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*—talk about the data example prediction that will/was done using sck-kit. Then talk about what linear regression, why it is used.....**

this paper talks about linear regression in the context of deep learning. Linear regrression is a statictical term that uses a dependent and independent variable to make predictions. in these paper, the use case of linear regression is used to predict data in machine learning. The data were gotten from kaggle.com. the data is a data of the titanic survivers and the model was trained using sci-kit or tensor flow.

data cleasing was done using pandas a python library to prepare the data for model training.

regression analysis as a whole is a stastical method used to understand the relation between dependent and independent variables.

with the rise in the amount of data we have access to and increase in the performance of computers, Machine Learning a subset of Artificial inteligence has seen a significant growth. machine learning refers to is the ability of applications to get better at doing things without necessarily a change in the code base them is something that has really been helpful.

even though machine learning isn't a new concept, it is fast gaining recognization and changing lives thanks to the increase in the processing power of computers over the years to be able to process big data at a level that has never been experienced.

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

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

I. Diagrams

Include diagram for machine learning automatically adapting to change?? get idea from fig 1-3 page 25 of [hands-on machine learning with scikit-learn]

II. 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 generation from global air fleet is projected to reach 98 billion gigabytes [**?**]. Thanks to the recent advancements in the information processing capabilities of computers, making sense of this huge chunk of data is now possible using data analysis techniques and Machine learning to clean and make prediction from this datasets. Machine learning is the ability of machine -software application- 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 through their grounded mathemical foundations.

Linear regression is a statistical test used to find out the relationship between independent and dependent variables in a data set using mathematical formular. They can also be used in projecting new relationship between the dependent and independent variables that has not been discovered. Even though linear models - linear regression - are quite simple to develop and understand, as well as good at predicting linear relationships, their approximation of nonlinear relationships have been found to be mostly unsatisfactory [**?**]. This paper goes more into the mathematical details this paper gives a profound, yet understandable mathematical description of Linear regression, relevant parts of the framework -Sci-Kit- used to perform predictions on the example dataset in the context of Machine learning using the Python programming language.

III. Theoretical Background

A. Machine Learning

According to [**?**], "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 in writing application that are somethings difficulty or even impossible to implement using traditional algorithms. Methods used in training a Machine learning model can be grouped into three main categories.

- Supervised Learning
  This is a type of machine learning technique in which the algorithm is fed with an input and output dataset of desired solution. The desired output data is otherwise known as lables. These information help train the machine lerning model in the precise prediction of output when given a related output after being trained. Some

examples of common algotithms used for supervised learnings are regression analysis - linear regression, etc.-, decision trees, amongst others. [**?**] The two types 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 large amount of unlabeled datasets to train -properly find patterns in datasets.
- Semisupervised Learning
  this is a learning technique that uses both labeled -input and desired output- and unlabled -only input- datasets for training machine lerning 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 who's algorithm defines an artificial neural network -ANN- that is designed to emulate the way an human brain learns, and unsupervised or semisupervised learning technique to train [**?**]. On other hand, reinforcement learning is a reward based learning in the sense that the machine leaning is trained by being given a point for reaction to certain 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. [**?**]. 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 [**?**]. 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 said to be a simple linear regression, but if the independent variable is more than one, it is a multiple linear regression. In this paper, we will make analysis using the simple linear regression model.

C. 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 the actual value.

1) R^2: this is otherwise known as the coefficient of determination or the coefficient of multiple determination 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 [**?**]. A value of 1 refers to the fact that all the points fits 100% to the the regression line. The lesser the value, the farther the points are to the regression line. The coefficient of determination is defined by the following equation:

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y_i} - \overline{y})^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \tag{3}$$

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$^{\text{th}}$ observation computed by the regression equation, y_i is the observed value of the dependent variable for the i$^{\text{th}}$ observation and $\overline{y}$ is the mean of all n observations of the dependent variable [**?**]

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 the actual value of the ouput given the same input. The equation of the calculation is denoted by:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 \tag{4}$$

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

IV. Algorithm Implementation

All the algorithm needed for prediction using the linear regression model have 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 algorithms that are used in machine learning. This library provide functionalities for solving machine learning jobs like regression, classification, clustering, model selection, pre-processing - like spliting data sets into test and train subsets -amongst others [**?**]. The scikit-learn API are designed around this 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 gives baseline solution for tasks at hand -and nonproliferation of classes - datasets are represented as NumPy arrays or SciPy sparse matrices [**?**].

2) Numpy: NumPy is an open-source Python library that that provides routines that allows for fast operations on multidimensional arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic 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 exploration, cleaning and processing of tabular data structures in Python[2].

4) Matplotlib: Matplotlib is a Python library used along with Numpy for creating static, animated and interactive visualization of data[3].

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

V. Practical Example

In this paper, the california housing datasets from [**?**] included in the scikit sklearn datasets library is used to predict train and prediction the average prices of housing in california using linear regression.

A. Preparing Data

The dataset has 20,640 instance number, 8 numeric predictive atrributes and 1 target attribute which is 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 [**?**]. 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.
- 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.

This dataset can download from scikit learn online sklearn.datasets repository by using the following code "from sklearn.datasets import $fetch\_california\_housingcalifornia = fetch\_california\_housing()$"

The data used in this example is gotten from sckit learn. this returns the following datasets with each attribute meaning....get from this link https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html.

- Figure 1 shows the import statments for the functions and libraries that are used.the libraries were imported
- Figure 2 shows the overview of the data as a pandas dataframe which include the target feature column.
- Figure 3 shows a table of the pearson correlation of the attribes to each other using pandas library. The closer the value are to one, the more thery are related to each other. It can be seen from the figure that the median income in a block has the highest correlation to the target variable. Which is why we will be using it as the variable for training our model. This visualization can also be seen in the heatmap drawn with seaborn in figure 4.
- Figure 5 shows the code snipet that was used to transform the X variable to a 2d array as expected by scikit learn regression function. X variable is a 2d array respresenting the median income per block and the y variable is the target representing the median house value in unit of 100,000.
- Create an instance of the linear regression class which is used for training the model.
- Figure 6 shows the coefficients and intercepts of the equation for predicting variables calculated by the sklearn fit() method.
- Try out the prediction using the test dataset. Figure 7 shows the first ten predicted value and the actual value expected. There seems to be some offset in the predictions.
- Figure 8 shows the R^2 and MSE value of the prediction. The value explains the reason why there were some offsets in the prediction. The model can be said to have underfitted the training data because the feature used did not provide enough information to make a good prediction as we understand that the house prices are influenced by many factors unlike just one variable that was used.
- Figure 9 shows a plot of the regression line of first 30 predicted values to their actual target marked in red. It can be seen that some of the target value are quite far away to the regression line drawn.
- Datasets were then split and randomized into 20 percent for training and testing the model.
- the dataset was created as a panda data frame to allow easy manipulation I guess. then the target column was added to the dataset columns which will be the y variable
- a description of the data was manage to get a quick statistics on the data being worked on.
- the input variable is the AveRooms ..... it total-room, which the totatl number of rooms in a block,
- We'll randomize the data, just to be sure not to get any pathological ordering effects that might harm the performance of Stochastic Gradient Descent.[google colab]
- use the

B. Result and discussion

From the result of the predictions, it can be seen that the model underfitted the training data and as a result was not able to make the right prediction of the target value. The MSE value

[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

was 0.45 which showed that there were significant errors in predicting the values.

VI. Conclusion

finish by highlighting the strenth of linear regression as well as the shorcomings.

```
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. 1. Import statements

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Lati
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	
...	...	...	...	...	...	...	
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	

20640 rows × 9 columns

Fig. 2. Dataset overview including the target column

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccu
MedInc	1.000000	-0.119034	0.326895	-0.062040	0.004834	0.01876
HouseAge	-0.119034	1.000000	-0.153277	-0.077747	-0.296244	0.01319
AveRooms	0.326895	-0.153277	1.000000	0.847621	-0.072213	-0.00485
AveBedrms	-0.062040	-0.077747	0.847621	1.000000	-0.066197	-0.00618
Population	0.004834	-0.296244	-0.072213	-0.066197	1.000000	0.06986
AveOccup	0.018766	0.013191	-0.004852	-0.006181	0.069863	1.00000
Latitude	-0.079809	0.011173	0.106389	0.069721	-0.108785	0.00236
Longitude	-0.015176	-0.108197	-0.027540	0.013344	0.099773	0.00247
target	0.688075	0.105623	0.151948	-0.046701	-0.024650	-0.02373

Fig. 3. Pearson correlation of the dataset attributes

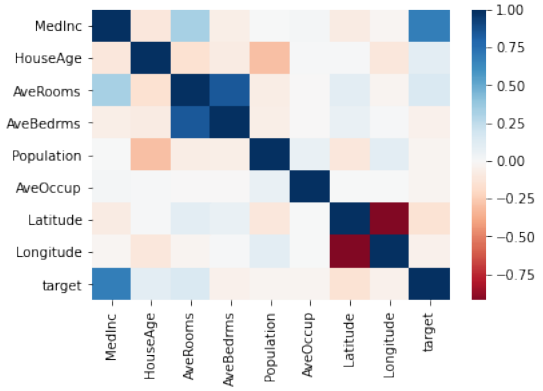

Fig. 4. Pearson correlation heatmap of the dataset attributes

```
#reshape the X
X = observations = len(dataset)
X = dataset['AveRooms'].values.reshape((observations,1))
y = dataset.target
X
```

Fig. 5. Code snippet for reshaping and extracting the X and y variable for the regression analysis

```
#coefficients of the calculations
print('Coefficients:', linear_regression.coef_)
print('Intercept:', linear_regression.intercept_)
```

```
Coefficients: [0.42032178]
Intercept: 0.4432063522765708
```

Fig. 6. Coefficients and intercept of the equation as calculated by sklearn

```
#try out prediction
y_predicted = linear_regression.predict(X_test)
y_predicted[:10]
```

```
array([2.18829831, 2.87249809, 2.27105966, 1.47345706, 2.54687481,
       1.52267674, 2.99292028, 2.93466369, 3.10691155, 2.08948066])
```

```
[16] #check prediction with the actual expected value
     y_test[:10]
```

```
14740    1.369
10101    2.413
20566    2.007
2670     0.725
15709    4.600
439      1.200
845      2.470
3768     3.369
964      3.397
8681     2.656
Name: target, dtype: float64
```

Fig. 7. Predicted value vs actual value

```
print('Mean squared error (MSE): %.2f'
      % mean_squared_error(y_test, y_predicted))
print('Coefficient of determination (R^2): %.2f'
      % r2_score(y_test, y_predicted))
#the coefficients represent the weit=ghts of the variables
```

```
Mean squared error (MSE): 0.72
Coefficient of determination (R^2): 0.45
```

Fig. 8. R^2 and MSE value of the model

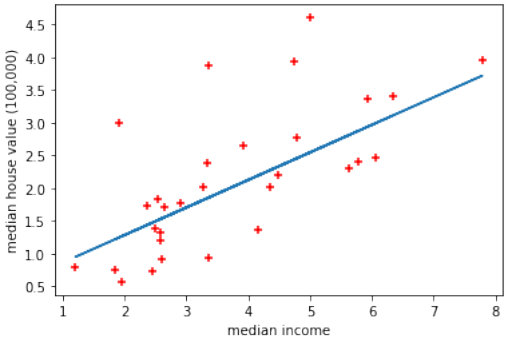

Fig. 9. Plot of the regression line to the actual target value