SUPERVISED LEARNING REGRESSION ALGORITHM SIMPLE LINEAR REGRESSION

Salimov Jamshid Obid o'g'li
Assistant
jamshidsalimov8@gmail.com

Xolmurodov Abduxoliq Nodir o'g'li
Student
abdixoliq994@gmail.com

Bobomurotov Jafarbek Jasurjonovich
Student
jafarbekbobomurotov@gmail.com
Jizzakh branch of National University of Uzbekistan

Annotation
In this article, the main task of Simple Linear Regression is to establish a linear relationship between (x) the independent variable and (y) the dependent variable. And predicting the value of the dependent variable based on the value of the independent variable

Keywords: regression,model,predictor, not available,test set, training set.

English mathematician and statistician Sir Francis Galton, who first introduced the concept in the late 19th century. Galton used the method to study the relationship between the heights of parents and their children, and he coined the term "regression" to describe how the children's heights tended to move towards the average height of the population, or the mean.

Later, the work of the American statistician and social scientist Francis Anscombe in the 20th century contributed to the popularization of simple linear regression, particularly through his influential paper "Graphs in Statistical Analysis" published in 1973. In this paper, Anscombe showed the importance of visualizing data and using regression analysis to model the relationship between variables.

Today, simple linear regression is widely used in various fields such as finance, economics, social sciences, and engineering to analyze the relationship between two variables and to make predictions based on that relationship.

SIMPLE LINEAR REGRESSION

Simple Linear Regression is a type of Regression algorithms that models the relationship between a dependent variable and a single independent variable. The relationship shown by a Simple Linear Regression model is linear or a sloped straight line, hence it is called Simple Linear Regression.

Simple Linear regression algorithm has mainly two objectives:

Model the relationship between the two variables. Such as the relationship between Income and expenditure, experience and Salary, etc.

Forecasting new observations. Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

Simple Linear Regression Model:

The Simple Linear Regression model can be represented using the below equation:

$$\hat{y} = \theta_0 + \theta_1 x_1$$

- $\hat{y}$ - **predicted values**
- θ_0 - **It is the intercept of the Regression line (can be obtained putting x=0)**
- θ_1 - **It is the slope of the regression line, which tells whether th line is increasing or decreasing.**
- x_1 - **independent variable**

Implementation of Simple Linear Regression Algorithm using Python

To implement the Simple Linear regression model in machine learning using Python, we need to follow the below steps:

Data Pre-processing- this includes removing unusual values, extracting the desired data, removing NaN values, and determining their data type and converting them to the desired data type.

Find a correlation between columns and visual analysis

```
# we call the necessary libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as mtp
%matplotlib inline
```

```
#download the required DataSet
df=pd.read_csv("https://raw.githubusercontent.com/JamshidSalimov/Ai-Fayls/master/my_rooms.csv", index_col=0)
df.head()
```

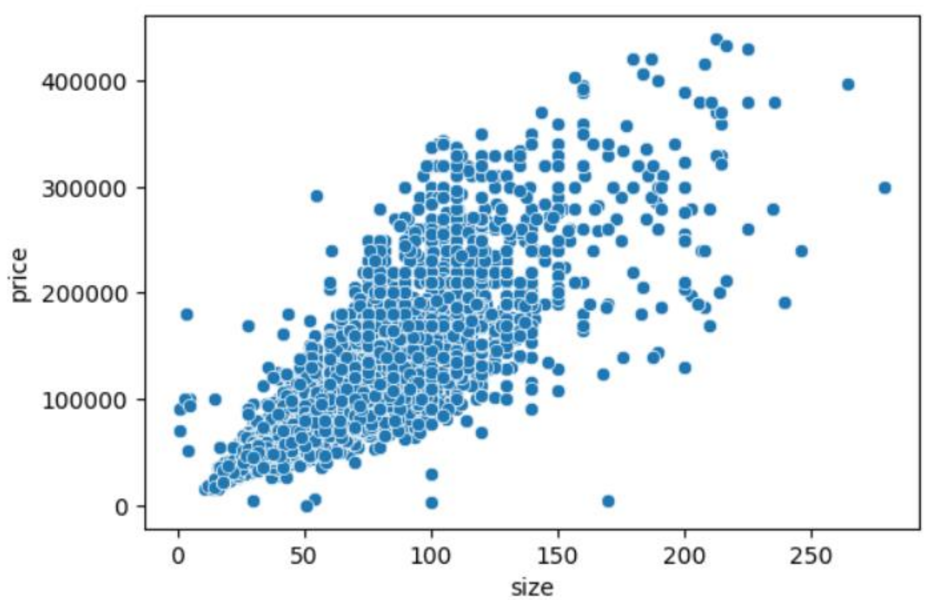
	district	size	level	max_levels	price
0	Юнусабадский	57.0	4	4	104000.0
1	Яккасарайский	52.0	4	5	112000.0
2	Чиланзарский	42.0	4	4	74000.0
3	Чиланзарский	65.0	1	4	99000.0
4	Чиланзарский	70.0	3	5	110000.0

```
# Find a correlation between columns
df.corrwith(df["price"])
```

```
size          0.794788
level         0.079509
max_levels    0.254600
price         1.000000
dtype: float64
```

The column data with the highest correlation is selected as the independent variable. And this is the **size** column data

```
[36] # visual analysis
plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x="size",y="price")
plt.show()
```



Splitting the dataset into training and test set.

```
[39] # Splitting the dataset into training and test set.
     from sklearn.model_selection import train_test_split
     traen_set, test_set =train_test_split(df, test_size=0.15, random_state=40)
```

Using the train_test_split function, we split 15% of the data for training the model and 85% of the data for testing the model

Fitting the Simple Linear Regression to the Training Set:

Fit our model to the training dataset. To do so, we will import the LinearRegression class of the linear_model library from the scikit learn. After importing the class, we are going to create an object of the class named as a regressor.

```
[1] #Fitting the Simple Linear Regression model to the training dataset
    from sklearn import linear_model
    LR_model= linear_model.LinearRegression()
```

```
[ ] # Convert size and price column data to array view
    x_train=np.asanyarray(traen_set[["size"]])
    y_train=np.asanyarray(traen_set[["price"]])
```

```
# Fitting the Simple Linear Regression
LR_model.fit(x_train,y_train)

▾ LinearRegression
LinearRegression()
```

The **LinearRegression()** model also works for **Simple Linear Regression** and defines θ_0 and θ_1. It is also possible to calculate T0 and T1 in SLR as follows.

$$\theta_1 = \frac{\sum_{i=1}^{s}(x_i - \tilde{x})(y_i - \tilde{y})}{\sum_{i=1}^{s}(x_i - \tilde{x})^2}$$

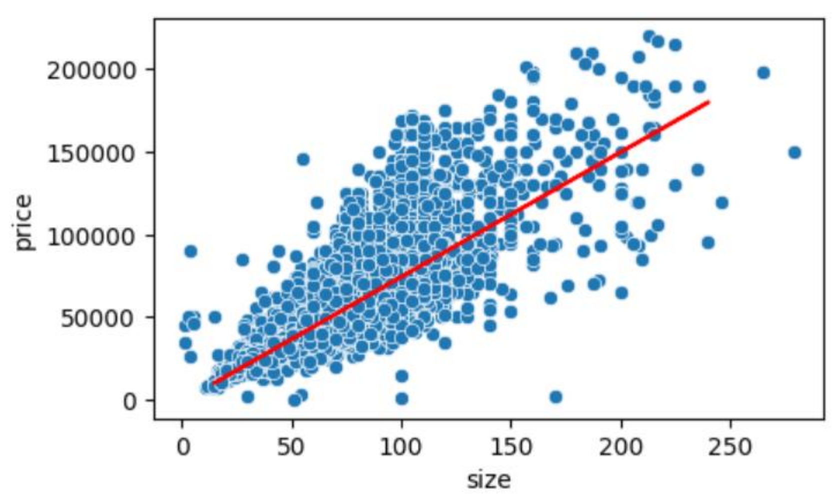
$$\theta_0 = \bar{y} - \theta_1 \tilde{x}$$

```
# tetta1 va tetta0 ni modeldan ajratib olish
tetta1= LR_model.coef_[0][0]
tetta0= LR_model.intercept_[0]
print(tetta1)
print(tetta0)
```

```
753.2195147265451
-955.7940269998871
```

Visualizing the Training set results:

```
[25] # Draw a graph based on thetta1 and thetta0
     plt.figure(figsize=(5,3))
     sns.scatterplot(data=df, x="size",y="price")
     plt.plot(x_train,tetta0+x_train*tetta1, color="r")
     plt.show()
```

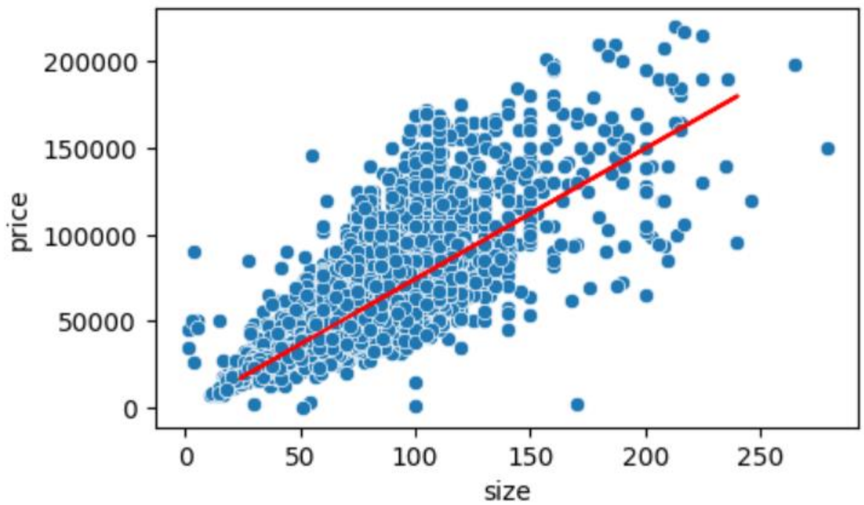



```
# Convert size and price column data to array view
x_test=np.asanyarray(test_set[["size"]])
y_test=np.asanyarray(test_set[["price"]])
```

Visualizing the Test set results:

EMERGENT: JOURNAL OF EDUCATIONAL DISCOVERIES AND LIFELONG LEARNING (EJEDL)
ISSN: 2776-0995 Volume 4, Issue 4, April, 2023

```
# visualizing the Test set results:
plt.figure(figsize=(5,3))
sns.scatterplot(data=df, x="size",y="price")
plt.plot(x_test,tetta0+x_test*tetta1, color="r")
plt.show()
```



Determination of predictive values using the model

```
# Determination of predictive values using the model
y_pridect=LR_model.predict(x_test)
y_pridect
```

```
array([[ 39718.05976823],
       [ 30679.42559152],
       [ 26160.10850316],
       [ 44237.37685659],
       [ 35198.74267987],
       [ 50263.13297441],
       [ 27666.54753261],
       [ 48003.47443023],
```

Model evaluation (Calculate the model error)
RMSE (root mean square error)
MAE (mean absolute error)

$$\mathrm{MAE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^{m} \left| h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right|$$

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

```python
# call the necessary functions of sklearn to calculate the error
from sklearn.metrics import mean_absolute_error, mean_squared_error
MAE=mean_absolute_error(y_pridect, y_test)
RMSE= np.sqrt(mean_squared_error(y_pridect, y_test))
print("MAE",MAE)
print("RMSE",RMSE)
```

```
MAE 7651.904713026272
RMSE 12977.44580854305
```

References

1. Amrullayevich K. A., Obid o'g'li S. J. ELEKTRON TALIM MUHITIDA TALABALARDA AXBOROT BILAN ISHLASH KOMPETENTLIKNI SHAKLLANTIRISH //International Journal of Contemporary Scientific and Technical Research. – 2022. – С. 641-645.
2. Obid o'g A. S. J. et al. Numpy Library Capabilities. Vectorized Calculation In Numpy Va Type Of Information //Eurasian Research Bulletin. – 2022. – Т. 15. – С. 132-137.
3. Javlon X. et al. Классификатор движения рук с использованием биомиметического распознавания образов с помощью сверточных нейронных сетей с методом динамического порога для извлечения движения с использованием датчиков EF //Journal of new century innovations. – 2022. – Т. 19. – №. 6. – С. 352-357.
4. Фитратович В. и др. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МНОГОФАЗНОЙ ФИЛЬТРАЦИИ В НЕФТЕГАЗОВОМ ПЛАСТЕ ПРИ ЕГО ЗАВОДНЕНИИ //INTERNATIONAL CONFERENCES ON LEARNING AND TEACHING. – 2022. – Т. 1. – №. 4. – С. 520-525.
5. Yusupbekov N. et al. EXERGET ANALYSIS OF ENERGY TECHNOLOGICAL INSTALLATIONS //InterConf. – 2021.
6. Ergashev, S. B., & Baxtiyor o'g'li, E. S. (2022). DESIGN OF AUTOMATED ENTERPRISE INFORMATION SYSTEMS USING UML DIAGRAMS IN THE CREATION OF APPLICATIONS. Innovative Technologica: Methodical Research Journal, 3(12), 25-33.
7. Tulqinova, Z., & Ergashev, S. (2023). DESIGN OF INFORMATION SYSTEMS. BPMN 2.0 THERMAL POWER STATIONS. Science and innovation, 2(A2), 250-257.

8. Naim o'g'li, M. D., & Baxtiyor o'g'li, E. S. (2022). DATA SCIENCE METHODOLOGY IN LEARNING PROGRAMMING. JOURNAL OF INNOVATIONS IN SCIENTIFIC AND EDUCATIONAL RESEARCH, 2(13), 207-210.
9. Тавбоев С. А. и др. НЕКОТОРЫЕ МЕТОДЫ И ЗАДАЧИ ЦИФРОВОЙ ОБРАБОТКИ И РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ //International Journal of Contemporary Scientific and Technical Research. – 2022. – С. 334-339.
10. Abdurahimovich A. A., Kamoliddin o'g'li M. A. SANOQ SISTEMALARIDA VAQT TUSHUNCHASI //International Journal of Contemporary Scientific and Technical Research. – 2022. – С. 331-334.
11. Tavboyev Sirojiddin Akhbutayevich, Mamaraimov Abror Kamoliddin ugli, and Karshibaev Nizomiddin Abdumalikovich, "Algorithms for Selecting the Contour Lines of Images Based on the Theory of Fuzzy Sets", TJET, vol. 15, pp. 31–40, Dec. 2022.
12. Mamatkulovich B. B. A DESIGN OF SMALL SCALE DEEP CNN MODEL FOR FACIAL EXPRESSION RECOGNITION USING THE LOW-RESOLUTION IMAGE DATASETS //MODELS AND METHODS FOR INCREASING THE EFFICIENCY OF INNOVATIVE RESEARCH. – 2023. – Т. 2. – №. 19. – С. 284-288.
13. Mamatkulovich B. B. AUTOMATIC STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITON //Next Scientists Conferences. – 2022. – С. 6-22.
14. Mamatkulovich B. B. LIGHTWEIGHT RESIDUAL LAYERS BASED CONVOLUTIONAL NEURAL NETWORKS FOR TRAFFIC SIGN RECOGNITION //European International Journal of Multidisciplinary Research and Management Studies. – 2022. – Т. 2. – №. 05. – С. 88-94.
15. Amrullayevich K. A., Mamatkulovich B. B. TALABALARDA AXBOROT BILAN ISHLASH KOMPETENTSIYASINI SHAKILLANTIRISHDA DIDAKTIK VOSITALARINING METODIK XUSUSIYATLARIDAN FOYDALANISH //International Journal of Contemporary Scientific and Technical Research. – 2022. – С. 645-650.
16. Ikromovich H. O., Mamatkulovich B. B. FACIAL RECOGNITION USING TRANSFER LEARNING IN THE DEEP CNN //Open Access Repository. – 2023. – Т. 4. – №. 3. – С. 502-507.
17. Mamatkulovich B. B., Shuhrat o'g'li M. S., Jasurjonovich B. J. SPECIAL DEEP CNN DESIGN FOR FACIAL EXPRESSION CLASSIFICATION WITH A SMALL AMOUNT OF DATA //Open Access Repository. – 2023. – Т. 4. – №. 3. – С. 472-478.
18. Mamatkulovich B. B., Alijon o'g'li H. A. Facial Image-Based Gender and Age Estimation //Eurasian Scientific Herald. – 2023. – Т. 18. – С. 47-50.
19. Choryorqulov G'.H., & Qosimov N.S. (2023). ELEKTRON JADVAL MODELINING TAVSIFLANISHI. PEDAGOGS Jurnali, 30(3), 67–73. Retrieved from.