# Day-10

# Mark Statement

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\mark.csv")
d
```

Out[2]:

| | Student_ID | Test_1 | Test_2 | Test_3 | Test_4 | Test_5 | Test_6 | Test_7 | Test_8 | Test_9 | Test_ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 22000 | 78 | 87 | 91 | 91 | 88 | 98 | 94 | 100 | 100 | 1 |
| **1** | 22001 | 79 | 71 | 81 | 72 | 73 | 68 | 59 | 69 | 59 | |
| **2** | 22002 | 66 | 65 | 70 | 74 | 78 | 86 | 87 | 96 | 88 | |
| **3** | 22003 | 60 | 58 | 54 | 61 | 54 | 57 | 64 | 62 | 72 | |
| **4** | 22004 | 99 | 95 | 96 | 93 | 97 | 89 | 92 | 98 | 91 | |
| **5** | 22005 | 41 | 36 | 35 | 28 | 35 | 36 | 27 | 26 | 19 | |
| **6** | 22006 | 47 | 50 | 47 | 57 | 62 | 64 | 71 | 75 | 85 | |
| **7** | 22007 | 84 | 74 | 70 | 68 | 58 | 59 | 56 | 56 | 64 | |
| **8** | 22008 | 74 | 64 | 58 | 57 | 53 | 51 | 47 | 45 | 42 | |
| **9** | 22009 | 87 | 81 | 73 | 74 | 71 | 63 | 53 | 45 | 39 | |
| **10** | 22010 | 40 | 34 | 37 | 33 | 31 | 35 | 39 | 38 | 40 | |
| **11** | 22011 | 91 | 84 | 78 | 74 | 76 | 80 | 80 | 73 | 75 | |
| **12** | 22012 | 81 | 83 | 93 | 88 | 89 | 90 | 99 | 99 | 95 | |
| **13** | 22013 | 52 | 50 | 42 | 38 | 33 | 30 | 28 | 22 | 12 | |
| **14** | 22014 | 63 | 67 | 65 | 74 | 80 | 86 | 95 | 96 | 92 | |
| **15** | 22015 | 76 | 82 | 88 | 94 | 85 | 76 | 70 | 60 | 50 | |
| **16** | 22016 | 83 | 78 | 71 | 71 | 77 | 72 | 66 | 75 | 66 | |
| **17** | 22017 | 55 | 45 | 43 | 38 | 43 | 35 | 44 | 37 | 45 | |
| **18** | 22018 | 71 | 67 | 76 | 74 | 64 | 61 | 57 | 64 | 61 | |
| **19** | 22019 | 62 | 61 | 53 | 49 | 54 | 59 | 68 | 74 | 65 | |
| **20** | 22020 | 44 | 38 | 36 | 34 | 26 | 34 | 39 | 44 | 36 | |
| **21** | 22021 | 50 | 56 | 53 | 46 | 41 | 38 | 47 | 39 | 44 | |
| **22** | 22022 | 57 | 48 | 40 | 45 | 43 | 36 | 26 | 19 | 9 | |
| **23** | 22023 | 59 | 56 | 52 | 44 | 50 | 40 | 45 | 46 | 54 | |
| **24** | 22024 | 84 | 92 | 89 | 80 | 90 | 80 | 84 | 74 | 68 | |
| **25** | 22025 | 74 | 80 | 86 | 87 | 90 | 100 | 95 | 87 | 85 | |
| **26** | 22026 | 92 | 84 | 74 | 83 | 93 | 83 | 75 | 82 | 81 | |
| **27** | 22027 | 63 | 70 | 74 | 65 | 64 | 55 | 61 | 58 | 48 | |
| **28** | 22028 | 78 | 77 | 69 | 76 | 78 | 74 | 67 | 69 | 78 | |
| **29** | 22029 | 55 | 58 | 59 | 67 | 71 | 62 | 53 | 61 | 67 | |
| **30** | 22030 | 54 | 54 | 48 | 38 | 35 | 45 | 46 | 47 | 41 | |
| **31** | 22031 | 84 | 93 | 97 | 89 | 86 | 95 | 100 | 100 | 100 | |
| **32** | 22032 | 95 | 100 | 94 | 100 | 98 | 99 | 100 | 90 | 80 | |
| **33** | 22033 | 64 | 61 | 63 | 73 | 63 | 68 | 64 | 58 | 50 | |
| **34** | 22034 | 76 | 79 | 73 | 77 | 83 | 86 | 95 | 89 | 90 | |
| **35** | 22035 | 78 | 71 | 61 | 55 | 54 | 48 | 41 | 32 | 41 | |
| **36** | 22036 | 95 | 89 | 91 | 84 | 89 | 94 | 85 | 91 | 100 | 1 |

| | Student_ID | Test_1 | Test_2 | Test_3 | Test_4 | Test_5 | Test_6 | Test_7 | Test_8 | Test_9 | Test_ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **37** | 22037 | 99 | 89 | 79 | 87 | 87 | 81 | 82 | 74 | 64 | |
| **38** | 22038 | 82 | 83 | 85 | 86 | 89 | 80 | 88 | 95 | 87 | |
| **39** | 22039 | 65 | 56 | 64 | 62 | 58 | 51 | 61 | 68 | 70 | |
| **40** | 22040 | 100 | 93 | 92 | 86 | 84 | 76 | 82 | 74 | 79 | |
| **41** | 22041 | 78 | 72 | 73 | 79 | 81 | 73 | 71 | 77 | 83 | |
| **42** | 22042 | 98 | 100 | 100 | 93 | 94 | 92 | 100 | 100 | 98 | |
| **43** | 22043 | 58 | 62 | 67 | 77 | 71 | 63 | 64 | 73 | 83 | |
| **44** | 22044 | 96 | 92 | 94 | 100 | 99 | 95 | 98 | 92 | 84 | |
| **45** | 22045 | 86 | 87 | 85 | 84 | 85 | 91 | 86 | 82 | 85 | |
| **46** | 22046 | 48 | 55 | 46 | 40 | 34 | 29 | 37 | 34 | 39 | |
| **47** | 22047 | 56 | 52 | 54 | 47 | 40 | 35 | 43 | 44 | 40 | |
| **48** | 22048 | 42 | 44 | 46 | 53 | 62 | 59 | 57 | 53 | 43 | |
| **49** | 22049 | 64 | 54 | 49 | 59 | 54 | 55 | 57 | 59 | 63 | |
| **50** | 22050 | 50 | 44 | 37 | 29 | 37 | 46 | 53 | 57 | 55 | |
| **51** | 22051 | 70 | 60 | 70 | 62 | 67 | 67 | 68 | 67 | 72 | |
| **52** | 22052 | 63 | 73 | 70 | 63 | 60 | 67 | 61 | 59 | 52 | |
| **53** | 22053 | 92 | 100 | 100 | 100 | 100 | 100 | 92 | 87 | 94 | 1 |
| **54** | 22054 | 64 | 55 | 54 | 61 | 63 | 57 | 47 | 37 | 44 | |
| **55** | 22055 | 60 | 66 | 68 | 58 | 49 | 47 | 39 | 29 | 39 | |

In [3]:

```
d.columns
```

Out[3]:

```
Index(['Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5',
       'Test_6', 'Test_7', 'Test_8', 'Test_9', 'Test_10', 'Test_11',
       'Test_12'],
      dtype='object')
```

In [4]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56 entries, 0 to 55
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Student_ID  56 non-null     int64
 1   Test_1      56 non-null     int64
 2   Test_2      56 non-null     int64
 3   Test_3      56 non-null     int64
 4   Test_4      56 non-null     int64
 5   Test_5      56 non-null     int64
 6   Test_6      56 non-null     int64
 7   Test_7      56 non-null     int64
 8   Test_8      56 non-null     int64
 9   Test_9      56 non-null     int64
 10  Test_10     56 non-null     int64
 11  Test_11     56 non-null     int64
 12  Test_12     56 non-null     int64
dtypes: int64(13)
memory usage: 5.8 KB
```

In [6]:

```
x=d[['Student_ID', 'Test_1', 'Test_2', 'Test_3', 'Test_4', 'Test_5',
       'Test_6', 'Test_7', 'Test_8', 'Test_9', 'Test_10', 'Test_11']]
y=d['Test_12']
```

In [7]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [8]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[8]:

```
LinearRegression()
```

In [9]:

```
print(lr.intercept_)
```

```
-1849.7127665451817
```

In [10]:

```
print(lr.score(x_test,y_test))
```

```
0.8463060688560902
```

In [11]:

```
print(lr.score(x_train,y_train))
```

```
0.9707262440457627
```

# Ridge Regression

In [12]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [13]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[13]:

```
0.85216338029586
```

# Lasso Regression

In [14]:

```
la=Lasso(alpha=10)
```

In [15]:

```
la.fit(x_train,y_train)
```

Out[15]:

```
Lasso(alpha=10)
```

In [16]:

```
la.score(x_test,y_test)
```

Out[16]:

```
0.9050813226633148
```

# Elastic Regreesion

In [17]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[17]:

```
ElasticNet()
```

In [18]:

```
predict=(en.predict(x_test))
print(predict)
```

```
[ 53.99551515  91.66016743  80.98632267  57.04957395  74.78444536
  49.32974666 100.76654073  54.15337542  69.98292897  90.99528472
  56.57662918  22.38439359  72.23408141  97.63364372  61.41091127
  77.20932895  78.16031466]
```

In [19]:

```
print(en.score(x_test,y_test))
```

```
0.8793449004604774
```

# Evaluation Method

In [20]:

```
from sklearn import metrics
```

In [21]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,predict))
```

```
Mean Absolute Error: 4.891554567442895
```

In [22]:

```
print("Root Mean Square Error:",np.sqrt(metrics.mean_squared_error(y_test,predict)))
```

```
Root Mean Square Error: 6.2325165210068825
```

In [23]:

```
print("Mean Square Error:",metrics.mean_squared_error(y_test,predict))
```

```
Mean Square Error: 38.84426218462374
```

In [ ]:

```

```