Principal Component Analysis On Asteroid Data (August 2019)

AASHISH UPADHYAY,

Author , AI Tech Systems,

California,USA

e-mail – aashish31f@gmail.com

https://ai-techsystems.com/

***Abstract*—This project report draws a comparison between classification using the original features and different no. of principal components computed using the PCA algorithm on the basis of accuracy of model , training time etc. The classification algorithm used here is logistic regression. The data is about Asteroids - NeoWs. NeoWs (Near Earth Object Web Service) is a RESTful web service for near earth Asteroid information. With NeoWs a user can: search for Asteroids based on their closest approach date to Earth, lookup a specific Asteroid with its NASA JPL small body id, as well as browse the overall dataset.**

***Index Terms*—logistic regression, PCA**

I. Introduction

This project report lays down a comparison between classification of asteroid data using original set of features and that after dimensionality reduction using PCA algorithm. The classification algorithm used is logistic regression.

The performance of models has been analyzed on the basis of accuracy achieved and training time and conclusions have been drawn on that basis.

II. Steps Involved In Making The Project

A. Introduction to The Dataset Used

The dataset used in this project has been taken from Kaggle. The link to which is:
https://www.kaggle.com/shrutimehta/nasa-asteroids-classification .
The dataset contains various columns containing the details linked to some asteroids. On the basis of these features, one has to predict whether a specific asteroid is dangerous or not. It is a simple classification problem, which requires the data to be classified into 2 categories – hazardous asteroids and non-hazardous asteroids.

B. Data Preprocessing

The various columns present in the dataset are:
1. Neo Reference ID
2. Name
3. Absolute Magnitude
4. Est Dia in KM(min)
5. Est Dia in KM(max)
6. Est Dia in M(min)
7. Est Dia in M(max)
8. Est Dia in Miles(min)
9. Est Dia in Miles(max)
10. Est Dia in Feet(min)
11. Est Dia in Feet(max)
12. Close Approach Date
13. Epoch Date Close Approach
14. Relative Velocity km per sec
15. Relative Velocity km per hr
16. Miles per hour
17. Miss Dist.(Astronomical)
18. Miss Dist.(lunar)
19. Miss Dist.(kilometers)
20. Miss Dist.(miles)
21. Orbiting Body
22. Orbit ID
23. Orbit Determination Date
24. Orbit Uncertainity
25. Minimum Orbit Intersection
26. Jupiter Tisserand Invariant
27. Epoch Osculation
28. Eccentricity
29. Semi Major Axis
30. Inclination
31. Asc Node Longitude
32. Orbital Period
33. Perihelion Distance
34. Perihelion Arg
35. Aphelion Dist
36. Perihelion Time
37. Mean Anomaly
38. Mean Motion
39. Equinox
40. Hazardous

Here the last feature ‘Hazardous’ is a dependent variable (target variable). All other features are independent variables.
Some of the features which were less useful like :
'Neo Reference ID', 'Name', 'Equinox', 'Orbit Determination Date', 'Close Approach Date' , 'Orbiting Body' and 'Orbit ID' were removed as the information received through them

was irrelevant for our purpose and for smooth working of the algorithm .

C. *Logistic Regression Algorithm*

This is a binary classification algorithm that uses sigmoid function to classify between two categories.

$$\text{Sigmoid}(x) = 1/(1+\exp(-x)) \quad - (1)$$

The Loss function in case of logistic regression is :

$$L(a^{(i)},y^{(i)}) = -(1/m)\sum ((y^{(i)}\log(a^{(i)})+ (1-y^{(i)})(1-\log(a^{(i)}))- (2)$$

Algorithm for logistic regression :

1. Initialize the weight matrix 'W' and intercept b.
2. Compute $Z = W^T X + b$
3. Now compute a=sigmoid(z) using equation(1).
4. Calculate the value of cost function in equation(2).
5. Compute the derivatives of W and b with respect to the loss function dL/dW and dL/db.
6. Update the parameters according to the following rule :
   W = W - learning_rate*(dL/dW)
   b = b - learning_rate*(dL/db)
7. Repeat steps 1-6 multiple times to get suitable values of W and b .

D. *Principal Component Analysis Algoritthm*

As described in the article in [2], the main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.

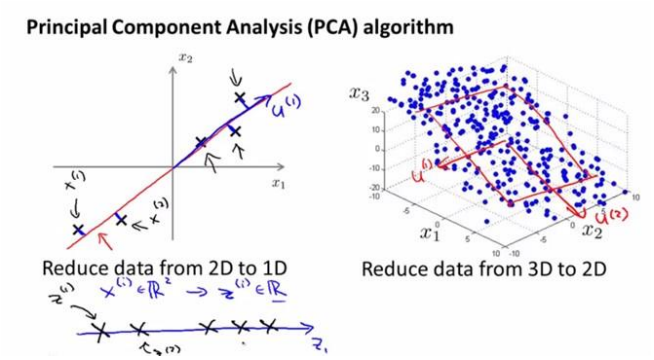

Fig 1. Illustration showing PCA
Image Source: Machine Learning Lectures by Prof. Andrew NG at Stanford University

Algorithm For Principal Component Analysis[2]:

1. Normalize the data .
2. Calculate the covariance matrix :
   $Cov(X) = (1/m) \times (XX^T)$
   m : No. of instances
3. Calculate Eigenvalues and Eigenvector .
4. Choosing components and forming a feture vector.
5. Forming principal components :
   $NewData = FeatureVector^T \times ScaledData^T$

III. Comparison

A. *Model with original features*

No. of features	:	32
Algorithm used	:	Logistic Regression
Optimizer	:	Gradient Descent
Learning Rate	:	0.3
No. of Epochs	:	16000
Accuracy achieved	:	94.96587030716723 %
Training Time	:	21.33051371574402 s

B. *Models with Different no. of Principal Components*

No. of features	:	32
Algorithm used	:	Logistic Regression
Optimizer	:	Gradient Descent
Learning Rate	:	0.3
No. of Epochs	:	16000

a. This is a list of information preserved with respect to original features after taking different no. of components in Principal Component Analysis algorithm :

```
for 1 components information preserved is : 31.157393578536798
for 2 components information preserved is : 48.80792296533600%
for 3 components information preserved is : 61.7658150515288 %
for 4 components information preserved is : 71.12839170550475%
for 5 components information preserved is : 77.06430665149496%
for 6 components information preserved is : 81.00980371343763%
for 7 components information preserved is : 84.3781490999543%
```

```
for 8 components information preserved is : 87.57935425259407%
for 9 components information preserved is : 90.62325350145363%
for 10 components information preserved is : 93.61176244511 94%
for 11 components information preserved is : 96.0476400824742%
for 12 components information preserved is : 97.7960875794382%
for 13 components information preserved is : 98.6964690795885%
for 14 components information preserved is : 99.3171914190535%
for 15 components information preserved is : 99.8385923084624%
for 16 components information preserved is : 99.93830988661 87%
for 17 components information preserved is : 99.9953869202934%
for 18 components information preserved is : 99.9989911003990%
for 19 components information preserved is : 99.9999999999997%
for 20 components information preserved is : 99.9999999999998%
for 21 components information preserved is : 99.9999999999998%
for 22 components information preserved is : 99.9999999999999%
for 23 components information preserved is : 99.9999999999999%
for 24 components information preserved is : 99.9999999999999%
for 25 components information preserved is : 99.9999999999999%
for 26 components information preserved is : 99.9999999999999%
for 27 components information preserved is : 99.9999999999999%
for 28 components information preserved is : 99.9999999999999%
for 29 components information preserved is : 99.9999999999999%
for 30 components information preserved is : 99.9999999999999%
for 31 components information preserved is : 99.9999999999999%
```

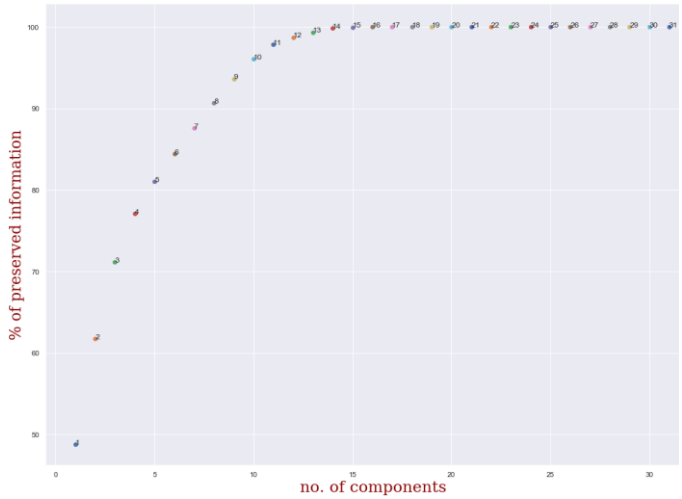

Fig 2. Graph showing the preserved information with different no. of components

b. This is a comparison between the accuracy achieved and training time while training the model with different no. of principal components.

Table I. Accuracy achieved and training time corresponding to no. of principal components .

NO. OF PRINCIPAL COMPONENTS	ACCURACY ACHIEVED (%)	TRAINING TIME (seconds)
1	*82.33788395904436 %*	*6.70512604713439s*
2	*82.16723549488054 %*	*8.10849738121032s*
3	*82.081911262798%*	*7.67784261703491s*
4	*82.593856655290%*	*5.5882503986s*
5	*82.5085324232082 %*	*5.87372088432312s*
6	*82.42320819112628 %*	*6.23962020874023s*
7	*83.53242320819113 %*	*6.90574359893798s*
8	*83.53242320819113 %*	*6.57174062728881s*
9	*83.44709897610922 %*	*7.00446057319641s*
10	*83.3617747440273 %*	*6.7526652812957s*
11	*83.3617747440027%*	*7.0622615814208s*
12	*88.31058020477815 %*	*7.2548096179962s*
13	*92.4061433447099 %*	*7.4429860115051s*
14	*94.19795221843003 %*	*7.9299955368041s*
15	***95.05119453924915 %***	***8.7681729793548s***
16	*95.05119453924915 %*	*10.716922521591s*
17	*94.96587030716724 %*	*9.0508036613464s*
18	*94.96587030716724 %*	*8.7018036842346s*
19	*94.96587030716724 %*	*9.2254827022552s*
20	*94.96587030716724 %*	*9.5695662498474s*
21	*94.96587030716724 %*	*12.020801782608s*
22	*94.96587030716724 %*	*10.354511499404s*
23	*94.96587030716724 %*	*10.679635524749s*
24	*94.96587030716724 %*	*11.163479089736s*
25	*94.96587030716724 %*	*11.754928588867s*
26	*94.96587030716724 %*	*12.388617277145s*
27	*94.96587030716724 %*	*14.316374778747s*
28	*94.96587030716724 %*	*16.364429712295s*
29	*94.96587030716724 %*	*13.837416172027s*
30	*94.96587030716724 %*	*14.848315000534s*
31	*94.96587030716724 %*	*14.207343339920s*

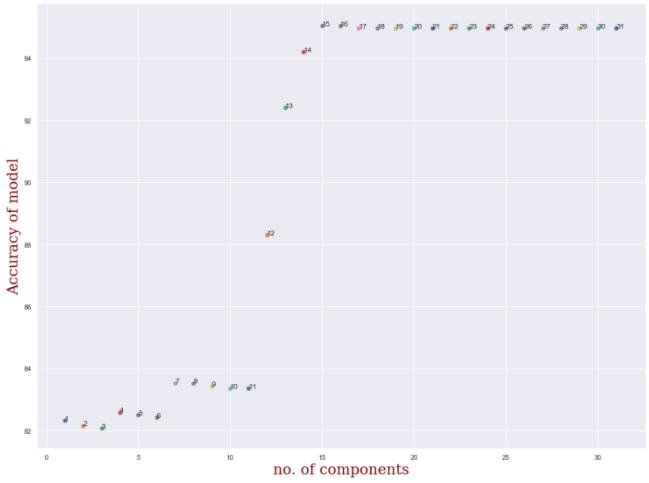

Fig 3. Graph showing the accuracies achieved by models having different no. of principal components

C. Results

a) ORIGINAL MODEL

- NO. OF FEATURES : 32
- ACCURACY OF THE MODEL : 94.96587030716723 %
- TRAINING TIME : 21.33051371 s

b) 90% INFORMATION PRESERVATION

- NO. OF PRINCIPAL COMPONENTS REQUIRED : 9
- ACCURACY OF THE MODEL : 83.44709897610922 %
- TRAINING TIME : 7.004460573196411 s

c) 50% INFORMATION PRESERVATION

- NO. OF PRINCIPAL COMPONENTS REQUIRED : 3
- ACCURACY OF THE MODEL : 82.08191126279864 %
- TRAINING TIME : 7.677842617034912 s

d) MAXIMUM ACCURACY ACHIEVED

- NO. OF PRINCIPAL COMPONENTS REQUIRED : 15
- ACCURACY OF THE MODEL : 95.05119453924915 %
- TRAINING TIME : 8.7681729793548 s

IV. CONCLUSION

In this project, I tried to draw a comparison between the classification models applied to the original set of features and that to at the reduced set of features obtained using Principal Component Analysis Algorithm on the asteroid dataset on the basis of accuracy achieved and the training time required. For this purpose, first I preprocessed the data by excluding some irrelevant features. Then, I applied logistic regression algorithm to the original set of features and that to the features obtained using PCA algorithm. Finally, the comparison between above mentioned items was done.

It was found that the model trained with principal components took much less time to train. The accuracies were less in some cases (Although, the difference between the accuracy of model trained on original dataset and that trained on reduced features was less than 12%) while better in some other cases. This is actually a great performance as we are able to successfully reduce the dimensions without much affecting the accuracy and at the same time decreasing the training time by almost 33% to 66% (approximately).

V. REFERENCES

[1] https://www.kaggle.com/shrutimehta/nasa-asteroids-classification

[2] https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial