set and 20% test set

        X_train, X_test, y_train, y_test = ''#Enter your code here
```

```
In [ ]: # Great -- Now we have training set + test set and we are ready to implement our first ML model
        # We will start with linear model from Scikit-Learn or SKLearn.
        # A linear regressor model can be imported using the following command
        from sklearn.linear_model import LinearRegression

        #Declare a linear regression model
        linModel = LinearRegression()
```

```
In [ ]: #Ques 2.d Use linModel to fit linear model on X_train and Y_train
        #Hint - google sklearn LinearRegression fit function
```

```
In [ ]: #Ques 2.e Evaluate your fit using score function or compute mean squared error
```

```
In [ ]: #Ques 2.f Make predictions on test set. This is the most crucial step in any data science project i.e. ability of your
        # model to make good predictions on the data that your model hasn't seen. Error on test set can help you to decide
        # between different models, hyperparameter selection (more in the coming labs).

        # use predict function to compute y_test_est
        y_test_est = '' # enter your code here

        #compute mean squared error
        err_test = '' #Enter your code here
```