PREDICTION OF ASTEROID DIAMETER WITH THE HELP OF MULTI-LAYER PERCEPTRON REGRESSOR

VICTOR BASU

Computer Science and Engineering Jalpaiguri Government Engineering College West Bengal, India
E-mail: basu369victor@gmail.com

Abstract - Predicting the diameter of an asteroid with help of artificial neural network technique. We have used Multilayer Perceptron Regressor algorithm to estimate the diameter of the asteroid with higher accuracy and least error.

I. INTRODUCTION

In this research paper, we are going to discuss how the concept of artificial neural network could be utilized to estimate the diameter of an asteroid. In this research, we have used the Multilayer Perceptron algorithm as the base algorithm to predict the diameter. We have used different algorithms to test and evaluate the performance of the model with the same dataset but Multilayer Perceptron algorithm performed best in these type of situations with higher ac- curacy and least error while prediction. The dataset is we have used is officially maintained by NASA Jet Propulsion Laboratory. In this dataset we have considered all types of asteroids such as asteroids which are grouped as Near Earth Objects(NEO), Potentially Hazardous Objects(PHA), we have also considered all the possible asteroid orbitclasses as mentioned in the official website of JPL(Jet Propulsion Laboratory). The columns of the dataset also contain all the physical and basic properties of an asteroid. We have used Mean Absolute Error, Mean Squared Error,Median Absolute Error, Explained Variance Score and R2-Score as metrics to evaluate and compare the performance of different regression algorithm against the same dataset. The R2-Score which we have achieved through Multilayer Perceptron is 0.9665626238, along with it we have achieved Explained Variance Score of 0.9665631410, the Mean Absolute Error for this model is 0.4306106593, Mean Squared Error is 3.3754211434 and Median Absolute Error is0.2242921644.

II. MODEL IMPLEMENTATION WITH THE HELPOF MULTILAYER PERCEPTRON REGRESSOR ALGORITHM

A. Why did we choose the diameter of an asteroid as the target parameter

The diameter of an asteroid is one the most important physical parameter of an asteroid, it is used to calculate many other physical and basic parameters of the asteroids like calculating the rotation period of asteroid, and also used to detect if an asteroid is potentially hazardous or not if it is found to be a Near Earth Object. The diameter is also used in many other researches about the asteroid.

There are many methods used to estimate the diameter or size of the asteroids, the latest technique that is used by space scientists to estimate the size of the asteroid is through absolute magnitude(H) and geometric albedo(a) where diameter(d) is considered as a function of (H) and (a).The dataset which I have used in this research is officially maintained by JPL(Jet Propulsion Laboratory) of NASA, and the database is based on the observations made through NASA telescopes. But when we analyzed that we found out that there are 646785 rows where the diameter is not specified out of 786226 rows. Although we were not able to find out any valid reason for which the diameters in these cases were not calculated. The main objective of this research is to show that it possible to estimate the diameter of an asteroid with the power of artificial neural network.

B. Implementation details

We have used python as the programming language to implement the model. The libraries which we have basically used in python are numpy and pandas for data analysing, sklearn for pre-processing, importing regressor algorithm and model evaluation, matplotlib and seaborn for data visualization.

C. Dataset Description

The dataset consists of 786226 rows and 22 columns. The columns consists of orbital details of the asteroid like semi-major axis(a),eccentricity(e), inclination(i), perihelion distance(q), condition code of orbit, orbital period and num- ber of observations used, aphelion distance(Q). The column also consist of some basic characteristics of asteroids like it is physically hazardous or not(PHA), it falls under near earth object or not (NEO), its absolute magnitude(H), geo- metric albedo value(albedo).The whole dataset is available in official website of NASA's Jet Propulsion Laboratory[6].

D. Dataset Pre-Processing

Although the dataset is very well defined still there are certain faults in the dataset which were needed to be corrected to create a better regressor model. As it is been mentioned above that there are 646785 rows in which diameter is not mentioned. So we have removed those rows from the dataset, we have also removed those rows in which albedo value is missing. Missing values of absolute magnitude(H) is filled with the mode of the entire column of ”H”. Columns like “condition_code”, ”neo”, ”pha” are transformed from object type to integer type. Columns like ”extent”, ”rot_per”,”n_del_obs_used”,”n_dop_obs used” were also removed from the dataset because almost eighty percent of the data were missing from these columns.

E. Data Visualizations

All the visualizations are done with seaborn library in python. The visualization made form the dataset indicates relationship between different parameters of an asteroid.

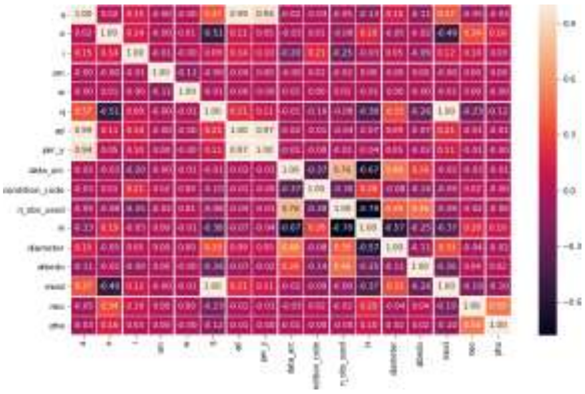

Fig. 1. Correlation matrix heatmap of preprocessed dataset

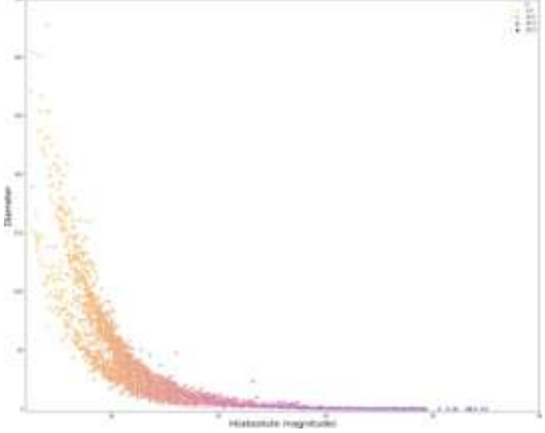

Fig. 2. Scatter plot between asteroid diameter and absolute magnitude(H)

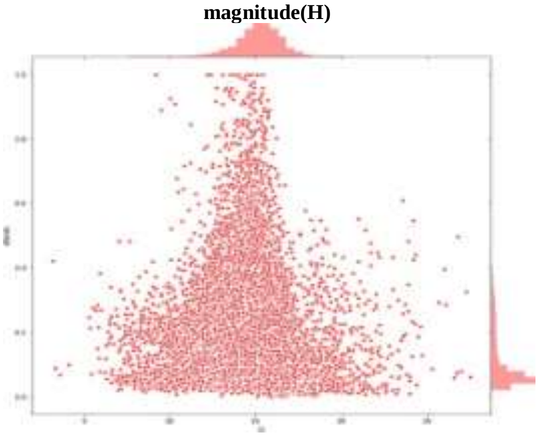

Fig.3. Joint plot between asteroid geometric albedo and absolute magnitude(H)

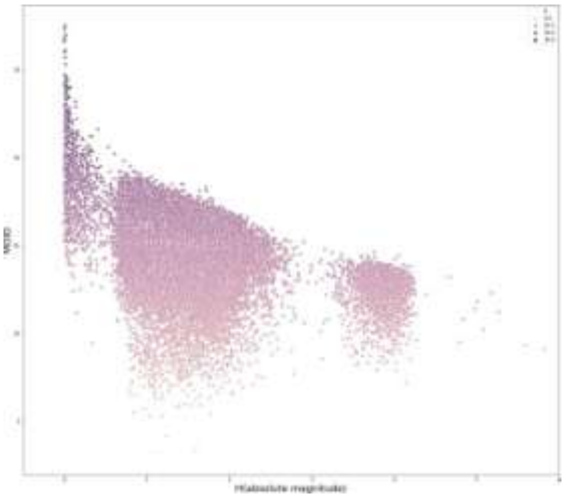

Fig. 4. Scatter plot between asteroid MOID and absolute magnitude(H)

F. Model Implementation and its Detailed Description

Machine Learning algorithms are designed to learn from data for a given situation and predict result for an unknown data from similar type of situation. These algorithms are also used to solve Regression problems.

For estimation of the diameter of an asteroid we have used **MultiLayer Perceptron Regeressor** as the base Supervised learning algorithm. Now let us discuss why did we choose MultiLayer Perceptron algorithm as the base algorithm to predict the diameter of an asteroid in outer space. There are certain metrics based on which we evaluate the performance of a model. Some of the metrics are discussed below which we have used in this research to evaluate and compare different model performances are discussed below.

- **Mean Absolute Error -** It is the mean of the absolute value of each difference between actual value and predicted value for that instance on all instance of the testdataset.
- **Mean Squared Error -** It is the average of squared error that is used as the loss function for least squared regression.
- **Median Absolute Error -** It is the median of all absolute difference between the target and predicted value.
- **Explained variance score -** It is a measure how far observed value differ from the average of predicted value.
- **r2 Score -** It is known as coefficient of determination. It is a statistical measure of how close are the data fitted to the regression line.

Here we have used mean_absolute_error, mean_squared _error,median_absolute_error, explained_variance_score, r2 _score from sklearn.metrics to evaluate mean absolute error, mean squared error, median absolute error, variance score and r2 score respectively.

Analysis of different algorithm for prediction of diameter of an asteroid and finally comparing them with MultiLayer Perceptronalgorithm:-

Before fitting and evaluating the model we have also pre-processed the training and test data with **MaxAbsScaler** algorithm, which is imported from sklearn.preprocessing. This estimator scales and translates each feature individually such that the maximal absolute value of each feature in the training set will be 1.0. It does not shift/center the data, and thus does not destroy any sparsity.

Gradient Boosting Regressor - Gradient boosting is a machine learning technique for regression and classification problems, where it works on reducing error sequentially. It builds an additive model in a forward stage wise manner. To implement it we have imported Gradient Boosting Regressor from sklearn.ensemble. We have used ”huber” as the loss function to be optimized. ”max_features” is set to ”auto” which means max features = n_features. ”learning rate” is set to 0.4. ”warm_start” is set to ”True” as a result the regressor model to reuse the solution of the previous call to fit and add more estimators to the ensemble. Model evaluation in case of Gradient Boosting Regressor is given below.

- Mean Absolute Error:0.4607025067
- Mean Squared Error : 6.8237488992
- Median Absolute Error :0.2290790857
- Explained Variance Score :0.9324036603
- r2-Score :0.9324030249

XGBoōst Regressor - It is an extreme gradient boosting algorithm. To implement it we have imported XGB Regressor from xgboost. We have set ”n estimators” which is the number of trees in a forest to 1000. Model evaluation in case of XG Boosting Regressor is given below.

- Mean Absolute Error:0.4164732536
- Mean Squared Error : 5.3267230111
- Median Absolute Error :0.2137639141
- Explained Variance Score :0.9472328399
- r2-Score :0.9472327649

Random Forest Regressor - It is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. To implement it we have imported Random Forest Regressor from sklearn.ensemble. Here ”n_estimators” is set to 100, ”warm_start” is set to ”True”, ”oob_score” is set to ”True” to use out-of-bag samples to estimate the r2 on unseen data. Model evaluation in case of Random Forest Regressor is given below.

- Mean Absolute Error:0.4040615222
- Mean Squared Error : 6.7417979790
- Median Absolute Error :0.2009400000
- Explained Variance Score :0.9332152304
- r2-Score :0.9332148418

AdaBoost Regressor - It is a boosting algorithm which boosts the performance of the base estimator model. It is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the

regressor on the same dataset and the weights of instances are adjusted according to the error of the current prediction. To implement it we have imported AdaBoost Regressor from sklearn.ensemble. Here the ’base estimator’ which is the base estimator from which the boosted ensemble is built is taken as a Decision Tree Regressor which is imported from sklearn.trees and in this Decision Tree Classifier we have set ”max_features” to ”auto”, ”max_depth” which is the maximum depth of the tree is set to 30. In AdaBoost

Regressor ”n_estimators” is set to 100, ”loss” function is set to ”exponential” and ”learning_rate” is set to 0.5. Model evaluation in case of AdaBoost Regressor is given below.

- Mean Absolute Error:0.4111519233
- Mean Squared Error : 6.3205563183
- Median Absolute Error :0.1960000000
- Explained Variance Score :0.9373877402
- r2-Score :0.9373877184

MultiLayer Perceptron Regressor - It is basically based on the concept of artificial neural networks. It utilizes **back propagation** for training, this model optimizes the squared- loss using stochastic gradient descent. To implement it we have imported MLPRegressor from sklearn.neural network. It has ā linear activation function in all neurons which is a linear function which maps the weighted inputs to the output of each neuron, here it is set to ”tanh”, the hyperbolic tan function, returns f(x) = tanh(x).”max_iter” which is the maximum number of iterations is set to 1000 and ”warm start” is set to ”True”. Model evaluation in case of MLP Regressor is given below.

- Mean Absolute Error:0.4306106593
- Mean Squared Error : 3.3754211434
- Median Absolute Error :0.2242921644
- Explained Variance Score :0.9665631410
- r2-Score :0.9665626238

From the above observations taken we could conclude that the MultiLayer Perceptron algorithm performs the best in this case. Therefore we came up with the result that MultiLayer Perceptron algorithm should be the base supervised learning algorithm to predict the diameter of an asteroid.

G. Visualization of Model performance with the help of Residual Plot

X-axis indicates the **Predicted value** and Y-axis indicates the **Residual value**, Residual value = Actual value - Pre- dicted value

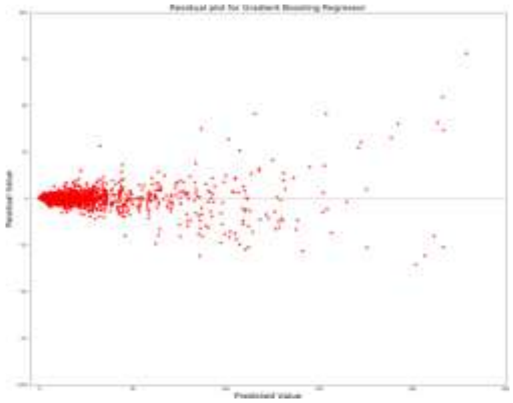

Fig. 5. Residual Plot for Gradient Boosting Regressor

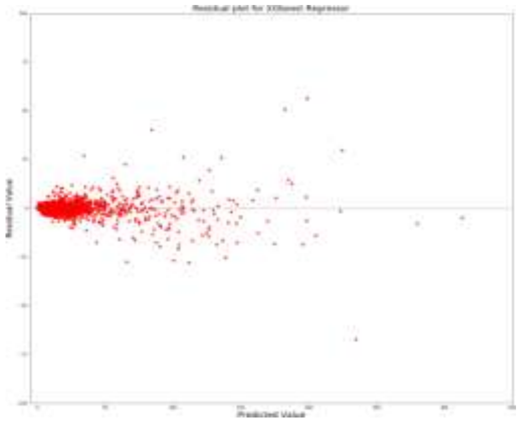

Fig. 6. Residual Plot for XGBoost Regressor

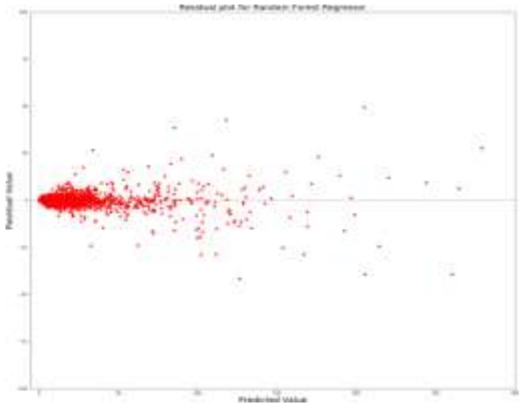

Fig. 7. Residual Plot for Random Forest Regressor

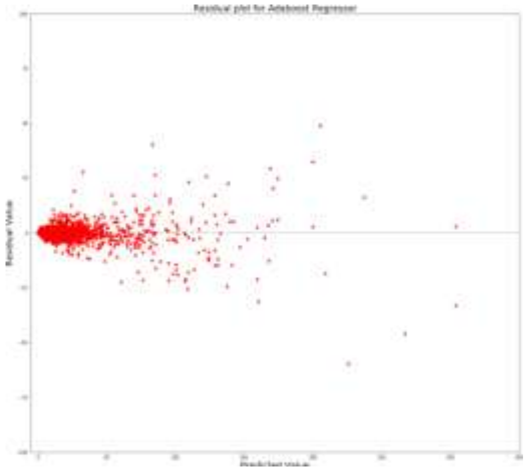

Fig. 8. Residual Plot for AdaBoost Regressor

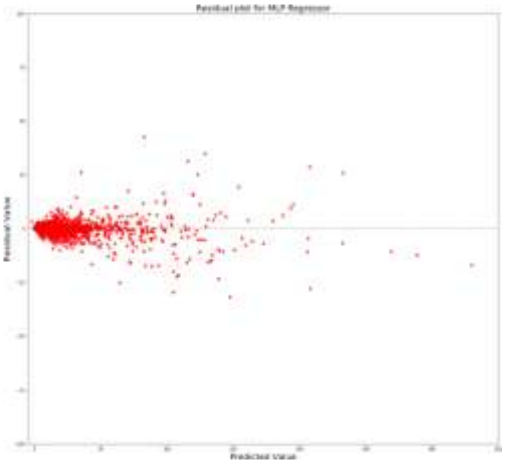

Fig. 9. Residual Plot for MLP Regressor

H. How does MultiLayer Perceptron algorithm works?

A MultiLayer Perceptron is a feed forward artificial neural network that maps a set of input data into a set of appropriate data. It consists of minimum three layers the input layer, the output layer and the hidden layer. The hidden layers could be one or more. Each node of a layer is connected with certain weight $w_{ij}$to every node of its frontal layer except the output layer. In order to activate a neuron in the leading layer the summation of the product of Activation value of the neurons in the preceding layer and Weight connecting to the neuron in the leading layer must be greater than activation value of the target neuron.

$$Z^{(L)} = w_{j0}{}^{(L)}.a_0{}^{(L-1)} + w_{j1}{}^{(L)}.a_1{}^{(L-1)} + w_{j2}{}^{(L)}.a_2{}^{(L-1)} + \text{...........} + w_{jn}{}^{(L)}.a_n{}^{(L-1)} \quad (1)$$

L indicates a particular layer of neurons.

Here the threshold activation value each neuron would be between 1 and 1, So we need a **Activation Function** to squeeze the value between -1 and 1. Here we have used tan has the activation function. It is also sigmoidal

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

So,

$$a^{(L)} = \sigma\ (Z^{(L)} + b_{\,j}{}^{(L)})$$

Where, b is a bias, which indicates how high the weighted sum needs to be before the neurons get meaningfully active. Therefore the process of activating neurons flows from input layer to inner layer. Activation of neuron in a layer highly depends on the behavior of all the neurons in the preceding layer.

MLP uses the technique of **backpropagation** for training. Learning occurs by changing the connecting weights after each iteration based on the amount of error in output com- pared to the expected result.

Total error for each output neuron,

$$C_{total} = \sum \frac{(t-y)^2}{2}$$

Where, t is the target value and y is the value produced by the perceptron. Now we need to upgrade the weights w_{ij} using **gradient descent**, Δw_{ij} is added to the old weight, and the product of the learning rate and the gradient, multiplied by -1 to decrease the value of C with each iteration.

$$\Delta w_{ij} = -\eta \frac{\partial C}{\partial w_{ij}}$$

I. Device used to compute the model and evaluate it

We have implemented everything on a machine, whose configurations are given below,

Processor: Intel(R) Core(TM)i5-5200U CPU: 2.20 GHz

System Type: 64-bit Operating System, x64-based processor

Operating System: Windows 8.1 pro

CONCLUSION

Space researches could proceed further with the help of machine learning algorithms. Here we accept MultiLayer Perceptron algorithm to be the best to tackle these types of a problem after all types of evaluation. The model which has been discussed in this paper could be improved further with higher scores and accuracy and less errors with the help of other machine learning algorithm or with advanced processing technique to pre-process data or through deep learning.

ACKNOWLEDGEMENT

This research is successful all due to Jet Propulsion Lab- oratory of NASA for maintaining such a wonderful dataset. The information present in the dataset was the main reason for the success of the research. Nowadays space scientists and astronomers are constantly trying hard to know more about these asteroids, their characteristics, are they a risk to earth or not ? So these scientists and astronomers are also a source of inspiration behind this research project.

REFERENCES

List of references which helped us a lot throughout the research are given below:

[1] E.R. Buhrke; J.L. LoCicero.1992.”A learning algo- rithm for multi-layer perceptron networks with non-differentiable nonlinearities”. [Proceedings 1992] IJCNN International Joint Conference on Neural Networks.

[2] Gurpreet Singh; Manoj Sachan.2014.”Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition”. 2014 IEEE International Conference on Computational Intelligence and Computing Research.

[3] Mehrshad Salmasi; H. Mahdavi-Nasab; H. Pourghassem.2011.”Comparison of Multilayer Perceptron and Generalized Regression Neural Networks in Active Noise Control”. 2011 Third Pacific-Asia Conference on Circuits, Communications and System(PACCS).

[4] T. Yidirim; H.K. Cigizoglu.2002.”Comparison of generalized regression neural network and MLP performances on hydrologic data forecasting”. Proceedings of the 9th International Conference on Neural Information Processing, 2002.ICONIP ’02.

[5] M.L. Vaughn; J.G. Franks.2003.”Explaining how a multi-layer perceptron predicts helicopter airframe load spectra from continuously valued flight parameter data”. Proceedings of the International Joint Confer- ence on Neural Networks,2003.

[6] https://ssd.jpl.nasa.gov/sbdb_query.cgi

★★★