Linear Regression: Deep Learning

Olaniyi Bayonle Alao
Summer Term, 2021
Bachelor of Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
olaniyi-bayonle.alao@stud.hshl.de

***Abstract*—Machine learning refers to applications' ability to get better at doing things without necessarily changing the codebase. Even though machine learning is not a new concept, it is fast gaining recognition and changing lives thanks to the increase in the processing power of computers over the years to process big data at a level that has never been experienced. This paper talks about linear regression in the context of machine learning. Linear regression is a statistical term that uses a dependent and independent variable to make predictions. This paper explores the theoretical background of linear regression, solves a regression problem using machine learning for training a model, and predicts values using the trained model using Scikit-Learn, a scientific library for machine learning in python. The paper uses the California housing datasets from the Scikit-Learn dataset repository for training the linear regression machine learning model. As a measure of how good the machine learning model has learnt, mean squared error and coefficient of determination were explained and determined. In conclusion, the paper evaluates the trained linear regression model to understand how well it performed in making predictions.**

***Index Terms*—machine learning, linear regression, deep learning, sci-kit**

I. Introduction

Due to an explosive increase in the amount of data generated from different internet-connected devices - Cyber-Physical-Systems (CPS) -, there has been an increase in the need to make sense of this "big data" to ensure the proper and productive functioning of businesses. By 2026, the data generated from the global air fleet is projected to reach 98 billion gigabytes [1]. Thanks to the recent advancements in the information processing capabilities of computers, making sense of this enormous chunk of data is now possible using data analysis techniques and machine learning algorithms to clean and make a prediction from these datasets. Machine learning is a machine's - software application - ability to learn and make predictions without being programmed to do so. They are able to learn and get better with an increase in the amount of data they are being fed through their grounded mathematical foundations.

Linear regression is a statistical test used to ascertain the relationship between independent and dependent variables in a dataset using a mathematical formula. They can also be used in projecting a new relationship between the dependent and independent variables that have not been discovered. Even though linear models - linear regression - are pretty simple to develop and understand, as well as good at predicting linear relationships, their approximation of nonlinear relationships is mostly unsatisfactory [7].

This paper gives a profound yet understandable mathematical description of Linear regression. Relevant parts of the scientific machine learning framework -SciKit Learn- used to perform linear regression predictions on the example dataset in machine learning using Python programming language.

The theoretical background section of this paper gives in-depth information about fundamental theories in understanding the machine learning field and linear regression. The algorithm implementation section of the paper discusses the libraries and dataset used in implementing the linear regression machine learning model trained in this paper. A practical example of preparing the dataset, splitting the dataset, training the machine learning model, as well as testing out the trained model, its accuracy and interpretation of the predictions were presented in the practical and result and discussion section of the paper.

Unless otherwise stated, the main reference for facts, mathematical models and terminologies used in this research paper is from [8].

II. Theoretical Background

A. Machine Learning

According to [3], "Machine learning is a branch of artificial intelligence (AI) focused on building applications that learn from data and improve their accuracy over time without being programmed to do so". Examples of applications built using machine learning trained models are email filters with the ability to distinguish between desired emails and spams, auto-suggestion/auto-correct in many typing applications, self-driving vehicles, amongst others. Machine learning techniques have the advantage of automatically adapting to change detected in trends from data. Likewise, they help reduce the complexity of writing applications that are sometimes difficult or even impossible to implement using conventional algorithms. The approach of solving an application's complexity using machine learning is shown in figure 1. As depicted in figure 1, machine learning algorithms do the job of find solutions to the studied problem and evaluation of the solution is done before being deployed. Methods used in training a Machine learning model can be grouped into three main categories.

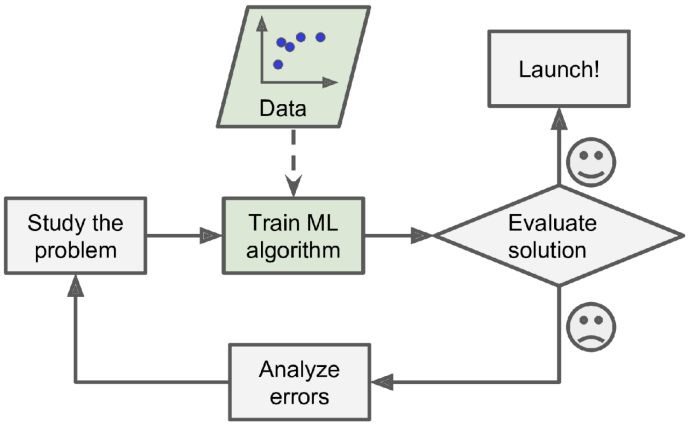

Fig. 1. Machine Learning approach [4]

- Supervised Learning
  This is a type of machine learning technique in which the algorithm is fed with an input and output dataset of the desired solution. The desired output data is otherwise known as labels. This information help train the machine learning model in the precise output prediction when given a related output after being trained. Some examples of standard algorithms used for supervised learnings are regression analysis - linear regression, logistic regression -, decision trees. [11] The two kinds of supervised learning techniques used are classification and regression.
- Unsupervised Learning
  This refers to the machine learning technique in which the algorithm is only fed with input data for the training process. The model learns "unsupervised" by finding out and grouping patterns in inputted datasets. Unlike supervised learning, they require many unlabeled datasets to train - properly find patterns in datasets.
- Semisupervised Learning
  This learning technique uses both labeled - input and desired output - and unlabeled - only input - datasets for training machine learning models.

The other methods used in training a machine learning model are reinforcement learning and deep learning. Deep learning is a subset of machine learning whose algorithm defines an artificial neural network - ANN - that is designed to emulate the way a human brain learns and uses unsupervised or semisupervised learning technique to train [3]. On the other hand, reinforcement learning is reward-based learning because the machine learning model is trained by being given a point for reaction to specific events.

B. Linear Regression Model

Regression is a predictive analysis with a long but glorious history from its successful applications to problems in the statistics and economics domain. Regression is a kind of supervised learning technique for determining the best fit line to describe patterns in data: linear regression uses a straight line to describe these patterns. [11]. The best fit line is the line that reduces the summed squared difference between the value of the line of a certain value $\boldsymbol{x}$ and its corresponding $\boldsymbol{y}$ values [8]. The mathematical expression for linear regression is:

$$y = \beta X + \beta_0 \tag{1}$$

Where y is the dependent variable, X is the independent variable of the equation, β is a cofficient which reperesnts the slope of the regression line, β_0 is a constant value called the **bias**. Equation (1) is the same as the equation of a straight line in linear algebra.

The independent variable X in (1) is calculated using the mathematical expression below:

$$X = \frac{\sum (x - \overline{x})(y - \overline{y})}{\sum (x - \overline{x})^2} \tag{2}$$

Where $\overline{x}$ and $\overline{y}$ are the mean values of all the respective x and y variables.

Linear regression analysis can be sub-divided into simple and multiple linear regression depending on the number of their independent variables. If the independent variable is one, the regression is a simple linear regression, but it is multiple linear regression if the independent variable is more than one. In this paper, we will make an analysis using the multiple linear regression model.

C. Pearson's Correlation Coefficient

According to [6], "Pearson's correlation coefficient (r) is a measure of the linear association of two variables". Equation (3) shows the formula for calculating Pearson's correlation coefficient

$$r = \frac{\sum (x - \overline{x})(y - \overline{y})}{\sqrt{\sum (x - \overline{x})^2 \sum (y - \overline{y})^2}} \tag{3}$$

Where:

x = values of the x variable
y = values of the y variable
$\overline{x}$ = mean values of all the x variables
$\overline{y}$ = mean values of all the y variables
r = pearson's correlation coefficient

Correlation analysis usually starts with a graphical representation of the relation of data pairs using a scatter diagram. The correlation coefficient varies from a value of –1 to +1. A value of 1 indicates that the linear model accurately describes the relationship between the dependent and independent variables. The lower the value gets towards the negative value indicates that the dependent variable decreases with an increase in the value of the independent variable, meaning that there is a negative linear correlation between the variables. The correlation coefficient value of 0 indicates no correlation between the dependent and independent variables.

D. Model Evaluation

The prediction made by the model using equation (1) can be evaluated using the statistical methods below to find out how close the predicted value is to the actual value.

1) R^2: otherwise known as the coefficient of determination or the coefficient of multiple determination, which is a measure of how close the best fit line is to the original data using a simple mean. The value of the output of this calculation ranges from 0 to 1 [10]. A value of 1 refers to the fact that all the points fit 100% to the regression line. The lesser the value, the farther the points are to the regression line. The following equation defines the coefficient of determination:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y_i} - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{4}$$

Where n is the number of observations, i.e. total number of variables, $\hat{y_i}$ is the estimated value of the dependent variable for the i[th] observation computed by the regression equation, y_i is the observed value of the dependent variable for the i[th] observation and $\overline{y}$ is the mean of all n observations of the dependent variable [10]

2) Mean Squared Error: Mean Squared Error (MSE) is a performance measure that helps determine how much error is made in the predicted value in relation to the actual value of the output given the same input. The equation of the calculation is denoted by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{5}$$

where x_i is a vector of all i[ith] observation, $\hat{f}$ is the prediction function. The lower the value of MSE, the more accurate the prediction is.

III. Algorithm Implementation

All the algorithm needed for prediction using the linear regression model has been implemented in machine learning libraries like Scikit-Learn, which we will be using in this paper.

A. Libraries and Tools used

1) Scikit Learn: Scikit learn is an open-source python library that provides an implementation of different mathematical models used for solving machine learning problems. This library provides functionalities for solving machine learning jobs like regression, classification, clustering, model selection, pre-processing - like splitting datasets into test and train subsets - amongst others [9]. The Scikit-Learn APIs are designed around these main design principles, which are consistency - all objects (basic or composite) share a consistent interface composed of a limited set of methods -, inspection - parameters are exposed as public attributes -, composition, sensible defaults - provides understandable default parameters which give baseline solution for tasks at hand -and nonproliferation of classes - datasets are represented as NumPy arrays or SciPy sparse matrices [2].

2) Numpy: NumPy is an open-source Python library that provides methods that allow for fast operations on multidimensional arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, introductory linear algebra, basic statistical operations, random simulation[1].

3) Pandas: Pandas is an open-source data analysis and manipulation python library that allows high-performance, easy examination, cleaning and processing of tabular data structures in Python[2]. The panda library finds its strength in helping to transform datasets to desired formats for machine learning jobs or other applications.

4) Matplotlib: Matplotlib is a Python library used along with Numpy for creating static, animated and interactive visualisation of data[3]. This library is helpful in creating and customising data visualisations.

5) Seaborn: Seaborn is a Python library that builds upon the functionalities of Matplotlib and integrates closely with pandas data structures for making statistical graphics to datasets[4].

Figure 2 shows the import statments for the functions and libraries mentioned above in python.

```
from sklearn.datasets import fetch_california_housing
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
%matplotlib inline
```

Fig. 2. Import statements

IV. Practical Example

In this paper, the California housing datasets from [5] included in the Scikit-Learn sklearn datasets library is used to train and predict the average prices of housing in California using linear regression.

A. Inspecting Dataset

The dataset has 20,640 instance number, 8 numeric predictive attributes and 1 target attribute, i.e. the expected output of the prediction, which is the median house value for California districts. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people); the datasets were obtained in a 1990 census [5]. The predictive attributes have the following information:

- MedInc: represents the median income in a block.
- HouseAge: represents the median house age in block.
- AveRooms: represents the average number of rooms.

[1]https://numpy.org/doc/stable/user/whatisnumpy.html
[2]https://pandas.pydata.org/docs/
[3]https://matplotlib.org/
[4]https://seaborn.pydata.org/introduction.html

- AveBedrms: represents the average number of bedrooms.
- Population: represents the population in the block.
- AveOccup: represents the average house occupancy.
- Latitude: represents the house block latitude.
- Longitude: represents the house block longitude.

A summary of these attributes and the corresponding target values as pandas data frame is represented in figure 3. The target attribute represents the median house value in a unit of 100,000.

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422
...	...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894

20640 rows × 9 columns

Fig. 3. Dataset overview including the target column

The correlation between the attributes is shown in figure 4 and 5. Figure 4 shows the Pearson correlation between the attributes as a heatmap and figure 5 as a numerical table. The darker the blue colour and the closer the value is to 1, the higher the correlation of the attribute to each other. It can be seen from figure 4 that the median income in a block has the highest correlation to the target value followed by house age and average rooms. The other values show little or no useful correlations to the target value because of their negative values.

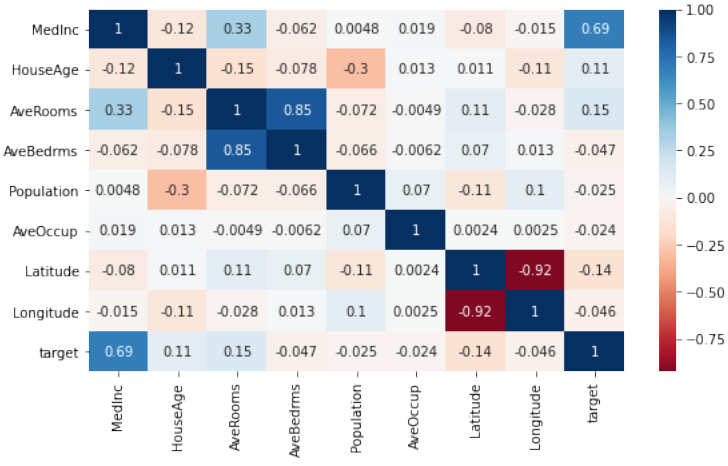

Fig. 4. Pearson correlation heatmap of the dataset attributes

B. Split Dataset and Train Model

The dataset is split into training and testing data using the $train\_test\_split$ function provided by the $sklearn.model\_selection$ class. The dataset was split into

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target
MedInc	1.000000	-0.119034	0.326895	-0.062040	0.004834	0.018766	-0.079809	-0.015176	0.688075
HouseAge	-0.119034	1.000000	-0.153277	-0.077747	-0.296244	0.013191	0.011173	-0.108197	0.105623
AveRooms	0.326895	-0.153277	1.000000	0.847621	-0.072213	-0.004852	0.106389	-0.027540	0.151948
AveBedrms	-0.062040	-0.077747	0.847621	1.000000	-0.066197	-0.006181	0.069721	0.013344	-0.046701
Population	0.004834	-0.296244	-0.072213	-0.066197	1.000000	0.069863	-0.108785	0.099773	-0.024650
AveOccup	0.018766	0.013191	-0.004852	-0.006181	0.069863	1.000000	0.002366	0.002476	-0.023737
Latitude	-0.079809	0.011173	0.106389	0.069721	-0.108785	0.002366	1.000000	-0.924664	-0.144160
Longitude	-0.015176	-0.108197	-0.027540	0.013344	0.099773	0.002476	-0.924664	1.000000	-0.045967
target	0.688075	0.105623	0.151948	-0.046701	-0.024650	-0.023737	-0.144160	-0.045967	1.000000

Fig. 5. Table of the Pearson correlation of dataset attributes

20% - 4,218 - for testing purposes and 80% -16, 512 - for training purposes. Splitting the dataset allows setting aside some dataset for testing purposes to evaluate the trained model's accuracy efficiently. If the dataset were not split, there is probability of the model predicting the target value through memorising and we want to avoid this.

An instance of the linear regression model class was instantiated, and the training datasets were fed into the fit function, which is an implementation of the linear regression formula. The value of the coefficients after training the model is [4.48674910e-01 9.72425752e-03 -1.23323343e-01 7.83144907e-01 -2.02962058e-06 -3.52631849e-03 -4.19792487e-01 -4.33708065e-01]. Training the linear regression machine learning model took 27.8 ms as calculated using the $\%\%time$ magic function for the IPhyton coding environment. This value shows that the Scikit-learn linear regression algorithm has a swift runtime as compared with other learning algorithms as proven in [8]. The value of the intercept was found to be -37.023277706063894. Ideally, these are the values needed if a manual calculation of the target value y is to be made.

C. Result and Discussion

With the regression model successfully trained, the testing datasets can now be fed to the model to predict the target values. The coefficient of determination R^2 of training datasets for the model was found to be 0.6125511913966952. When predictions were made using the testing dataset, there were some offsets in the predicted values to the target value as shown in figure 6 which shows a graph of the first 20 values. The Mean squared error of the predicted value to the target value was found to be 0.56 and the coefficient of determination to be 0.58. These values explain the reason why there were some offsets in the predictions.

With the result of the predictions from the model, it can be said that the model has under fitted the training data because the features used did not have an accurate linear correlation. This can be seen in figure 5 which showed some negative values. We can also understand that many not-so-known factors influence house prices that were not included in the datasets or could not be found using linear regression.

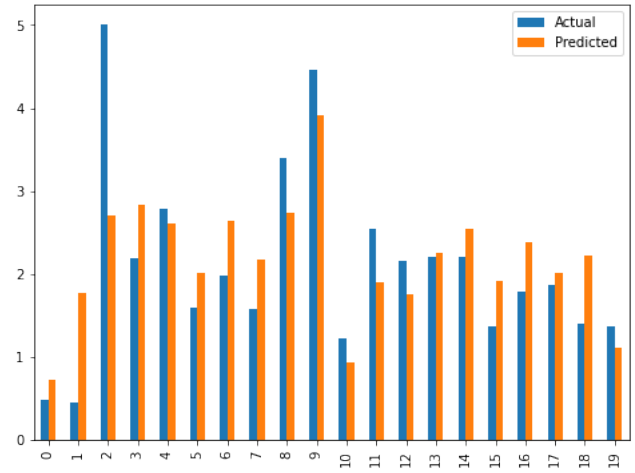

Fig. 6. Graph of first 20 actual values vs predicted values by the trained model

V. Conclusion

The paper's primary aim is to understand linear regression in the context of machine learning and how to train a regression machine learning model to predict the average price of a house for a district block in California.

Different sections in the paper gave essential information on machine learning and linear regression to achieve this aim. A practical example was done using the different feature points that made up the California house price dataset utilising the Scikit learn python library, which implements the necessary mathematical formulas needed to make the statistical prediction job. The Scikit-learn linear regression algorithm showed a swift runtime of 27.8 ms in training the machine learning model with a dataset of 16, 512. The coefficient of determination of the training dataset was found to be R^2 = 0.6125511913966952. Predictions made using the testing dataset gave R^2 = 0.58 and mean squared error MSE = 0.56. These values showed that the model has around 57% accuracy of correctly predicting the dependent variable.

In conclusion, as can be highlighted from the linear regression equation, it is a reasonably straightforward regression model to get started within machine learning. However, it is pretty poor at predicting datasets that do not have solid linear correlation as depicted in the trained model in this paper. As a suggestion, different regression algorithms should be tried out in training linearly related datasets to find the best fit for the desired prediction job.

References

[1] Mro survey 2016.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[3] IBM Cloud Education. What is machine learning?, Jul 2020.

[4] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[5] R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[6] Wilhelm Kirch, editor. *Pearson's Correlation Coefficient*, pages 1090–1091. Springer Netherlands, Dordrecht, 2008.

[7] K.-Y Lee, K.-H Kim, J.-J Kang, S.-J Choi, Y.-S Im, Y.-D Lee, and Y.-S Lim. Comparison and analysis of linear regression & artificial neural network. *International Journal of Applied Engineering Research*, 12:9820–9825, 01 2017.

[8] Luca Massaron and Alberto Boschetti. *Regression Analysis with Python*. Packt Publishing Ltd, 2016.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Astrid Schneider, Gerhard Hommel, and Maria Blettner. Linear Regression Analysis. *Dtsch Arztebl International*, 107(44):776–782, 2010.

[11] Oliver Theobald. *Machine learning for absolute beginners: a plain English introduction*. Scatterplot press, 2017.