# UNIT – III

# Regression

CONTENTS:

- Introduction to Regression analysis
- measure of linear relationship
- Regression with stats models
- Determining coefficient
- meaning and significance of coefficients
- coefficient calculation with least square method
- Types of regression, Simple Linear Regression, Using Multiple features, Polynomial Regression
- Metrics for Regression: MSE, RMSE, MAE.

## 3.1 Introduction to Regression analysis

Predictive modeling is the problem of developing a model using historical data to make a prediction on new data where we do not have the answer.

Predictive modeling can be described as the mathematical problem of approximating a mapping function (f) from input variables (X) to output variables (y). This is called the problem of function approximation.

Regression is a type of Machine learning which helps in finding the relationship between independent and dependent variable.

In simple words, Regression can be defined as a Machine learning problem where we have to predict discrete values like price, Rating, Fees, etc.

Regression is one of the most common models of machine learning. It differs from classification models because it estimates a numerical value, whereas classification models identify which category an observation belongs to.

The main uses of regression analysis are forecasting, time series modeling and finding the cause and effect relationship between variables.

## Terminologies Related to the Regression Analysis:

- **Dependent Variable:** The main factor in Regression analysis which we want to predict or understand is called the dependent variable. It is also called **target variable**.
- **Independent Variable:** The factors which affect the dependent variables or which are used to predict the values of the dependent variables are called independent variable, also called as a **predictor**.
- **Outliers:** Outlier is an observation which contains either very low value or very high value in comparison to other observed values. An outlier may hamper the result, so it should be avoided.
- **Multicollinearity:** If the independent variables are highly correlated with each other than other variables, then such condition is called Multicollinearity. It should not be present in the dataset, because it creates problem while ranking the most affecting variable.
- **Underfitting and Overfitting:** If our algorithm works well with the training dataset but not well with test dataset, then such problem is called **Overfitting**. And if our algorithm does not perform well even with training dataset, then such problem is called **underfitting**.

**Why Is It Important?**

Regression has a wide range of real-life applications. It is essential for any machine learning problem that involves continuous numbers – this includes, but is not limited to, a host of examples, including:

- Financial forecasting (like house price estimates, or stock prices)
- Sales and promotions forecasting
- Testing automobiles
- Weather analysis and prediction
- Time series forecasting

As well as telling you whether a significant relationship exists between two or more variables, regression analysis can give specific details about that relationship. Specifically, it can estimate the strength of impact that multiple variables will have on a dependent variable. If you change the value of one variable (price, say), regression analysis should tell you what effect that will have on the dependent variable (sales).

Businesses can use regression analysis to test the effects of variables as measured on different scales. With it in your toolbox, you can assess the best set of variables to use when building predictive models, greatly increasing the accuracy of your forecasting.

Finally, regression analysis is the best way of solving regression problems in machine learning using data modeling. By plotting data points on a chart and running the best fit line through them, you can predict each data point's likelihood of error: the further away from the line they lie, the higher their error of prediction (this best fit line is also known as a regression line).

## 3.2 MEASURE OF LINEAR RELATIONSHIP

## Measures of Relationship

statistical measures show a relationship between two or more variables or two or more sets of data. For example, generally there is a high relationship or correlation between parent's education and academic achievement. On the other hand, there is generally no relationship or correlation between a person's height and academic achievement. The major statistical measure of relationship is the correlation coefficient.

## Correlational Coefficient

Correlation is the relationship between two or more variables or sets of data. It is expressed in the form of a coefficient with +1.00 indicating a perfect positive correlation; -1.00 indicating a perfect inverse correlation; 0.00 indicating a complete lack of a relationship.

Note: A simplified method of determining the magnitude of a correlation is as follows:

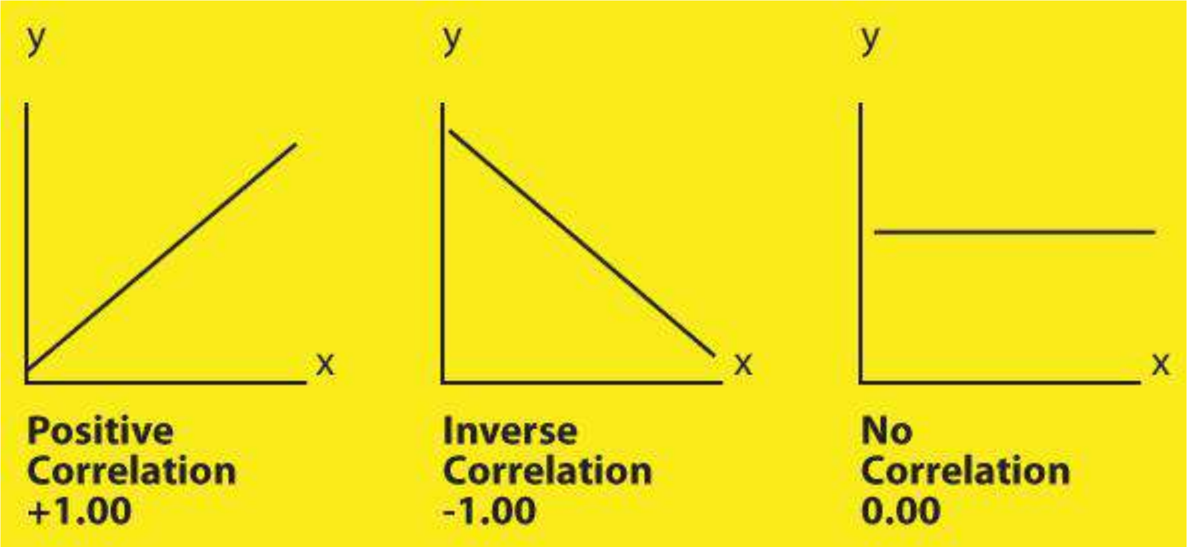
•.00 - .20 Negligible<br>
• .20 - .40 Low<br>
* .40 - .60 Moderate<br>
* .60 - .80 Substantial<br>
* .80 - 1.0 High

1. Pearson's Product Moment Coefficient (r) is the most often used and most precise coefficient; and generally used with continuous variables.

2. Spearman Rank Order Coefficient (p) is a form of the Pearson's Product Moment Coefficient which can be used with ordinal or ranked data.

3. Phi Correlation Coefficient is a form of the Pearson's Product Moment Coefficient which can be used with dichotomous variables (i.e. pass/fail, male/female).

   For example, r = .66, p < .01 where r is the correlational coefficient and p is the level (.01) of statistical significance.



**Linear (Line) Representations of Correlation Coefficients**

## Linear Regression and Multiple Regression

1. Linear regression is the use of correlation coefficients to plot a line illustrating the linear relationship of two variables X and Y. It is based on the slope of the line which is represented by the formula : Y = c+ mX where

   * Y = dependent variable<br>
   * X = independent variable
   * m = slope of the line
   * c = constant or Y intercept

   Regression is used extensively in making predictions based on finding unknown Y values from known X values.

   For example, the linear regression formula for predicting college GPA from known high school grade point averages would be displayed as follows:
   College GPA = a + b(High School GPA)

2. Multiple Regression is the same as regression except that it attempts to predict Y from two or more independent X variables. The formula for multiple regression is an extension of the linear regression formula: Y = a + b1 X1 + b2 X2 + ....

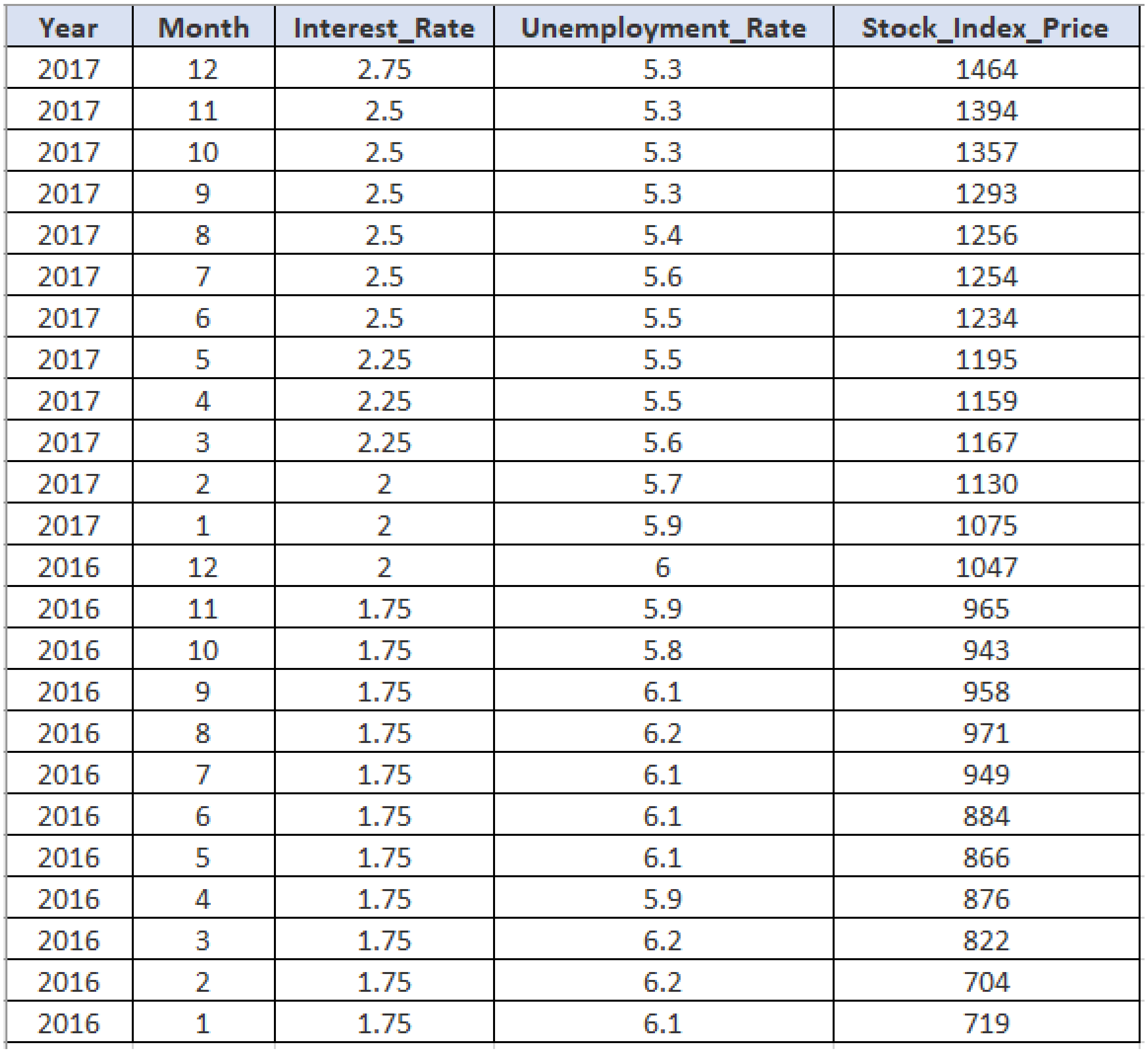
Multiple regression is used extensively in making predictions based on finding unknown Y values from known X values. For example, the multiple regression formula for predicting college GPA from known high school grade point averages and SAT scores would be displayed as follows:

College GPA = a + b1(High School GPA) + b2(SAT Score)

## 3.3 REGRESSION WITH STATS MODELS

stats models is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct

statsmodels supports specifying models using pandas DataFrames. Here is a simple example using ordinary least squares.



| Year | Month | Interest_Rate | Unemployment_Rate | Stock_Index_Price |
|---|---|---|---|---|
| 2017 | 12 | 2.75 | 5.3 | 1464 |
| 2017 | 11 | 2.5 | 5.3 | 1394 |
| 2017 | 10 | 2.5 | 5.3 | 1357 |
| 2017 | 9 | 2.5 | 5.3 | 1293 |
| 2017 | 8 | 2.5 | 5.4 | 1256 |
| 2017 | 7 | 2.5 | 5.6 | 1254 |
| 2017 | 6 | 2.5 | 5.5 | 1234 |
| 2017 | 5 | 2.25 | 5.5 | 1195 |
| 2017 | 4 | 2.25 | 5.5 | 1159 |
| 2017 | 3 | 2.25 | 5.6 | 1167 |
| 2017 | 2 | 2 | 5.7 | 1130 |
| 2017 | 1 | 2 | 5.9 | 1075 |
| 2016 | 12 | 2 | 6 | 1047 |
| 2016 | 11 | 1.75 | 5.9 | 965 |
| 2016 | 10 | 1.75 | 5.8 | 943 |
| 2016 | 9 | 1.75 | 6.1 | 958 |
| 2016 | 8 | 1.75 | 6.2 | 971 |
| 2016 | 7 | 1.75 | 6.1 | 949 |
| 2016 | 6 | 1.75 | 6.1 | 884 |
| 2016 | 5 | 1.75 | 6.1 | 866 |
| 2016 | 4 | 1.75 | 5.9 | 876 |
| 2016 | 3 | 1.75 | 6.2 | 822 |
| 2016 | 2 | 1.75 | 6.2 | 704 |
| 2016 | 1 | 1.75 | 6.1 | 719 |

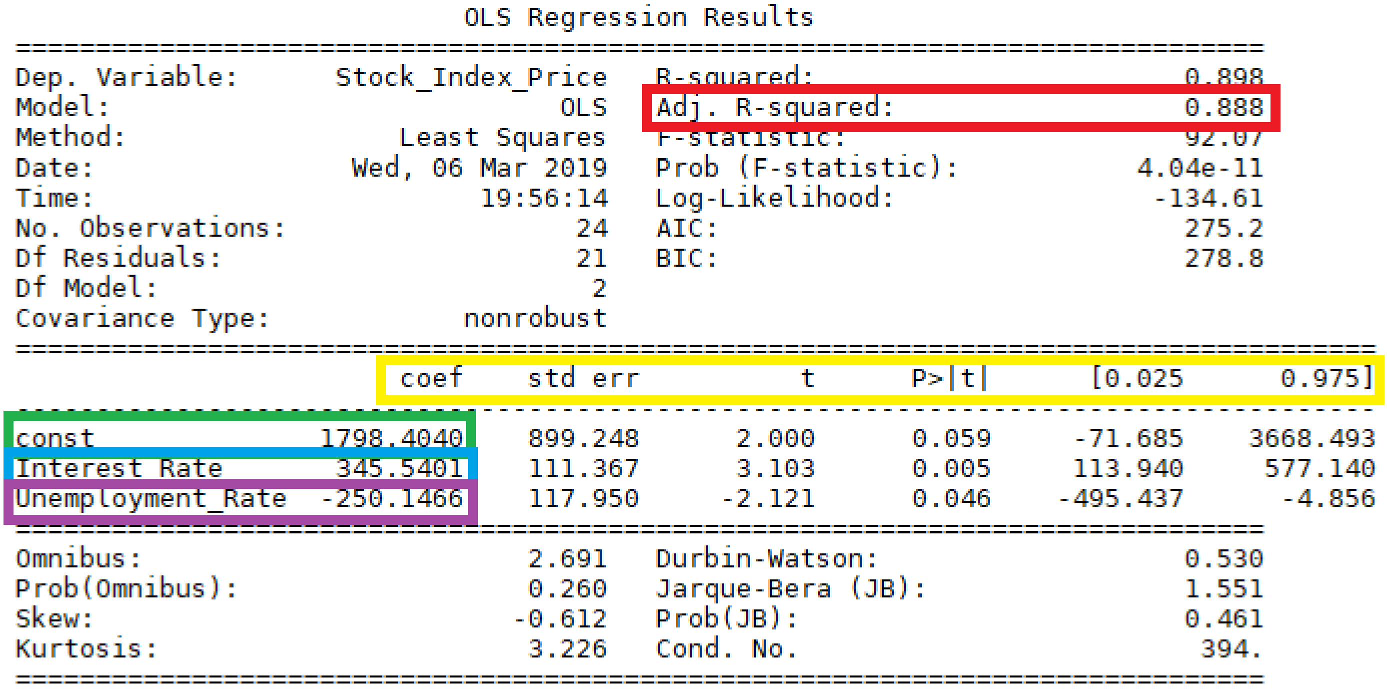
The goal here is to predict/estimate the stock index price based on two macroeconomics variables: the interest rate and the unemployment rate.

We will use pandas DataFrame to capture the above data in Python.

Here is the complete syntax to perform the linear regression in Python using statsmodels (for larger datasets, you may consider to import your data.

```
from pandas import DataFrame

import statsmodels.api as sm



Stock_Market = {'Year':
[2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2016,2016,2016,2016,2016,2016,
2016,2016,2016,2016,2016,2016],

                'Month': [12, 11,10,9,8,7,6,5,4,3,2,1,12,11,10,9,8,7,6,5,4,3,2,1],

                'Interest_Rate':
[2.75,2.5,2.5,2.5,2.5,2.5,2.5,2.25,2.25,2.25,2,2,2,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,
1.75,1.75,1.75],

                'Unemployment_Rate':
[5.3,5.3,5.3,5.3,5.4,5.6,5.5,5.5,5.5,5.6,5.7,5.9,6,5.9,5.8,6.1,6.2,6.1,6.1,6.1,5.9,6.2,6.2,
6.1],

                'Stock_Index_Price':
[1464,1394,1357,1293,1256,1254,1234,1195,1159,1167,1130,1075,1047,965,943,958,971,949,884,8
66,876,822,704,719]

                }

df =
DataFrame(Stock_Market,columns=['Year','Month','Interest_Rate','Unemployment_Rate','Stock_I
ndex_Price'])

X = df[['Interest_Rate','Unemployment_Rate']] # here we have 2 variables for the multiple
linear regression. If you just want to use one variable for simple linear regression, then
use X = df['Interest_Rate'] for example

Y = df['Stock_Index_Price']

X = sm.add_constant(X) # adding a constant

model = sm.OLS(Y, X).fit()

predictions = model.predict(X)

print_model = model.summary()

print(print_model)
```

This is the result that you’ll get once you run the Python code:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Stock_Index_Price   R-squared:                       0.898
Model:                            OLS   Adj. R-squared:                  0.888
Method:                 Least Squares   F-statistic:                     92.07
Date:                Wed, 06 Mar 2019   Prob (F-statistic):           4.04e-11
Time:                        19:56:14   Log-Likelihood:                -134.61
No. Observations:                  24   AIC:                             275.2
Df Residuals:                      21   BIC:                             278.8
Df Model:                           2
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              1798.4040    899.248      2.000      0.059     -71.685    3668.493
Interest_Rate       345.5401    111.367      3.103      0.005     113.940     577.140
Unemployment_Rate  -250.1466    117.950     -2.121      0.046    -495.437      -4.856
==============================================================================
Omnibus:                        2.691   Durbin-Watson:                   0.530
Prob(Omnibus):                  0.260   Jarque-Bera (JB):                1.551
Skew:                          -0.612   Prob(JB):                        0.461
Kurtosis:                       3.226   Cond. No.                         394.
==============================================================================
```

**Interpreting the Regression Results**

1. **Adjusted. R-squared** reflects the fit of the model. R-squared values range from 0 to 1, where a higher value generally indicates a better fit, assuming certain conditions are met.
2. **const coefficient** is your Y-intercept. It means that if both the Interest_Rate and Unemployment_Rate coefficients are zero, then the expected output (i.e., the Y) would be equal to the const coefficient.
3. **Interest_Rate coefficient** represents the change in the output Y due to a change of one unit in the interest rate (everything else held constant)
4. **Unemployment_Rate coefficient** represents the change in the output Y due to a change of one unit in the unemployment rate (everything else held constant)
5. **std err** reflects the level of accuracy of the coefficients. The lower it is, the higher is the level of accuracy
6. **P >|t|** is your *p-value*. A p-value of less than 0.05 is considered to be statistically significant
7. **Confidence Interval** represents the range in which our coefficients are likely to fall (with a likelihood of 95%)

## 3.4 Determining coefficient

The coefficient of determination (R² or r-squared) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, the coefficient of determination tells one how well the data fits the model. R-squared (or $R^2$), assesses how strong the linear relationship is between two variables The coefficient of determination can take any values between 0 to 1.

**Interpretation of the Coefficient of Determination (R²)**

The most common interpretation of the coefficient of determination is how well the regression model fits the observed data. Generally, a higher coefficient indicates a better fit for the model.

More specifically, R-squared gives you the percentage variation in y explained by x-variables. The range is 0 to 1 (i.e. 0% to 100% of the variation in y can be explained by the x-variables).

However, it is not always the case that a high r-squared is good for the regression model. The quality of the coefficient depends on several factors, including the units of measure of the variables, the nature of the variables employed in the model, and the applied data transformation. Thus, sometimes, a high coefficient can indicate issues with the regression model.

### Calculation of the Coefficient

Mathematically, the coefficient of determination can be found using the following formula:

Where:

$$\text{Coefficient of Determination } (R^2) = 1 - \frac{SS_{\text{regression}}}{SS_{\text{total}}}$$

- **$SS_{\text{regression}}$** – The sum of squares due to regression (explained sum of squares)

- **$SS_{\text{total}}$** – The total sum of squares

Although the terms “total sum of squares” and “sum of squares due to regression” seem confusing, the variables’ meanings are straightforward.

The total sum of squares measures the variation in the observed data (data used in regression modeling). The sum of squares due to regression measures how well the regression model represents the data that were used for modeling.

**Adjusted Coefficient of Determination** (Adjusted R-squared)

The Adjusted Coefficient of Determination (Adjusted R-squared) is an adjustment for the Coefficient of Determination that takes into account **the number of variables in a data set.** It also penalizes you for points that don’t fit the model.

Few values in a data set (a too-small sample size) can lead to misleading statistics, at the same time **too many** data points can also lead to problems. Every time we add a data point in regression analysis, $R^2$ will increase. **$R^2$ never decreases.** Therefore, the more points you add, the better the regression will seem to “fit” your data. If your data doesn’t quite fit a line, we keep on adding data until you have a better fit.

Some of the points you add will be significant (fit the model) and others will not. $R^2$ doesn’t care about the insignificant points. **The more you add, the higher the coefficient of determination**.

The adjusted $R^2$ can be used to include a **more appropriate** number of variables. The adjusted $R^2$ will increase only if a new data point improves the regression more than you would expect by chance. $R^2$ doesn’t include all data points. Negative values will likely happen if $R^2$ is close to zero — after the adjustment, the value will dip below zero a little.

## 3.6 Least Square Method

The least square method is the process of finding the best-fitting curve or line of best fit for a set of data points by reducing the sum of the squares of the offsets (residual part) of the points from the curve. During the process of finding the relation between two variables, the trend of outcomes are estimated quantitatively. This process is termed as regression analysis. The method of curve fitting is an approach to regression analysis. This method of fitting equations which approximates the curves to given raw data is the least squares.

The method of least squares actually defines the solution for the minimization of the sum of squares of deviations or the errors in the result of each equation.

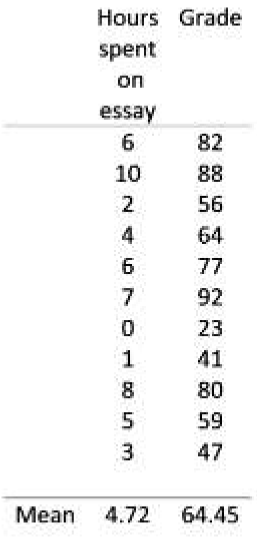
The least-squares method is often applied in data fitting. The best fit result is assumed to reduce the sum of squared errors or residuals which are stated to be the differences between the observed or experimental value and corresponding fitted value given in the model.

**Least squares regression equations**

The premise of a regression model is to examine the impact of one or more independent variables (in this case time spent writing an essay) on a dependent variable of interest (in this case essay grades). Linear regression analyses such as these are based on a simple equation:

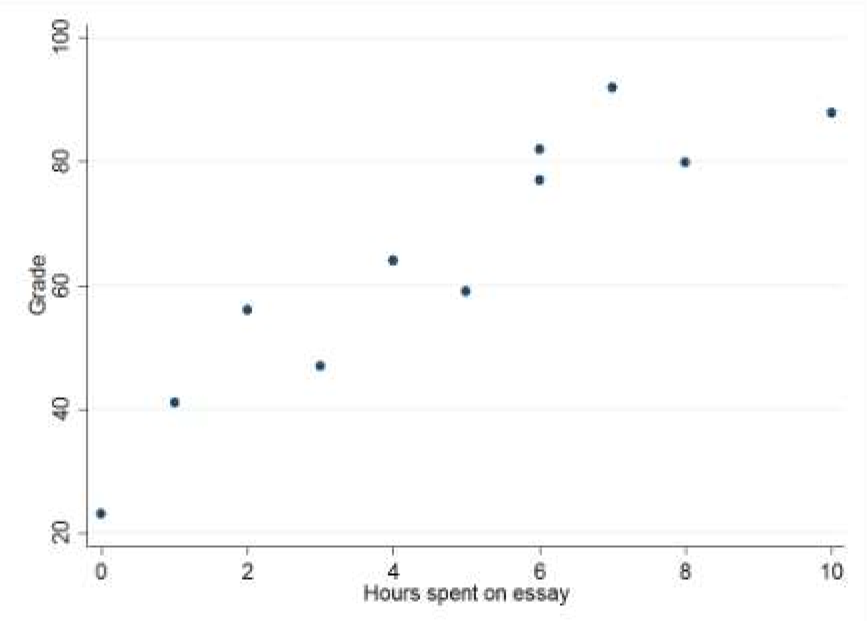
**Y = a + bX**

Example: For the table given below estimate a score for someone who had spent exactly 2.3 hours on an essay.



| | Hours spent on essay | Grade |
|---|---|---|
| | 6 | 82 |
| | 10 | 88 |
| | 2 | 56 |
| | 4 | 64 |
| | 6 | 77 |
| | 7 | 92 |
| | 0 | 23 |
| | 1 | 41 |
| | 8 | 80 |
| | 5 | 59 |
| | 3 | 47 |
| Mean | 4.72 | 64.45 |

Step1: plot the graph with hours spent on essay on x-axis as input and Grade on y-axis as output



Step2: Linear regression analyses such as these are based on a simple equation

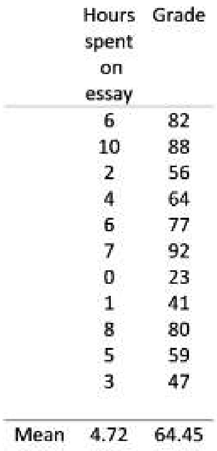
**Y = a + bX**

Where,

**Y – Essay Grade** ***a*** **– Intercept** ***b*** **– Coefficient** **X – Time spent on Essay**

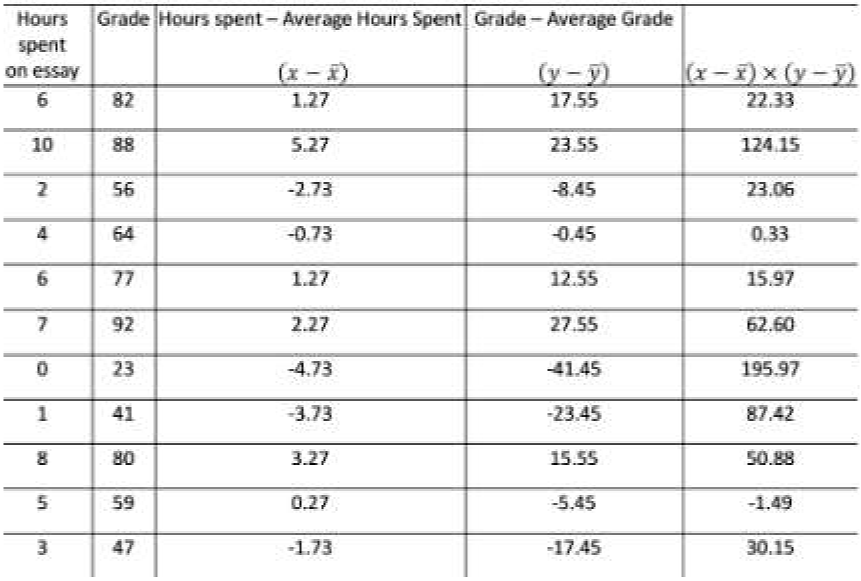
$$b = \frac{\sum(x - \bar{x}) * (y - \bar{y})}{\sum(x - \bar{x})^2}$$

Step3: When calculating least squares regressions by hand, the first step is to **find the means of the dependent and independent variables**.



| | Hours spent on essay | Grade |
|---|---|---|
| | 6 | 82 |
| | 10 | 88 |
| | 2 | 56 |
| | 4 | 64 |
| | 6 | 77 |
| | 7 | 92 |
| | 0 | 23 |
| | 1 | 41 |
| | 8 | 80 |
| | 5 | 59 |
| | 3 | 47 |
| Mean | 4.72 | 64.45 |

Step4: The next step is to calculate the difference between each value and the mean value for both the dependent and the independent variable. In this case this means we subtract 64.45 from each test score and 4.72 from each time data point. Additionally, we want to find the product of multiplying these two differences together.



| Hours spent on essay | Grade | Hours spent – Average Hours Spent<br>$(x-\bar{x})$ | Grade – Average Grade<br>$(y-\bar{y})$ | $(x-\bar{x}) \times (y-\bar{y})$ |
|---|---|---|---|---|
| 6 | 82 | 1.27 | 17.55 | 22.33 |
| 10 | 88 | 5.27 | 23.55 | 124.15 |
| 2 | 56 | -2.73 | -8.45 | 23.06 |
| 4 | 64 | -0.73 | -0.45 | 0.33 |
| 6 | 77 | 1.27 | 12.55 | 15.97 |
| 7 | 92 | 2.27 | 27.55 | 62.60 |
| 0 | 23 | -4.73 | -41.45 | 195.97 |
| 1 | 41 | -3.73 | -23.45 | 87.42 |
| 8 | 80 | 3.27 | 15.55 | 50.88 |
| 5 | 59 | 0.27 | -5.45 | -1.49 |
| 3 | 47 | -1.73 | -17.45 | 30.15 |

You should notice that as some scores are lower than the mean score, we end up with negative values. By squaring these differences, we end up with a standardized measure of deviation from the mean regardless of whether the values are more or less than the mean. Let's remind ourselves of the equation we need to calculate ***b***.

$$\sum(x-\bar{x}) * (y-\bar{y}) = 611.36$$

And

$$\sum(x-\bar{x})\ \text{^}2 = 94.18$$

$$b = \frac{611.36}{94.18} = 6.49$$

Step5: The final step is to calculate the intercept, which we can do using the initial regression equation with the values of test score and time spent set as their respective means, along with our newly calculated coefficient.

**64.45= a + 6.49*4.72**

We can then solve this for ***a***:

**64.45 = a + 30.63**

**a = 64.45 – 30.63**

**a = 30.18**

If we wanted to know the predicted grade of someone who spends 2.35 hours on their essay, all we need to do is swap that in for **X**.

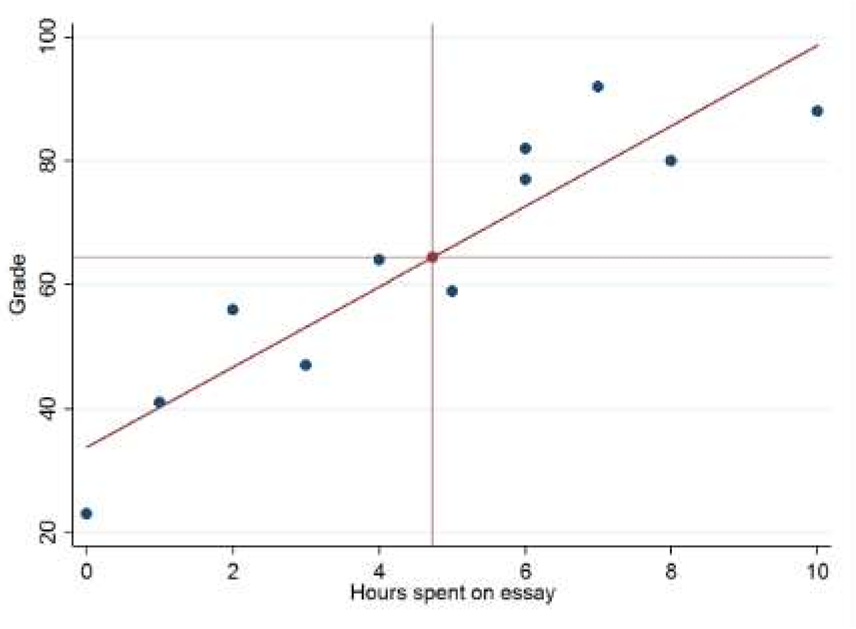
**y=30.18 + 6.49 * X**

**y = 30.18 + (6.49 * 2.35)**

**y = 45.43**

### Drawing a least squares regression line by hand

The best fit line should cross the means of both the time spent on the essay and the mean grade received.



### What are the disadvantages of least-squares regression?

As noticed, a model shown above as this has its limitations. For example, if a student had spent 20 hours on an essay, their predicted score would be 160, which doesn't really make sense on a typical 0-100 scale. It's always important to understand the realistic real-world limitations of a model and ensure that it's not being used to answer questions that it's not suited for.

## 3.7 Different Types of Regression

There are many types of regression analysis techniques, and the use of each method depends 1upon the number of factors. These factors include the type of target variable, shape of the regression line, and the number of independent variables.

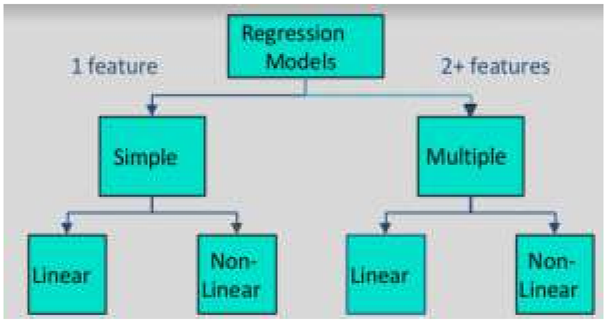
Outliers such as these can have a disproportionate effect on our data. In this case, it's important to organize your data and validate your model depending on what your data looks like to make sure it is the right approach to take.

Based on independent variable

1. Simple
2. Multiple

Based on shape of the regression line

1. Linear
2. Non-linear



Based on type of target variable

1. Linear regression
2. Logistic regression
3. Polynomial regression
4. Ridge regression
5. Lasso regression

Linear regression and logistic regression are two **types of regression analysis** techniques that are used to solve the regression problem using machine learning. They are the most prominent techniques of regression. But, there are many types of regression analysis techniques in machine learning, and their usage varies according to the nature of the data involved.

### 1. Linear regression

One of the most basic types of regression in machine learning, linear regression comprises a predictor variable and a dependent variable related to each other in a linear fashion. Linear regression involves the use of a best fit line

You should use linear regression when your variables are related linearly. For example, if you are forecasting the effect of increased advertising spend on sales. However, this analysis is susceptible to outliers, so it should not be used to analyze big data sets.

### Linear Regression Real Life Example

- Businesses often use linear regression to understand the relationship between advertising spending and revenue.
- Medical researchers often use linear regression to understand the relationship between drug dosage and blood pressure of patients.
- Data scientists for professional sports teams often use linear regression to measure the effect that different training regimens have on player performance.

### 2. Logistic regression

Does your dependent variable have a discrete value? In other words, can it only have one of two values (either 0 or 1, true or false, black or white, spam or not spam, and so on)? In that case, you might want to use logistic regression to analyze your data.

Logistic regression uses a sigmoid curve to show the relationship between the target and independent variables. However, caution should be exercised: logistic regression works best with large data sets that have an almost equal occurrence of values in target variables. The dataset should not contain a high correlation between independent variables (a phenomenon known as multicollinearity), as this will create a problem when ranking the variables.

### Logistic Regression Real Life Example

- Medical researchers want to know how exercise and weight impact the probability of having a heart attack. To understand the relationship between the predictor variables and the probability of having a heart attack, researchers can perform logistic regression.
- A business wants to know whether word count and country of origin impact the probability that an email is spam.
- A credit card company wants to know whether transaction amount and credit score impact the probability of a given transaction being fraudulent.

**3. Ridge regression**

If, however, you do have a high correlation between independent variables, ridge regression is a more suitable tool. It is known as a regularization technique, and is used to reduce the complexity of the model. It introduces a small amount of bias (known as the 'ridge regression penalty') which, using a bias matrix, makes the model less susceptible to overfitting.

**4. Lasso regression** Like ridge regression, lasso regression is another regularization technique that reduces the model's complexity. It does so by prohibiting the absolute size of the regression coefficient. This causes the coefficient value to become closer to zero, which does not happen with ridge regression.

The advantages:

It can use feature selection, letting you select a set of features from the dataset to build the model. By only using the required features – and setting the rest as zero – lasso regression avoids overfitting.

**5. Polynomial regression**

Polynomial regression models a non-linear dataset using a linear model. It is the equivalent of making a square peg fit into a round hole. It works in a similar way to multiple linear regression (which is just linear regression but with multiple independent variables), but uses a non-linear curve. It is used when data points are present in a non-linear fashion.

The model transforms these data points into polynomial features of a given degree, and models them using a linear model. This involves best fitting them using a polynomial line, which is curved, rather than the straight line seen in linear regression. However, this model can be prone to overfitting, so you are advised to analyze the curve towards the end to avoid odd-looking results.

There are more types of regression analysis than those listed here, but these five are probably the most commonly used. Make sure you pick the right one, and it can unlock the full potential of your data, setting you on the path to greater insights.

**Which of the following is a regression task?**

- Predicting age of a person
- Predicting nationality of a person
- Predicting whether stock price of a company will increase tomorrow
- Predicting whether a document is related to sighting of UFOs?

**Solution :** Predicting age of a person (because it is a real value, predicting nationality is categorical, whether stock price will increase is discreet-yes/no answer, predicting whether a document is related to UFO is again discreet- a yes/no answer).

Regression is related to classification, but the two are different. In simple terms, classification forecasts whether something will happen, while regression forecasts how much something will happen.

Regression cannot deal with non-linear relationships or interactions.

## 3.8 REGRESSION METRICS

Model evaluation is very important in Machine learning. It helps us to understand the performance of our model and makes it easy to present our model to other people. There are many different evaluation metrics out there but only some of them are suitable to be used for regression.

**There are 3 main metrics for model evaluation in regression:**

1. R Square/Adjusted R Square

2. Mean Square Error(MSE)/Root Mean Square Error(RMSE)

3. Mean Absolute Error(MAE)

### 1. R Square/Adjusted R Square

R Square measures how much variability in dependent variable can be explained by the model. It is the square of the Correlation Coefficient(R) and that is why it is called R Square.

R Square is  calculated by the sum of squared of prediction error divided by the total sum of the square which replaces the calculated prediction with mean. R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.

R Square is a good measure to determine how well the model fits the dependent variables. However, it does not take into consideration of overfitting problem. If your regression model has many independent variables, because the model is too complicated, it may fit very well to the training data but performs badly for testing data. That is why Adjusted R Square is introduced because it will penalize additional independent variables added to the model and adjust the metric to prevent overfitting issues**.**

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R square formula

While R Square is a relative measure of how well the model fits dependent variables, Mean Square Error is an absolute measure of the goodness for the fit.

R-squared is always between 0 and 100%:

- 0% indicates that the model explains none of the variability of the response data around its mean.

- 100% indicates that the model explains all the variability of the response data around its mean.

**Python Code:**

```
import numpy as np

from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error

x=np.array([1,2,3,4])

y=np.array([4,3,2,1])

r=r2_score(x,y)

print("r squared ",r)
```

**Use cases:**

**case 1:** $Y=[1,2,3,4]$

$Y^{\wedge}=[1,2,3,4]$

$R^2=1.0$

**case 2:** $Y=[1,2,3,4]$

$Y^{\wedge}=[3,1,4,2]$

$R^2=-1.0$

**case 3:** $Y=[1,2,3,4]$

$Y^{\wedge}=[4,3,2,1]$

$R^2=-3.0$

### *2. Mean Square Error (MSE)/Root Mean Square Error(RMSE)*

MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points. It gives you an absolute number on how much your predicted results deviate from the actual number. You cannot interpret many insights from one single result but it gives you a real number to compare against other model results and help you select the best regression model.

Root Mean Square Error(RMSE) is the square root of MSE. It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and makes it easier for interpretation

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Mean Square Error formula

**Python Code:**

```
import numpy as np

from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error

x=np.array([1,2,3,4,5,6,7])

y=np.array([1,2,3,4,5,6,7])

m2=mean_squared_error(x,y)

print("mse ",m2)
```

use  cases:

**Case1:** Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[1,2,3,4,5,6,7]

MSE=0

**Case2:** Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[7,6,5,4,3,2,1]

MSE=16.0

**Case3:** Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[6,4,3,2,7,1,5]

MSE=9.42

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average of squared differences between prediction and actual observation.

**Python Code:**

```
import numpy as np

from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error

x=np.array([1,2,3,4,5,6,7])

y=np.array([1,2,3,4,5,6,7])

m2=mean_squared_error(x,y)

rmse=np.sqrt(m2)

print("rmse value ",rmse)
```

**Use cases:**

Case1: Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[1,2,3,4,5,6,7]

RMSE=0

Case2:  Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[7,6,5,4,3,2,1]

RMSE=4.0

Case3: Y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[6,4,3,2,7,1,5]

RMSE=3.7

3. ***Mean Absolute Error(MAE)***

Mean Absolute Error(MAE) is similar to Mean Square Error(MSE). However, instead of the sum of square of error in MSE, MAE is taking the sum of the absolute value of error.

Compare to MSE or RMSE, MAE is a more direct representation of sum of error terms. MSE gives larger penalization to big prediction error by square it while MAE treats all errors the same.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Mean Absolute Error formula

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

**Python code:**

import numpy as np

from sklearn.metrics import r2_score

from sklearn.metrics import mean_squared_error

from sklearn.metrics import mean_absolute_error

```
x=np.array([1,2,3,4,5,6,7])

y=np.array([6,4,3,2,7,1,5])

mae=mean_absolute_error(x,y)

print("mean_absolute_error ",mae)
```

**Use case 1:** y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[1,2,3,4,5,6,7]

MAE=0.0

**Use case 2:** y=[1,2,3,4,5,6,7]

$Y^{\wedge}$= [6,4,3,2,7,1,5]

MAE=2.57

**Use case 3:** y=[1,2,3,4,5,6,7]

$Y^{\wedge}$=[7,6,5,4,3,2,1]

MAE=3.42

R Square/Adjusted R Square is better used to explain the model to other people because you can explain the number as a percentage of the output variability. MSE, RMSE, or MAE are better be used to compare performance between different regression models.

Adjusted R square is the only metric here that considers the overfitting problem. R Square has a direct library in Python to calculate but no direct library to calculate Adjusted R square except using the statsmodel results. If you really want to calculate Adjusted R Square, you can use statsmodel or use its mathematic formula directly.