HOF UNIVERSITY OF APPLIED SCIENCES

PROJECT DOCUMENTATION

Solution of "Titanic: Machine Learning from Disaster" from scratch

Author:
Rayhan PATOARY

Supervisor:
Prof. Dr. Gefei ZHANG

The project submitted in fulfillment of the requirements for Analytical information System course

in the

MASTER IN SOFTWARE ENGINEERING FOR INDUSTRIAL APPLICATIONS

November 7, 2020

Declaration of Authorship

I, Rayhan PATOARY, declare that this work titled, “Solution of "Titanic: Machine Learning from Disaster" from scratch” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly to Complete AIS course in this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

HOF UNIVERSITY OF APPLIED SCIENCES

Abstract

MASTER IN SOFTWARE ENGINEERING FOR INDUSTRIAL APPLICATIONS

Solution of "Titanic: Machine Learning from Disaster" from scratch

by Rayhan PATOARY

Machine learning gained a significant position in prediction. The Titanic incident is a Known misfortune worldwide. There is a need of Solution from scretch that can effectively predict the Survival cases of titanic. In this project, I applied Perceptron algorithm to predidict the survival of passengers. Our objective is to develop an optimized and efficient machine learning (ML) model without using any existing Library which can effectually recognize and predict the result. The Submission result shows that the model is 77 percent accurate. . .

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. for the teaching of my Analytical Information system course. His patience ,motivation and guidance helped me in all the time of learning Machine learning.

I also thank to my fellow Md Sanwar Hossain with whom I have learned perceptron algorithm through group study. . . .

Contents

List of Figures

Chapter 1

Background Scenario

The RMS Titanic was a British traveler liner that sank in the North Atlantic Ocean in the early morning of 15 April 1912, after it slammed into an icy mass during its first venture from Southampton to New York City. There were an expected 2,224 travelers and group on board the boat, and more than 1,500 kicked the bucket, making it one of the deadliest business peacetime oceanic catastrophes in current history. The RMS Titanic was the biggest boat above water at the time it entered administration and was the second of three Olympic-class sea liners worked by the White Star Line. The Titanic was worked by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her draftsman, kicked the bucket in the calamity.

Now I have to analyze the passenger data to find out the probability of drowning. This means that it is my job to find out who drowned and who survived. When drowned, most men die. Because of the small boats that were there, women and children were given priority. Now we can easily say that men will die. And women will mostly survive. That means sex is very important. The children also got priority in getting on the small boats. Now age is important in case of sex male. Those under the age of 18 are more likely to survive.

Chapter 2

Materials and Methodology

2.1 Datasets

I downloaded three Datasets from Kaggle, named train.csv , test.csv and gender$_{s}ubmission.csv.Where, trainh$
$PassengerId : UniqueIdentificationnumber,$
$Survived : DeadorAlived,$
$Pclass : PassengerClass(1 = 1st; 2 = 2nd; 3 = 3rd),$
$Name : Name,$
$Sex : Gender,$
$Age : NumaricValue,$
$SibSp : NumberofSiblings/Spouses,$
$Parch : NumberofParents/Children$
$Ticket : TicketNumber,$
$Fare : PassengerFare,$
$Cabin : Cabin,$
$Embarked : PortofEmbarkation(C = Cherbourg; Q = Queenstown; S = Southampton)$

2.2 Used Tools and technology

2.2.1 Python 3.8

Language of high-level and general purpose programming. The design philosophy of Python, developed by Guido van Rossum and first published in 1991, emphasizes code readability with its notable use of large whitespace.

2.2.2 Jupyter Notebook

The Jupyter Notebook is an open-source web-based application that allows us to create and share live code, calculations, visualizations, and narrative text documents. Data cleaning and conversion, numerical simulation, mathematical modeling, visualization of data, machine learning, and used in much more.

2.3 Supervised learning

As the name suggests, supervised learning takes place under the supervision of a teacher. It is based on this learning process. The input vector is presented to the network, which will generate an output vector, during the training of ANN under supervised learning. Compared with the desired / target output vector, this output vector is. If there is a difference between them, an error signal is produced Based on real output and the vector of desired / target output. Some of the widely used

algorithms of supervised learning are
k-Nearest Neighbours
Decision Trees
Naive Bayes
Logistic Regression
Support Vector Machines

2.4 Selected Algorithm

I used Perceptron Algorithm to solve this Titanic problem.cause this is easier to write from scratch.

Chapter 3

Description of the Algorithm

3.1 Artificial Neural Network

The ANN Artificial Neural Network is an effective computer system, the core theme of which is borrowed from the biological neural network analogy. ANNs are often referred to as "artificial neural systems," or "parallel distributed processing systems," or "connectionist systems." In order to allow communication between the units, ANN acquires a wide number of units that are interconnected in some pattern. These modules, also referred to as nodes or neurons, are basic parallel operating processors.Model of Artificial Neural Network:

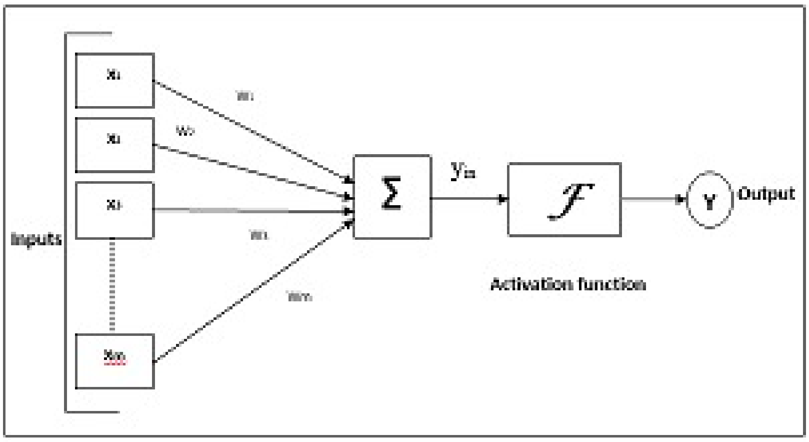


FIGURE 3.1: Artificial Neural Network.

3.2 Perceptron

The basic operational unit of artificial neural networks is the perceptron. It uses the rules of supervised learning and can classify the data into two classes.Perceptron operational characteristics: It consists of a single neuron with an arbitrary number of inputs along with adjustable weights, but depending on the threshold, the output of the neuron is 1 or 0. It also has a bias, the weight of which is always 1. A schematic representation of the perceptron appears in the following diagram. Thus, Perceptron has the following three fundamental elements:

Links-It will have a set of links that carry a weight that always carries weight, including a bias 1.

Adder-It adds the input after their respective weights are multiplied.

Activation function- which restricts neuron production. The most basic activation feature is a step function of Heaviside that has two possible outputs. If the input is positive, this function returns 1, and for any negative input, 0.

Perceptron Mechanism:

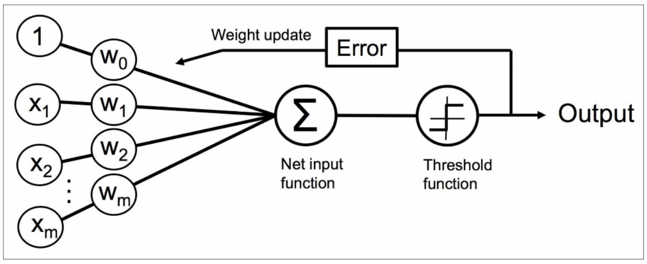


FIGURE 3.2: Perceptron.

3.3 Sigmoid Function

The main reason why we use sigmoid function is because it exists between (0 to 1). Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice

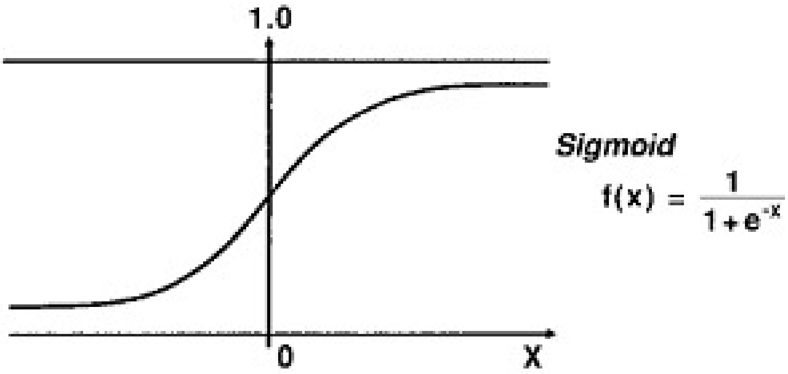


FIGURE 3.3: sigmoid function.

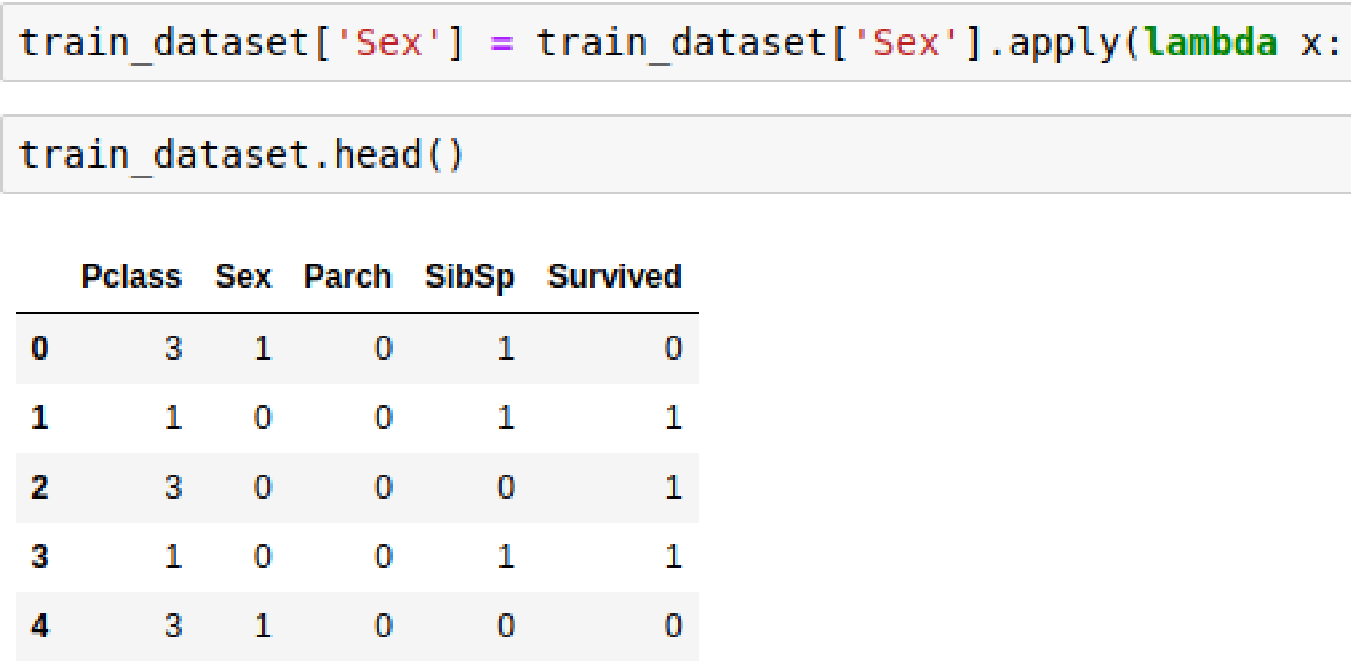
Chapter 4

Implementation

4.1 Clean Train data

Some data is giving duplicate values and some data is not related to our target column. In this step we have created a dataset with the necessary data by omitting them.

4.2 Replace True False with O and 1



```
train_dataset['Sex'] = train_dataset['Sex'].apply(lambda x:
```

```
train_dataset.head()
```

	Pclass	Sex	Parch	SibSp	Survived
0	3	1	0	1	0
1	1	0	0	1	1
2	3	0	0	0	1
3	1	0	0	1	1
4	3	1	0	0	0

FIGURE 4.1: value replacement

4.3 Shuffle train Data

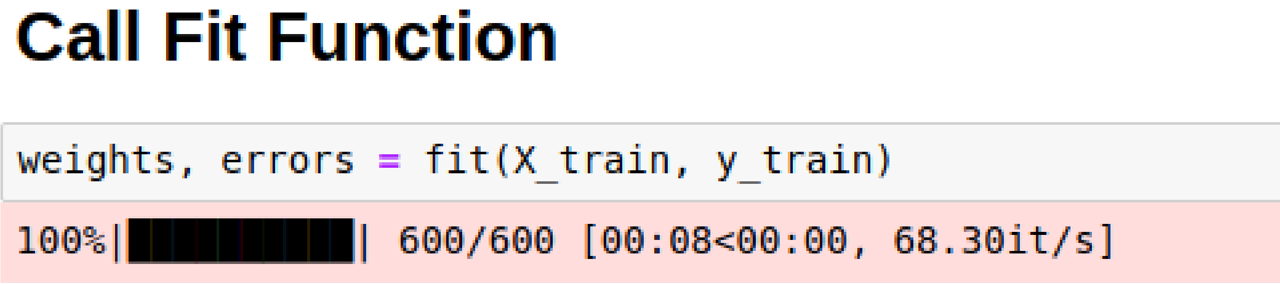
For best accuracy of the model, it’s always recommended that training data should have all flavours of data. Shuffling of training data helps us in achieving this target.

4.4 Split Dataset

In this step I splited the training data into two set. Here 70 percent was for train and 30 percent was for test.

4.5 Fit Model

In this step I trained my model with my splited train data.



Call Fit Function

```
weights, errors = fit(X_train, y_train)
```

```
100%|██████████| 600/600 [00:08<00:00, 68.30it/s]
```

FIGURE 4.2: Model fit progress bar

4.6 Predict our test data

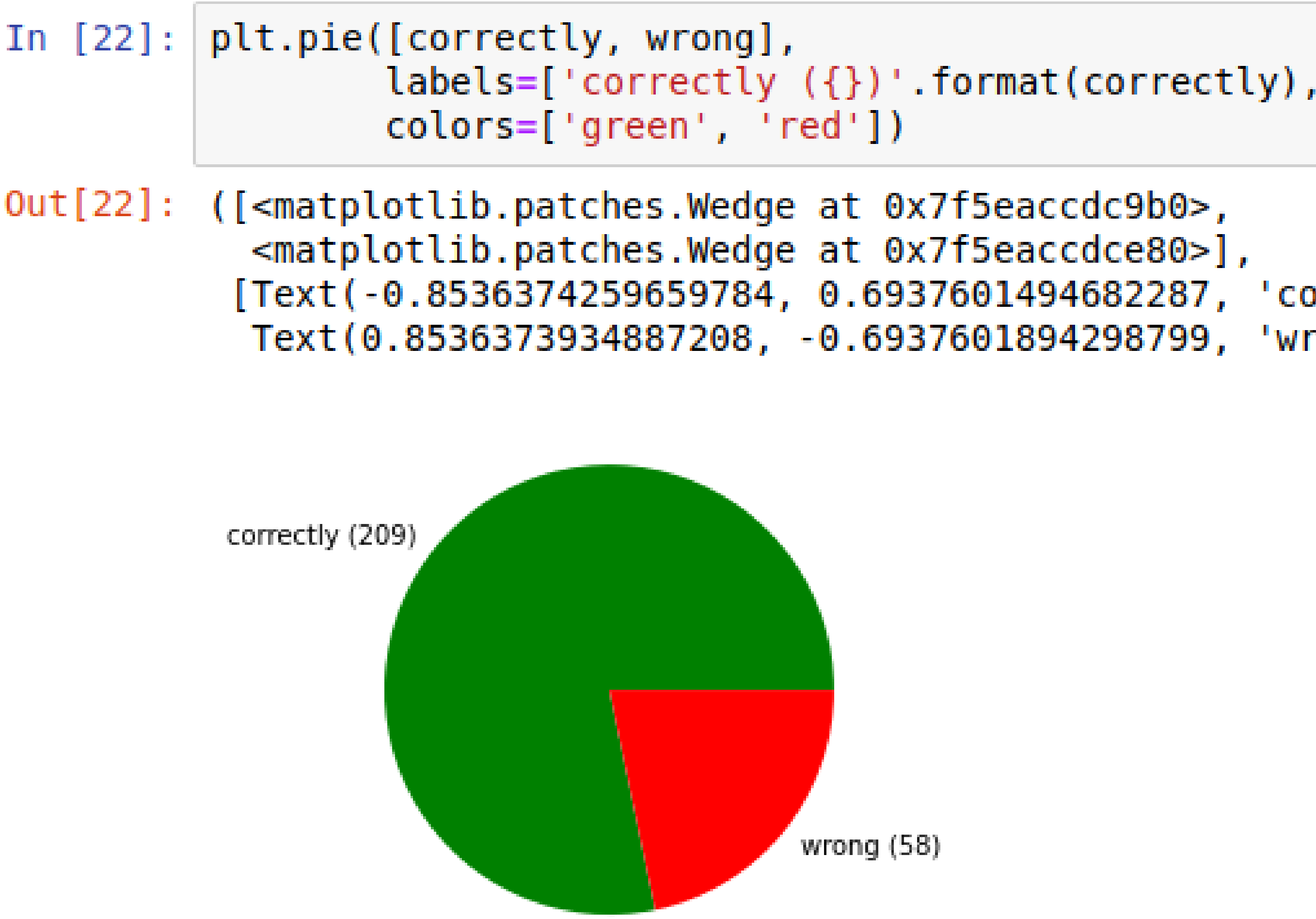
At first, I simplified my test data as like as my train dataset.Then initially I imagined Everybody Survived.At last I made The Final pediction on Test data and generated a csv file.

Chapter 5

Results and Discussion

5.1 Measurement

Within 267 data our model gives 209 correct prediction and 58 wrong prediction.



```
In [22]: plt.pie([correctly, wrong],
                 labels=['correctly ({})'.format(correctly),
                 colors=['green', 'red'])

Out[22]: ([<matplotlib.patches.Wedge at 0x7f5eaccdc9b0>,
           <matplotlib.patches.Wedge at 0x7f5eaccdce80>],
          [Text(-0.8536374259659784, 0.6937601494682287, 'co
           Text(0.8536373934887208, -0.6937601894298799, 'wr
```

FIGURE 5.1: Accuracy.

5.2 Submission

My Kaggle Submission Score is 0.77272.

Chapter 6

Conclusion

The main contribution of this project are as follows; first, I wrote a fit function from scratch then train the model with given data.At last I predicted the test data and submitted the solution in kaggle.

Chapter 7

References

1.https://www.kaggle.com/
2.https://www.tutorialspoint.com/
3.https://www.datacamp.com/
4.https://stackoverflow.com/