News Posts' Shares predicted through Regression and Classification

Under the guidance of
Prof R Jha

Presented by

Divyanshu Shekhar
(BE/10253/2013)
Amit Kr. Bhatnagar
(BE/10264/2013)

Rise of Machine Learning

- Grew out of work in AI
- New capability of Computers

Wide Applications of ML

- Database Mining
  - Large datasets from the growth of automation/web.
  - E.g web click data, medical records, engineering
- Applications can’t program by hand
  - E.g. Autonomous Helicopter, Handwriting Recognition, Natural Language Processing, Computer Vision.
- Self Customizing Programs
  - E.g. Amazon, Google Recommendations
- Understanding Human Learning (Brain, Real AI)
- Large scale companies e.g. Google, facebook etc are tremendously investing in this modern area.

What is Machine Learning?

- “ML is the field of study that gives computers the ability to learn without being explicitly programmed” - Arthur Samuel.
- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E” - Tom Mitchell.
- AI Dream- To build machines as intelligent as human. The best solution is to learn algorithms trying to mimic how the human brain learns.

Classification of ML problems

- Supervised Learning
  - In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

- Unsupervised Learning
  - Concern is to find a structure in the dataset where we don't necessarily know the effects of the variables and possess little or no Idea what our results should look like.
  - e.g. Social Media, Market Segmentation

Supervised Learning

- Regression
  - results within a continuous output
  - e.g - given a picture of person, predict the age of the person.
- Classification
  - Discrete Output
  - e.g. - Given a patient with tumor, predict whether the tumor is benign or malignant.

Literature Review I

- A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance
- This paper's conclusion tells that classification algorithm produces a better performance than the regression algorithms.
- This helps in solidifying the concept that prediction must be based on several techniques and the analysis of those results obtained.

Literature Review II

- A Comparison Of Logistic Regression, Neural Networks, and Classification trees in a study to predict success of actuarial students.
- The conclusion of this papers says that although logistic regression method works well for a variety of problems but more accurate results can be obtained by deploying different other algorithms

Selection of Dataset

While selecting the dataset, few points were kept in mind

- Selecting a dataset which can be utilised for regression and classification problem as well.
- Have training examples in the order of 10, 000
- Multivariable features

Online News Popularity

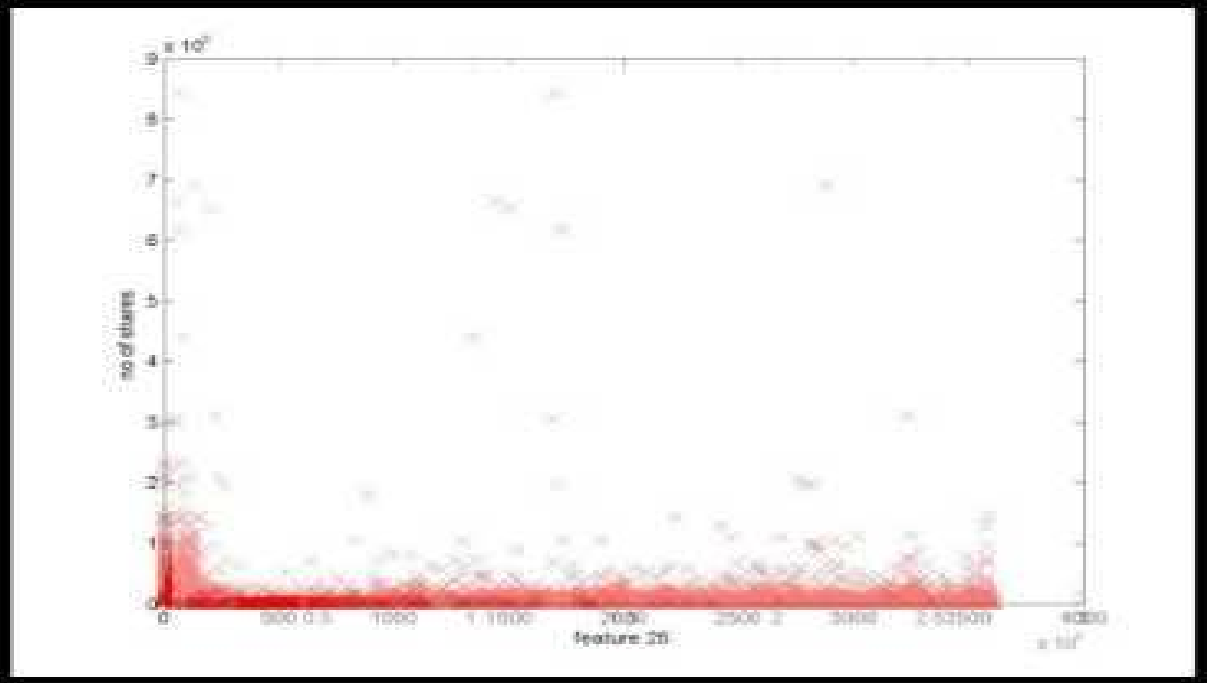
- Set of features about articles published by Mashable in two years.
- Number of instances- 39k
- 58 predictive fields, 1 goal field(no of shares)
- No of words, no of images, etc are few among the predictive features.

- Using ML, “no of shares” would be predicted using different techniques.
- Datset gathered from https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity

Techniques Implemented

- Linear Decision Boundary
  - Linear Regression using Gradient Descent Method
  - Linear Regression using Normal Method.
- Polynomial Decision Boundary
  - Gradient Descent Method.
  - Normal Method.
- Logistic Regression

Visualaisation of Data



Gradient Descent Method

- Hypothesis Function
- Cost Function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- Feature Scaling
- Gradient Descent Convergence

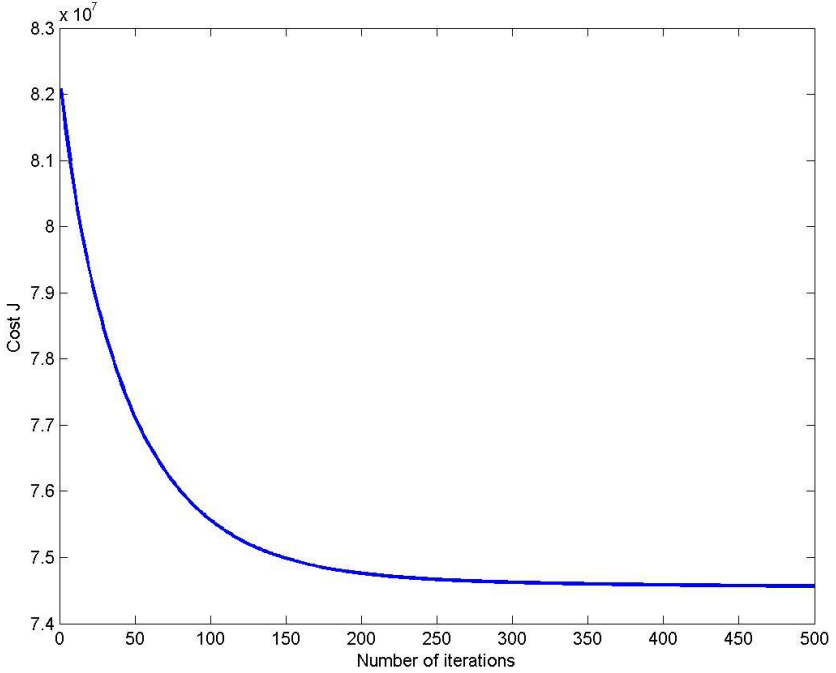
repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

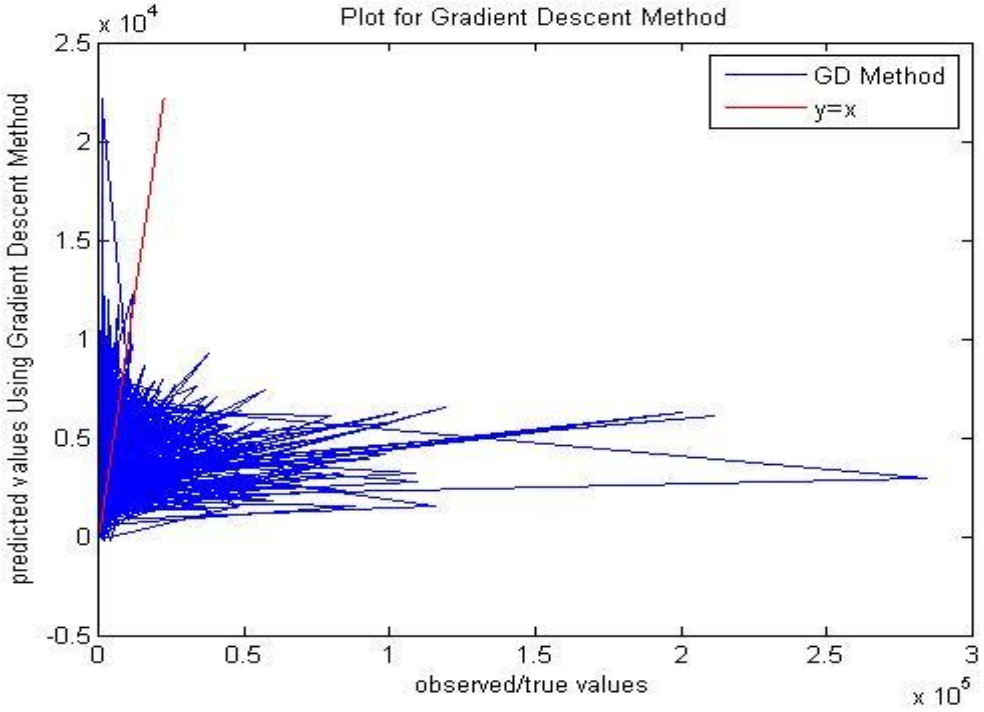
Cost Function for gradient Descent Method

- Cost Function decreases with iterations.
- Learning rate, alpha= 0.03



Prediction(GD Method)

- Training Accuracy- close to zero
- Accuracy is found to be approx 6% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 12%



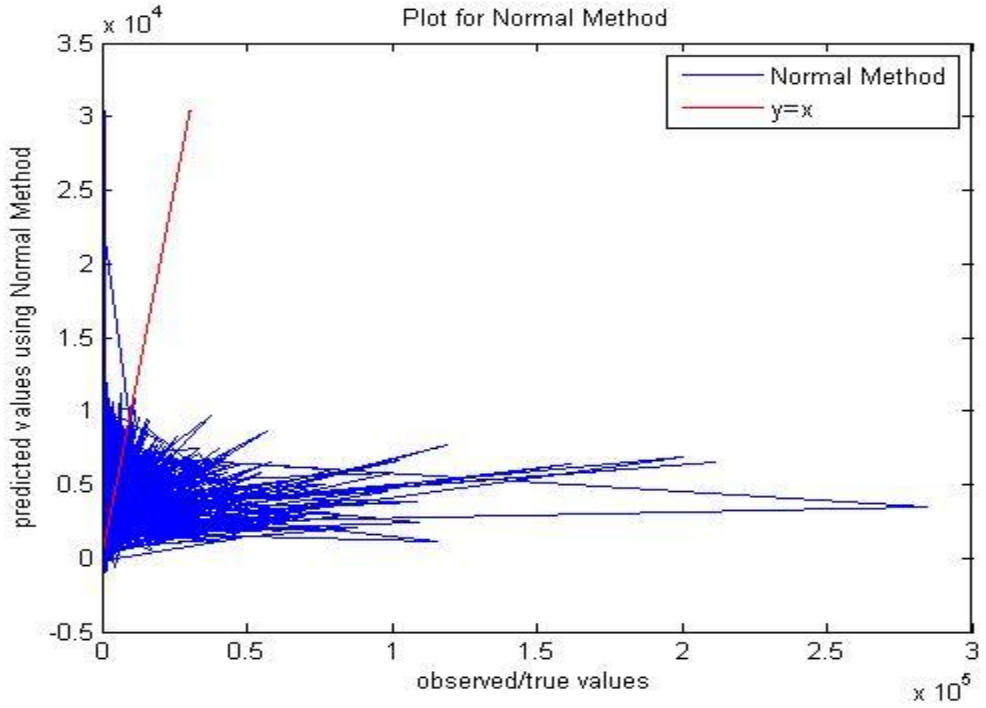
Normal Method

- Method of finding optimum theta without iterations.
- No need of feature scaling
-

$$\theta = (X^T X)^{-1} X^T y$$

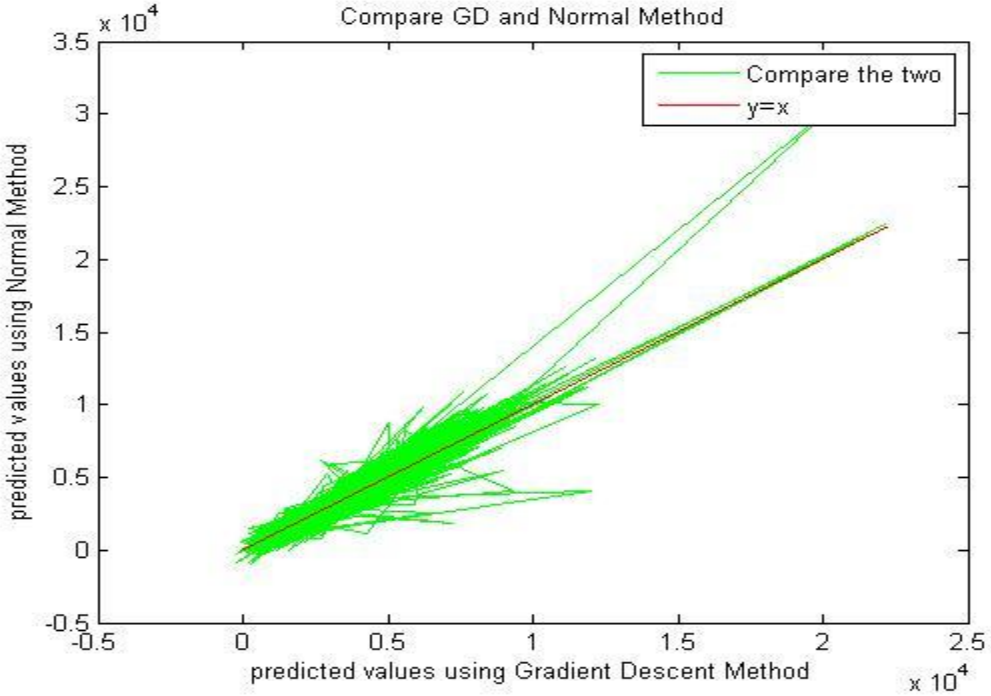
Prediction(Normal Method)

- Training Accuracy close to zero.
- Accuracy is found to be approx 7% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 13%



Compare GD and Normal Method Results

- Similarity in the results by 2 methods can be seen here.
- Several “theta” or weight values were almost same in the two cases.

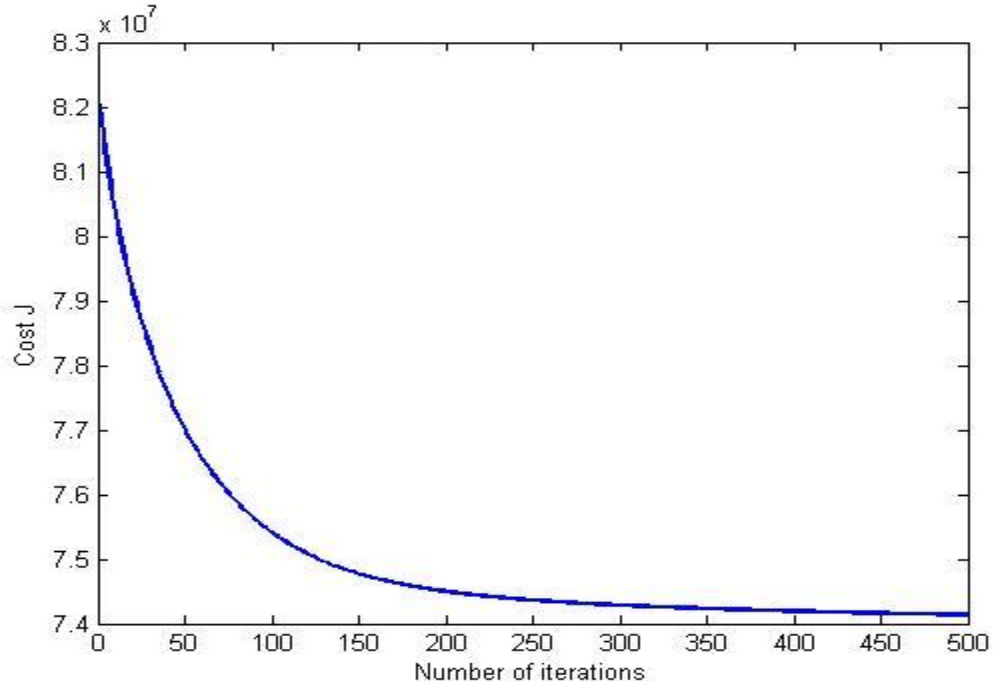


Polynomial Regression

- Hypothesis function need not be linear, if it doesn’t fit the data well.
- We can change the curve of our hypothesis by making it quadratic, cubic or any other form.

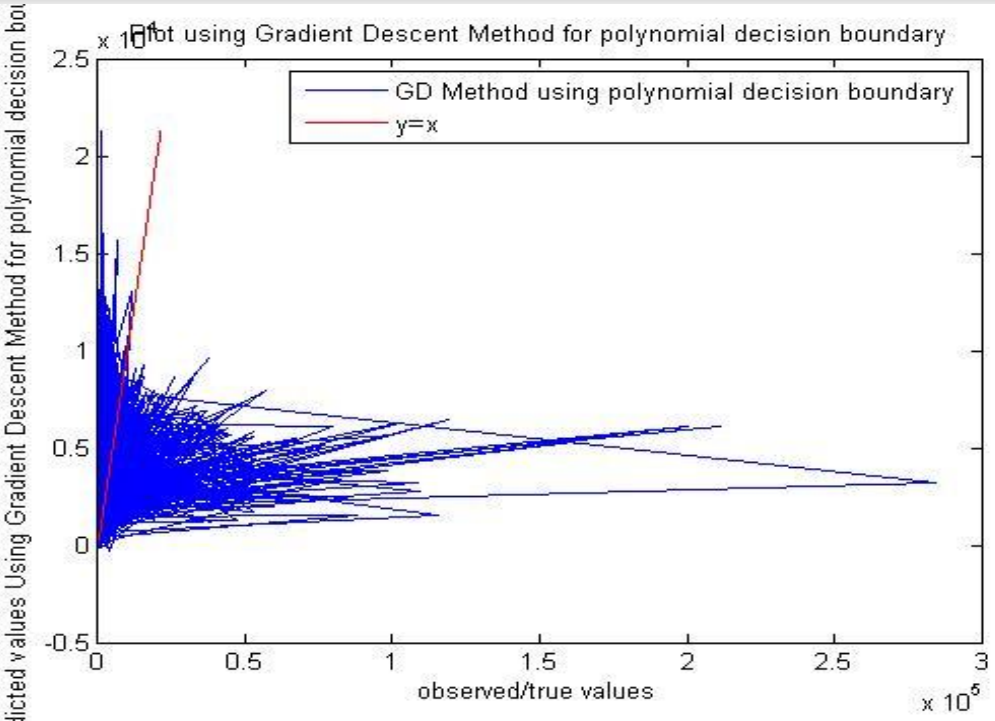
Cost Function(GD) for Polynomial Regression

- Cost Function Decreasing
- Learning Rate, alpha= 0.01



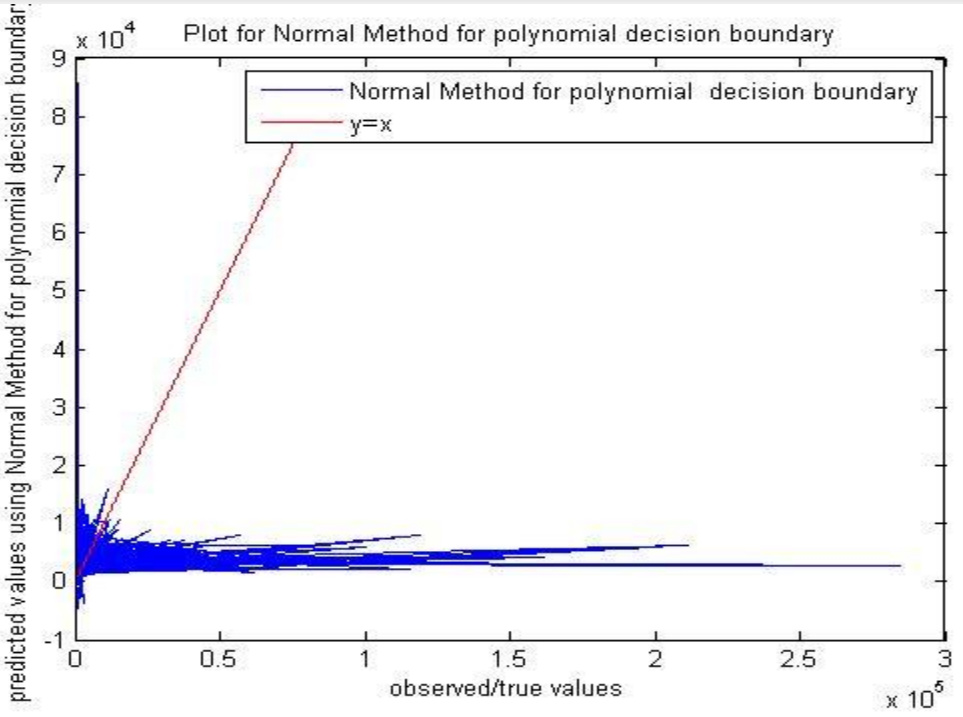
Prediction(GD) for Polynomial Regression

- Training Accuracy close to zero.
- Accuracy is found to be approx 6% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 12%

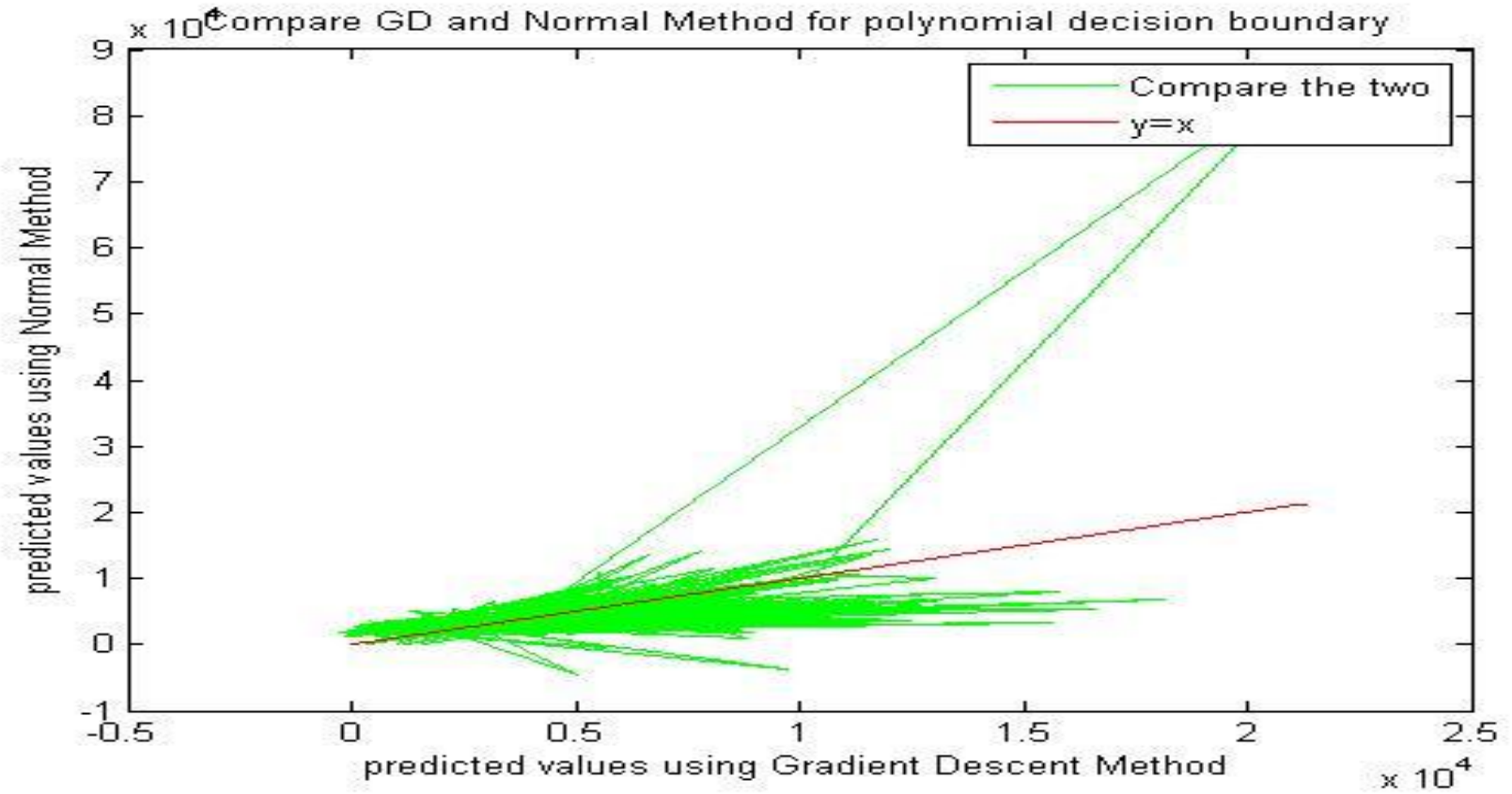


Prediction(Normal) for Polynomial Regression

- Training Accuracy close to zero.
- Accuracy is found to be approx 5% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 9%



Prediction for GD and Normal Method for Polynomial Regression



Logistic Regression(OneVsAll)

- Hypothesis function must satisfy
  $0 \leq h\theta(x) \leq 1$
- Sigmoid function is used in calculation of hypothesis function.
- The cost function is calculated as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

- For One vs All method, multiple classes were formed and the theta or weights values are found for each of the classes.

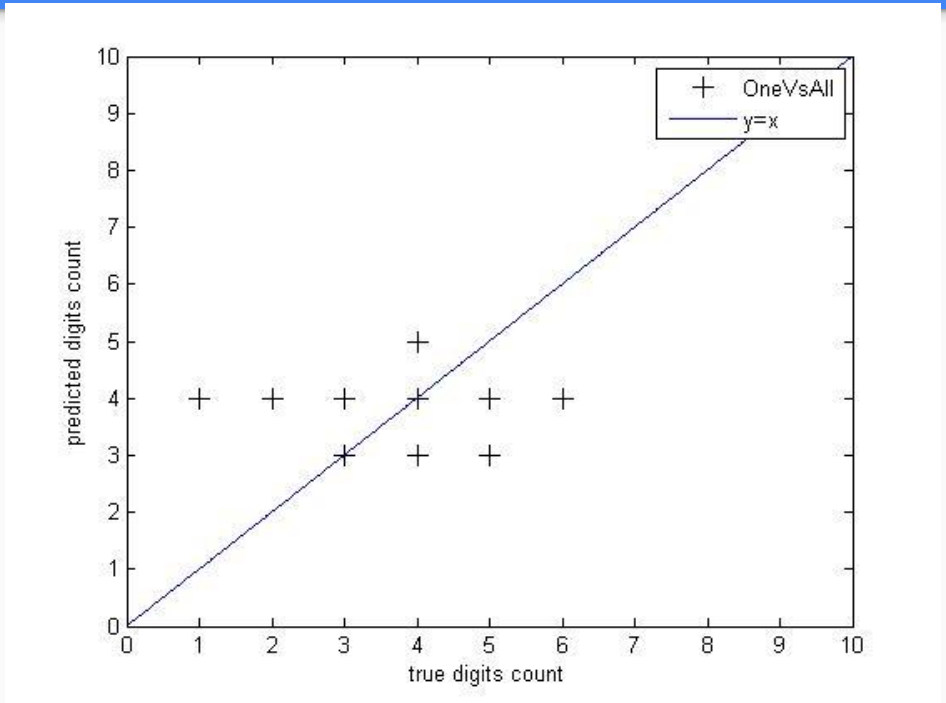
OneVsAll

- In the dataset, a new field containing the number of digits in the number of shares field is created. This way the dataset can be classified into 6 classes.
- For every instance in the test suite, hypothesis values for all the corresponding classes are calculated and prediction is made by choosing the class with highest hypothesis value.
- For regularisation, cost function also includes lambda*summation(theta).

OneVsAll with and without regularisation

- On applying the above algorithm without regularisation term, the training accuracy came out to be approx 64%.

- On using the regularisation parameter, lambda= 1 or 2 doesn't give any better result other than increasing the training accuracy negligibly.



Conclusion so far

- Among the algorithms implemented so far, classification algorithms are giving much better result than the regression algorithms as evident from the training accuracy obtained earlier in case of different algorithms' implementations and corresponding plots.
- Need to search for other algorithms
- Possibly, many more suitable instances required to learn properly.
- Manipulation of features are necessary.

Features Extraction

- Because of training accuracy obtained was not so great, one of the most feasible solution was to extract features.
- Extraction of common quantitative features for message content such as: no. of words, unique words, etc.
- Tf-idf (Term frequency - Inverse document frequency)

Example of extracted features:

- Training- “Companies would get a tax break for helping workers pay off their student loans.”
- Training-”Students’ success depend on their hard work.”
- Vocabulary-(0: Company, 1:tax, 2:workers, 3:pay, 4:student, 5:loans, 6:success, 7:depend, 8:hard ,9:work)
- Test- “A company’s success depends upon workers.”
- Vector- (0-1,1-0, 2-1,3-0, 4-0, 5-0, 6-0, 7-0, 8-0, 9-0.)

Tf-idf

- A way to convert the textual representation of information into a Vector Space Model(VSM), or into sparse features.
- To model the document into a vector space ,firstly, create a dictionary of terms taken from data set.
- Term frequency:

$$\mathrm{tf}(t,d) = \sum_{x \in d} \mathrm{fr}(x,t)$$

where the $\mathrm{fr}(x,t)$ is a simple function defined as:

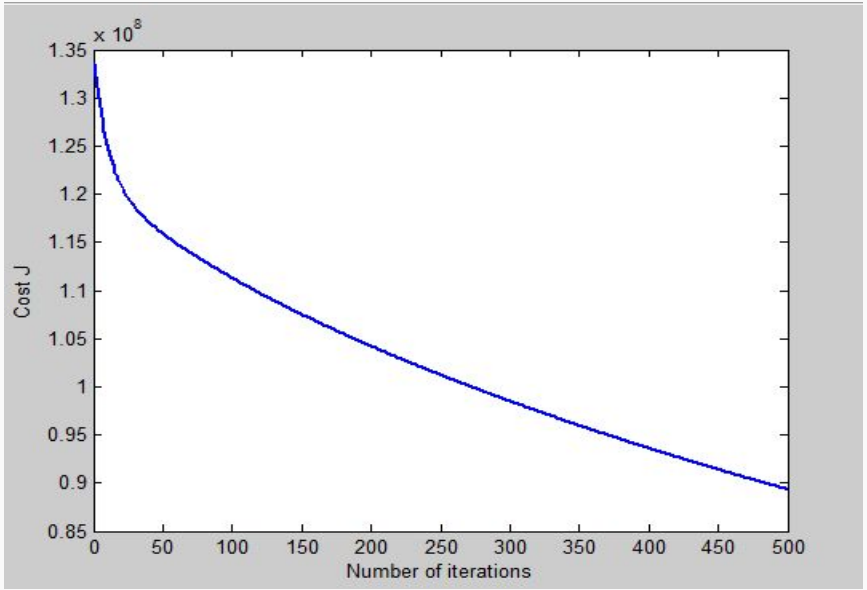
$$\mathrm{fr}(x,t) = \begin{cases} 1, & \text{if } x = t \\ 0, & \text{otherwise} \end{cases}$$

Document vector:

$$\vec{v_{d_n}} = (\mathrm{tf}(t_1,d_n), \mathrm{tf}(t_2,d_n), \mathrm{tf}(t_3,d_n), \ldots, \mathrm{tf}(t_n,d_n))$$

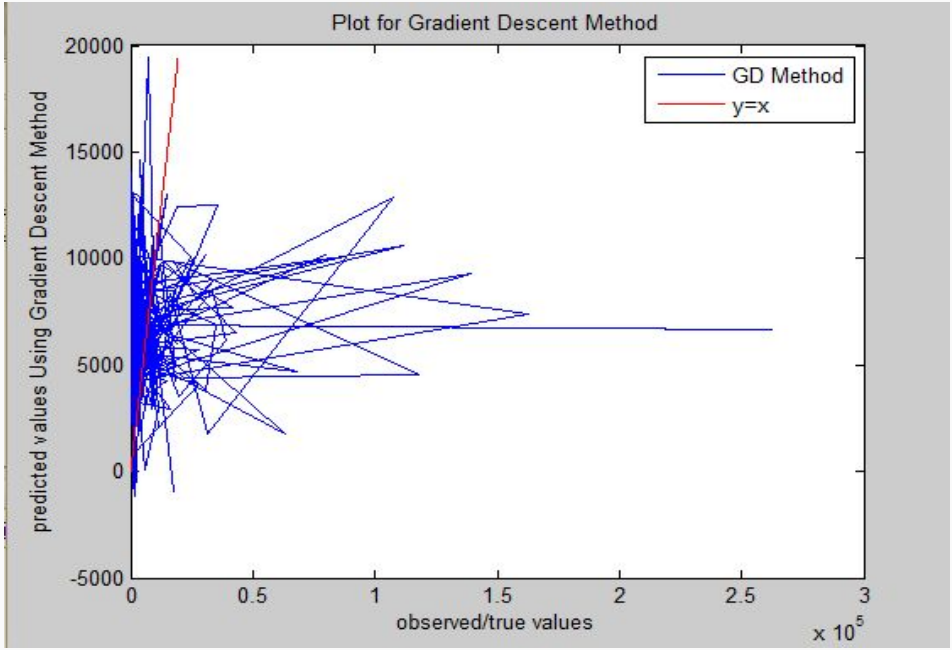
Cost Function for gradient Descent Method

- Cost Function decreases with iterations.
- Learning rate, alpha= 0.01



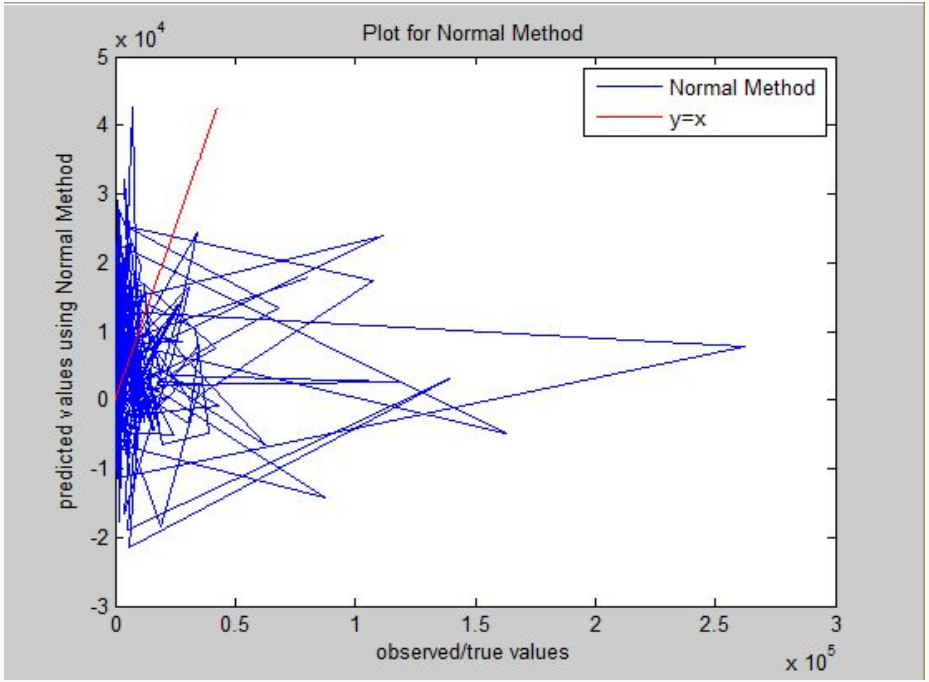
Prediction(GD) for Regression

- Training Accuracy close to zero.
- Accuracy is found to be approx 8% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 15%

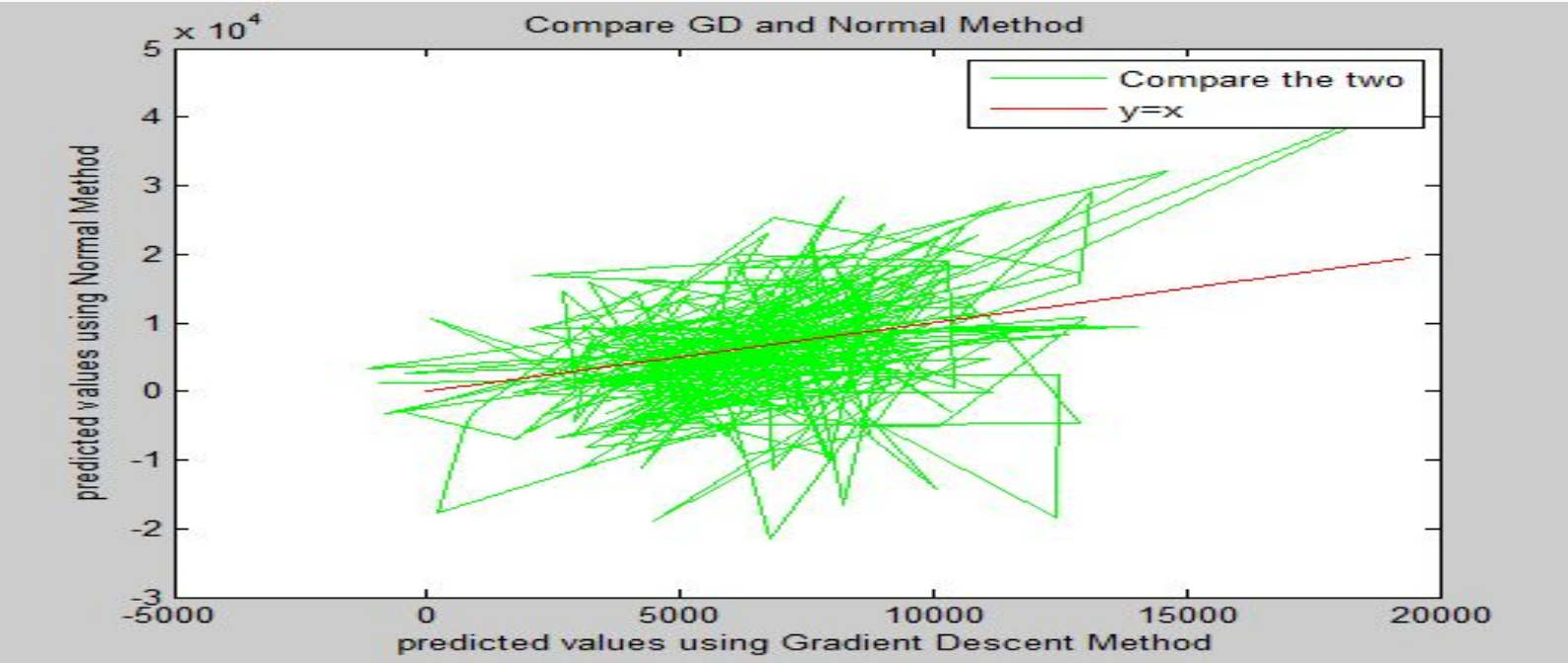


Prediction(Normal) for Regression

- Training Accuracy close to zero.
- Accuracy is found to be approx 8% when 10% of shift in predicted values are acceptable.
- When 20% of shift is acceptable, accuracy is approx 16%



Prediction for GD and Normal Method for Regression



Conclusion

- Training accuracy did improve but wasn’t good enough.
- Other algorithms’ implementations may help.
- More features need to be extracted using other methods.

Future Works

- Finding the most effective algorithm to predict the number of shares and studying the behaviour of these algorithms.
- Employing more techniques to extract features.
- Applying several techniques such as Neural Network, SVM etc
- Creating an analyser of graph behaviour of different features and utilising it to form hypothesis function.

References

- https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity
- http://www.educationaldatamining.org/EDM2015/uploads/papers/paper_158.pdf
- http://blog.christianperone.com/2011/09/machine-learning-text-feature-extraction-tf-idf-part-i/
- http://www.nedsi.org/proc/2007/proc/p061011026.pdf
- Huge thanks to Andrew Ng for his course materials on Coursera.

Thank you