

# Implementation of Linear Regression using Least Squares and Gradient Descent in Python

Ahmad Farhan AlShammari  
Department of Computer and Information Systems  
College of Business Studies, PAAET  
Kuwait

## ABSTRACT

The goal of this research is to develop a linear regression program using least squares and gradient descent in Python. Linear regression helps to find the line that best fits to the data points. The linear regression model is based on a linear polynomial of slope ( $m$ ) and intercept ( $c$ ). Least squares is used to minimize the error between the observed and predicted points. Gradient descent is used to find the optimal solution that provides the minimum value of error function.

The basic steps of linear regression using least squares and gradient descent are explained: preparing observed points, initializing slope and intercept, computing predicted points, computing partial derivatives, updating slope and intercept, computing error function, making equation of line, and plotting predicted line.

The developed program was tested on an experimental dataset from Kaggle. The program successfully performed the basic steps of linear regression and provided the required results.

## Keywords

Artificial Intelligence, Machine Learning, Linear Regression, Least Squares, Mean Squared Error, Gradient Descent, Python, Programming.

## 1. INTRODUCTION

In recent years, machine learning has played a major role in the development of computer systems. Machine learning (ML) is a branch of Artificial Intelligence (AI) which is focused on the study of computer algorithms to improve the performance of computer programs [1-10].

Linear regression is one of the important applications of machine learning. It is a common area that is sharing knowledge between various fields such as machine learning, programming, data science, mathematics, statistics, and numerical methods [11-13, 14, 15, 16, 17].

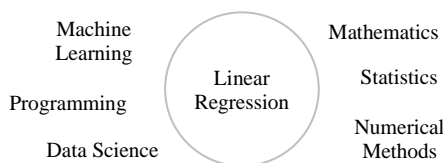


Fig 1: Field of Linear Regression

In this paper, linear regression is applied using least squares and gradient descent to find the optimal line that best fits to the data points using a linear polynomial. Linear regression is extensively used in data analysis. It has a wide range of applications in many fields like: engineering, industry,

business, education, medicine, public health, agriculture, environment, climate change, etc.

## 2. LITERATURE REVIEW

The review of literature revealed the major contributions in the field of linear regression using least squares and gradient descent [18-30].

Linear regression is a well-known method in mathematics. It was first performed by the great mathematicians Legendre (1805) and Gauss (1809) to predict the motion of planets [31]. It was used to model the relationship between two variables.

The linear regression model is based on a linear polynomial as shown in the following form:

$$y = mx + c$$

where ( $m$ ) is the slope of the line, and ( $c$ ) is the intercept with the  $y$ -axis. The linear function is explained in the following diagram:

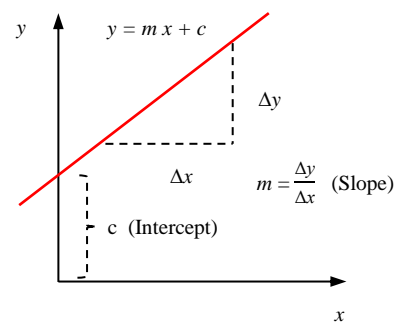


Fig 2: Equation of Line

A line can have different shapes depending on the values of slope and intercept as shown in the following diagram:

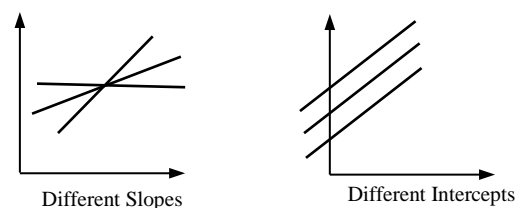


Fig 3: Different Shapes of Line

The fundamental concepts of linear regression using least squares and gradient descent are explained in the following section.

## Linear Regression:

Linear regression is an algorithm used to find the line that best fits to the data points. It is used to model the relationship between the independent variable ( $x$ ) and the dependent variable ( $y$ ) using a linear polynomial of slope ( $m$ ) and intercept ( $c$ ). Linear regression is widely used in data analysis because of its simplicity and efficiency.

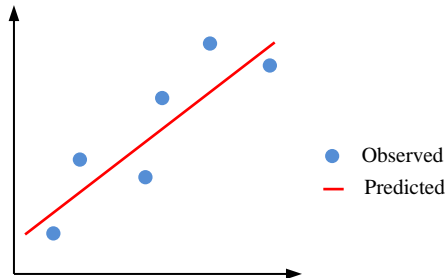


Fig 4: Explanation of Linear Regression

To find the optimal line that best fits to the data points, the concept of least squares is used to minimize the variance between the observed and predicted points.

The concept of least squares is explained in the following section.

## Least Squares:

Least Squares is a mathematical method used to minimize the error between the observed and predicted points. The error function is defined by the Mean Squared Error (MSE). It is computed by the following formula:

$$\begin{aligned} \text{MSE} &= \left(\frac{1}{n}\right) \sum (y - y_p)^2 \quad (1) \\ &= \left(\frac{1}{n}\right) \left[ (y_0 - y_{p0})^2 + (y_1 - y_{p1})^2 + \dots + (y_n - y_{pn})^2 \right] \end{aligned}$$

The differences between the observed and predicted points are shown in the following diagram:

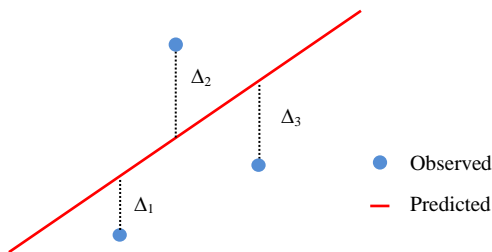


Fig 5: Explanation of Least Squares

## Gradient Descent:

Gradient descent is an optimization method used to find the optimal solution that provides the minimum value to the error function.

In general, the gradient descent method starts by giving an initial value to the required parameter ( $p$ ). Then, the partial derivative of error function ( $E$ ) with respect to parameter is used to update the value of parameter. The iterative process

continues until the optimal solution, that provides the minimum value to the error function, is reached.

The concept of gradient descent is explained in the following diagram:

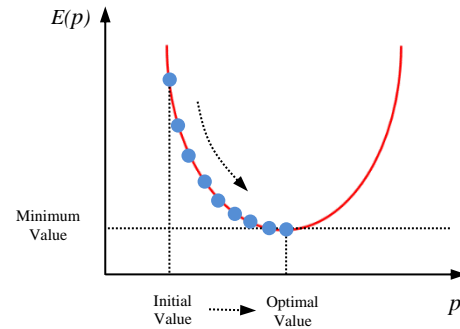


Fig 6: Explanation of Gradient Descent

The parameter ( $p$ ) is updated by the following formula:

$$p_{new} = p_{old} - \alpha \left( \frac{\partial E}{\partial p} \right) \quad (2)$$

where ( $p$ ) is the required parameter, ( $\alpha$ ) is the learning rate,  $\left( \frac{\partial E}{\partial p} \right)$  is the partial derivative of error function ( $E$ ) with respect to parameter ( $p$ ).

Now, applying that to the error function (MSE). The partial derivative of (MSE) with respect to slope ( $m$ ) is given by the following formula:

$$\frac{\partial \text{MSE}}{\partial m} = \left( \frac{-1}{n} \right) \sum (y - y_p)(x) \quad (3)$$

And, the partial derivative of (MSE) with respect to intercept ( $c$ ) is given by the following formula:

$$\frac{\partial \text{MSE}}{\partial c} = \left( \frac{-1}{n} \right) \sum (y - y_p) \quad (4)$$

By using formula (2), the slope ( $m$ ) is updated by the following formula:

$$m_{new} = m_{old} - \alpha \left( \frac{\partial \text{MSE}}{\partial m} \right) \quad (5)$$

And, the intercept ( $c$ ) is updated by the following formula:

$$c_{new} = c_{old} - \alpha \left( \frac{\partial \text{MSE}}{\partial c} \right) \quad (6)$$

The steps of the gradient descent method are explained in the following algorithm:

---

### Algorithm 1: Gradient Descent Method

---

```
# initialize slope
m = 0
# initialize intercept
c = 0
# learning rate
α = 0.0001
# number of iterations
```

```

Nt = 1000
for t = 0 to Nt do
    # compute predicted points
    yp = m * x + c
    # compute partial derivative w.r.t. slope
    dm = (-1/n) * Σ (y - yp) * x
    # compute partial derivative w.r.t. intercept
    dc = (-1/n) * Σ (y - yp)
    # update slope
    m = m - α * dm
    # update intercept
    c = c - α * dc
    # compute error function
    MSE = (1/n) * Σ (y - yp)2
end for

```

### R-Squared:

R-Squared ( $R^2$ ) is a statistical measure used to evaluate the linear regression model. It is computed by the following formula:

$$R^2 = 1 - \left( \frac{\sum (y - y_p)^2}{\sum (y - \bar{y})^2} \right) \quad (7)$$

where ( $y$ ) is the observed  $y$ -value, ( $y_p$ ) is the predicted  $y$ -value, and  $\bar{y}$  is the average of the observed  $y$ -values.

The  $R^2$  can take values in the range [0, 1], where (0) indicates that the line does not fit to the data points and (1) indicates that the line fits to the data points.

### Linear Regression System:

In the linear regression system:

**Input:** Observed points ( $X, Y$ ).

**Output:** Predicted line.

**Processing:** The observed points are obtained and prepared for processing. First, the slope and intercept are initialized to zero. Then, the predicted points are computed for the current values of slope and intercept. After that, the partial derivatives of error function are used to update the values of slope and intercept. During that, the error function is computed to make sure that the model is converging well to the optimal solution. Finally, the optimal values of slope and intercept are obtained to form the equation of line and plot the predicted line.

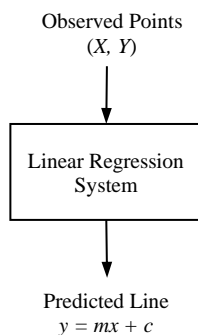


Fig 7: Diagram of Linear Regression System

### Python:

Python [32] is a general high-level programming language. It is simple, easy to learn, and powerful. It is the most preferred programming language by the developers of machine learning applications.

Python provides many useful additional libraries such as: Numpy [33], Pandas [34], Matplotlib [35], NLTK [36], SciPy [37], and SK Learn [38].

In this research, the standard functions of Python are applied without using any additional library.

### 3. RESEARCH METHODOLOGY

The basic steps of linear regression are: (1) preparing observed points, (2) initializing slope and intercept, (3) computing predicted points, (4) computing partial derivatives, (5) updating slope and intercept, (6) computing error function, (7) making equation of line, and (8) plotting predicted line.

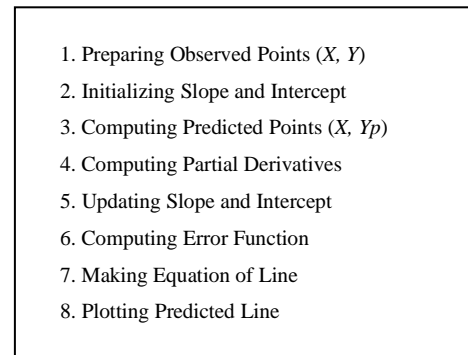


Fig 8: Steps of Linear Regression

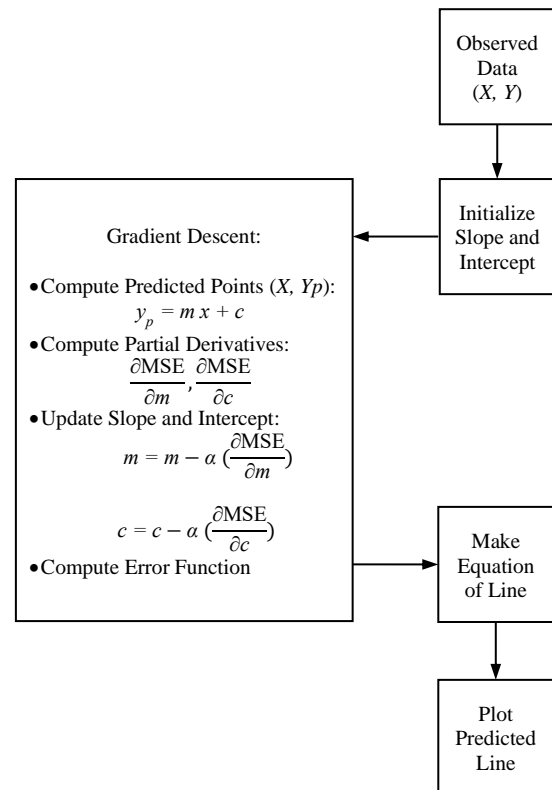


Fig 9: Flowchart of Linear Regression

The steps of linear regression using least squares and gradient descent are explained in the following section.

### 1. Preparing Observed Points:

The observed points ( $X, Y$ ) are obtained from the original source and converted into lists in the following form:

```
X = [x0, x1, x2, ..., xn]
Y = [y0, y1, y2, ..., yn]
```

### 2. Initializing Slope and Intercept:

The slope ( $m$ ) and intercept ( $c$ ) are initialized to zero as shown in the following code:

```
m = 0
c = 0
```

### 3. Computing Predicted Points:

The predicted points ( $X, Yp$ ) are computed for the current values of slope ( $m$ ) and intercept ( $c$ ). It is done by the following code:

```
for i in range(n):
    Yp[i] = m*X[i] + c
```

### 4. Computing Partial Derivatives:

The partial derivative of error function with respect to slope ( $dm$ ) is computed by the following code:

```
def compute_dm(X, Y, Yp):
    sum = 0
    for i in range(n):
        sum += (Y[i] - Yp[i])*X[i]
    return (-1/n)*sum
```

The partial derivative of error function with respect to intercept ( $dc$ ) is computed by the following code:

```
def compute_dc(Y, Yp):
    sum = 0
    for i in range(n):
        sum += (Y[i] - Yp[i])
    return (-1/n)*sum
```

### 5. Updating Slope and Intercept:

The values of slope ( $m$ ) and intercept ( $c$ ) are updated by the following code:

```
m = m - alpha * dm
c = c - alpha * dc
```

### 6. Computing Error Function:

The error function (MSE) is computed by the following code:

```
def compute_MSE(Y, Yp):
    sum = 0
    for i in range(n):
        sum += (Y[i] - Yp[i])**2
    return sum/n
```

### 7. Making Equation of Line:

The equation of line is obtained as shown in the following form:

```
y = m * x + c
```

### 8. Plotting Predicted Line:

The predicted line is plotted using the "matplotlib" library. It is done by the following code:

```
import matplotlib.pyplot as plt

plt.scatter(X, Y)
plt.plot(X, Yp, color="red")
plt.show()
```

## 4. RESULTS AND DISCUSSION

The developed program was tested on an experimental dataset from Kaggle [39]. The program performed the basic steps of linear regression using least squares and gradient descent and provided the required results. The program output is explained in the following section.

### Observed Points:

The observed points ( $X, Y$ ) are printed as shown in the following view:

	X	Y
0	25.128	53.454
1	31.588	50.393
2	32.502	31.707
3	32.669	45.571
4	32.94	67.171
5	33.094	50.72
6	33.645	69.9
7	33.864	52.725
8	34.333	55.723
9	35.568	41.413
10	35.678	52.722
...		

### Processing Gradient Descent:

The gradient descent method is processed (1,000) iterations. For each iteration; the slope ( $m$ ), intercept ( $c$ ), and error function (MSE) are printed as shown in the following view:

t	m	c	MSE
0	0.368535	0.007274	5565.107834
100	1.478861	0.032102	112.648861
200	1.478802	0.035105	112.647057
300	1.478743	0.038107	112.645254
400	1.478684	0.041108	112.643452
500	1.478625	0.044107	112.641652
600	1.478566	0.047106	112.639853
700	1.478507	0.050103	112.638055
800	1.478448	0.053099	112.636259
900	1.47839	0.056095	112.634464

A summary of the results is shown in the following view:

```
Nt = 1000
m = 1.478331327454368
c = 0.0590585566568512
MSE = 112.63268870038829
R2 = 0.59001
```

### Predicted Points:

The final values of the predicted points ( $X, Yp$ ) are printed as shown in the following view:

	X	Yp
0	25.128	37.207
1	31.588	46.757
2	32.502	48.108
3	32.669	48.355
4	32.94	48.756

5	33.094	48.983
6	33.645	49.797
7	33.864	50.122
8	34.333	50.815
9	35.568	52.64
10	35.678	52.803
...		

### Error Function Plot:

The error function (MSE) is plotted as shown in the following chart:

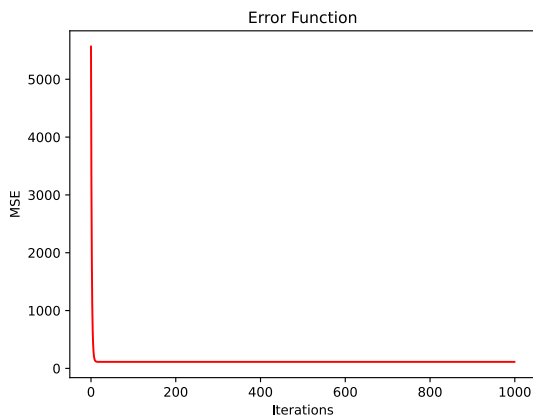


Fig 10: Error Function Plot

The error function plot shows that MSE is decreasing with iterations which indicates that the linear regression model is converging well to the optimal solution.

### Equation of Line:

The equation of line is formed and printed as shown in the following view:

$$y = 1.478331 x + 0.059059$$

### Predicted Line:

The predicted line is plotted as shown in the following chart:

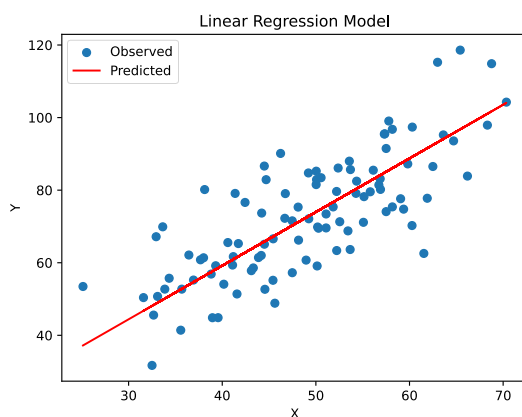


Fig 11: Linear Regression Model

The  $R^2$  value is (0.59001) which indicates that the predicted line fits to the observed points.

### Improving the Results:

To improve the results of the program, the number of iterations is increased 10 times to (10,000) iterations. The results of run (2) are shown in the following view:

Nt	=	10000
m	=	1.4731251028822563
c	=	0.3239430786286303
MSE	=	112.47669301199636
R2	=	0.590577

The equation of line is shown in the following view:

$$y = 1.473125 x + 0.323943$$

The  $R^2$  value is (0.590577) which indicates that the resulting line fits to the data points better than the previous run.

To improve the results more, the number of iterations is increased 100 times to (100,000) iterations. The results of run (3) are shown in the following view:

Nt	=	100000
m	=	1.4297282427708158
c	=	2.531907100632597
MSE	=	111.38251276556622
R2	=	0.59456

The equation of line is shown in the following view:

$$y = 1.429728 x + 2.531907$$

The  $R^2$  value is (0.59456) which indicates that the resulting line fits to the data points better than the previous runs.

A comparison of  $R^2$  between the three runs of the linear regression model is shown in the following table:

Table 1: Comparison of  $R^2$  between The Three Runs

	Linear Regression Model		
	Run 1	Run 2	Run 3
$R^2$	0.59001	0.590577	0.59456

In summary, the program output clearly shows that the program has successfully performed the basic steps of linear regression using least squares and gradient descent and provided the required results.

## 5. CONCLUSION

Machine learning is playing a major role in the development of computer systems. Linear regression is an important application of machine learning. It helps to find the line that best fits the data points. Linear regression is used to model the relationship between the independent variable (x) and the dependent variable (y) using a linear polynomial with slope (m) and intercept (c). Least squares is used to minimize the error between the observed and predicted points. Gradient descent is used to find the optimal solution that provides the minimum value of error function.

In this research, the author developed a program to perform linear regression using least squares and gradient descent in Python. The developed program performed the basic steps of linear regression: preparing observed points, initializing slope and intercept, computing predicted points, computing partial derivatives, updating slope and intercept, computing error function, making equation of line, and plotting predicted line.

The program was tested on an experimental dataset from Kaggle and provided the required results: computed slope and

intercept, computed error function, predicted points, equation of line, and predicted line.

In future work, more research is needed to improve and develop the current methods of linear regression using least squares and gradient descent. In addition, they should be more investigated on different fields, domains, and datasets.

## 6. REFERENCES

- [1] Sammut, C., & Webb, G. I. (2011). "Encyclopedia of Machine Learning". Springer Science & Business Media.
- [2] Jung, A. (2022). "Machine Learning: The Basics". Singapore: Springer.
- [3] Kubat, M. (2021). "An Introduction to Machine Learning". Cham, Switzerland: Springer International Publishing.
- [4] Dey, A. (2016). "Machine Learning Algorithms: A Review". *International Journal of Computer Science and Information Technologies*, 7 (3), 1174-1179.
- [5] Jordan, M. I., & Mitchell, T. M. (2015). "Machine Learning: Trends, Perspectives, and Prospects". *Science*, 349 (6245), 255-260.
- [6] Forsyth, D. (2019). "Applied Machine Learning". Cham: Springer International Publishing.
- [7] Chopra, D., & Khurana, R. (2023). "Introduction to Machine Learning with Python". Bentham Science Publishers.
- [8] Sarker, I. H. (2021). "Machine Learning: Algorithms, Real-world Applications and Research Directions". *SN Computer Science*, 2(3), 160.
- [9] Das, S., Dey, A., Pal, A., & Roy, N. (2015). "Applications of Artificial Intelligence in Machine Learning: Review and Prospect". *International Journal of Computer Applications*, 115(9), 31-41.
- [10] Dhall, D., Kaur, R., & Juneja, M. (2020). "Machine Learning: A Review of the Algorithms and its Applications". *Proceedings of ICRI 2019: Recent Innovations in Computing*, 47-63.
- [11] Raschka, S. (2015). "Python Machine Learning". Packt Publishing.
- [12] Müller, A. C., & Guido, S. (2016). "Introduction to Machine Learning with Python: A Guide for Data Scientists". O'Reilly Media.
- [13] Swamynathan, M. (2019). "Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics using Python". Apress.
- [14] Brandt, S. (2014). "Data Analysis: Statistical and Computational Methods for Scientists and Engineers". Springer.
- [15] VanderPlas, J. (2017). "Python Data Science Handbook: Essential Tools for Working with Data". O'Reilly Media.
- [16] Atkinson, K. (1989). "An Introduction to Numerical Analysis". John Wiley & Sons.
- [17] Chapra, S. C. (2010). "Numerical Methods for Engineers". McGraw-Hill.
- [18] Gray, J.B. (2002). "Introduction to Linear Regression Analysis". *Technometrics*, 44, 191-192.
- [19] Groß, J. (2003). "Linear Regression". Springer Science & Business Media.
- [20] Olive, D. J. (2017). "Linear Regression". Berlin: Springer International Publishing.
- [21] Yan, X., & Su, X. (2009). "Linear Regression Analysis: Theory and Computing". World Scientific.
- [22] Su, X., Yan, X., & Tsai, C. L. (2012). "Linear Regression". *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3), 275-294.
- [23] Montgomery, D.C., Peck, E.A., Vining G. G. (20012). "Introduction to Linear Regression Analysis". *Wiley Series in Probability and Statistics: John Wiley & Sons*.
- [24] Kutner, N., Nachtsheim, C., & Neter, J. (2004). "Applied Linear Regression Models". *McGraw-Hill/Irwin Series: Operations and Decision Sciences*.
- [25] Leemis, L.M. (1991). "Applied Linear Regression Models". *Journal of Quality Technology*, 23, 76-77.
- [26] Seber, G. A., & Lee, A. J. (2003). "Linear Regression Analysis". John Wiley & Sons.
- [27] Neter, J., Wasserman, W., & Kutner, M. H. (1983). "Applied Linear Regression Models". Irwin.
- [28] Weisberg, S. (2005). "Applied Linear Regression". John Wiley & Sons.
- [29] Malik, M.B. (2005). "Applied Linear Regression". *Technometrics*, 47, 371-372.
- [30] Maulud, D., & Abdulazeez, A. M. (2020). "A Review on Linear Regression Comprehensive in Machine Learning". *Journal of Applied Science and Technology Trends*, 1(4), 140-147.
- [31] Stigler, Stephen M. (1981). "Gauss and the Invention of Least Squares". *The Annals of Statistics*. 9 (3): 465–474.
- [32] Python: <https://www.python.org>
- [33] Numpy: <https://www.numpy.org>
- [34] Pandas: <https://pandas.pydata.org>
- [35] Matplotlib: <https://www.matplotlib.org>
- [36] NLTK: <https://www.nltk.org>
- [37] SciPy: <https://scipy.org>
- [38] SK Learn: <https://scikit-learn.org>
- [39] Kaggle: <https://www.kaggle.com>