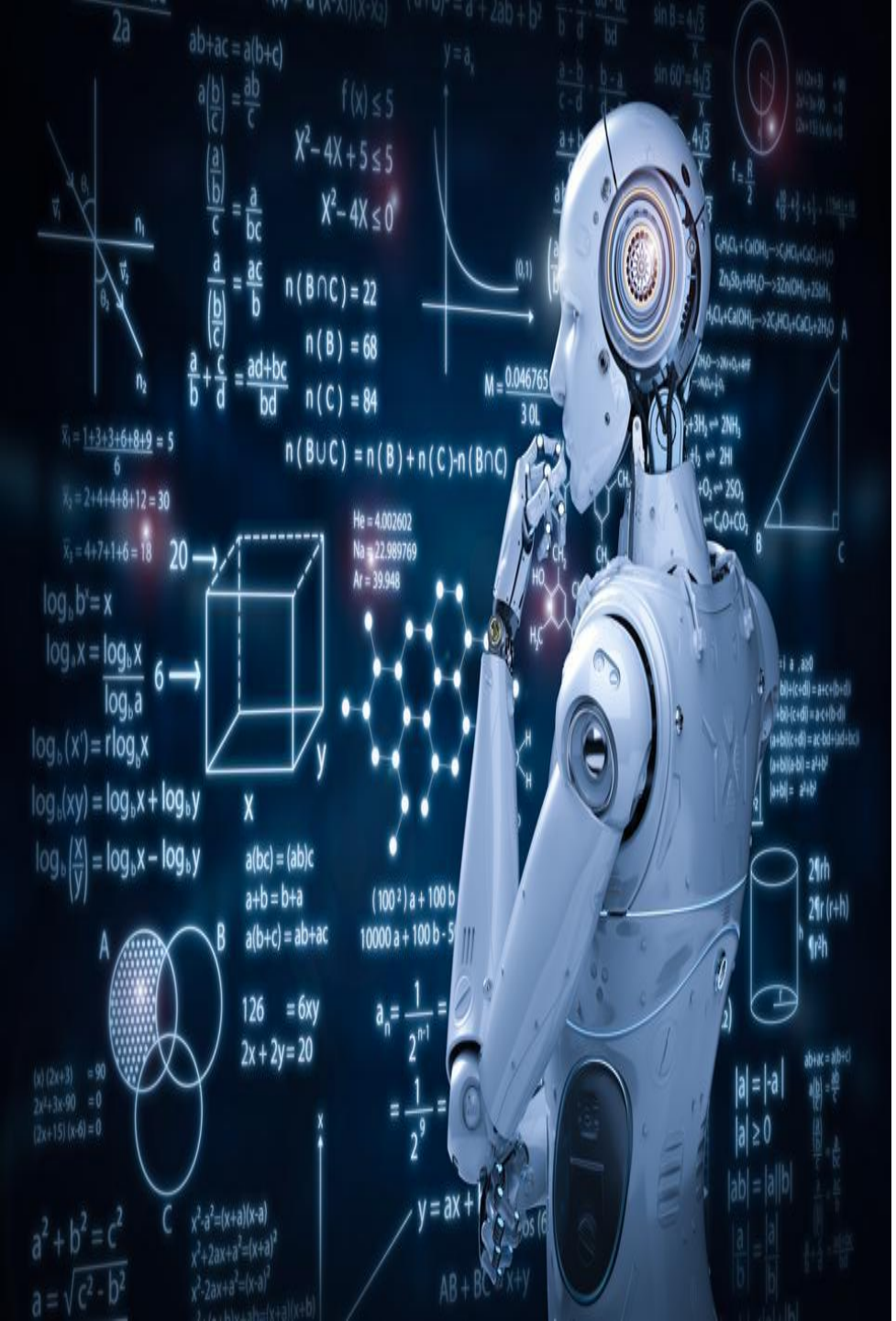


Practical Machine Learning

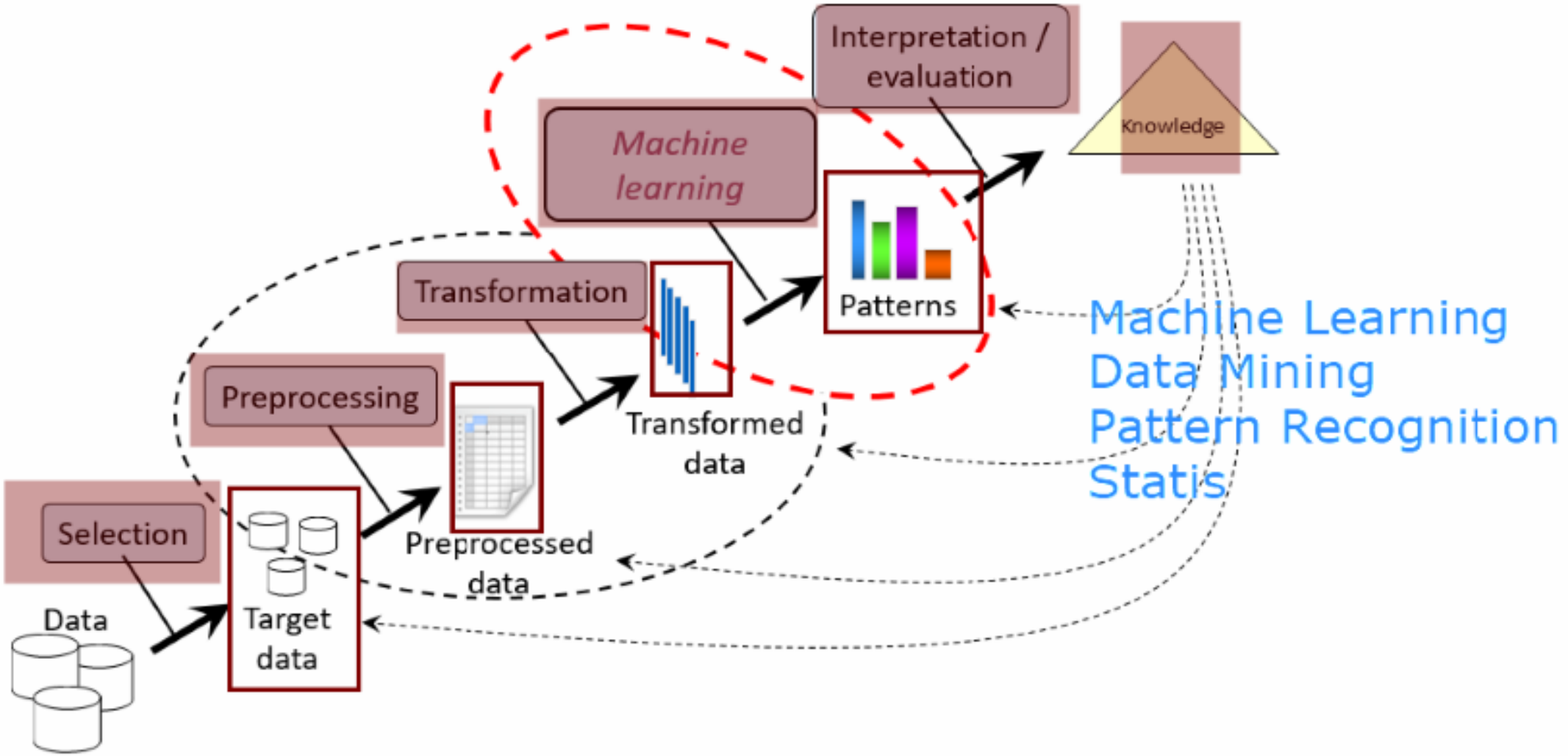
Day 4: Mar22 DBDA

Kiran Waghmare

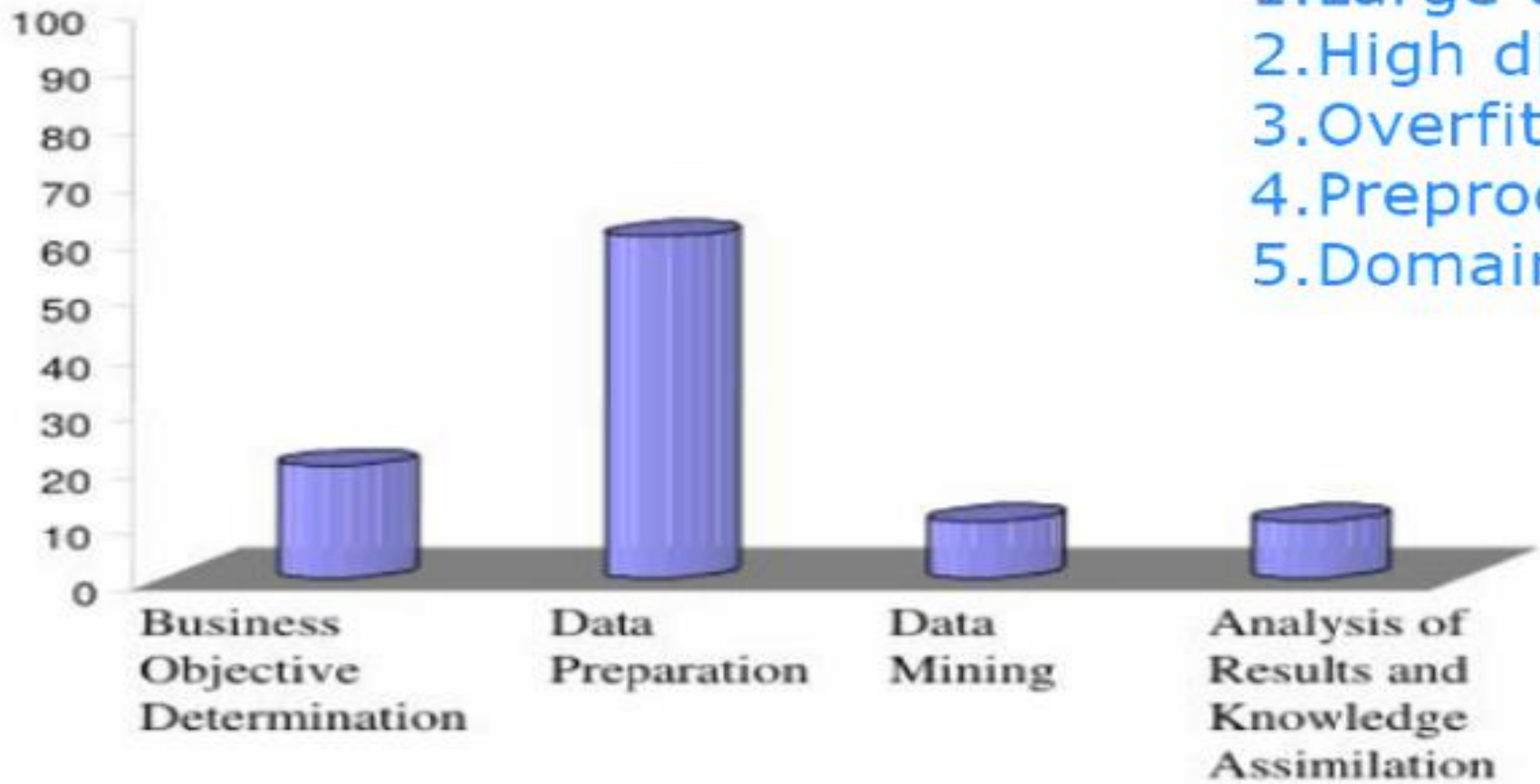
Agenda

- Knowledge Extraction
- Regression Analysis

Stages of knowledge extraction

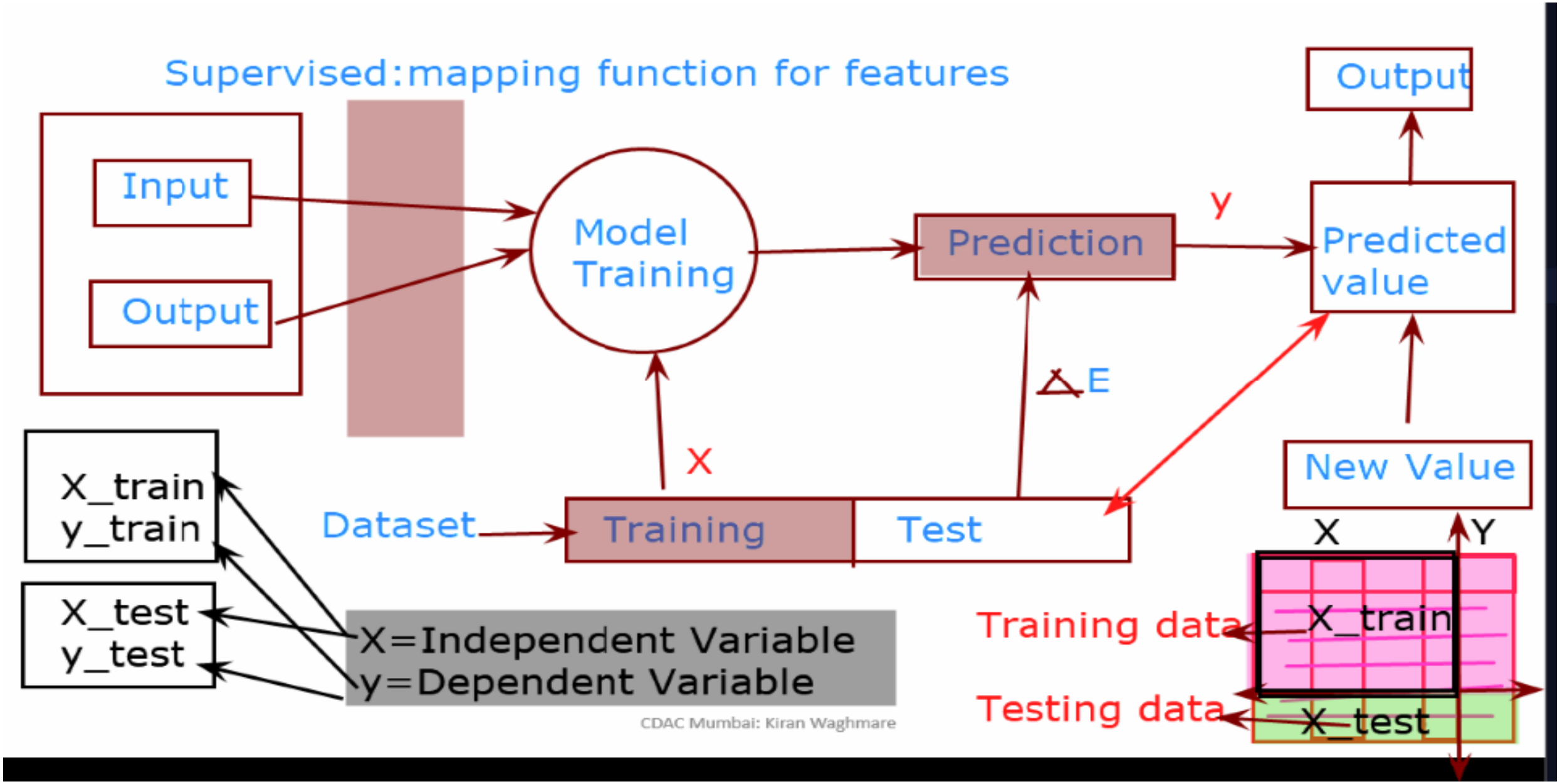


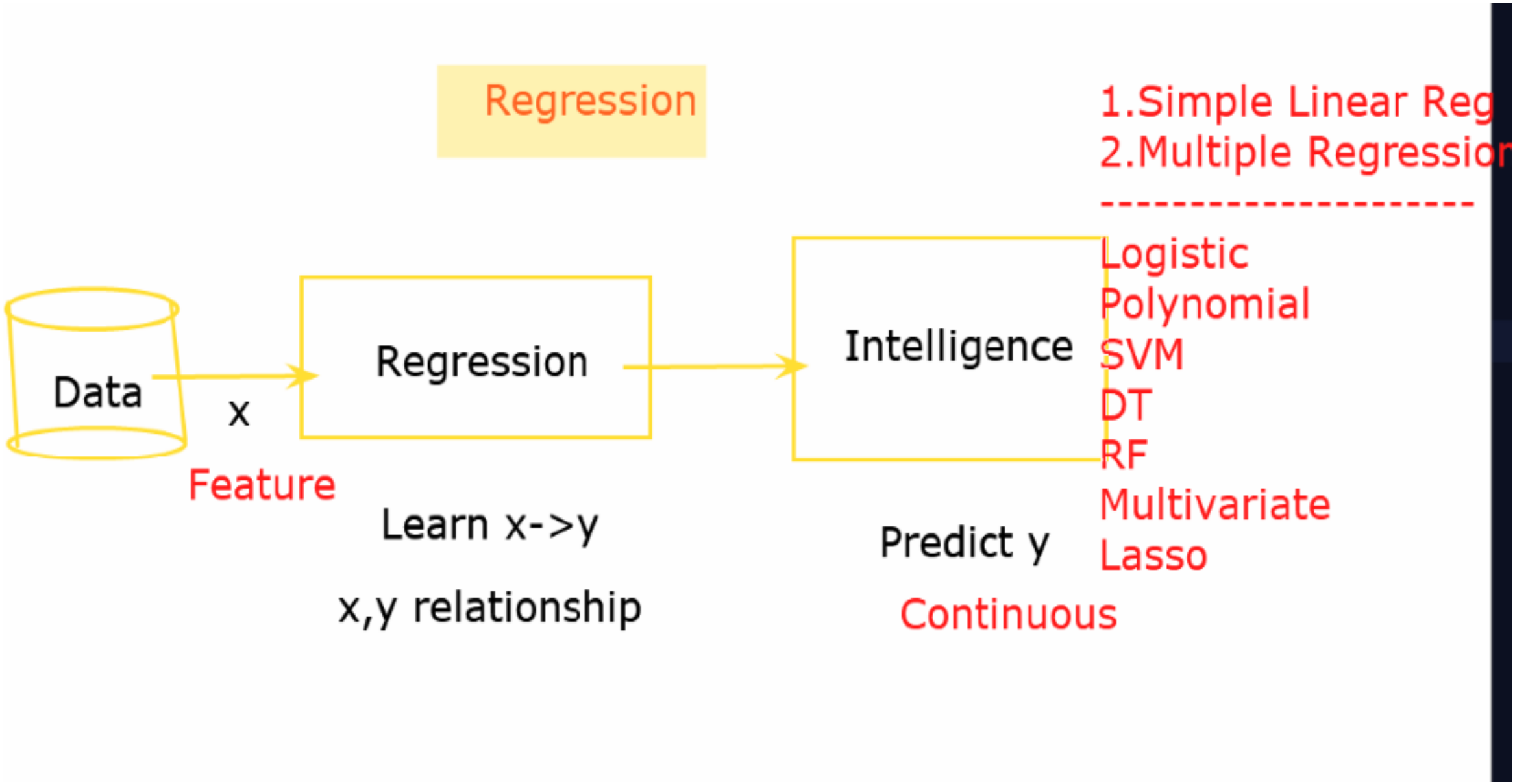
1.Large data
2.High dimensional
3.Overfitting
4.Preprocessing
5.Domain knw

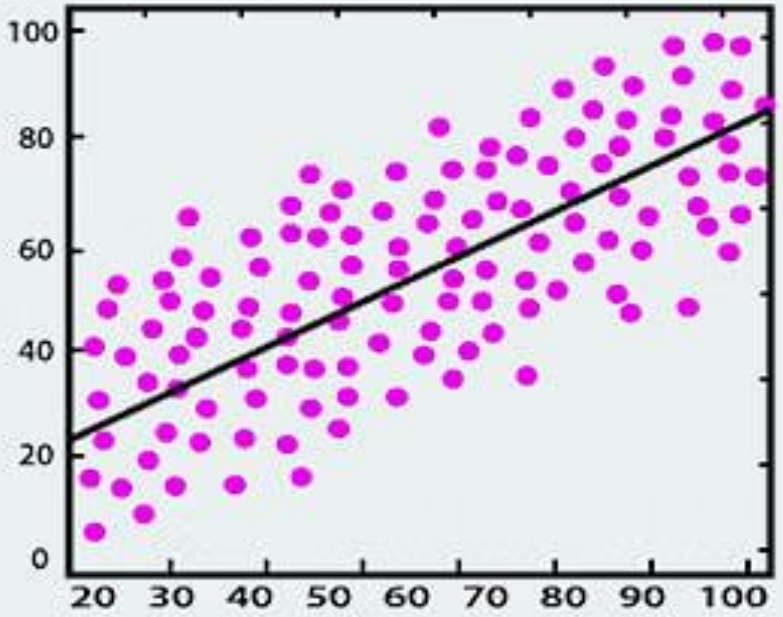


Effort for each data-mining process step

KDD: Knowledge Data Discovery

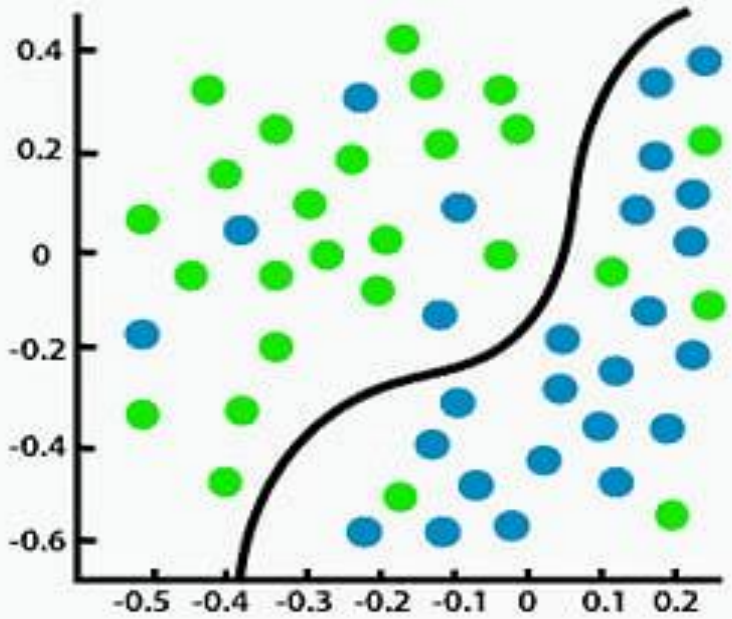

Supervised:mapping function for features
Input
Output
Model Training
Prediction
y
Output
Predicted value
E
X
New Value
X_train
y_train
X_test
y_test
Dataset
Training
Test
X=Independent Variable
y=Dependent Variable
CDAC Mumbai: Kiran Waghmare
X
Y
Training data
X_train
Testing data
X_test


Regression
1.Simple Linear Reg
2.Multiple Regressior
Logistic
Polynomial
SVM
DT
RF
Multivariate
Lasso
Data
x
Feature
Regression
Learn x->y
x,y relationship
Intelligence
Predict y
Continuous



Regression

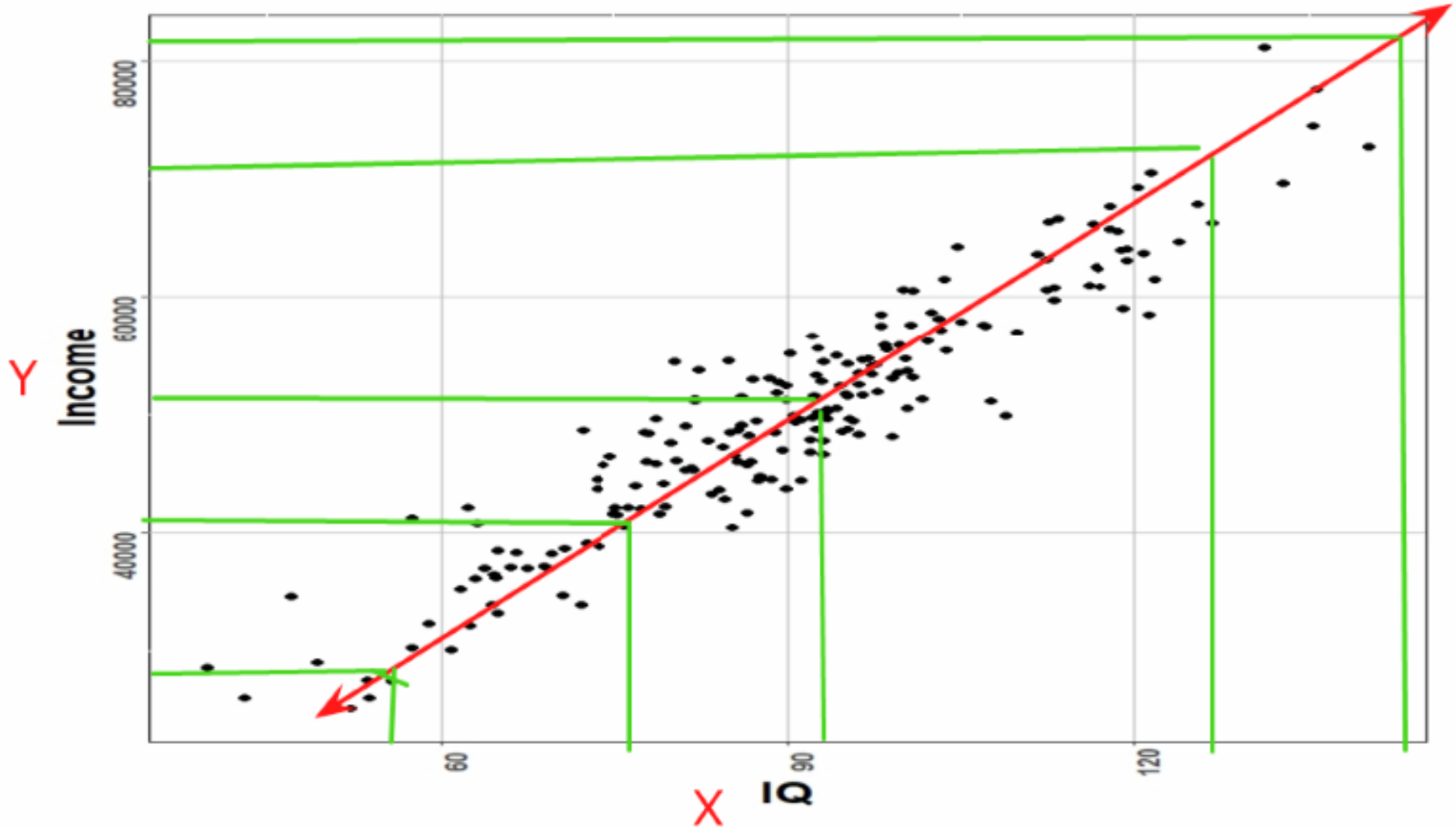
versus



Classification

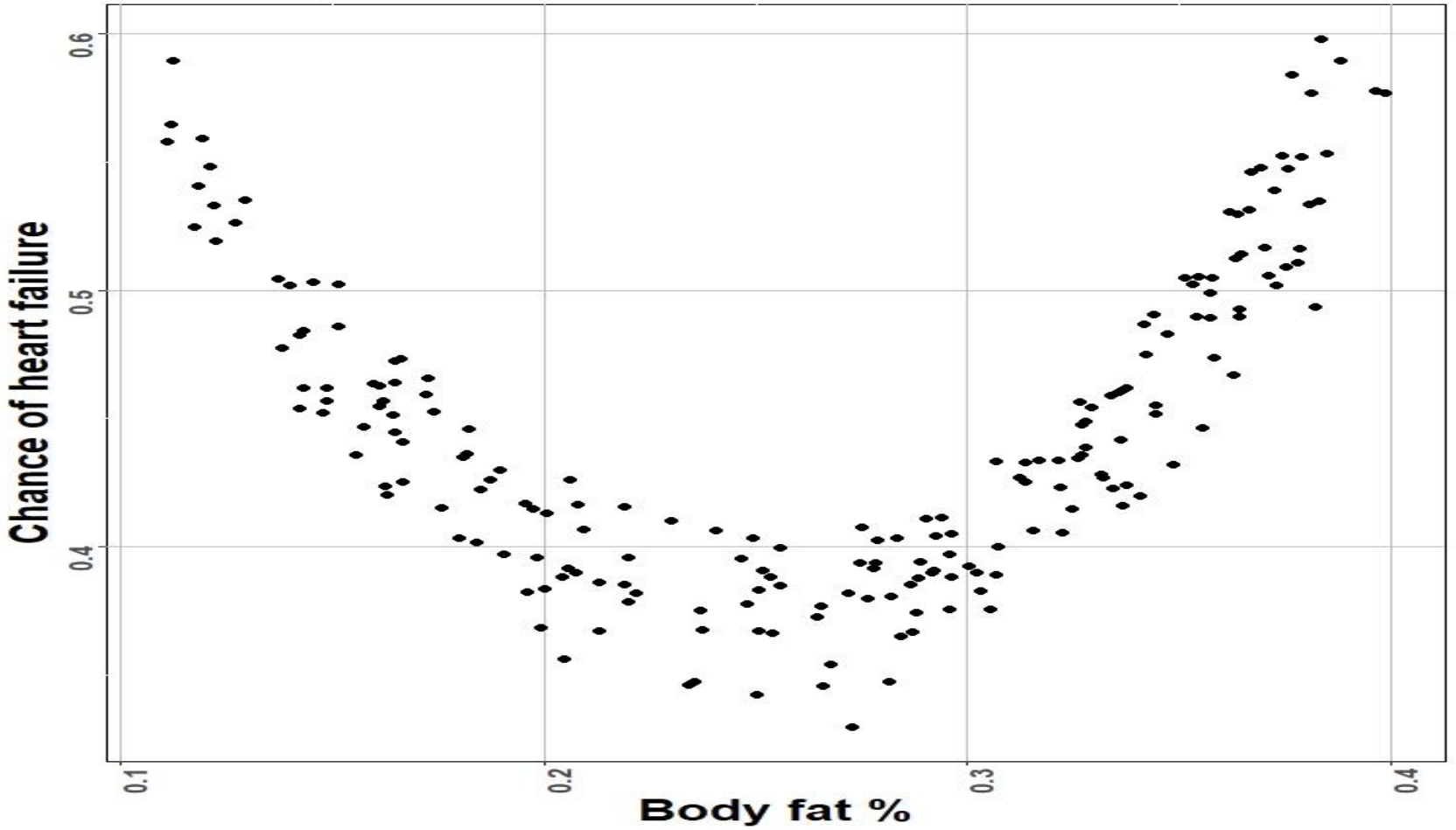
Simple Linear Regression

Displaying the data



Displaying the data

It is important to perform a scatterplot because it helps us to see if the relationship is linear.



Linear model

In regression, the relationship between Y and X is modelled in the following form:

$$Y = a + b * X + E$$

where:

- **Y** is the dependent variable (Income in the example)
- **X** is the independent variable (IQ in the example)
- **a** is an intercept
- **b** is the coefficient
- **E** is an error term for each observation (since there is additional variation not explained by income)

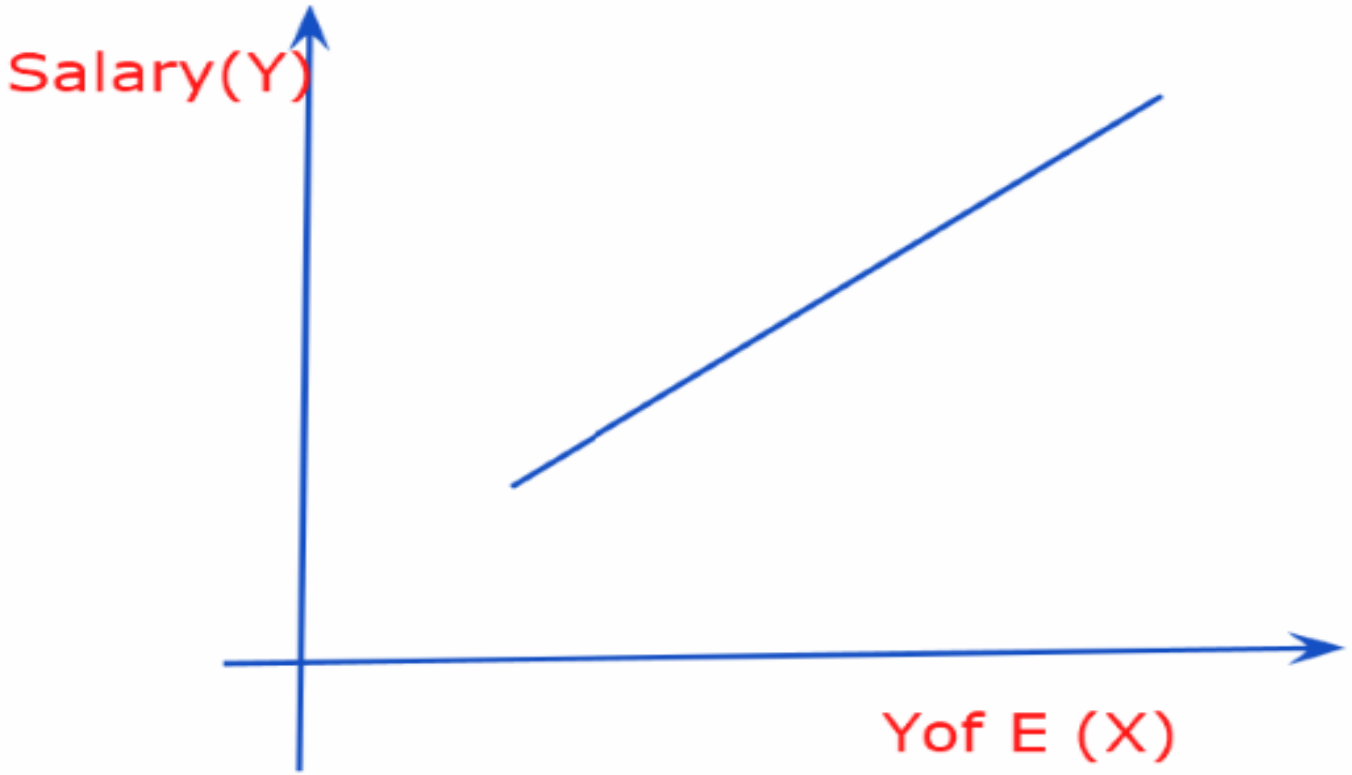
Import Dataset

```
In [3]: dataset=pd.read_csv('D:\Test\Salary_Data.csv')

In [4]: dataset
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0



```
In [7]: y
```

```
Out[7]: array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
                54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
                61111.,  67938.,  66029.,  83088.,  81363.,  93940.,  91738.,
                98273., 101302., 113812., 109431., 105582., 116969., 112635.,
               122391., 121872.])
```

Splitting the dataset (Training and Testing)

```
In [ ]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train,y_test =train_test_split(X,y,test_size=1/3, random_state=0)
```

Training the Simple Regression Model

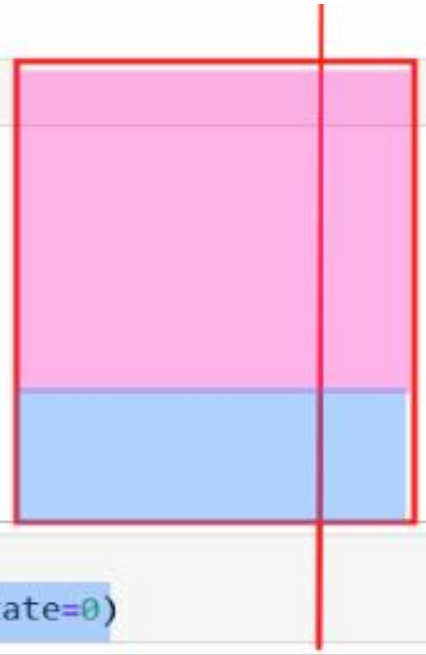
```
In [ ]: from sklearn.linear_model import LinearRegression
        reg=LinearRegression()
        reg.fit(X_train, y_train)
```

Prediction of Testing dataset

```
In [ ]: reg.predict(X_test)
```

```
In [7]: y

Out[7]: array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
                54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
                61111.,  67938.,  66029.,  83088.,  81363.,  93940.,  91738.,
                98273., 101302., 113812., 109431., 105582., 116969., 112635.,
               122391., 121872.])
```



Splitting the dataset (Training and Testing)

```
In [ ]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train,y_test =train_test_split(X,y,test_size=1/3, random_state=0)
```

Training the Simple Regression Model

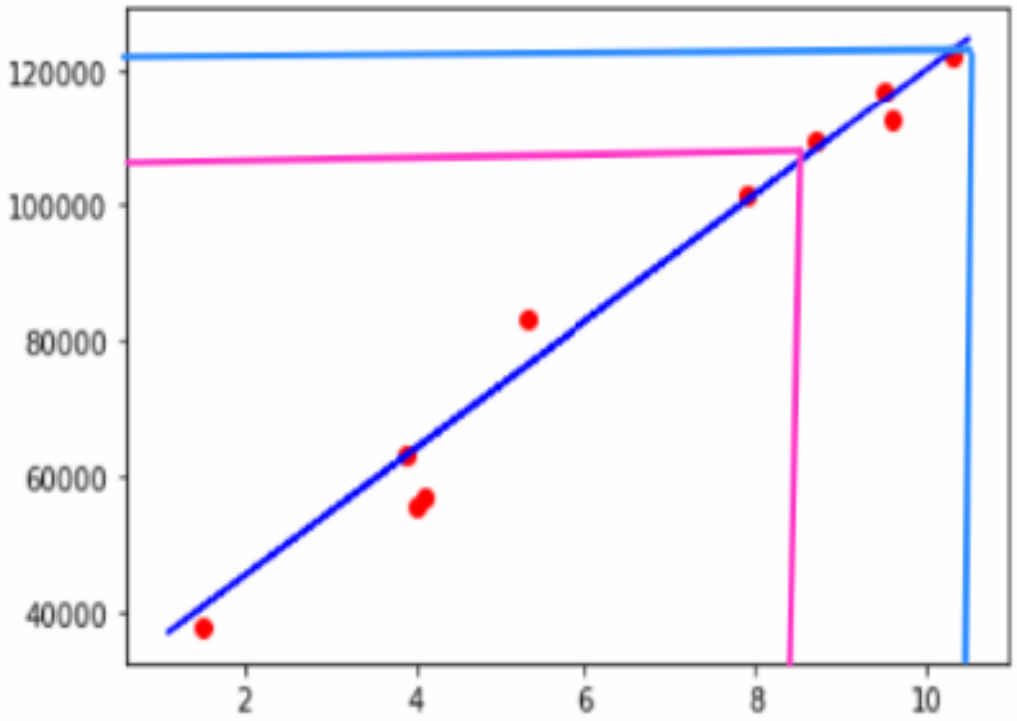
```
In [ ]: from sklearn.linear_model import LinearRegression
        reg=LinearRegression()
        reg.fit(X_train, y_train)
```

Prediction of Testing dataset

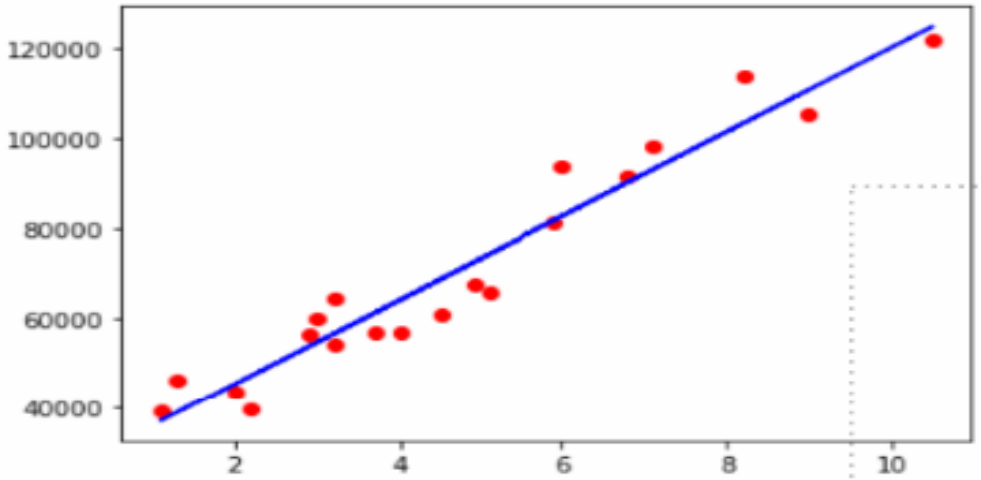
```
In [ ]: reg.predict(X_test)
```

In [12]:
```
plt.scatter(X_test,y_test,color='red')
plt.plot(X_train, reg.predict(X_train), color='blue')
```

Out[12]:
```
[<matplotlib.lines.Line2D at 0x1e10b3cf490>]
```



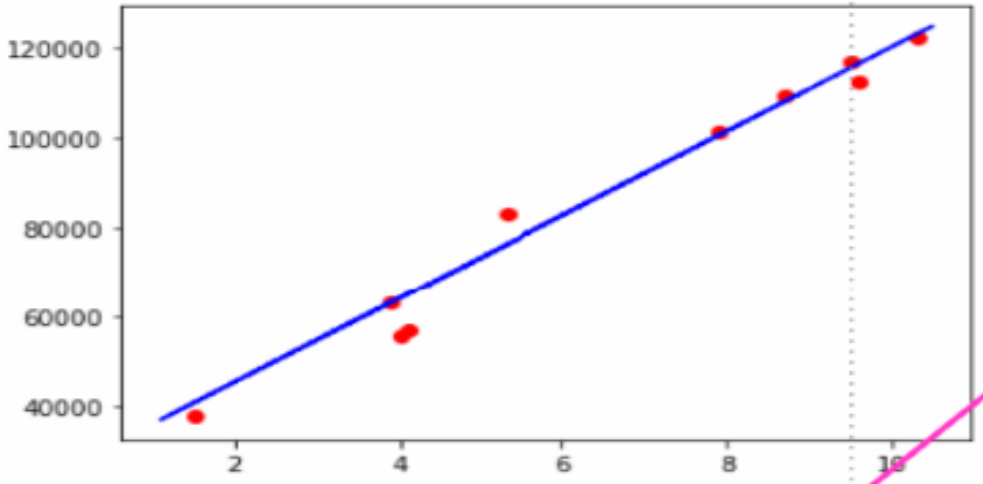
In [ ]:

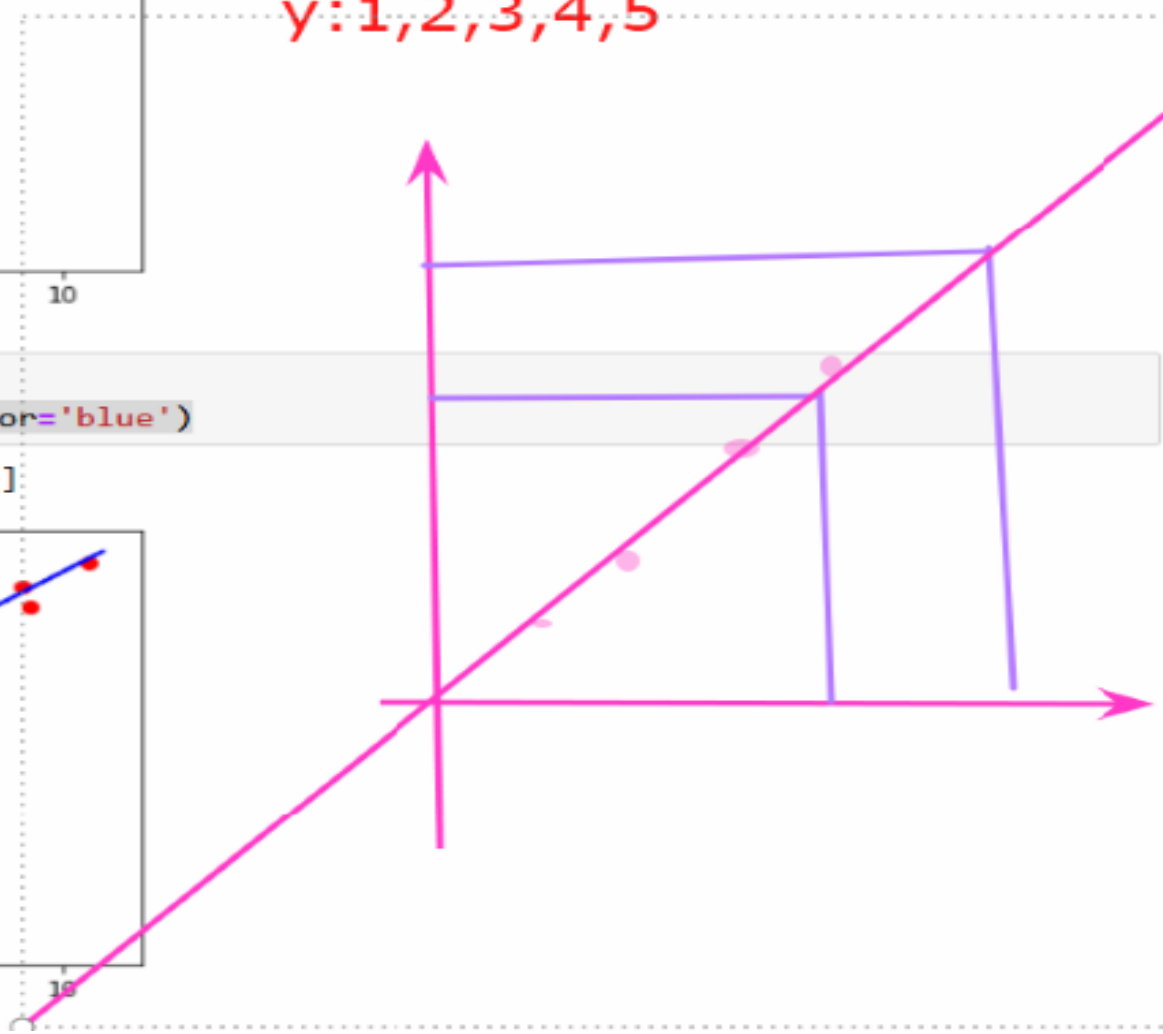


x:1,2,3,4,5
y:1,2,3,4,5

```
In [12]: plt.scatter(X_test,y_test,color='red')
         plt.plot(X_train, reg.predict(X_train), color='blue')
```

```
Out[12]: [<matplotlib.lines.Line2D at 0x1e10b3cf490>]
```





Assumptions of regression